

BAB II

TINJAUAN PUSTAKA

2.1. Teori Dasar

2.1.1. Pengertian Sistem

Dalam Hamidini, (2017), menyatakan bahwa “sistem ialah sekumpulan elemen bersifat data, program jaringan yang saling berkaitan, sumber daya manusia, teknologi, hardware dan yang berinteraksi secara keseluruhan dalam menggapai objek khusus yang sama.”

Menurut Moch.Irfan, (2014), “Sistem ialah sekumpulan yang berasal dari beberapa bagian terkait yang bekerja sama dan secara bersama-sama mencapai tujuan sistem.”

Muhamad Muslihudin, (2016), mendefinisikan sistem adalah “Komponen atau jaringan sekelompok program, mereka saling berhubungan dan membentuk jaringan dalam menggapai tujuan tertentu.”

Sesuai dengan yang telah diuraikan ahli-ahli di atas, maka penulis bisa menarik kesimpulan sebagai berikut: Konsep suatu sistem merupakan kumpulan atau sekelompok komponen atau jaringan yang saling berinteraksi, bergantung dan mempunyai hubungan, serta dapat bekerja sama dalam meraih tujuan.

2.1.2. Pengertian Informasi

Pada Pratama, (2014), Informasi merupakan hasil olahan knowledge dari satu atau lebih sumber, lantas dikelola untuk beri tambahan nilai, arti dan manfaat.

Dalam Moch.Irfan, (2014), Informasi ialah sejenis data atau objek yang diproses sebelumnya yang dapat diklasifikasikan dengan benar untuk

membuatnya bermakna bagi penerima, dan kemudian bagi penerima adalah pengetahuan tentang masalah tertentu, yang membantu membuat keputusan yang tepat.

Menurut Djahir, (2014), menjelaskan bahwa informasi ialah hasil berasal dari pengolahan knowledge menjadi wujud tertentu, wujud yang lebih bermanfaat bagi penerimanya, yang melukiskan peristiwa aktual dan bisa digunakan sebagai alat pengambilan ketetapan.

Sesuai dengan pemikiran pakar yang telah dikutip, bisa diartikan informasi ialah data yang telah diproses, terstruktur dan disusun dari berbagai sumber jadi wujud yang sangat perlu bagi pengguna dan penerima informasi, yang dimaksudkan untuk pengambilan ketetapan saat ini dan berkelanjutan yang akan mampir.

2.1.3. Pengertian Sistem Informasi

Dalam Mahatmyo, (2014). Sistem informasi ialah rangkaian prosedur resmi yang mana informasi telah terkumpul, diproses dan dikomunikasikan informasi kepada pemakai.

Pada Pratama, (2014) menjelaskan bahwa proses informasi adalah kombinasi berasal dari empat bagian utama yang meliputi software, hardware, infrastruktur serta sumber kekuatan manusia yang telah dilatih. Keempat hal tersebut dihubungkan dengan membentuk proses yang sanggup mengubah knowledge jadi informasi yang berguna.

Sesuai dengan pengartian di atas maka peneliti membuat kesimpulan konsep sistem informasi ialah sistem yang memberikan informasi manajemen pada saat

pengambilan keputusan tentang menjalankan suatu bisnis, dimana sistem tersebut merupakan gabungan antara SDM, teknologi dan proses yang terorganisir.

2.1.4. Pencarian Informasi

Pengumpulan informasi adalah harapan untuk belajar lebih banyak tentang sesuatu, seseorang atau masalah. Ini termasuk mencari informasi secara mendalam, yang berada di luar cakupan pertanyaan biasa, atau di luar apa yang diperlukan untuk pekerjaan itu, termasuk menggali informasi yang akurat. Pada saat yang sama, proses pencarian informasi ialah aktivitas mengumpulkan informasi menjadi hal yang berguna, lalu diserap oleh seseorang dan dijadikan sebagai pengetahuan. Dapat dilihat bagaimana teori kognisi menjadi bagian dari proses interaksi antara pengguna dan sistem informasi, serta bagaimana struktur kognitif pengguna berubah melalui informasi yang ditemukan (Paliwahet, I. N. S., Sukarsa, I. M., & Putra, 2017).

2.2. Teori Khusus

2.2.1. Metode *Haversine*

Metode *Haversine* berguna dalam menghitung jarak geografis pada *node* pada permukaan lintang bumi (*longitude*) dan bujur (*latitude*). Sebagai variabel masukan, rumus Haversine navigasi persamaan penting yang memilih jarak lingkaran pada dua titik pada permukaan bola (Bumi), berdasarkan lintang dan bujur. *Haversine* terdapat banyak keunggulan, dibandingkan penghitungan jarak tanah lainnya, seperti penghitungan yang sederhana dan akurat serta tingkat kesalahan laju analisis yang rendah (Esenbuğa, Ö. G., Akoğuz, A., Çolak, E., Varol, B., & Erol, 2016). Beraneka ragam pemakaian algoritma *haversine* seperti

mencari rute sekolah pada penelitian (Dauni, P., Firdaus, M. D., Asfariani, R., Saputra, M. I. N., Hidayat, A. A., & Zulfikar, 2019), mencari masjid pada penelitian masjid (Setyorini, I., & Ramayanti, 2019), dan mencari *service center* pada penelitian (Putra, D., Daniawan, B., Suwitno, S., & Wijaya, 2019).

Dengan pemikiran bumi itu bulat prima, R6,371 km serta di wilayah dua titik terhadap lintang dan bujur (koordinat bola) berturut-turut ialah Lon1, Lat1 dan Lon2, Lat2 sehingga bisa disimpulkan rumus *Haversine* adalah sebagai berikut.

$$x = (lon2 - lon1) \cos\left(\frac{lat1+lat2}{2}\right);$$

$$y = (lat2 - lat1);$$

Rumus 2.1 Rumus Haversine

$$d = \sqrt{(x^2 + y^2)}R$$

Keterangan :

x : *Longitude* (Lintang)

y : *Lattitude* (Bujur)

d : Jarak

R : Radius Bumi =6.371 km

1⁰ : 0.0174532925 radian.

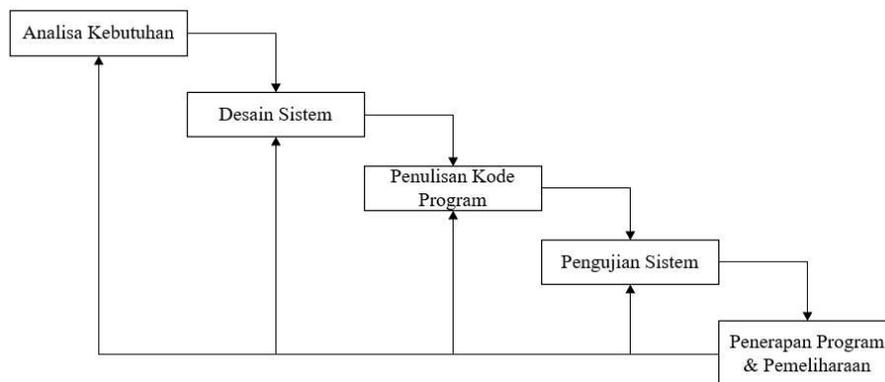
2.2.2. System Development Life Cycle (SDLC)

Dalam Pressman (2015), Model siklus hidup pengembangan sistem ini disebut juga model waterfall atau siklus hidup klasik, yaitu metode sekuensial sistematis.

Ada sebagian model SDLC. Air terjun adalah model yang terlampaui kondang dan banyak digunakan. Dengan menggunakan siklus kaskade SDLC,

sistem pembangunan terbagi menjadi beberapa bagian langkah. Dalam sistem besar, tiap-tiap langkah dilaksanakan oleh tim yang tidak serupa:

1. Analisis sistem, yakni kesimpulan alur pekerja manajemen yang masih berlangsung.
2. Menentukan keperluan sistem, yakni pembahasan rinci tentang apa yang diperlukan dalam mengembangkan sistem dan mengembangkan rancangan yang berhubungan desain sistem.
3. Desain sistem, yakni merancang alur kerja didalam pengelolaan dan desain perangkat lunak yang dibutuhkan untuk membangun sistem TI.
4. Pengembangan sistem, yakni step pembuatan sistem IT bersama penulisan program yang diperlukan.
5. Pengujian sistem, yakni untuk menguji sistem yang sudah siap apakah berjalan dengan baik.
6. Penerapan serta pemeliharaan program, yakni harus melakukan implementasi dan pemeliharaan sistem yang sudah siap dipakai.



Gambar 2.1 Metode *Waterfall*

Sumber: Data Olahan Peneliti, (2021)

2.2.3. Unified Modelling Language (UML)

Dalam Nugroho, (2015) Pernyataan itu menyebutkan bahwa bersama dengan UML ini diinginkan mampu mengurangi kebingungan di dalam bahasa pemodelan yang berlangsung di lingkungan industri, dan juga diinginkan untuk memecahkan persoalan notasi UML dan mekanisme type saat pertukaran.

Menurut A.S, Rosa dan Shalahuddin, (2013) Ditegaskan: “UML (Unified Modeling Language) ialah bahasa standar telah banyak dipakai oleh industri dalam menetapkan prasyarat, pemikiran, desain, serta gambaran arsitektur pemrograman berorientasi objek”. UML ialah "bahasa" yang sudah berubah ke dalam bentuk visualisasi, standar industri untuk desain sistem dan perangkat lunak dokumentasi. UML sediakan standar dalam perancangan *style* sistem.

a. Tujuan UML

- 1) Ekspresif bahasa pemodelan visual menyediakan siap pakai untuk mengembangkan model untuk pemahaman umum.
- 2) Menyajikan bahasa pemodelan yang terlepas dari beragam bahasa pemrograman serta rekayasa proses.
- 3) Menggabungkan praktek termasuk dalam pemodelan.

b. Diagram UML

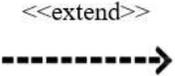
Di dalam UML terdapat beberapa jenis diagram yaitu:

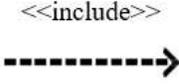
1) Use Case Diagram

Merupakan gambaran fungsionalitas yang diharapkan berasal dari sistem. Penekanannya ialah pada "apa" dari sistem, bukanlah "bagaimana". Kasus penggunaan mewakili interaksi antara peserta dan sistem. Kasus

penggunaan adalah tugas khusus, seperti masuk ke sistem, membuat atau membuat daftar belanja, dll. Peranannya ialah kumpulan manusia, teknologi, dan sistem yang bekerjasama dalam mengerjakan tugas tertentu. Simbol diagram *use case*, antara lain:

Tabel 2.1 Simbol *Use Case Diagram*

No	Nama	Gambar	Keterangan
1	<i>Use Case</i>		Fungsi yang di sediakan oleh proses sebagai unit untuk bertukar pesan pada entitas atau aktor; kebanyakan dinyatakan sebagai kata kerja di dalam kalimat pemakaian nama masalah awal.
2	Aktor / <i>Actor</i>		Orang, proses, atau sistem lain yang berinteraksi bersama dengan sistem Info dibuat, jadi meskipun lambang aktor adalah gambar orang, belum tentu aktor; biasanya diberikan sebagai kata benda dalam kata-kata di mulai bersama dengan nama aktor.
3	Asosiasi / <i>Association</i>		Komunikasi antara aktor dan kasus penggunaan yang terlibat kasus penggunaan atau kasus penggunaan untuk berinteraksi dengan aktor.
4	Ekstensi / <i>Extend</i>		Kasus penggunaan tambahan untuk kasus penggunaan di mana kasus penggunaan tambahan dapat ada dengan sendirinya, bahkan tanpa kasus penggunaan tambahan; mirip dengan prinsip pewarisan dalam pemrograman
5	Generalisasi / <i>Generalization</i>		Hubungan generalisasi dan spesialisasi (-Jenderal khusus) antara dua kasus penggunaan di mana fungsi adalah fungsi yang lebih umum daripada yang lain.

6	Menggunakan / <i>Include / uses</i>		Hubungan dengan penggunaan-kasus kasus penggunaan tambahan di mana penggunaan kasus penggunaan ditulis kasus memerlukan itu untuk melakukan fungsinya atau sebagai syarat pemicu untuk kasus penggunaan. Sertakan berarti bahwa kasus penggunaan yang ditambahkan akan selalu dipanggil saat kasus penggunaan tambahan dijalankan.
---	--	---	--

Sumber: Data olahan peneliti, (2021)

2) *Activity Diagram*

Merupakan gambaran aliran kegiatan yang berbeda pada sistem perancangan, bagaimana setiap aliran dimulai, keputusan itu bisa terjadi, dan bagaimana aliran ujungnya. Simbol diagram aktivitas, termasuk.:

Tabel 2.2 Simbol *Activity Diagram*

No.	Nama	Gambar	Keterangan
1.	Activity		Tindakan yang dilakukan oleh sistem, aksi umumnya diawali bersama dengan kata kerja.
2.	Decision		Asosiasi perdagangan terkecuali tersedia lebih dari satu pilihan.
3.	Status Awal		Keadaan awal dari aktivitas sistem, aktivitas diagram punya status awal.
4.	Status Akhir		Status akhir yang dilakukan oleh sistem, diagram aktivitas memiliki status akhir.
5.	Join		Bergabung dengan federasi yang menggabungkan lebih dari satu kegiatan ke dalam satu.

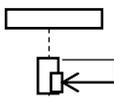
Sumber: Data olahan peneliti, (2021)

3) *Sequence Diagram*

Merupakan gambaran hubungan pada objek dalam dan di kurang lebih sistem (pemakai dan layar) dalam bentuk pesan yang disajikan selama

waktu. Diagram kronologis terdiri dari dimensi vertikal (waktu) juga dimensi horizontal (objek yang berkaitan). Masing-masing pesan yang dikirim dapat merespon (kembali) terhadap skenario yang terancang dalam diagram *use case*. Hubungan yang timbul dapat berupa instance dari objek atau metode statis kelas. Di bawah ini ialah simbol diagram urutan.

Tabel 2.3 Simbol *Sequence Diagram*

No	Nama	Gambar	Keterangan
1.	Aktor		Aktor adalah pengguna sistem, pengguna dapat berarti orang, mesin, atau sistem atau subsistem lain dari model apa pun yang berinteraksi dengan sistem batas sistem.
2.	Message		Menunjukkan arus informasi atau mengontrol transaksi antar item
3.	Self-Message		Mencerminkan proses baru atau metode panggilan operasi garis hidup. Ini adalah spesifikasi pesan yang biasanya ada dalam diagram urutan
4.	Boundary		Batas adalah objek stereotip yang memodelkan batas sistem. Ini membutuhkan layar antarmuka pengguna
5.	Control		Kontrol adalah objek stereotip untuk mengendalikan atau mengatur entitas
6.	Entity		Merupakan tabel dalam database, itu adalah model penyimpanan data yang dapat menangkap dan menyimpan informasi dalam sistem

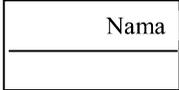
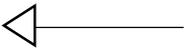
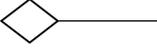
Sumber: Data olahan peneliti, (2021)

4) *Class Diagram*

Diagram kelas spesifikasi, jika sebuah organisasi, maka akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan

desain berorientasi objek. Kelas menjelaskan status proses dan menyediakan layanan untuk status operasi.

Tabel 2.4 Simbol-simbol *Class Diagram*

No	Nama	Gambar	Keterangan
1.	Kelas		Kelas pada stuktur sistem
2.	Properti		Atribut kelas yang mewakili batas nilai
3.	<i>Association</i>		Hubungan antara kelas memiliki akal sehat, dan asosiasi biasanya disertai dengan banyaknya suatu
4.	<i>Directed Association</i>		Hubungan antara kelas dengan satu jenis makna digunakan oleh yang lain. Biasanya disertai dengan keserbaragaman
5.	<i>Dependency</i>		Arti hubungan antar kelas dan ketergantungan antar kelas
6.	Generalisasi		Hubungan antara kelas dengan makna generalisasi-spesialisasi (generalisasi khusus)
7.	Agregasi		Hubungan antar kelas dan arti semua bagian (whole parts)

Sumber: Data olahan peneliti, (2021)

2.2.4. Android

Hubungan antara kelas dalam hal seluruh anggota (semua bagian) Android ialah sistem operasi mobile berbasis Linux termasuk sistem operasi, *middleware*, serta aplikasi. Android mengimbuhkan pengembang platform terbuka dalam mengembangkan aplikasi. Awalnya, Google Inc. membeli Android Inc., Lalu melakukan pengembangan perangkat lunak pada *smartphone*. Selanjutnya,

pengembangan Android, dibentuk *Open Handset Alliance*, konsorsium berasal dari 34 perusahaan *hardware*, *software*, serta telekomunikasi. Selama rilis pertama berasal dari Android pada 5 November 2007, Android dan *Open Handset Alliance* menginformasikan bahwa mereka menopang perkembangan *open source* dalam perangkat *mobile*. Di sisi lain, Google sediakan kode Android di bawah lisensi *Apache*, lisensi *software*, juga platform terbuka dalam perangkat *mobile* (Juansyah, 2015).



Gambar 2.2 Logo Android

Sumber: (Wikipedia, 2020)

Berikut ini perkembangan versi dari tahun ketahun, berikut ini adalah perkembangan versi Android:

- a) Android tahun 2008 versi 1.0 (Apple Pie)
- b) Android tahun 2008 versi 1.1 (Banana Bread)
- c) Android tahun 2009 versi 1.5 (Cupcake)
- d) Android tahun 2009 versi 1.6 (Donut)
- e) Android tahun 2009 versi 2.0-2.1 (Eclair)
- f) Android tahun 2010 versi 2.2 (Froyo)
- g) Android tahun 2010 versi 2.3-2.3.7 (Gingerbread)
- h) Android tahun 2011 versi 3.1-3.2 (Honeycomb)

- i) Android tahun 2011 versi 4.0.3-4.0.4 (Ice Cream Sandwich)
- j) Android tahun 2012 versi 4.1.x,4.2.x,4.3.x (Jelly Bean)
- k) Android tahun 2013versi 4.4 (KitKat)
- l) Android tahun 2014 versi 5.x (Lollipop)
- m) Android tahun 2015 versi 6.0 (Marshmallow)
- n) Android tahun 2016 versi 7.0 (Nougat)
- o) Android tahun 2017 versi 8.0 (Oreo)
- p) Android tahun 2018 versi 9.0 (Pie)
- q) Android tahun 2019 versi 10.0 (Android 10)

2.2.4.1. Kelebihan Android

- a. Transisi yang lebih baik dan bisa digunakan untuk mendukung aplikasi android multitasking. Dalam android honey comb, pengguna dapat dengan mudah beralih antar-aplikasi dengan menekan ikon di layar ponsel.
- b. Memiliki daya tampung yang lebih bagus sehingga dimana pemakai dapat menggunakan widget tanpa perlu membuka aplikasi. Misalnya, Gmail email widget Google, pemakai tidak harus *open* aplikasi Gmail demi ingin lihat isinya.
- c. Android bisa disebut dengan *user friendly* karena sangat mudah untuk digunakan. Bahkan untuk pengguna yang belum pernah menggunakannya tidak memerlukan waktu yang lama untuk mempelajarinya.
- d. Android memiliki desain tampilan yang unggul dan menarik untuk dilihat sehingga mudah untuk dimengerti.

2.2.4.2. Kekurangan Android

- a. Selalu terhubung ke Internet. Kebanyakan Android membutuhkan simultan aktivasi yang terkoneksi internet. Artinya pengguna android wajib menyiapkan paket GPRS, sesuai dengan keperluan dan GPRS akan terus menerus dihidupkan, sehingga baterai akan lebih boros. memang mudah dan gratis untuk mendapatkan aplikasi di ponsel android, namun hasil di setiap aplikasi selalu ditampilkan iklan (Sulihati, 2016).
- b. Pemakai android selalu mengalami cepatnya kehabisan baterai. Penyebab utama dari melemahnya baterai ponsel yaitu paket data yang selalu menyala serta aplikasi terus berjalan. Solusi yang tepat untuk menanganin masalah ini adalah dengan memangkas aktivitas pada ponsel android.
- c. Seringnya menghadapi ponsel yang lamban atau lelet. Ini berhubungan dengan detail ponsel android karena pemakaian aplikasi yang tidak cocok dengan Ram atau kebanyakan membuka aplikasi. Oleh karena itu, ada baiknya menggunakan aplikasi yang sesuai dengan procesor android.

2.2.5. *Blackbox Testing*

Uji *black box* ialah strategi pengujian yang hanya berfokus pada faktor fungsional dan spesifikasi perangkat lunak.

Pengujian *black box* dapat mengidentifikasi jenis-jenis kesalahan, termasuk kesalahan fungsional, kesalahan antarmuka, kesalahan pemodelan data, juga kesalahan mengakses sumber data eksternal. (Agarwal, B. B., Tayal, S. P., & Gupta, 2010).

Uji *black box* yang dibutuhkan untuk menambah kualitas dan kehandalan *software*. Semakin tinggi kompleksitas *software*, maka akan diperlukan uji yang lebih rinci. Hal ini dapat dilihat dalam proses memilih jenis input yang memungkinkan untuk pengujian.

2.3. Penelitian Terdahulu

Berikut ini uraian penelitian yang dilakukan oleh peneliti terdahulu dengan beragam judul dan hasil :

1. Penelitian yang dilakukan oleh Alam, C. N., Manaf, K., Atmadja, A. R., & Aurum, (2016) dengan judul “*Implementation of Haversine Formula for Counting Event Visitor in The Radius Based on Android Application*” dan ISSN 978-1-4673-8443-8. Layanan berbasis lokasi (LBS) adalah satu dari teknologi baru yang memperoleh posisi dikehadiran geografis perangkat seluler. Kehadiran teknologi LBS dapat digunakan untuk pemetaan informasi suatu daerah seperti adanya acara tersebut. Keberadaan acara dipengaruhi oleh banyaknya pengunjung acara tersebut, karena dapat menarik minat masyarakat untuk menghadiri acara tersebut. Rumus Haversine ialah metode yang dipergunakan dalam menghitung jarak dari dua titik yang berbeda. Pada penelitian ini, formula Haversine diterapkan untuk menghitung jarak antara pengguna dengan serangkaian peristiwa yang ada. Untuk menentukan keberadaan pengunjung dilakukan dengan menghitung hasil formula haversine dengan area radius masing-masing event tersebut. Dari perhitungan, jika pengguna berada di area radius, jumlah pengunjung acara akan dihitung dan diperbarui. Dalam penelitian ini, kami menerapkan metode ini dalam aplikasi android. Oleh karena itu, formula haversine dapat digunakan untuk perhitungan

jarak dan dapat diimplementasikan untuk menentukan jumlah pengunjung yang masuk ke dalam acara.

2. Penelitian yang dilakukan oleh Khairina, D. M., Ramadhinata, F. W., & Hatta, (2017) dengan judul “Pencarian Lokasi Jalur Nugraha Ekakurir (JNE) Terdekat Menggunakan Haversine Formula (Studi Kasus Kota Samarinda)” dan ISSN 2085-4218. Jasa kurir ialah aktivitas yang mempromosikan jasa kirim, ambil, serta antar barang sesuai dengan tujuan. Keperluan jasa kurir yang terus bertambah, melahirkan banyak perusahaan yang ingin mendirikan jasa kirim barang, seperti JNE yang merupakan perusahaan besar dan mempunyai cabang yang lebar di Indonesia, termasuk di Kota Samarinda. Penelitian ini memiliki tujuan untuk mencari letak JNE terdekat berbasis web yang bisa memenuhi informasi serta bisa diperoleh dengan gampang tiap waktunya menggunakan media internet oleh pemakai yang hendak mencari informasi letak JNE dengan pencarian jarak terdekat sesuai lokasi pemakai ke lokasi JNE yang ingin didatangi. Metode Haversine Formula bisa dipergunakan dalam menghitung jarak antara dua titik, sesuai dengan posisi garis lintang latitude dan posisi garis bujur longitude sebagai variabel inputan. Haversine Formula akan diterapkan dalam menilai lokasi JNE terdekat di Samarinda. Kesimpulan yang didapat ialah aplikasi sistem berbasis web ini bisa memberikan solusi untuk pemakai yang mencari informasi lokasi JNE di Samarinda menggunakan pencarian jarak terdekat sesuai dengan lokasi pemakai ke lokasi JNE yang ingin didatangi via media internet.

3. Penelitian yang dilakukan oleh Fauzi, A., Fernando, F., & Raharjo, (2018) dengan judul “Penerapan Metode Haversine Formula Pada Aplikasi Pencarian

Lokasi Tempat Tambah Ban Kendaraan Bermotor Berbasis Mobile Android” dan ISSN 2550-0120. Intinya, dikutip dari survei yang dilakukan Jakarta Transportation Statistics Jakarta - BPS Jakarta menjelaskan pengguna kendaraan bermotor di wilayah DKI Jakarta pada 2014 sebanyak 13.084.372 dengan pertumbuhan tahunan 10, 54% dan ditambah dengan mobil penumpang, mobil beban, bus mobil dan rensusu sebesar 17.523.967 dengan jumlah banyak tentu akan menyebabkan banyak masalah antara cacat dan yang paling ditakuti adalah kebocoran probadi ban kendaraan , untuk memudahkan pengendara motor pribadi dalam mencari tempat pemberian makan ban diperlukan penerapan informasi perbaikan ban kendaraan sehingga pengendara tidak perlu berjalan jauh untuk mencari tempat perbaikan ban padahal kemungkinan posisi pengendara tidak terlalu jauh dengan lokasi yang mereka cari, aplikasi akan menemukan jarak terdekat dari tempat tambah ban yang telah didaftarkan oleh penjual jasa.

4. Penelitian yang dilakukan oleh Sitohang, S., & Tampubolon, (2019) dengan judul “Rancang Bangun Aplikasi e-GISH Pencarian Lokasi Hydrant di Kota Batam” dan ISSN 978-602-52829-1-1. Batam ialah kota industri yang padat dan berpotensi terkena kebakaran. Beraneka ragam sarana prasana yang diperlukan untuk menanggulangi kebakaran sudah tersedia dari pemerintah Kota Batam, seperti *hydrant*. *Hydrant* ialah sarana prasana yang tersedia yang menjadi sumber air ketika dilakukan langkah memadamkan api. Permasalahan yang dialami ialah lokasi *hydrant* yang tidak dipetakan secara baik sehingga kesulitan dalam melakukan pencarian keberadaanya secara tepat. Terutama di Batam masih pada masa membangun infrastruktur yang akhirnya berkemungkinannya pergeseran

letak *hydrant*. Penelitian ini memperoleh tujuan untuk merancang bangun aplikasi e-GISH yang mempermudah pemadaman api saat mencari lokasi *hydrant*. Aplikasi ini didambakan bisa memberi ruang dalam tata-kelola *hydrant* yang lebih baik. Pengumpulan data dilakukan menggunakan teknik wawancara juga observasi langsung ke lapangan. Wawancara bertujuan memperoleh keperluan pihak pemadam untuk membuat aplikasi. Observasi langsung bertujuan mencari koordinat latitude dan longitude *hydrant* di Kota Batam serta diisi ke database aplikasi. Aplikasi e-GISH dibuat berbasis web serta memakai layanan *google map*. Layanan map diperoleh dari *mapcoordinates.net*. Database aplikasi dibuat memakai MySQL. Rancang bangun aplikasi e-GISH didambakan sebagai awal dalam memfasilitasi sistem informasi *hydrant* juga tata kelola *hydrant* yang lebih baik. e-GISH akan sangat menolong pihak pemadam untuk mencari tempat *hydrant* terdekat dengan tempat kebakaran.

5. Penelitian yang dilakukan oleh Rianto, F., & Rahmatulloh, (2019) dengan judul “Implementasi Formula Haversine Dan Komunikasi Data *Real-Time* Menggunakan Websocket Di Sistem Pengawasan Warga Negara Asing” dan ISSN 2406-7857. Warga Negara Asing (WNA) sesudah masuk Indonesia berdasarkan langkah hukum yang berlangsung, sehingga WNA bisa dimanapun. Oleh karena itu, muncullah permasalahan saat pihak pengawasan WNA tidak bisa dihubungi WNA yang sedang memiliki masalah mengenai imigrasi serta tidak tahu di mana letak WNA. Satu dari sekian banyak pemecahan masalah adalah pengimplementasian komunikasi data *real time* dalam mengamati letak keberadaan WNA. Sebelum data ke tempat tinggal tersebut, diperlukan evaluasi

dalam memutuskan tempat tinggal WNA mana terlebih dahulu yang ingin di kunjungi. Mengetahui jarak tempat tinggal WNA bisa dijadikan poin penambah untuk evaluasi. Metode haversine dilakukan implementasi menggunakan cara menghitung jarak antara pihak yang terlibat pengawasan WNA dengan masing-masing alamat tempat tinggal WNA. Hasil uji perbandingan metode haversine menggunakan *google maps* dalam penelitian ini ialah memiliki rata-rata selisih sebesar 0,039 km pada hal jarak sesuai dengan garis lurus. Teknologi komunikasi data *real time* memakai *websocket* pada pengimplementasian dalam sistem pengawasan WNA, berisikan aplikasi berbasis web serta android yang saling terkait ke dalam jaringan *websocket*. Rata-rata profiling menghasilkan kecepatan proses tulis data kurang lebih 15 menit dari *smartphone* ke *firebase realtime database* ialah 1,4 milidetik dengan interval pengiriman data dari *smartphone* 10 detik sekali serta rata-rata hasil kecepatan penerimaan data dari *firebase realtime database* ke sistem pengawasan WNA ialah 0,36 detik dengan interval penerimaan data dari *firebase realtime database* 10 detik sekali.

6. Penelitian yang dilakukan oleh Sumaryo, R. Y., Harsadi, P., & Nugroho, (2020) dengan judul “Implementasi Algoritma Dijkstra Dan Metode Haversine Pada Penentuan Jalur Terpendek Pendakian Gunung Merapi Jalur Selo Berbasis Android” dan ISSN 2620-7532. Pendakian gunung adalah satu dari beberapa olahraga yang memiliki risiko tinggi dan teknik khusus. Banyak faktor yang muncul hilangnya pendaki gunung, misalnya sesat atau keluar jalur akibat dari kondisi alam serta ingin coba trek baru tanpa adanya persiapan yang matang. Sedangkan, informasi mengenai rute pendakian gunung biasanya dapat dari brosur

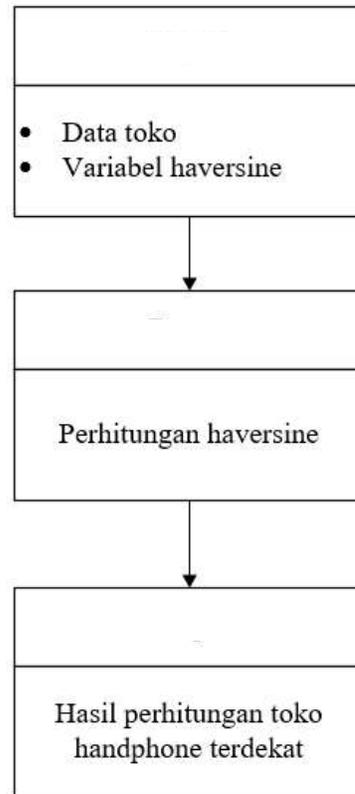
atau peta di *Mount Base Camp*. Lebih lanjut, ada kasus-kasus di mana pendaki gunung wajib mendapatkan jalur terdekat dengan *base camp* terdekat apabila kondisi alam kurang bersahabat. Akibat tidak terdapat begitu banyaknya aplikasi mengenai kartografi gunung serta pentingnya alur terdekat yang jadi latar belakang pengimplementasian algoritma Dijkstra berbasis Android, dapat memberikan keputusan jalur terpendek ke Gunung Merapi melalui Selo. Penelitian ini menggunakan metode algoritma Dijkstra dalam mengambil keputusan tingkat paling pendek juga menggunakan metode Haversine dalam memberi nilai node yang dilewati. Penelitian ini memperoleh tujuan ialah untuk mengetahui jalur terpendek menuju Gunung Merapi melalui Selo. Pemakaian aplikasi Google *map* serta GPS dipergunakan dalam pemetaan juga penemuan koordinat lokasi. Hasil penelitian adalah aplikasi ini bisa dipergunakan dengan tidak sulit memakai ponsel dan uji kelayakan ditunjukkan nilai "Baik" rata-rata 77%.

7. Penelitian yang dilakukan oleh Chandra Husada et al., (2020) dengan judul "Implementasi Haversine Formula untuk Pembuatan SIG Jarak Terdekat ke RS Rujukan COVID-19" dan ISSN 2580-0760. Telah dibuat Sistem Informasi Geografis (SIG) berbasis Haversine formula dalam pencarian lokasi terdekat rumah sakit rujukan penanganan COVID-19 di Kota Semarang. Tujuan penelitian ini, yaitu (1) menentukan jarak terdekat dan membandingkan hasil tersebut dengan hasil perhitungan Find Nearest Tool dan aplikasi Google Maps dan (2) merancang bangun sebuah SIG. Untuk pencapaian tujuan penelitian tersebut, dilakukan melalui (i) pembuatan data primer dan sekunder dan pengolahan data,

(ii) pengukuran tingkat keakuratan dengan Haversine formula, maka perlu dibandingkan terhadap Find Nearest Tool maupun aplikasi Google Maps. Sistem Informasi Geografis dibangun setelah diperoleh hasil perhitungan berdasarkan jarak terdekat dari lokasi user. Berdasarkan penerapan Haversine formula dapat dihasilkan perhitungan jarak dari titik awal lokasi user menuju rumah sakit rujukan. Perbandingan hasil pengukuran jarak lokasi terdekat antara SIG hasil rancangan berbasis Haversine formula dan Find Nearest Tool diperoleh nilai selisih rata-rata sebesar 13 meter, dengan nilai selisih terkecil 3 meter dan nilai selisih terbesar 40 meter. Selain itu terdapat juga selisih antara hasil perhitungan berbasis Haversine formula terhadap hasil pengukuran jarak garis lurus dengan aplikasi Google Maps, dengan nilai selisih terkecil 0 meter dan nilai selisih terbesar 5 meter, sehingga nilai selisih rata-rata sebesar 3 meter. Pembuatan SIG juga telah diperoleh melalui rancang bangun use case, activity, dan class diagram yang dilengkapi dengan user interface. Berdasarkan hasil tersebut dapat disimpulkan, bahwa pembuatan SIG berbasis Haversine formula dapat digunakan sebagai “Sistem Informasi Geografis Pencarian Rumah Sakit Rujukan Penanganan COVID-19 di Kota Semarang” berdasarkan jarak terdekat dari lokasi user.

2.4. Kerangka Pemikiran

Kerangka berpikir yaitu suatu gambaran yang menjelaskan mengenai alur logika berjalannya penelitian secara garis besar. Berikut tahap-tahap yang dilakukan dalam kerangka berpikir penelitian, antara lain:



Gambar 2.3 Kerangka Berpikir

Sumber: Data olahan peneliti, (2021)

1. Input

Ialah satu dari banyaknya proses penelitian yang utama di antarproses lainnya, admin dapat menginput data toko dan variable haversine yang akan digunakan pada sistem yang dibangun.

2. Proses

Tahapan ini dilakukan dengan melakukan perhitungan menggunakan rumus haversine untuk mencari jarak terdekat. Sistem akan menggunakan variable yang dijadikan parameter untuk pencarian toko handphone terdekat.

3. Output

Tahapan ini adalah hasil akhir dari sistem yaitu hasil output dari perhitungan jarak terdekat toko handphone, sehingga user dapat melihat hasil jarak terdekat toko handphone yang dapat dituju oleh user.