

BAB II

TINJAUAN PUSTAKA

II.1 Penelitian Terdahulu

Pada penelitian terkait yang dilakukan oleh Usman Ependi, Suyanto; (2016) dengan judul penelitian **Implementasi Location Based Service Pada Aplikasi Mobile Pencarian Halte BRT Transmusi Palembang**. Perkembangan teknologi saat ini seperti teknologi *mobile*, Salah satunya teknologi *smartphone* Android, dengan adanya kemajuan teknologi Aplikasi ini dapat melakukan pencarian lokasi layanan *halte BRT* menggunakan *smartphone* android.

Pada penelitian terkait yang dilakukan oleh Dona Marcelina^{[1]*}, Evi Yulianti^[2]; (2020) dengan judul penelitian **Aplikasi Pencarian Rute Terpendek Lokasi Kuliner Khas Palembang Menggunakan Algoritma *Euclidean Distance* dan A*(Star)**. pencarian rute terpendek dengan menggunakan estimasi nilai *cost* terkecil untuk mencapai tujuan yang ingin dicapai, dan memiliki nilai heuristik yang digunakan sebagai dasar pertimbangan. Hasil akhir penelitian menunjukkan bahwa algoritma yang dilakukan memiliki tingkat akurasi tinggi dalam menentukan rute terpendek lokasi kuliner palembang yaitu Nilai *Mean absolute percentage error (MAPE)* sebesar 4,4%.

Pada penelitian terkait yang dilakukan oleh Nopriyanus V Y R Bani, Ahmad Faisol Faisol, F. X Ariwibisono; (2020), dengan judul penelitian **APLIKASI PENCARIAN TEMPAT WISATA MENGGUNAKAN METODE LOCATION ¹¹ BASED SERVICE PADA WILAYAH KABUPATEN TIMOR TENGAH SELATAN BERBASIS ANDROID** Sistem yang akan dibangun pada penelitian ini yaitu implementasi *Location Based Services* pada perangkat android dimana pengguna dapat mencari dan menemukan lokasi pariwisata yang ada di dalam aplikasi yang akan dikembangkan.

Pada penelitian terkait yang dilakukan oleh oleh Canggih Ajika Pamungkas ;(2019) dengan judul penelitian **APLIKASI PENGHITUNG JARAK KOORDINAT BERDASARKAN LATITUDE DAN LONGITUDE DENGAN METODE EUCLIDEAN DISTANCE DAN METODE HAVERSINE**. Proses penelitian yang telah dilakukan dalam pengembangan aplikasi *aplikasi* penghitung jarak berbasis android, dapat ditarik kesimpulan yaitu bahwa Pengujian menunjukkan hasil sama saat perhitungan jarak antara metode *EuclideanDistance* dan metode *Haversine*. Rekapitulasi hasil pengujian dengan *metode black box*, metode ini digunakan untuk mengetahui apakah sistem berfungsi sesuai dengan yang diharapkan. Dalam testing menunjukkan aplikasi telah benar dan tidak memiliki kesalahan baik dari segi logika maupun fungsi dan layak untuk diimplementasikan.

Pada penelitian terkait yang dilakukan oleh Edi Budiman ; (2016) dengan judul

penelitian **Pemanfaatan Teknologi *Location Based Service* Dalam Pengembangan Aplikasi Profil Kampus Universitas Mulawarman Berbasis Mobile.**

Pada penelitian terkait yang dilakukan oleh Canggih Ajika Pamungkas ; (2019) dengan judul **Aplikasi penghitung jarak koordinat berdasarkan latitude dan longitude dengan metode Euclidean distance dan metode haversine.**

Berdasarkan proses dan analisa dari penelitian ini maka dapat diambil kesimpulan sebagai berikut:

1. Pemanfaatan teknologi Location Based Service dalam pengembangan aplikasi profil kampus universitas mulawarman memadukan teknologi Geographic Information System, Internet Service, dan Mobile Devices ini membantu memberikan informasi profil dan lokasi fakultas serta fasilitas-fasilitas di Universitas Mulawarman secara mudah, cepat dan akurat.
2. . Aplikasi yang dikembangkan dapat berjalan dengan baik pada berbagai macam tipe perangkat Android maupun PC dan Laptop dengan memanfaatkan emulator android, sehingga pengguna diberikan kenyamanan dan tidak terpaku pada satu macam perangkat android saja

II.2 Aplikasi

Secara istilah pengertian aplikasi adalah suatu subkelas perangkat lunak computer yang memanfaatkan kemampuan computer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak system yang mengintegrasikan berbagai kemampuan computer, tapi tidak

secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. (Andri Novriadi Lomboan, dkk; 2018)

II.3 Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi.”.Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri. Pada awalnya dikembangkan oleh *Android Inc*, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian dibeli oleh *Google Inc*. Untuk pengembangannya, dibentuklah *Open Handset Alliance* (OHA), konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

. 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*. Ada beberapajenis versi android termasuk *Android Honeycomb*, *Android Cream Sandwich*, *Android Jelly Bean*, *Android Kitkat*, *Adroid Lollipop*, dan *Android Oreo*. (Andri Novriadi Lomboan, dkk; 2018)

II.4 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 mudah menyesuaikan ke platform baru. *Android* terdiri dari satu tumpukan yang lengkap, mulai dari *boot loader*, *device driver*, dan fungsi-fungsi pustaka, hingga perangkat lunak API (*Application Programming Interface*), termasuk aplikasi SDK (*Software Development Kit*). *Android* bukanlah satu perangkat tertentu, melainkan sebuah *platform* yang dapat digunakan dan diadaptasikan untuk mendukung berbagai 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.5 *Firestore Realtime Database*

Firestore Realtime Database sendiri merupakan basis data berbasis *cloud* NoSQL yang menyinkronkan data di semua klien secara *realtime*, dan menyediakan fungsionalitas *offline*. Data yang diinputkan disimpan ke dalam *database Realtime* sebagai JSON. Semua klien yang terhubung dan saling berbagi dalam satu waktu,

secara otomatis akan menerima pembaruan dengan data terbaru(Kurniawati dan Lukman Bacthiar ; 2020 : 59).

II.6 Pendidikan Anak Usia Dini (PAUD)

Pendidikan anak usia dini (PAUD) adalah suatu upaya pembinaan yang ditujukan kepada anak sejak lahir sampai dengan usia enam tahun yang dilakukan melalui pemberian rangsangan pendidikan untuk membantu pertumbuhan dan perkembangan jasmani dan rohani agar anak memiliki kesiapan dalam memasuki pendidikan lebih lanjut (UU Nomor 20 Tahun 2003 Bab I Pasal 1 ayat 14).

Penyelenggaraan Pendidikan Anak Usia Dini dapat dilakukan dalam bentuk formal, nonformal dan informal. Setiap bentuk penyelenggaraan memiliki kekhasan tersendiri. Penyelenggaraan pendidikan anak usia dini pada jalur formal adalah Taman Kanak-kanak (TK) atau RA dan lembaga sejenis. (Tatik Arianti, dkk; 2019).

II.8 Database

Pengertian database menurut Jogiyanto dalam buku analisis & desain 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 danKarolina Talia Sela ; 2018).

II.9 UML (*Unified Modeling language*)

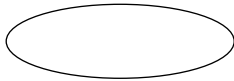
Winda Aprianti & Umi Maliha (2016) Unified Modeling Language (UML) adalah standarisasi bahasa pemodelan untuk membangun perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek. Diagram-diagram yang digunakan pada UML antara lain adalah class diagram, object diagram, use case diagram, activity diagram, dan sequence diagram.

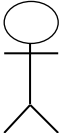
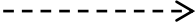
1. *Use Case Diagram*

Use Case diagram merupakan pemodelan untuk kelakuan (*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 Diagram*

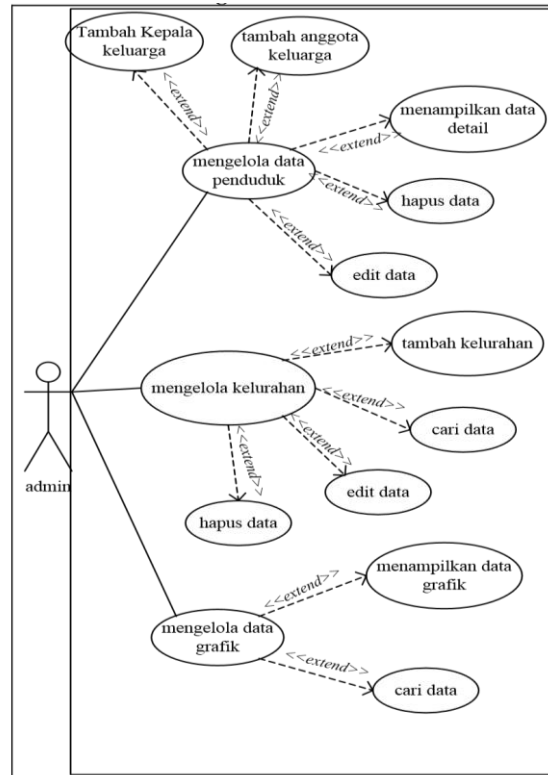
Gambar	Keterangan
	<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>Use</i>

	<i>Case.</i>
	Aktor adalah <i>abstraction</i> 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 <i>Use Case</i>, tetapi tidak memiliki <i>control</i> terhadap <i>Use Case</i>.
	Asosiasi antara aktor dan <i>Use Case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>Use Case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i>, merupakan di dalam <i>Use Case</i> lain (<i>required</i>) atau pemanggilan <i>Use Case</i> oleh <i>Use Case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.

←-----	<i>Extend</i> , merupakan perluasan dari <i>Use Case</i> lain jika kondisi atau syarat terpenuhi.
--------	---

(Sumber : Ade Hendini; 2016)

Contoh kasus;



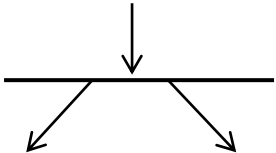
(Sumber: Winda Aprianti & Umi Maliha 2016:24)

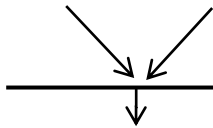
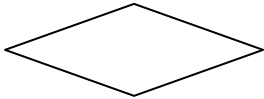
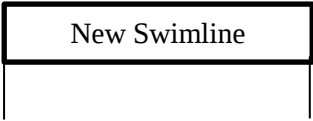
Use Case diagram yang disajikan pada Gambar mendeskripsikan interaksi aktor, yaitu admin sistem informasi data penduduk yang dapat mengelola data penduduk, mengelola kelurahan, dan mengelola data grafik. Pengelolaan data penduduk meliputi tambah kepala keluarga, tambah anggota keluarga, menampilkan, mengedit, dan menghapus data. Sedangkan pengelolaan kelurahan meliputi tambah, cari, *edit*, dan hapus data kelurahan.

2. 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 :

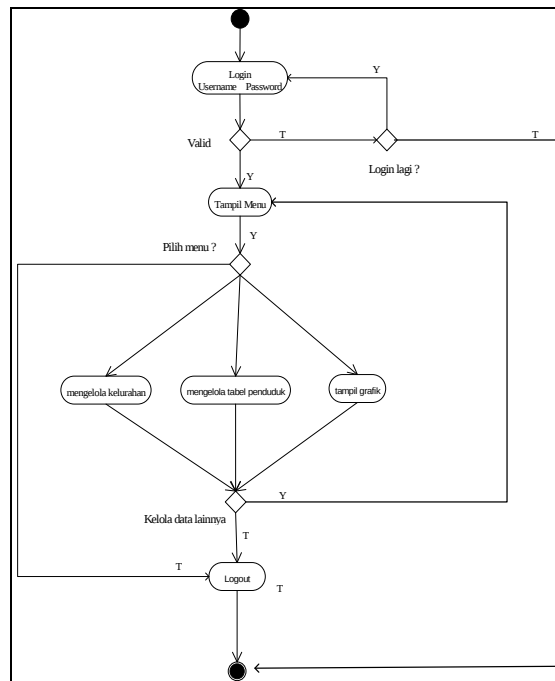
Tabel II.2. Simbol Diagram Aktivitas

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk

	menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

(Sumber : Ade Hendini; 2016)

Contoh kasus;



(Sumber: Winda Aprianti & Umi Maliha 2016:25)

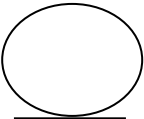
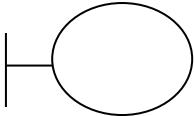
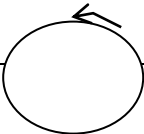
Activity diagram yang diilustrasikan pada Gambar merupakan diagram yang

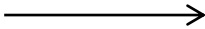
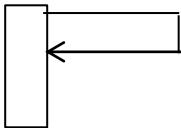
menggambarkan *workflow* (aliran kerja) dari Sistem Informasi Kepadatan Penduduk di Setiap Kelurahan atau Desa pada Badan Pemberdayaan Masyarakat dan Pemerintah Desa (BPMPD) Studi Kasus pada Kecamatan Bati-Bati Kabupaten Tanah Laut.

3. Diagram Urutan (*Sequence Diagram*)

Diagram urutan menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam diagram urutan, yaitu

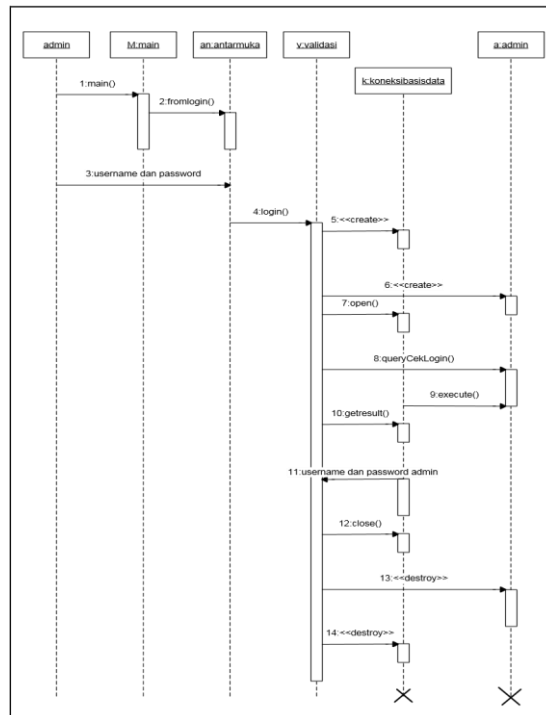
Tabel II.3. Simbol Diagram Urutan

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>formentry</i> dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi

	yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Ade Hendini; 2016)

Contoh kasus :



(Sumber:Winda Aprianti& Umi Maliha 2016:25)

Urutan proses pada *Sequence login* pada Gambar dimulai dari admin sebagai pengguna yang masuk ke antarmuka untuk masuk ke *form login*. Setelah itu *admini* memasukkan *username* dan *password* menuju validasi untuk *login* setelah itu data dikirim ke *database*.

4. 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 tanggung jawab entitas yang menentukan perilaku sistem.

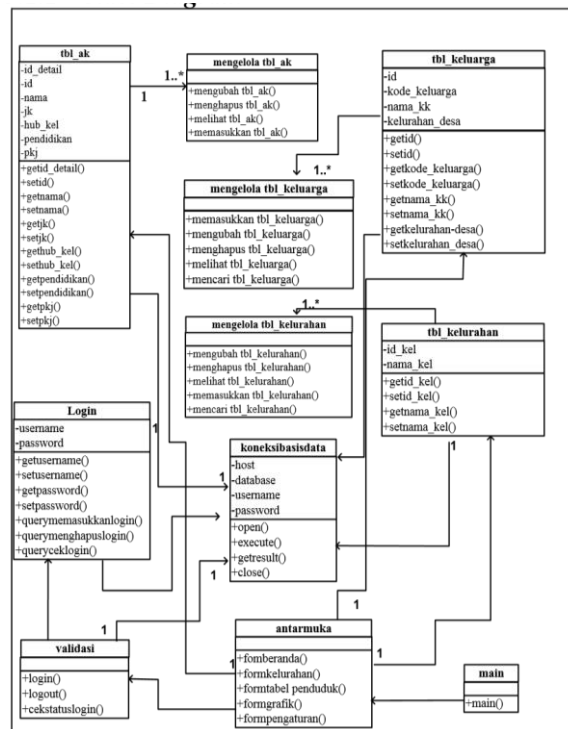
Class diagram merupakan diagram statis dari suatu aplikasi. Class Diagram tidak hanya digunakan untuk memvisualisasikan, menggambarkan, dan mendokumentasikan berbagai aspek sistem tetapi juga untuk membangun kode eksekusi (*executable code*) dari aplikasi perangkat lunak (Munawar ; 2018 : 101).

Tabel II.4. Simbol *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 : Ade Hendini; 2016)

Contoh kasus:



(Sumber: Winda Aprianti & Umi Maliha 2016:24)

Class diagram disajikan pada Gambar 1 terdiri dari 11 kelas yang meliputi kelas *Main*, *Antarmuka*, *login*, *KoneksiBasisData*, *Validasi*, *tbl_keluarga*, *mengelola tbl_keluarga*, *tbl_kelurahan*, *mengelola tbl_kelurahan*, *tbl_ak*, dan *mengelola tbl_ak*

