BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Adapun penelitian terdahulu yang akan digunakan sebagai sumber acuan yang relevan dan terkini yaitu:

Luqman Affandi, Ridwan Rismanto, dan Muhammad Mustofa Firmansyah (2017) dengan judul "APLIKASI PENGIRIMAN PESANAN MAKANAN MENGGUNAKAN ALGORITMA DJIKSTRA". Berdasarkan penelitian ini, didapatkan kesimpulan yaitu dengan membandingkan penerapan algoritma Djikstra yang dihitung menggunakan Excel dan melalui sistem, didapatkan hasil perhitungan yang sama, sehingga sistem ini dinilai berhasil mencari urutan rute terpendek dari beberapa titik tujuan dengan akurat. Penerapan yang mengabungkan layanan Google API yang di-*filter* menggunakan algoritma Djikstra juga dapat menghasilkan pencarian rute urut dari beberapa tujuan sekaligus. Dan aplikasi yang dibangun juga sudah mampu mencatat pesanan dari pelanggan dan data pesanan tersebut ditampilkan pada halaman kurir, dimana kurir sudah bisa mengatur status pesanan, guna untuk menangani rute pengiriman makanan yang sudah siap dikirimkan.

Embun Fajar Wati (2018) dengan judul "APLIKASI SISTEM LAYANAN PESAN ANTAR MAKANAN BERBASIS ANDROID PADA KEDAI AYAM REMUK". Berdasarkan penelitian ini, didapatkan kesimpulan yaitu dengan adanya aplikasi layanan antar pesan berbasis android diharapkan dapat memperlancar

kinerja perusahaan. Aplikasi layanan antar pesan yang dibuat ini juga memudahkan dan mempercepat dalam proses pemesanan dan pengiriman. Dan keamanan data akan lebih terjamin dengan adanya aplikasi yang berbeda sesuai kebutuhan *user*.

Randika Gumilar Saputra (2018) dengan judul "APLIKASI LAYANAN PESAN ANTAR AYAM POTONG BERBASIS ANDROID". Berdasarkan penelitian ini, didapatkan kesimpulan yaitu telah berhasilnya dibangun aplikasi layanan pesan antar ayam potong berbasis android untuk mempermudah dalam pemesanan dan pemasaran ayam potong secara *online*. Aplikasi layanan pesan antar ayam potong berbasis android ini juga dibangun untuk dapat memberikan informasi kepada konsumen dengan melihat menu-menu ayam potong, harga, ukuran dan jenis ayam sesuai dengan kriteria yang diinginkan. Dan berdasarkan hasil pengujian fungsional yaitu pengujian pada versi android, resolusi layar dan densitas layar, pengujian user interface dan pengujian fungsi dan menu aplikasi didapatkan hasil berjalan dengan yang baik.

Canggih Ajika Pamungkas (2019) dengan judul "APLIKASI PENGHITUNG JARAK KOORDINAT BERDASARKAN LATITUDE DAN LONGITUDE DENGAN METODE EUCLIDEAN DISTANCE DAN METODE HAVERSINE". Berdasarkan penelitian ini, didapatkan kesimpulan yaitu telah berhasilnya dibangun aplikasi layanan pesan antar ayam potong berbasis android untuk tujuan pembandingan antara metode *Euclidian Distance* dan metode *Haversine*. Pengujian menunjukkan hasil sama saat perhitungan jarak antara metode *Euclidean Distance* dan metode *Haversine*. Penelitian ini menyajikan pembahasan mengenai pembangunan Aplikasi Berbasis Android Penghitung Jarak

Koordinat Berdasarkan Latitude dan Longitude menggunakan metode Euclidean Distance dan metode Haversinememanfaatkan API Google Maps. Pengembangan yang dilakukan mengunakan HTML, PHP, CSS dan Javascript. Aplikasi web tersebut kemudian dibentuk menjadi file *.apk dengan softwareWebsite 2 APK Builder.

Dona Marcelina, Evi Yulianti (2020) dengan judul "APLIKASI PENCARIAN RUTE TERPENDEK LOKASI KULINER KHAS PALEMBANG MENGGUNAKAN ALGORITMA EUCLIDEAN DISTANCE DAN A* (STAR)". Berdasarkan penelitian ini, didapatkan kesimpulan yaitu bahwasanya metode yang dilakukan memiliki tingkat akurasi tinggi. Peneliti menyarankan untuk penelitian selanjutnya dapat mengembangkan aplikasi ini seperti dengan menambahan berbagai fitur baru lainnya.

II.2. Landasan Teori

Berikut merupakan beberapa teori yang sesuai dan signifikan dengan objek penelitian yang mendukung penelitian ini, yang didapatkan dari berbagai sumber terpercaya sebagai landasan dalam melakukan perancangan sistem.

II.2.1. Android

Android merupakan salah satu sistem operasi yang sedang *booming* saat ini. Kelebihan Android dibandingkan sistem operasi *smartphone* lainnya adalah Android berbasis *open source code* sehingga memudahkan para pengembang untuk menciptakan dan memodifikasi aplikasi atau fitur-fitur yang belum ada di sistem operasi Android sesuai dengan keinginan mereka sendiri (Pangkey, Poekoel, &

Lantang, 2016). Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Android adalah sistem operasi yang menghidupkan lebih dari satu miliar *smartphone* dan tablet. Karena perangkat ini membuat hidup kita begitu manis, maka setiap versi Android dinamai dari makanan penutup (*dessert*).

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 bergerak. Android umum digunakan di *smartphone*. Fungsinya sama seperti sistem operasi Symbian di Nokia, iOS di Apple, dan BlackBerry OS. (Wahyu Widodo, 2017).

II.2.2. Android Studio

Android Studio yang merupakan Lingkungan Pengembangan Terpadu - *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas saat membuat aplikasi Android, misalnya sistem versi berbasis Gradle yang fleksibel, *emulator* yang cepat dan kaya fitur, lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android, instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru, template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh, alat pengujian dan kerangka kerja yang

ekstensif, alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain, dukungan C++ dan NDK, dukungan bawaan untuk Google Cloud Platform, mempermudah pengintegrasian Google Cloud Messaging dan App Engine. Tahap selanjutnya setelah pemaparan materi adalah praktik langsung. Untuk mengimplementasikan program aplikasi yang telah dirancang, maka diperlukan sebuah alat bantu berupa komputer, yang mana untuk mengoperasikan komputer itu sendiri yang memerlukan tiga buah komponen pendukung seperti hardware, software, dan brainware. Android studio adalah IDE resmi untuk pengembangan aplikasi Android dan bersifat open source atau gratis (Juansyah, 2015).

Android merupakan generasi baru *platform mobile*, yang memberikan pengembang untuk melakukan pengembangan sesuai dengan yang diharapkan. Sistem oprasi yang mendasari android dilisensikan dibawah GNU, *General Public License* v2.0 (GPLv2), yang sering dikenal dengan istilah "*copyleft*" lisensi dimana setiap perbaikan pihak ke tiga harus terus jatuh dibawah *terms*. Aplikasi android dapat dikembangkan pada sistem operasi berikut:

1. Windows XP, Vista / 7, 8 / 8.1, 10
2. Mac OS X (Mac OS X 10.4.8 atau lebih baru)
3. Linux

II.2.3. Android SDK

Android SDK (*Software Development Kit*) adalah *tools* API (*Application Programming Interface*) yang diperlukan untuk memulai pengembangan suatu aplikasi pada *platform* android menggunakan bahasa pemrograman Java. Android

merupakan *subset* perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di-*release* oleh *Google*. Saat ini disediakan Android SDK sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* android menggunakan bahasa pemrograman Java. Sebagai *platform* aplikasi netral, android memberi anda kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan aplikasi bawaan *handphone / smartphone.* (Dedy Abdullah, dkk, 2018).

SDK Tools dan SDK Build Tools Android terdiri dari alat-alat untuk pengembangan aplikasi dan untuk pengujian hasil jalannya kode program yang telah dituliskan (Sasongko, 2017:147) perintah-perintah java yang dituliskan pada saat pembuatan aplikasi Android tidak akan dikenal tanpa menggunkan alat ini. Untuk peta dapat dilakukan secara mudah melalui layanan gratis dari Google. Suatu dokumentasi yang terdiri dari *interface*, fungsi, kelas, struktur dan sebagainya untuk membangun sebuah perangkat lunak.

II.2.4. *Android Development Tools* (ADT)

Android Development Tools (ADT) yang lebih dikenal plugin Eclipse. ADT adalah *plugins* di Eclipse yang harus kita *install* sehingga Android SDK dapat dihubungkan dengan IDE Eclipse yang digunakan sebagai tempat *coding* aplikasi Android nantinya.

Menurut Setiawan, Mulyani, & Budihartanti (2014:151) mendefinisikan bahwa “Android SDK merupakan alat bantu dan API dalam mengembangkan aplikasi pada *platform* android menggunakan bahasa pemrograman Java”. *Tools* bagi para programmer yang ingin mengembangkan aplikasi berbasis Google

android, mencangkup seperangkat alat pengembangan yang komprehensif. IDE yang didukung secara resmi adalah Eclipse 3.2 atau lebih dengan menggunakan *plugin Android Development Tools* (ADT). (Setiawan, Mulyani, & Budihartanti 2014:151)

II.2.5. Google API

Dikutip dari Wikipedia Bahasa Indonesia, Google Maps adalah layanan pemetaan *web* yang dikembangkan oleh Google. Layanan ini memberikan citra satelit, peta jalan, panorama 360°, kondisi lalu lintas, dan perencanaan rute untuk bepergian dengan berjalan kaki, mobil, sepeda (versi beta), atau angkutan umum. API adalah kependekan dari *Application Programming Interface*. Dengan bahasa yang lebih sederhana, API adalah fungsi fungsi pemrograman yang disediakan oleh aplikasi atau *provider* aplikasi supaya layananan aplikasi tersebut bisa di integrasikan dengan aplikasi yang kita buat. Google Maps API adalah fungsi fungsi pemrograman yang disediakan oleh Google Maps agar *maps* dapat di integrasikan kedalam web atau aplikasi yang sedang dikembangkan.

Ada banyak API yang disediakan oleh Google, beberapa diantaranya adalah:

1. *Language* API: untuk memanfaatkan fitur translation yang dimiliki Google.
2. *Earth* API: untuk memanfatkan fitur yang ada pada Google Earth
3. Javascript API
4. *Maps* API: memanfaatkan fitur yang ada pada Google Maps
5. *Search* API: memanfaatkan fitur pencarian pada Google Search

6. *Visualization* API: membuat grafik maupun chart dengan Google API
7. YouTube API: memanfaat kan fitur yang ada pada YouTube misalnya untuk pencarian video

Google Maps adalah layanan gratis berupa peta interaktif yang disediakan oleh Google. Google Maps API (*Application Program Interface*) terdiri dari sekumpulan kode-kode blok yang nantinya dapat digunakan untuk memodifikasi petasesuai dengan kebutuhan pengguna. Untuk dapat menggunakan Google Maps API, pengembang aplikasi harus sudah memiliki akun Google yang akan digunakan untuk mendapatkan kode akses dari Google dalam bentuk *token* (Onsu et al., 2016).

II.2.6. *Database*

Database adalah sekumpulan tabel-tabel yang saling berelasi, relasi tersebut bisa ditunjukkan dengan kunci dari tiap tabel yang ada. Satu *database* menunjukkan satu lingkup perusahaan atau instansi (Gellysa dan Fauzi H., 2015).

Database juga merupakan kumpulan data yang umumnya menggambarkan aktifitas-aktifitas dan pelakunya dalam suatu organisasi. Sistem *database* merupakan sistem komputer yang digunakan untuk menyimpan dan mengelola data tersebut (Hendini, 2016).

II.2.7. MySQL

Pada perkembangannya, MySQL disebut juga SQL yang merupakan singkatan dari *Structured Query Languange*. SQL merupakan bahasa terstruktur yang khusus digunakan untuk mengolah *database*. SQL pertama kali didefinisikan oleh American National Standards Institute (ANSI) pada tahun 1986. MySQL

adalah sebuah sistem manajemen *database* yang bersifat *open source*. MySQL merupakan sistem manajemen *database* yang bersifat *relational*, artinya, data yang dikelola dalam *database* yang akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan jauh lebih cepat. MySQL dapat digunakan untuk mengelola *database* mulai dari yang kecil sampai dengan yang sangat besar. (Andi, Wahana Komputer, 2014:h.73). SQL juga dapat diartikan sebagai antar muka standar untuk sistem manajemen relasional, termasuk sistem yang beroperasi pada komputer pribadi. SQL memungkinkan seorang pengguna untuk mengetahui dimana lokasinya, atau bagaimana informasi tersebut disusun. SQL lebih mudah digunakan dibandingkan dengan bahasa pemrograman, tetapi rumit dibandingkan *software* lembar kerja dan pengolah data.

Sebuah pernyataan SQL yang sederhana dapat menghasilkan *set* permintaan untuk informasi yang tersimpan pada komputer yang berbeda diberbagai lokasi yang tersebar, sehingga membutuhkan waktu dan sumber daya komputasi yang banyak. SQLite dapat digunakan untuk investigasi interaktif, atau pembuatan laporan *ad-hoc* atau disisipkan dalam program aplikasi. SQL juga merupakan bahasa pemrograman yang dirancang khusus untuk mengirimankan suatu perintah *query* (pengaksesan data berdasarkan pengalamatan tertentu) terhadap sebuah *database*. Kebanyakan *software database* mengimplementasikan SQL secara sedikit berbeda, tapi seluruh *database* SQL mendukung *subset* standar yang ada. Jadi, SQL adalah permintaan yang melekat pada suatu *database* atau SMBD tertentu. Dengan kata lain, SQL adalah perintah atau bahasa yang melekat di dalam SMBD. Sebagai suatu bahasa permintaan, SQL didukung oleh SMBD, seperti

MySQL Server, MySQL, PostgreSQL, Interbase, dan Oracle. Selain itu SQL juga didukung oleh *database* bukan *server*, seperti MS *Access* maupun Paradox (Surniawan, Wahana Eri Mardiani, 2014:h,25,26).

MySQL adalah sistem manajemen *database* SQL yang bersifat open source dan paling populer saat ini. Sistem *Database* MySQL mendukung beberapa fitur sebagai *multithreaded*, *multi-user* dan SQL *database managemen system* (DBMS). *Database* ini buat untuk keperluan sistem *database* yang cepat, handal dan mudah digunakan (Madcoms, 2016).

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

Unified Modeling Language (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan *diagram* dan teks-teks pendukung. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek (Rosa A.S dan M. Shalahudin, 2014:133). Aplikasi yang akan dibuat dan perancangan UML (*Unified Modeling Language*) dirancang dengan empat model *diagram* (*Use Case Diagram, Class Diagram, Sequence Diagram, Activity Diagram*).

Menurut Afriansyah (2015:56), mendefinisikan bahwa "*Unified Modeling Language* (UML) merupakan sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan system piranti lunak".

II.2.8.1. *Use Case Diagram*

Use case adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari *use case symbol* namun dapat juga dilakukan dalam *activity diagram*. *Use case* digambarkan hanya yang dilihat dari luar oleh aktor (keadaan lingkungan sistem yang dilihat user) dan bukan bagaimana fungsi yang ada di dalam sistem.

` *Use case diagram* dapat menetukan pengguna potensial dan bagaimana mereka bereaksi terhadap aktivitas dan aliran proses dalam aplikasi berbasis *web* (Mubin dkk, 2016). Komponen-komponen dari *use case diagram* terdiri dari:

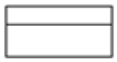
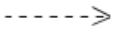
Tabel II.1. Komponen *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2	<< include >>	*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.
3	<< extend >>	*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
4		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
5		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
6		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

II.2.8.2. *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 (*Associations*), *Generalitation* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operation / Method*) dan *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau *attribute*. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality* (Gellysa dan Fauzi H., 2015). Komponen-komponen untuk *class diagram* terdiri dari:

Tabel II.2. Komponen *Class Diagram*

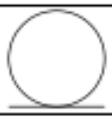
NO	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2		*N-Ary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
6		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya

II.2.8.3. *Sequence Diagram*

Sequence Diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek (Gellysa dan Fauzi H., 2015).

Sequence diagram digunakan untuk menjelaskan perilaku pada sebuah skenario dan menggambarkan bagaimana entitas dan sistem berinteraksi, termasuk pesan yang dipakai saat interaksi. Semua pesan digambbarkan dalam urutan pada eksekusi. Berikut komponen - komponen yang ada pada *sequence diagram*:

Tabel II.3. Komponen *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
		*Actor*	Digunakan untuk menggambarkan user / pemgguna.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3		*Boundary*	Digunakan untuk menggambarkan sebuah form.
4		*Control Class*	Digunakan untuk menghubungkan *boundary* dengan tabel.
5		*Entity Clas*	Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan.

II.2.8.4. Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Gellysa dan Fauzi H., 2015).

Activity diagram juga dapat menjelaskan metode paralel yang mungkin terjadi pada beberapa eksekusi. Activity diagram adalah state diagram khusus, yang mana state ini berfungsi sebagai action dan sebagian besar transisi ditrigger oleh akhir state sebelumnya (internal processing). Berikut komponen - komponen yang ada pada *activity diagram*:

Tabel II.4. Komponen *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Activity Final Node*	Bagaimana objek dibentuk dan diakhiri
5		*Decision*	Diguanakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
6		*Line Connector*	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya