



## **BAB II**

### **TINJAUAN PUSTAKA**

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Penelitian Terkait**

Untuk mendukung keberhasilan penelitian ini, penulis melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan. Beberapa uraian penelitian terdahulu yang menjadi acuan dalam penelitian ini yaitu:

Pada penelitian terkait yang dilakukan oleh Ferdian Octa Dinata dan Ratna Mutu Manikam (2019). Berdasarkan hasil dan pembahasan yang telah dilakukan dapat ditarik kesimpulan sebagai berikut: Aplikasi ini dapat digunakan oleh masyarakat dalam mencari informasi kepulauan seribu Jakarta. dan aplikasi ini dapat digunakan oleh pelanggan dalam melakukan pengecekan fasilitas yang ada di setiap pulau. Adapun yang membedakan pada rancangan yang di bangun adalah aplikasi pemetaan pariwisata di Sidikalang Kabupaten Dairi dengan *smartphone*.

Pada penelitian terkait yang dilakukan oleh Dewi Yanti (2018). Penelitian yang dihasilkan adalah Sistem Informasi Pariwisata Kabupaten Dairi dapat membantu mempromosikan Pariwisata Kabupaten Dairi kepada seluruh masyarakat di Tanah Air Indonesia bahkan masyarakat internasional. Sistem Informasi Pariwisata Kabupaten Dairi memberikan kemudahan kepada masyarakat dalam mencari informasi mengenai pariwisata, sejarah Dairi, seni budaya setempat. Adapun yang membedakan penelitian selanjutnya adalah

aplikasi yang dapat membantu memberikan informasi lokasi pariwisata dan jarak rute terpendek.

Pada penelitian terkait yang dilakukan oleh Sandro & Ririn (2017). Hasil penelitian ini adalah sebuah rancangan aplikasi sistem informasi kereta api di Jabodetabek yang dapat digunakan oleh pengguna untuk mengetahui informasi perjalanan kereta api. Aplikasi ini terintegrasi dengan layanan Google Map dalam penentuan jalur antara user dengan suatu tempat dan untuk menampilkan lokasi, informasi dan petunjuk tentang stasiun dan kereta.

Pada penelitian terkait yang dilakukan oleh Mukhlis, dkk (2017). Dari hasil penelitian aplikasi peta sekolah berbasis android dapat diambil yaitu aplikasi dapat menampilkan letak atau lokasi sekolah dan menampilkan informasi sekolah yang ada di Bengkalis serta aplikasi peta sekolah dapat dijalankan di berbagai tipe perangkat mobile, dan tampilan peta serta menampilkan informasi sesuai yang diinginkan. Adapun yang membedakan pada rancangan dari peneliti yaitu aplikasi yang menampilkan lokasi pariwisata dan informasi melalui *smartphone*.

Pada penelitian terkait yang dilakukan oleh Usman Efendi & Suyanto (2016). Penelitian ini mengembangkan aplikasi mobile pencarian halte BRT TransMusi berbasis lokasi. Dengan aplikasi ini pengguna dapat dengan mudah menemukan dan menuju halte terdekat dari posisinya berada karena aplikasi ini akan menampilkan peta jalan untuk menuju ke lokasi halte terdekat.

Pada penelitian terkait yang dilakukan oleh Dona Marcelina dan Evi Yulianti (2020). Berdasarkan hasil yang didapat dalam penelitian ini maka

disimpulkan bahwa Algoritma *Euclidean Distance* dan A\* (Star) dapat digunakan untuk melakukan pencarian rute terpendek lokasi kuliner yang ada di kota Palembang. Berdasarkan hasil akurasi yang dicapai pada pengujian menunjukkan bahwa Nilai MAPE (mean absolute percentage error) untuk evaluasi prediksi akurasinya yaitu 4,4%. Pengujian yang dilakukan menghasilkan kesimpulan bahwa metode yang dilakukan memiliki tingkat akurasi tinggi. Peneliti menyarankan untuk penelitian selanjutnya dapat mengembangkan aplikasi ini seperti dengan menambahkan fitur seperti suara untuk mengarahkan ke tempat lokasi tujuan dan juga menghitung kemacetan lalu lintas jalan.

Pada penelitian terkait yang dilakukan oleh Rezky Rizaldi, dkk. (2017). Penelitian ini menghasilkan Aplikasi ruang ganti virtual ini ini mampu memberikan rekomendasi ukuran pakaian kepada calon pembeli dengan tingkat akurasi sebesar 70 % dengan memperhatikan ketinggian peletakan Kinect serta jarak antara calon pembeli dan Kinect. Sistem akan optimal dalam memberikan rekomendasi sebesar 90,8 % jika peletakan Kinect setinggi 55 m. Sistem ini juga dilengkapi dengan fitur berganti pakaian. Selayaknya orang mencoba, sistem ini juga mengadopsi keinginan calon pembeli ketika mencoba pakaian, maka dia akan bergerak ke kanan, kiri atau bahkan membalik badan serta mendekat atau menjauh ke cermin. Hasil dari uji coba, memperlihatkan bahwa pakaian virtual masih tetap melekat pada badan calon pembeli walaupun dia telah melakukan pergerakan ke kanan, kiri, ataupun membalik badan. Dan pakaian virtual juga menunjukkan semakin membesar/mengecil jika calon pembeli mendekat/menjauh dari Kinect.

Pada penelitian terkait yang dilakukan oleh Anissa Shinta Ahmasetyosari dan Titin Fatimah (2018). Penelitian ini menghasilkan proses presensi menggunakan algoritma *Euclidean Distance* dan metode *Geofencing* maka hasil presensi dapat lebih akurat dan aman karena terhindar dari kecurangan-kecurangan yang biasa terjadi. b) Tidak didapatkan waktu jeda antara perubahan lokasi sebenarnya dengan perhitungan jarak. c) Penggunaan satu akun hanya dapat diakses pada satu perangkat saja tanpa mengganggu aktifitas lainnya dalam aplikasi, sehingga aplikasi dapat dipercaya dalam segala aktifitasnya. d) File keluaran berupa dokumen berekstensi .xls dapat digunakan sebagai laporan presensi pada suatu periode tertentu.

Pada penelitian terkait yang dilakukan oleh Fitriyani, dkk. (2017). Penelitian ini menghasilkan Aplikasi sistem pemilihan paket internet menggunakan algoritma *Euclidean Distance* dapat diterapkan untuk membantu konsumen khususnya konsumen BIP Ponsel Banjarbaru untuk menentukan paket internet yang sesuai untuk mereka gunakan. Dari Pengukuran nilai akurasi sistem rekomendasi (F1) menggunakan metode *precision* dan *recall* test, sistem yang dibangun memperoleh nilai sebesar 0.760, nilai ini cukup tinggi karena apabila nilai F1 semakin mendekati angka 1 maka semakin tinggi pula nilai akurasi dari sistem rekomendasi.

Pada penelitian terkait yang dilakukan oleh Yusup Miftahuddin, dkk. (2020). Penelitian ini menghasilkan perhitungan jarak yang dapat diimplementasikan pada sistem pengidentifikasi lokasi karyawan adalah *Euclidean* dan *Haversine* karena memiliki rata-rata selisih jarak dengan

perhitungan sebenarnya sebesar kurang dari 0,5 meter. Sedangkan metode perhitungan manhattan yang memiliki rata-rata selisih jarak dengan perhitungan sebenarnya sebesar 6,67 meter dinyatakan tidak cocok diterapkan dalam studi kasus pengidentifikasi lokasi karyawan, karena hal tersebut dapat dijadikan celah kecurangan. Kemudian, antara euclidean dan haversine yang memiliki selisih kesalahan yang kecil masih terdapat perbedaan yang menentukan. Dengan implementasi aplikasi di area perbatasan, haversine menghasilkan ketepatan keputusan yang lebih akurat.

## **II.2. Landasan Teori**

### **II.2.1. Vaksin**

Vaksin atau yang sering kali dikenal dengan nama imunisasi telah lama beredar dalam kehidupan kita sehari-hari. Dimulai dari vaksin hepatitis B yang diberikan kepada bayi yang baru lahir untuk memberikan kekebalan imunitas kepada bayi tersebut, sehingga peluang hidup sehat lebih besar, serta diikuti oleh berbagai jenis vaksin lain guna memberikan kekebalan imunitas bagi bayi agar mampu tumbuh sehat dan memberikan peluang hidup yang lebih panjang. (Kementerian Kesehatan RI, 2016)

Vaksin sendiri berasal dari bagian bakteri atau virus yang menyerang manusia, yang mana bagian tersebut dilemahkan dan disuntikkan ke dalam tubuh manusia dengan harapan tubuh akan membentuk antibodi terhadap bentuk bakteri atau virus serupa untuk kemudian mampu menciptakan imunitas terhadap paparan bakteri atau virus yang asli. Oleh karena itu, vaksin merupakan bagian penting

dalam peradaban manusia dalam menghadapi penyakit mematikan dan menghindari penyebaran wabah penyakit mematikan. (WHO, 2019).

### **II.2.2. Aplikasi**

Aplikasi adalah satu unit perangkat lunak yang dibuat untuk membantu melayani kebutuhan seseorang untuk melakukan aktivitas. Dengan adanya aplikasi maka kebutuhan akan pelayanan sebuah aktivitas menjadi lebih baik. Dimana setiap pekerjaan dapat dilakukan dengan mudah kalau menggunakan sebuah aplikasi (Agusdi Syafrizal: 2018).

### **II.2.3. Lokasi**

Lokasi adalah berbagai kegiatan perusahaan untuk membuat produk yang dihasilkan atau dijual terjangkau dan tersedia bagi pasar sasaran, dalam hal ini berhubungan dengan bagaimana cara penyampaian produk atau jasa kepada para konsumen dan dimana lokasi yang strategis. (Kotler dan Armstrong: 2018:51)

### **II.2.4. *Euclidean Distance***

*Euclidean Distance* adalah perhitungan jarak dari 2 buah titik dalam Euclidean space. Euclidean space diperkenalkan oleh Euclid, seorang matematikawan dari Yunani sekitar tahun 300 B.C.E. untuk mempelajari hubungan antara sudut dan jarak. Euclidean ini berkaitan dengan Teorema Phytagoras dan biasanya diterapkan pada 1, 2 dan 3 dimensi. Tapi juga sederhana jika diterapkan pada dimensi yang lebih tinggi. Pada 1 dimensi. Semisal ingin

menghitung jarak Euclidean 1 dimensi. Titip pertama adalah 4, titik (Kiki Setiawan, dkk ; 2018 : 3).

Rumus:

$$d = \sqrt{(x1 - x2) + (y1 - y2)}$$

Dimana:

d = Jarak *Euclidean* dalam derajat

x1 = *latitude* lokasi *i*

x2 = *longitude* lokasi *i*

### II.2.5. Puskesmas

puskesmas yaitu suatu unit pelaksana fungsional yang berfungsi sebagai pusat pembangunan kesehatan, pusat pembinaan peran serta masyarakat dalam bidang kesehatan serta pusat pelayanan kesehatan tingkat pertama yang menyelenggarakan kegiatannya secara menyeluruh terpadu yang berkesinambungan pada suatu masyarakat yang bertempat tinggal dalam suatu wilayah tertentu. Puskesmas (Pusat Kesehatan Masyarakat) adalah suatu organisasi kesehatan fungsional yang merupakan pusat pengembangan kesehatan masyarakat yang juga membina peran serta masyarakat di samping memberikan pelayanan secara menyeluruh dan terpadu kepada masyarakat di wilayah kerjanya dalam bentuk kegiatan pokok. Puskesmas merupakan unit pelaksana teknis dinas kesehatan kabupaten/kota yang bertanggung jawab menyelenggarakan pembangunan kesehatan di wilayah kerja (Effendi, 2018).

### **II.2.6. *Android***

*Android* merupakan sistem operasi yang dikembangkan untuk perangkat *mobile* berbasis Linux. Pada awalnya sistem operasi ini dikembangkan oleh Android Inc. yang kemudian dibeli oleh *Google* pada tahun 2005. Dalam usaha mengembangkan *Android*, pada tahun 2007 dibentuklah Open Handset Alliance (OHA), sebuah konsorsium dari beberapa perusahaan, yaitu *Texas Instruments*, *Broadcom Corporation*, *Google*, *HTC*, *Intel*, *LG*, *Marvell Technology Group*, *Motorola*, *Nvidia*, *Qualcomm*, *Samsung Electronics*, *Sprint Nextel*, dan *T-Mobile* dengan tujuan untuk mengembangkan standar terbuka untuk perangkat *mobile*. Pada tanggal 9 Desember 2008, diumumkan bahwa 14 orang anggota baru akan bergabung dengan proyek *Android*, termasuk *PacketVideo*, *ARM Holdings*, *Atheros Communications*, *Asustek Computer Inc*, *Garmin Ltd*, *Softbank*, *Sony Ericsson*, *Toshiba Corp*, dan *Vodafone Group Plc* (Efmi Maiyana; 2018).

### **II.2.7. *Android Studio***

*Android Studio* adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. Aplikasi *Android* yang dikembangkan menggunakan *Java* dan menyesuaikan ke platform baru. *Android* terdiri dari satu tumpukan yang lengkap, mulai dari *boot loader*, *device driver*, hingga perangkat lunak API (*Application Programming Interface*), aplikasi SDK (*Software Development Kit*). *Android* bukanlah satu perangkat tertentu, melainkan sebuah *platform* yang dapat digunakan untuk mendukung konfigurasi perangkat keras. Sistem operasi yang mendasari *Android* merupakan

lisensi di bawah naungan GNU, *General Public License Versi 2(GPLv2)*, yang biasa dikenal dengan istilah *Copy left*. Istilah *copy left* ini merupakan lisensi yang setiap perbaikan oleh pihak ketiga harus terus jatuh di bawah terms. Distribusi *Android* berada di bawah *lisensi Apache Software (ASL/Apache2)*, yang memungkinkan untuk distribusi kedua atau seterusnya. Pengembang aplikasi *Android* diperbolehkan untuk mendistribusikan aplikasi mereka di bawah skema lisensi apapun yang mereka inginkan. (Ganda Yoga Swara, dkk; 2017) .

#### **II.2.8. *Firestore Realtime Database***

*Firestore Realtime Database* merupakan basis data berbasis *cloud* NoSQL yang menyinkronkan data di semua klien secara *realtime*, menyediakan fungsionalitas *offline*. Data yang diinputkan disimpan ke dalam *database Realtime* sebagai JSON. Semua klien yang terhubung berbagi satu waktu, secara otomatis menerima pembaruan data terbaru (Kurniawati dan Lukman Bachtiar; 2020: 59).

#### **II.2.9. Peta**

Peta merupakan pengenalan lokasi yang digunakan untuk mengidentifikasi suatu daerah. Peta memiliki berbagai macam bentuk seperti yang dapat kita lihat peta yang sering digunakan sebagai bahan ajar mata pelajaran geografi berupa buku bernama atlas, berisi beberapa negara di dunia dan kota-kota yang terdapat di dalamnya. Pengertian peta adalah gambaran permukaan bumi pada bidang datar dengan skala tertentu melalui sistem proyeksi (Mukhlis, ddk; 2017).

### **II.2.10. Database**

Pengertian *database* menurut Jogiyanto dalam buku analisis & desain (2005:217) adalah :”merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya, tersimpan di simpanan luar komputer dan digunakan perangkat lunak tertentu untuk memanipulasiinya. *Database* juga merupakan salah satu komponen yang penting di sistem informasi (Leonard Tambunan dan Karolina Talia Sela ; 2018 : 132).

### **II.2.11. Unified Modeling Language (UML)**

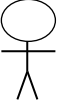
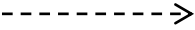

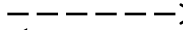
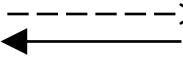

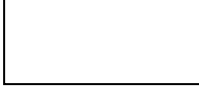
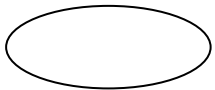


UML yaitu satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak yang terkait dengan objek. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan system yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. (Andikos, 2019 : 39).

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

#### **1. Use Case Diagram**

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah apa yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. use case diagram dapat digambarkan dengan sumber pada Tabel II.1.

**Table II.1. Simbol Use Case**

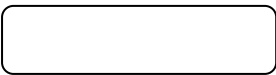
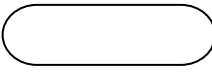



<b>Gambar</b>	<b>NAMA</b>	<b>Keterangan</b>
	Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
	<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
	<i>Include</i>	Menspesifikasikan bahwa use case sumber secara eksplisit.
	<i>Extend</i>	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan .
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur dari suatu actor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya ( <i>sinergi</i> ).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

(Sumber : Andikos, 2019 : 39)

2. Diagram Aktivitas (*Activity Diagram*)

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir, *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* dapat digambarkan dengan simbol-simbol pada tabel II.2.

**Tabel II.2. Simbol Activity Diagram**


Gambar	Nama	Keterangan
	Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain..
	Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	Initial Node	Bagaimana objek dibentuk atau diawali.
	Activity Final	Bagaimana objek dibentuk dan dihancurkan
	Fork Node	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran



(Sumber: Andikos, 2019: 39)

### 3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang diambarkan terhadap waktu. Sequence Diagram dapat digambarkan dengan simbol-simbol seperti pada Tabel II.3.

**Tabel II.3. Simbol Sequence Diagram**

Gambar	Nama	Keterangan
	Lifeline	Objek entity, antarmuka yang saling berinteraksi.

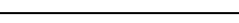

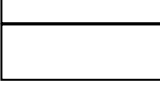

	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

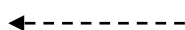
(Sumber : Andikos, 2019: 39)

#### 4. *Class Diagram* (Diagram Kelas)

Class adalah sebuah V yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desai berorientasi objek. Class diagram menggambar struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. Class diagram dapat digambarkan dengan simbol-simbol seperti pada Tabel II.4.

**Tabel II.4. *Class Diagram***

<b>Gambar</b>	<b>Nama</b>	<b>Keterangan</b>
	Generalization	Hubungan dimana objek anak (descendent) erbagai perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor).
	Nary Association	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	Class	Himpunan dari objek-objek yang berbagai atribut serta operasi yang sama.
	Collaboration	Deskripsi dari urutan aksi-aksi yang ditampilkan system yang menghasilkan suatu hasil yang terukur bagi suatu actor
	Realization	Operasi yang benar-benar



		dilakukan oleh suatu objek.
----->	Depedency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
_____	Assocation	Apa yang menghubungkan antara objek satu dengan objek lainnya.

(Sumber: Andikos, 2019: 39)

**Tabel II.5. Multiplicity Class Diagram**

<b>Multiplicity</b>	<b>Penjelasan</b>
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

(Sumber: Ade Hendini; 2016)