

## BAB II

### KAJIAN PUSTAKA

#### A. Kajian Teoritis

##### 1. Metode *Vincenty*

Metode *vincenty* adalah metode perhitungan jarak yang menganggap bumi berbentuk geometri tiga dimensi yang menyerupai bola yang sedikit pipih atau memanjang (Wulansari et al., 2015). Sedangkan, menurut Siregar (2023) “Formula *Vincenty* merupakan metode yang digunakan dalam geodesi untuk menghitung jarak dan azimuth antara dua titik di permukaan *ellipsoid (oblate sferoid)* yang dirumuskan oleh Thaddeus Vincenty (1975)”. Salah satu keunggulan metode *vincenty* adalah dapat memperhitungkan jarak yang presisi bahkan pada jarak yang jauh.

Menurut Panigrahi (2014) Thaddeus Vincenty merancang rumus untuk menghitung jarak *geodesik* antara sepasang titik di permukaan Bumi yang diberikan oleh garis lintang dan garis bujur. Rumus ini menggunakan model elipsoid Bumi yang akurat. Meskipun matematikanya agak sulit, rumus ini mudah untuk diprogram. Rumus *Vincenty* akurat hingga 0,5 mm pada elipsoid yang digunakan.

Metode *Vincenty* memiliki dua pendekatan: metode terbalik (*inverse*) dan metode langsung (*direct*). Untuk menentukan jarak antara dua titik berdasarkan koordinat lintang (*latitude*) dan bujur (*longitude*), digunakan metode terbalik. Sebaliknya, metode langsung dipakai untuk menentukan koordinat titik kedua jika diketahui koordinat titik pertama

beserta jarak dan azimuth awal. Jika diketahui koordinat lintang ( $\Phi$ ) dan bujur ( $L$ ) dari dua titik, yaitu titik 1 ( $\Phi_1, L_1$ ) dan titik 2 ( $\Phi_2, L_2$ ), maka metode terbalik dapat digunakan untuk menghitung jarak elipsoid ( $s$ ), azimuth maju ( $\alpha_1$ ) dan azimuth balik ( $\alpha_2$ ). Proses perhitungan ini bersifat *iteratif* untuk menemukan nilai  $\lambda$  (perbedaan bujur pada bola bantu) hingga mencapai tingkat konvergensi yang diinginkan, misalnya ketika perubahan  $\lambda$  lebih kecil dari 10<sup>-12</sup>.

Menurut Fauzi & Putranto (2025), menghitung jarak menggunakan *vincenty inverse* memiliki langkah-langkah sebagai berikut:

- a. Hitung  $U_1$ ,  $U_2$  dan  $L$ , menggunakan rumus 2.1, 2.2, dan 2.3

$$U_1 = \arctan((1 - f)\tan \Phi_1) \quad (2.1)$$

$$U_2 = \arctan((1 - f)\tan \Phi_2) \quad (2.2)$$

$$L = L_2 - L_1 \quad (2.3)$$

- b. Tentukan nilai awal  $\lambda = L$
- c. Selanjutnya, persamaan-persamaan berikut dievaluasi secara berulang (*iteratif*) sampai nilai  $\lambda$  mencapai konvergensi, dengan menggunakan rumus 2.4 sampai 2.11:

$$\begin{aligned} \sin \sigma & \\ &= \sqrt{(\cos U_2 \sin \lambda)^2 + (\cos U_1 \sin U_2 - \sin U_1 \cos U_2 \cos \lambda)} \end{aligned} \quad (2.4)$$

$$\cos \sigma = \sin U_1 \sin U_2 + \cos U_1 \cos U_2 \cos \lambda \quad (2.5)$$

$$\sigma = \arctan 2 (\sin \sigma, \cos \sigma) \quad (2.6)$$

$$\sin \alpha = \frac{\cos U_1 \cos U_2 \sin \lambda}{\sin \sigma} \quad (2.7)$$

$$\cos^2 \alpha = 1 - \sin^2 \alpha \quad (2.8)$$

$$\begin{aligned} \cos(2\sigma_m) &= \cos \sigma - \frac{2 \sin U_1 \sin U_2}{\cos^2 \alpha} \\ &= \cos \sigma - \frac{2 \sin U_1 \sin U_2}{1 - \sin^2 \alpha} \end{aligned} \quad (2.9)$$

$$C = \frac{f}{16} \cos^2 \alpha [4 + f (4 - 3 \cos^2 \alpha)] \quad (2.10)$$

$$\begin{aligned} \lambda &= L + (1 - C) f \sin \alpha \{ \sigma + C \sin \sigma \\ &\quad [\cos(2\sigma_m) C \cos \sigma (-1 + 2 \cos^2 (2\sigma_m))] \} \end{aligned} \quad (2.11)$$

- d. Setelah nilai  $\lambda$  mencapai konvergensi pada tingkat akurasi yang diinginkan ( $10^{-12}$  setara dengan 0,006 mm), maka langkah selanjutnya adalah melakukan evaluasi, dengan menggunakan rumus 2.12 sampai 2.16:

$$u^2 = \cos^2 \alpha \left( \frac{a^2 - b^2}{b^2} \right) \quad (2.12)$$

$$A = 1 + \frac{u^2}{16384} (4096 + u^2 [-768 + u^2 (320 - 175u^2)]) \quad (2.13)$$

$$B = \frac{u^2}{1024} (256 + u^2 [-128 + u^2 (74 - 47u^2)]) \quad (2.14)$$

$$\begin{aligned} \Delta \sigma &= B \sin \sigma \left\{ \cos(2\sigma_m) + \frac{1}{4} B \left( \cos \sigma [-1 + \right. \right. \\ &\quad \left. \left. 2 \cos^2(2\sigma_m)] - \frac{1}{6} B \cos[2\sigma_m] [-3 + \right. \right. \\ &\quad \left. \left. 4 \sin^2 \sigma] [-3 + 4 \cos^2(2\sigma_m)] \right) \right\} \end{aligned} \quad (2.15)$$

$$s = bA(\sigma - \Delta \sigma) \quad (2.16)$$

Keterangan:

$a = 6.378.137\text{m}$  : panjang sumbu semi-mayor *ellipsoid* (jari-jari di ekuator) dengan standar WGS-84.

$f = \frac{1}{298.257223563}$  : *flattening* (pemipihan) *ellipsoid*.

$b = (1 - f) a$  : panjang sumbu semi-kecil *ellipsoid* (jari-jari di kutub).

$\Phi_1, \Phi_2$  : garis lintang dari titik-titik tersebut.

$U_1, U_2$  : lintang yang dikurangi (lintang pada bola bantu).

$L_1, L_2$  : garis bujur dari titik-titik tersebut.

$L$  : perbedaan bujur dari dua titik.

$\lambda$  : Perbedaan bujur titik-titik pada bola bantu.

$s$  : jarak *ellipsoidal* antara dua titik.

$\sigma$  : pemisahan sudut antar titik.

$\sigma_1$  : pemisahan sudut antara titik dan khatulistiwa.

$\sigma_m$  : pemisahan sudut antara titik tengah garis dan khatulistiwa.

## 2. Sampah

Menurut UU No. 18 Tahun 2008 (dalam Febriyanto & Dewi 2021)

Sampah merupakan sisa-sisa berbentuk padat yang berasal dari kegiatan manusia sehari-hari atau dari proses alam. Namun, tidak semua sampah harus dimusnahkan. Dengan pengelolaan sampah yang benar dan berkelanjutan dapat mengurangi jumlah sampah yang dikirim ke tempat pembuangan sampah, sehingga dapat mengurangi gas rumah kaca dan gas berbahaya lain yang dilepaskan ke atmosfer (Pambudi, 2023). Sampah dapat

dibagi menjadi tiga jenis, yaitu sampah organik (sampah basah), sampah anorganik (sampah kering), dan sampah B3 (bahan berbahaya dan beracun).

Menurut Salim et al. (2020) Sampah adalah masalah penting yang perlu diperhatikan karena merupakan produk tak terhindarkan dari aktivitas manusia. Seiring meningkatnya tingkat konsumsi, volume sampah yang dihasilkan pun akan terus bertambah. Tanpa pengelolaan yang baik, penumpukan sampah ini dapat menyebabkan pencemaran lingkungan dan masalah kesehatan.

Dari berbagai penjelasan tersebut, dapat disimpulkan bahwa sampah merupakan sisa hasil kegiatan manusia yang jika dikelola dengan benar dapat mengurangi polusi yang ada.

### **3. Koperasi**

Koperasi merupakan kegiatan yang dijalankan oleh perseorangan atau kelompok yang mengutamakan kerja sama dan gotong royong, dengan berdasarkan persamaan derajat, hak dan kewajiban (Mawarni et al., 2024). Dengan visi yang jelas, koperasi memiliki potensi untuk berperan sebagai fondasi dalam memajukan perekonomian di Indonesia. Koperasi memegang peranan penting di Indonesia dengan memberdayakan UMKM, mendorong kemandirian masyarakat, menggerakkan ekonomi, serta menciptakan pasar baru (Ichsan et al., 2021).

Berdasarkan penjelasan tersebut, dapat disimpulkan koperasi merupakan kegiatan yang dijalankan oleh perseorangan dan kelompok yang

mengutamakan kerja sama dan gotong royong, yang bisa meningkatkan perekonomian di Indonesia dengan membantu usaha kecil dan menengah.

#### **4. Sistem Informasi Geografis (SIG)**

Sistem Informasi Geografis (SIG) merupakan teknologi berbasis komputer yang memiliki kemampuan untuk mengelola dan menganalisis data lokasi di permukaan bumi. Menurut Sahrudin & Sularno (2023) Sistem Informasi Geografis (SIG) adalah sebuah sistem berbasis komputer yang dirancang untuk bekerja dengan data bereferensi geografis. Fungsi utamanya adalah mengintegrasikan data spasial (peta) dengan data atribut (informasi deskriptif). Sistem ini memiliki kemampuan untuk menangkap, mengelola, menganalisis, dan memvisualisasikan data tersebut untuk mendukung perencanaan, pengelolaan, serta penelitian terkait masalah ke ruang. Secara sederhana, SIG dapat dipahami sebagai sistem yang mampu menampilkan peta digital yang dapat di kustomisasi.

Untuk dapat menggunakan SIG dengan baik perlu beberapa komponen. Empat komponen fundamental yaitu perangkat keras, perangkat lunak, organisasi, dan pengguna, di mana kombinasi yang tepat di antara keempatnya merupakan penentu kesuksesan implementasi proyek (Supuwiningsih & Rusli, 2020). Saat ini, terdapat berbagai aplikasi SIG yang populer, seperti QGIS, ArcGIS, Google Earth Pro, dan MapInfo Professional.

#### **5. *Location Based Service (LBS)***

*Location-Based Service (LBS)* adalah istilah umum untuk teknologi yang memanfaatkan data lokasi geografis dari perangkat pengguna untuk

menyediakan suatu layanan (Siregar et al., 2023). Teknologi ini menggunakan GPS, Wi-Fi atau jaringan seluler untuk mendapatkan koordinat lokasi secara *real-time*. Banyaknya penggunaan ponsel pintar dalam kehidupan sehari-hari telah memunculkan ratusan layanan berbasis lokasi, dan banyak dari layanan itu sebenarnya punya tujuan yang sama (Yus et al., 2020).

LBS juga merupakan implementasi dari teknologi Sistem Informasi Geografis (SIG) untuk perangkat seluler yang mengolah data lokasi menjadi peta digital. Sejak kemunculan salah satu LBS pertama seperti sistem E911 di Amerika Serikat pada tahun 1996, teknologi ini telah berkembang secara signifikan dan terus berlanjut hingga kini (Jiang et al., 2022). Perkembangan pesat tersebut mendorong inovasi beragam fitur, misalnya navigasi *real-time*, dan rekomendasi tempat. Dengan demikian, LBS kini menjadi komponen penting dalam banyak aplikasi modern

## 6. *Unified Modeling Language (UML)*

*Unified Modeling Language (UML)* merupakan suatu metode pemodelan yang digunakan dalam perancangan sistem berorientasi objek (Marthiawati et al., 2024). UML berfungsi memvisualisasikan proses kerja sistem, struktur data, serta interaksi antar sistem, sehingga dapat membantu dalam pengembangan sistem. Menurut Laia (2023) UML menawarkan serangkaian notasi grafis yang komprehensif untuk menjelaskan beragam aspek perangkat lunak, yang juga membantu melancarkan komunikasi

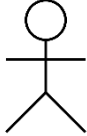



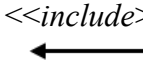
antara pengembang, analis bisnis, dan pihak berkepentingan lainnya.

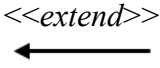
Diagram UML terdiri dari beberapa jenis, di antaranya:

a. *Use Case Diagram*

Diagram *use case* adalah sarana untuk menggambarkan persyaratan sebuah sistem yaitu sistem apa yang seharusnya digunakan (Destriana et al., 2022). Tujuannya adalah untuk memberikan gambaran dari sistem yang mudah dipahami oleh semua pemangku kepentingan. Simbol pada diagram *use case* dapat dilihat pada Tabel 2.1.

Tabel 2. 1 *Use Case Diagram*

No.	Simbol	Nama	Keterangan
1		<i>Actor</i>	Menunjukkan pihak luar (bisa orang atau sistem lain) yang menggunakan atau berinteraksi dengan sistem.
2		<i>Use Case</i>	Menjelaskan fungsi-fungsi yang disediakan oleh sistem untuk pengguna, tanpa merinci cara kerja teknis di baliknya.
3		<i>Association</i>	Merepresentasikan hubungan interaksi antara seorang Aktor dan sebuah <i>Use Case</i> yang diikutinya.
4		<i>Use Case Generalization</i>	Sebuah hubungan di mana satu <i>use case</i> anak (spesifik) mewarisi karakteristik dari <i>use case</i> induk (umum), sambil menambahkan atau mengubah beberapa perilakunya sendiri.
5		<i>Include</i>	Menunjukkan bahwa sebuah <i>use case</i> utama (dasar) secara wajib menyertakan atau menjalankan seluruh

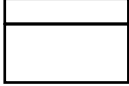



No.	Simbol	Nama	Keterangan
			fungsiionalitas dari <i>use case</i> lain (yang di <i>include</i> ).
6		<i>Extend</i>	Menambahkan fungsiionalitas opsional (perluasan) ke sebuah <i>use case</i> utama. Fungsiionalitas tambahan ini hanya akan dijalankan jika sebuah kondisi spesifik terpenuhi.


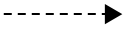

Sumber: Destriana et al., (2022)

b. *Class Diagram*

Diagram kelas (*class diagram*) menggambarkan struktur statis sebuah sistem dengan menunjukkan hubungan antar kelas beserta detail dari setiap kelas tersebut (Syarif & Pratama, 2021). Selain itu diagram ini juga memberikan aturan dan tanggung jawab pada setiap elemen untuk membentuk perilaku sistem. Simbol diagram kelas (*class diagram*) dapat dilihat pada tabel 2.2.

Tabel 2. 2 *Class Diagram*

No.	Simbol	Nama	Keterangan
1		<i>Class</i>	Struktur kelas yang membentuk arsitektur atau kerangka dasar dari sebuah sistem.
2		<i>Interface</i>	Konsep ini serupa dengan prinsip antarmuka ( <i>interface</i> ) dalam pemrograman berorientasi objek
3		<i>Association</i>	Relasi antar kelas yang paling umum, yang biasanya dilengkapi dengan <i>multiplicity</i>
4		<i>Direct Association</i>	Relasi yang menunjukkan bahwa sebuah kelas bergantung dengan fungsiionalitas dari kelas lain.



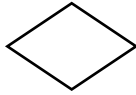
No.	Simbol	Nama	Keterangan
5		<i>Generalisasi</i>	Relasi Generalisasi yaitu sebuah hubungan di mana kelas anak (spesialisasi/khusus) mewarisi atribut dan metode dari kelas induk (generalisasi/umum)
6		<i>Dependency</i>	Hubungan yang menunjukkan bahwa satu kelas bergantung pada atau menggunakan fungsionalitas dari kelas lain'
7		<i>Aggregation</i>	Hubungan kepemilikan, di mana kelas "keseluruhan" terdiri dari kelas-kelas "bagian".



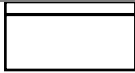
Sumber: Destriana et al., (2022)

### c. *Activity Diagram*

Menurut Hidayati et al. (2023) *Activity Diagram* adalah penggambaran *workflow* (aliran kerja) dari sebuah sistem. Diagram ini merinci urutan aktivitas, titik keputusan (*decision*), dan alur kontrol dari satu kegiatan ke kegiatan lainnya. Simbol pada *activity diagram* dapat dilihat pada tabel 2.3.

Tabel 2. 3 *Activity Diagram*

No.	Simbol	Nama	Keterangan
1		Status Awal	Menandakan titik dimulainya seluruh alur proses dalam diagram.
2		Aktivitas	Merepresentasikan sebuah tugas atau aksi yang dilakukan oleh sistem, umumnya diawali dengan kata kerja.
3		<i>Decision</i>	Titik percabangan kondisional di mana alur dapat terpecah menjadi beberapa jalur alternatif berdasarkan kondisi tertentu.


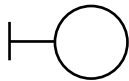

No.	Simbol	Nama	Keterangan
4		<i>Join</i>	Titik penggabungan kembali beberapa alur aktivitas yang sebelumnya berjalan secara paralel menjadi satu alur tunggal.
5		Status Akhir	Menandakan titik berakhirnya atau selesainya seluruh alur proses dalam diagram.
6		<i>Swimlane</i>	Partisi yang mengelompokkan aktivitas berdasarkan aktor atau unit bisnis yang bertanggung jawab atas aktivitas tersebut.

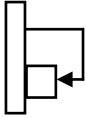


Sumber: Destriana et al., (2022)

#### d. *Sequence Diagram*

*Sequence diagram* adalah gambaran interaksi antar objek dengan menekankan pada urutan waktu komunikasi di antara objek-objek tersebut (Nabila et al., 2021). Dengan demikian, diagram ini memudahkan dalam memodelkan logika dari sebuah skenario atau *use case*. Simbol pada diagram *sequence* dapat dilihat pada Tabel 2.4.

Tabel 2. 4 *Sequence Diagram*

No.	Simbol	Nama	Keterangan
1		<i>Entity Class</i>	Merepresentasikan objek bisnis yang menyimpan dan mengelola data jangka panjang, yang sering kali terkait dengan basis data.
2		<i>Boundary Class</i>	Merepresentasikan antarmuka (interface) tempat aktor (pengguna) berinteraksi dengan sistem, contohnya seperti layar atau menu.
3		<i>Control Class</i>	Merepresentasikan objek yang mengelola alur kerja (workflow) atau logika bisnis untuk suatu fungsi atau use case.

No.	Simbol	Nama	Keterangan
4		<i>Recursive</i>	Pesan yang dikirimkan oleh objek kepada dirinya sendiri, yang biasanya menandakan pemanggilan metode internal.
5		<i>Activation</i>	Menunjukkan periode waktu di mana sebuah objek sedang aktif atau sedang menjalankan sebuah operasi/metode.
6		<i>Life Line</i>	Garis vertikal yang merepresentasikan keberadaan atau "masa hidup" sebuah objek selama interaksi berlangsung.

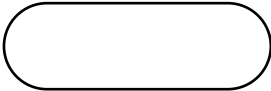



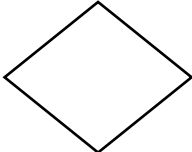
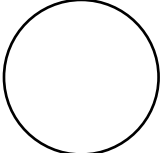
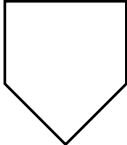
Sumber: Destriana et al., (2022)

## 7. Flowchart

Menurut Zalukhu et al. (2023) “*Flowchart* adalah gambaran dalam bentuk diagram alir dari algoritma-algoritma dalam suatu program, yang menyatakan arah alur program tersebut”. Tujuan utama dari diagram ini adalah untuk menyederhanakan rangkaian logika yang panjang dan rumit agar lebih mudah dipahami. Hal ini menjadikan *flowchart* sebagai alat komunikasi yang baik antara analis sistem, pemrogram, dan pengguna akhir.

Dari berbagai penjelasan tersebut, dapat disimpulkan bahwa *flowchart* adalah diagram grafis yang digunakan untuk menggambarkan langkah-langkah yang berurutan dalam sistem. *Flowchart*, dengan simbol-simbol standarnya, mampu mengilustrasikan tahapan penyelesaian masalah dan aliran data dengan jelas, menjadikannya alat komunikasi dan dokumentasi yang efektif. *Flowchart* bertujuan untuk menyederhanakan dan mempermudah pemahaman proses atau prosedur yang kompleks. Simbol pada *flowchart* dapat dilihat pada Tabel 2.5.

Tabel 2. 5 *Flowchart*

No	Simbol	Keterangan
1	 Terminal	Simbol yang digunakan untuk menunjukkan titik awal atau akhir dari sebuah program.
2	 <i>Flow Direction Symbol /Connecting Line</i>	Simbol yang digunakan dalam menggambarkan alur atau aliran program
3	 <i>Processing</i>	Simbol yang digunakan Untuk memproses perintah pengolahan data
4	 <i>Input Output Data</i>	Simbol yang digunakan yang untuk menggambarkan proses memasukkan dan mengeluarkan data
5	 <i>Decision</i>	Simbol yang digunakan untuk memberikan pilihan antara ya atau tidak
6	 <i>On Page Connector</i>	Simbol yang menghubungkan bagian <i>flowchart</i> dalam halaman sama yang berbeda-beda
7	 <i>Off Page Connector</i>	Simbol yang Menghubungkan bagian <i>flowchart</i> dalam halaman yang berbeda

Sumber: Zalukhu et al., (2023)

## 8. *Rapid Application Development (RAD)*

*Rapid Application Development (RAD)* adalah metodologi yang mempercepat pengembangan perangkat lunak melalui pembuatan prototipe secara *iteratif* (Sari et al., 2024). Pendekatan ini mengandalkan kolaborasi dengan pengguna untuk mendapatkan umpan balik. Keunggulan RAD adalah kemampuannya untuk menerima perubahan selama proses pengembangan tanpa mengorbankan kualitas produk akhir.

RAD juga dianggap sebagai metode yang efektif dalam meningkatkan kolaborasi antara pengembang dan pengguna. Hal ini dikarenakan RAD memfasilitasi interaksi yang lebih intens antara tim teknis dan pengguna akhir, sehingga menghasilkan solusi yang lebih sesuai dengan kebutuhan bisnis. Penggunaan prototipe dalam RAD memungkinkan evaluasi lebih awal, yang dapat mengurangi risiko kesalahan dalam tahap akhir pengembangan perangkat lunak. Dengan demikian, RAD memberikan solusi yang adaptif dan responsif terhadap kebutuhan.

### B. Kajian Empiris

Beberapa penelitian terdahulu yang relevan dengan penerapan metode *Vincenty* digunakan sebagai referensi dan pembanding dalam penelitian ini, di antaranya sebagai berikut.

Penelitian yang dilakukan oleh (Syahdan, 2024) yang membandingkan metode *vincenty* dan *haversine* untuk analisis penugasan agen terhadap nasabah. Penelitian tersebut menghasilkan akurasi dengan menggunakan metode *vincenty* terhadap standar *google maps*, mencapai di atas 89,17%.

Sedangkan saat menggunakan metode *haversine* sekitar 73,18% terhadap standar *google maps*. Dari penelitian ini didapatkan bahwa penggunaan metode *vincenty* memiliki akurasi lebih tinggi dengan standar *google maps*.

Penelitian yang dilakukan oleh (Windarni & Setiawan, 2022) yang membandingkan metode *vincenty* dan *geodesik* untuk mengukur jarak antar kantor kecamatan di kota Salatiga. Dalam jurnal ini didapatkan persentase menggunakan metode *vincenty* saat dibandingkan dengan *google maps* rata-rata 99,49%. Sedangkan persentase menggunakan metode *geodesik* saat dibandingkan dengan *google maps* rata-rata 99,58%. Dari penelitian ini didapatkan bahwa penggunaan metode *geodesik* memiliki persentase yang sedikit lebih baik dari metode *vincenty*.

Penelitian yang dilakukan oleh (Wijaya, 2024) di Universitas PGRI Semarang, implementasi algoritma *vincenty* dalam aplikasi presensi berbasis Android untuk pegawai Badan Pusat Statistik (BPS) Kabupaten Grobogan terbukti sangat efektif untuk meningkatkan akurasi dan efisiensi pencatatan kehadiran. Studi ini menunjukkan bahwa algoritma *vincenty* mampu menghitung jarak antara koordinat GPS pegawai dengan lokasi kantor secara akurat, sehingga sistem dapat memvalidasi kehadiran hanya jika pegawai berada dalam radius yang telah ditentukan. Hasil pengujian aplikasi menunjukkan tingkat keberhasilan fungsionalitas sebesar 100% melalui pengujian

Penelitian yang dilakukan oleh (Yanti et al., 2024) yang menganalisis penugasan agen terhadap lokasi nasabah menunjukkan bahwa metode *vincenty*

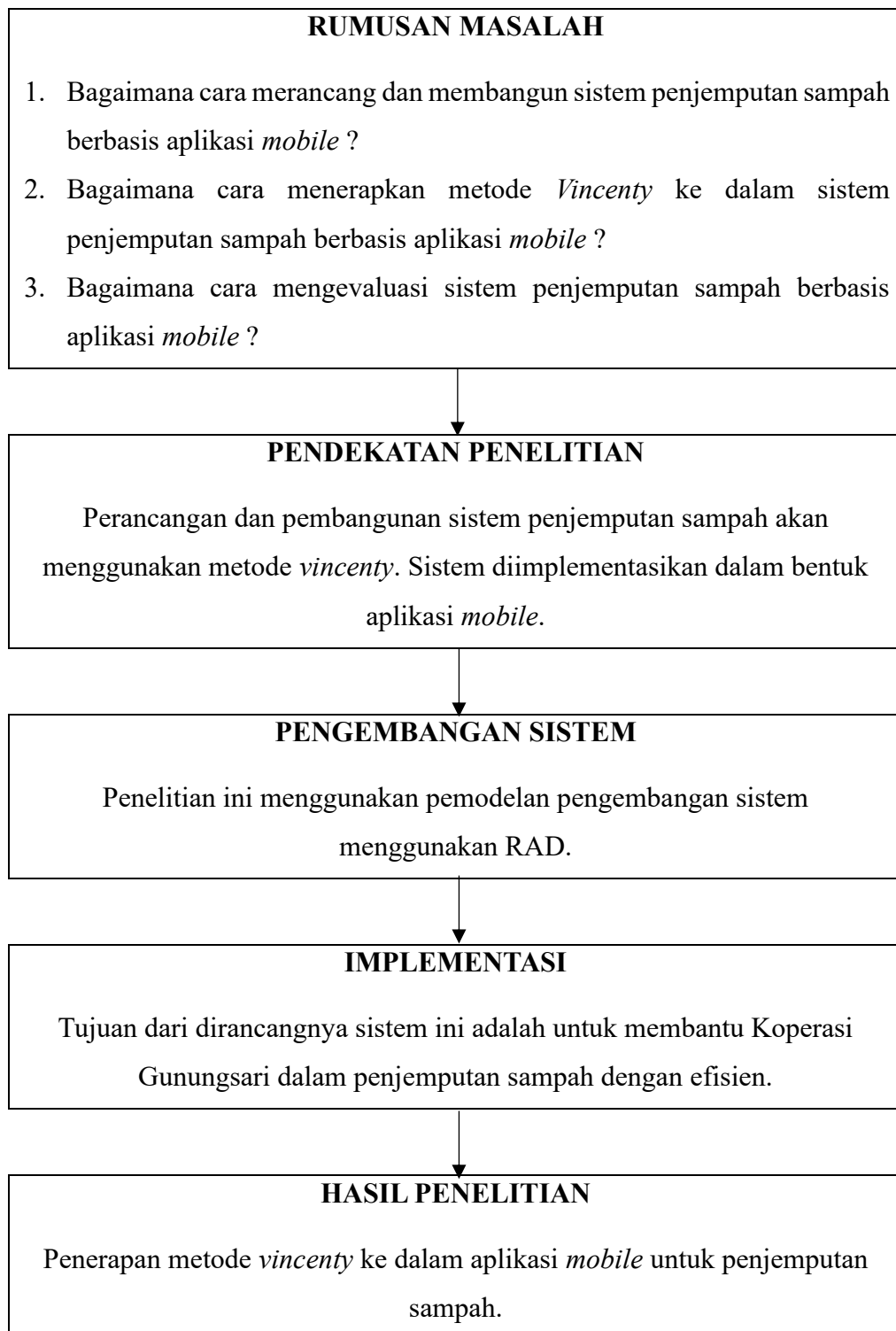
berhasil menghitung jarak antara agen dan nasabah dengan tingkat akurasi 97% jika dibandingkan dengan *google maps*. Penelitian ini juga mengidentifikasi bahwa dari 11 agen yang dianalisis, agen di Kelurahan Jenetallasa adalah yang paling banyak dihubungi oleh nasabah, dengan persentase 26,8%.

Sementara itu, penelitian lain oleh (Halim, 2024) membandingkan kinerja algoritma *vincenty* dan *hubeny* untuk kasus penentuan jarak terpendek antar rumah sakit di Kota Lhokseumawe. Hasilnya menunjukkan bahwa algoritma *vincenty* lebih akurat karena adanya proses iterasi, meskipun sedikit lebih lambat dari algoritma *hubeny*. Perbedaan jarak antara kedua algoritma tersebut sangat kecil, yaitu di bawah 1%, dengan rata-rata 0.009289%. Namun, ketika dibandingkan dengan jarak rute sebenarnya dari *google maps*, kedua algoritma menunjukkan penyimpangan yang signifikan, mendekati 30%, karena keduanya menghitung jarak dalam garis lurus antar titik. Meskipun demikian, penelitian tersebut menyimpulkan bahwa kedua algoritma masih layak digunakan untuk pemetaan pada wilayah seukuran kota. Dalam pencarian rute terpendek menggunakan algoritma.

Berdasarkan beberapa penelitian terdahulu tersebut, dapat disimpulkan bahwa metode *vincenty* secara konsisten terbukti sebagai salah satu metode yang paling akurat untuk menghitung jarak *geodesik* (garis lurus) di permukaan bumi. Keakuratannya yang tinggi, sering kali mencapai di atas 97% bila dibandingkan dengan referensi seperti *google maps*, membuatnya unggul dibandingkan metode yang lebih sederhana seperti *haversine* dan sangat sebanding dengan metode kompleks lainnya seperti *hubeny* dan *geodesic*.

Meskipun demikian, kelemahan utamanya adalah sifat perhitungannya yang *iteratif*, yang membuatnya sedikit lebih lambat secara komputasi. Perbedaan mendasar antara jarak garis lurus yang dihitung *vincenty* dan jarak rute perjalanan nyata dari *google maps* juga menjadi faktor penting yang perlu dipertimbangkan. Temuan ini memberikan landasan yang kuat untuk penelitian ini, memvalidasi pemilihan metode *vincenty* karena akurasinya dalam menentukan lokasi penjemputan sampah.

### C. Kerangka Berpikir



Gambar 2. 1 Kerangka Berpikir