



universitas
MALIKUSSALEH

TUGAS AKHIR

Sebagai Salah Satu Syarat Untuk Memperoleh Gelar

SARJANA KOMPUTER

Pada Program Studi Sistem Informasi – Universitas Malikussaleh

PERBANDINGAN ALGORITMA *VINCENTY* DAN *HUBENY*

DALAM MENENTUKAN JARAK TERPENDEK PADA

RUMAH SAKIT DI LHOKSEUMAWE

BERBASIS *ANDROID*

Oleh :

Ridwan Halim

190180077

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS TEKNIK

UNIVERSITAS MALIKUSSALEH

LHOKSEUMAWE

2024

LEMBAR PERNYATAAN ORISINALITAS

Saya yang bertanda tangan dibawah ini :

Nama : Ridwan Halim

NIM : 190180077

Fakultas/Jurusan : Teknik / Sistem Informasi

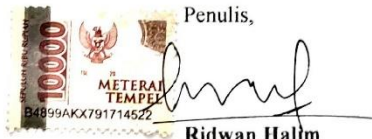
Dengan ini menyatakan skripsi yang berjudul:

Analisis Perbandingan Algoritma *Vincenty* Dan *Huheny* Dalam Menentukan Jarak Terpendek Pada Rumah Sakit Di Lhokseumawe Berbasis *Android.*, adalah hasil kerja tulisan saya sendiri didampingi dosen pembimbing bukan hasil plagiat dari karya tulis ilmiah orang lain.

Demikian surat pernyataan ini saya buat dengan sebenarnya, jika dikemudian hari ternyata terbukti bahwa skripsi yang saya tulis adalah plagiat, maka saya bersedia menerima sanksi sesuai aturan yang berlaku, dan saya bertanggung jawab secara mandiri tidak ada sangkut pautnya dengan Dosen Pembimbing dan kelembagaan Fakultas Teknik Universitas Malikussaleh.

Lhokseumawe, 24 Januari 2024

Penulis,



Ridwan Halim

NIM. 190180077

LEMBAR PENGESAHAN PEMBIMBING

Judul Tugas Akhir : Perbandingan Algoritma *Vincenty* Dan *Hubeny* Dalam
Menentukan Jarak Terpendek Pada Rumah Sakit Di
Lhokseumawe Berbasis *Android*

Nama : Ridwan Halim
NIM : 190180077
Program Studi : Sistem Informasi
Tanggal Sidang : 19 Desember 2023

Bukit Indah, 24 Januari 2024

Penulis

Ridwan Halim
NIM : 190180077

Menyetujui,

Pembimbing Utama,



Mutammimul Ula, S.Kom., M.Cs
NIP. 198508282008121003

Pembimbing Pendamping,





Desvina Yulisda, S.ST., M.S
NIP. 198912122019032020


LEMBAR PENGESAHAN KOMISI PENGUJI


Telah disidangkan pada
Tanggal 19 Desember 2023

KOMISI PENGUJI TUGAS AKHIR


Pembimbing Utama :
Mutammimul Ula, S.Kom., M.Cs
NIP. 198508282008121003 ()


Pembimbing Pendamping :
Desvina Yulisda, S.ST., M.S
NIP. 198912122019032020 ()


Penguji I :
Mochamad Ari Saptari, S.Kom., M.Kom
NIP. 198001052008121001 ()


Penguji II :
Veri Ilhadi, S.T., M.Kom
NIP. 199111292022031012 ()

LEMBAR ACC CETAK

Pembimbing Utama :
Mutammimul Ula, S.Kom., M.Cs
NIP. 198508282008121003 ()

Pembimbing Pendamping :
Desvina Yulisda, S.ST., M.S
NIP. 198912122019032020 ()

Penguji I :
Mochamad Ari Saptari, S.Kom., M.Kom
NIP. 198001052008121001 ()

Penguji II :
Veri Ilhadi, S.T., M.Kom
NIP. 199111292022031012 ()

LEMBAR PENGESAHAN SKRIPSI

Judul Skripsi : Perbandingan Algoritma *Vincenty* Dan *Hubeny*
Dalam Menentukan Jarak Terpendek Pada Rumah
Sakit Di Lhokseumawe Berbasis *Android*
Nama Mahasiswa : Ridwan Halim
NIM : 190180077
Program Studi : S1 Sistem Informasi
Jurusan : Teknik Elektro
Fakultas : Teknik
Perguruan Tinggi : Universitas Malikussaleh
Pembimbing Utama : Mutammimul Ula, S.Kom., M.Cs
Pembimbing Pendamping : Desvina Yulisda, S.ST., M.S
Ketua Penguji : Moch. Ari Saptari, S.Kom., M.Kom
Anggota Penguji : Veri Ilhadi, S.T., M.Kom

Lhokseumawe, 24 Januari 2024
Penulis,

Ridwan Halim
NIM 190180077

Pembimbing Utama,



Mutammimul Ula, S.Kom., M.Cs
NIP. 198508282008121003

Menyetujui:

Pembimbing Pendamping,



Desvina Yulisda, S.ST., M.S
NIP. 198912122019032020

Ketua Jurusan,



**Prof. Dr. Ir. Dahlan Abdullah, S.T.,
M.Kom, IPU, AS/AN Eng.**
NIP 1976092282002121005

Mengetahui:

Koordinator Program Studi,



Rizky Putra Fhonna, S.T., M.Kom.
NIP 199111192019031012

ABSTRAK

Sistem Informasi Geografis digunakan untuk membantu dalam hal pelayanan umum seperti melihat data rumah sakit karena sistem yang diterapkan ini sangatlah penting dalam sistem informasi. Dalam penelitian ini, peneliti merancang aplikasi berbasis *android* yang dapat digunakan untuk menentukan jarak antara dua titik antar rumah sakit di Lhokseumawe menggunakan algoritma *Vincenty* dan algoritma *Hubeny*. Dalam pengujian aplikasi, akan dilakukan pengujian terhadap kecepatan dan ketepatan dari kedua algoritma tersebut yang kemudian diolah menggunakan algoritma *ant colony* dalam menentukan jarak terpendek antara dua rumah sakit. Algoritma *ant colony* merupakan algoritma pencarian jarak terpendek yang dibuat berdasarkan prinsip semut dalam mencari makanan dengan cara meninggalkan feromon sebagai acuan semut lain dengan cara diiterasi hingga mendapatkan jarak terpendek. Dengan menggunakan *framework flutter* dan *library dart* yaitu *google maps* maka dapat dirancang aplikasi penentuan jarak terpendek pada rumah sakit di kota Lhokseumawe dengan implementasi dari algoritma *hubeny*, *vincenty* dan *ant colony*. Perbedaan jarak antara algoritma *hubeny* dan *vincenty* berada dibawah 1% dengan rata-rata perbedaannya 0.009289%. Untuk dibandingkan dengan jarak yang sebenarnya melalui *google map* kedua algoritma memiliki perbedaan sebesar hampir 30% yang mana jika kedua titik terlalu jauh maka akan terjadi penyimpangan jarak melebihi 30%. Kedua algoritma diatas masih layak untuk digunakan dengan catatan hanya bisa diaplikasikan pada daerah seukuran kota madya atau ibukota provinsi. Program algoritma *hubeny* lebih efisien dibandingkan algoritma *vincenty* dalam kecepatan pengekuasian program dengan rata-rata perbedaan dalam waktu yaitu algoritma *hubeny* lebih cepat 0.4584 milidetik dibandingkan algoritma *vincenty*. Dari hasil pengujian, penulis menyimpulkan bahwa algoritma *vincenty* lebih akurat dikarenakan terdapat iterasi yang dilakukan. Namun memiliki efisiensi waktu sedikit lebih lama dari pada algoritma *hubeny*. Perbedaan jarak terbaik algoritma *hubeny*, algoritma *vincenty* dan *google map* pada algoritma *ant colony* adalah algoritma *hubeny* : 19274,741553 meter dan algoritma *vincenty* : 19276,712964 meter, yang mana selisih keduanya hanya sebesar 1,971411096 meter dan rata – rata keduanya 19275,727258 meter dan *google map* dengan jarak terbaik : 25972 meter, yang mana perbedaan dengan rata- rata kedua algoritma sebelumnya 6696,272742 meter.

Kata Kunci : *Rumah Sakit, Vincenty, Hubeny, Ant Colony, Perbandingan*

ABSTRACT

Geographic Information Systems are used to assist in terms of public services such as viewing hospital data because the system applied is very important in information systems. In this study, researchers designed an android-based application that can be used to determine the distance between two points between hospitals in Lhokseumawe using the Vincenty algorithm and the Hubeny algorithm. . In testing the application, testing will be carried out on the speed and accuracy of the two algorithms which are then processed using the ant colony algorithm in determining the shortest distance between two hospitals. The ant colony algorithm is the shortest distance search algorithm made based on the principle of ants in finding food by leaving pheromones as a reference for other ants by iterating until they get the shortest distance. By using the flutter framework and the dart library, namely google maps, an application for determining the shortest distance to hospitals in Lhokseumawe city can be designed with the implementation of the hubeny, vincenty and ant colony algorithms. The difference in distance between the hubeny and vincenty algorithms is below 1% with an average difference of 0.009289%. To compare with the actual distance through google map, the two algorithms have a difference of almost 30%, which if the two points are too far away, there will be a deviation in distance exceeding 30%. The two algorithms above are still feasible to use with a note that they can only be applied to areas the size of a city or provincial capital. The hubeny algorithm program is more efficient than the vincenty algorithm in program execution speed with an average difference in time, namely the hubeny algorithm is 0.4584 milliseconds faster than the vincenty algorithm. From the test results, the author concludes that the vincenty algorithm is more accurate because there are iterations performed. However, it has a slightly longer time efficiency than the hubeny algorithm. The best distance difference of hubeny algorithm, vincenty algorithm and google map on ant colony algorithm is hubeny algorithm: 19274.741553 meters and vincenty algorithm: 19276,712964 meters, where the difference between the two is only 1,971411096 meters and the average of the two is 19275,727258 meters and google map with the best distance: 25972 meters, which is the difference with the average of the two previous algorithms 6696.272742 meters.

Keyword : Hospital, Vincenty, Hubeny, Ant Colony, Comparison

KATA PENGANTAR

Dengan memanjatkan puji dan syukur atas kehadiran Allah SWT yang telah memberikan segala rahmat dan hidayah-Nya kepada peneliti, sehingga peneliti dapat menyelesaikan Proposal tugas akhir dengan judul **“PERBANDINGAN ALGORITMA *VINCENTY* DAN *HUBENY* DALAM MENENTUKAN JARAK TERPENDEK PADA RUMAH SAKIT DI LHOKSEUMAWE BERBASIS *ANDROID*”** Laporan Tugas Akhir ini disusun sebagai salah satu syarat untuk mengerjakan skripsi pada program Strata-1 di Program Studi Sistem Informasi, Fakultas Teknik, Universitas Malikussaleh.

Pada kesempatan ini tak lupa peneliti mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Ir. Herman Fitra, S.T., M.T., IPM., ASEAN Eng selaku Rektor Universitas Malikussaleh.
2. Bapak Dr. Muhammad Daud, M.T selaku Dekan Fakultas Teknik Universitas Malikussaleh.
3. Bapak Prof. Dr. Dahlan Abdullah, S.T., M.Kom., IPU., ASEAN Eng selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Malikussaleh.
4. Bapak Rizky Putra Fhonna, S.T., M.Kom selaku Ketua Prodi Sistem Informasi Fakultas Teknik Universitas Malikussaleh.
5. Bapak Mutammimul Ula, S.Kom., M.Cs selaku Dosen Pembimbing Utama yang dengan sabar telah meluangkan waktu untuk memberikan arahan dan bimbingan selama menyusun Proposal tugas akhir ini.
6. Ibu Desvina Yulisda, S.ST., M.S. Kom selaku Dosen Pembimbing Pendamping yang dengan sabar telah meluangkan waktu untuk memberikan arahan dan bimbingan selama menyusun tugas akhir ini.
7. Orang Tua beserta kakak yang selalu memberikan doa dan dukungan sehingga peneliti dapat menyelesaikan tugas akhir ini.

8. Bapak / Ibu Dosen dan Staff Prodi Sistem Informasi Fakultas Teknik Universitas Malikussaleh yang telah banyak memberikan ilmu selama duduk di bangku perkuliahan.
9. Keluarga Besar Mahasiswa Sistem Informasi Fakultas Teknik Universitas Malikussaleh yang selalu mendukung dan membantu dalam menyelesaikan tugas akhir ini.
10. Rekan-rekan seperjuangan saya angkatan 2019 Prodi Sistem Informasi Fakultas Teknik Universitas Malikussaleh

Dengan segala kerendahan hati, peneliti menyadari dalam penyusunan Laporan Tugas Akhir ini masih jauh dari kata sempurna. Oleh karena itu, peneliti mengharapkan adanya kritik dan saran yang bersifat membangun. Akhir kata, semoga penyusunan skripsi ini bermanfaat bagi semua pihak.

Lhokseumawe, 5 April 2023

Penulis

Ridwan Halim

190180077

DAFTAR ISI

ABSTRAK	i
ABSTRACT	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR GAMBAR	vii
DAFTAR TABEL	viii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	2
1.4. Tujuan Penelitian	3
1.5. Manfaat Penelitian	3
1.6. Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA	5
2.1. Rumah sakit	5
2.2. <i>Hubeny</i>	5
2.2.1. Pengertian algoritma <i>hubeny</i>	5
2.2.2. Rumus algoritma <i>hubeny</i>	6
2.3. <i>Vincenty</i>	6
2.3.1. Pengertian algoritma <i>vincenty</i>	6
2.3.2. Rumus algoritma <i>vincenty</i>	6
2.4. <i>Ant Colony</i>	8
2.4.1. Pengertian algoritma <i>ant colony</i>	8
2.4.2. Rumus algoritma <i>ant colony</i>	8
2.5. <i>Android</i>	10
2.6. Bahasa Pemrograman	11
2.7. <i>Dart</i>	11
2.8. <i>Framework</i>	12
2.9. <i>Flutter</i>	12
2.10. <i>White Box Testing</i>	13

2.11. <i>Visual Studio Code</i>	13
2.12. Penulisan Terdahulu	13
BAB III METODOLOGI PENELITIAN	19
3.1. Tempat dan Jadwal Penelitian	19
3.2. Metode Penelitian	20
3.3. Teknik Pengumpulan Data	23
3.4. Kebutuhan Sistem	23
3.5. Skema Sistem	24
3.6. Data Penelitian	25
BAB IV HASIL DAN PEMBAHASAN	27
4.1. Implementasi Algoritma	27
4.1.1. Perhitungan algoritma <i>hubeny</i>	28
4.1.2. Perhitungan algoritma <i>vincenty</i>	30
4.1.3. Perhitungan algoritma <i>ant colony</i>	33
4.1.4. Analisa perbandingan algoritma <i>vincenty</i> dan <i>hubeny</i>	44
4.1.5. Analisa perbandingan hasil menggunakan algoritma <i>ant colony</i>	49
4.2. Implementasi Aplikasi	50
4.2.1. Tampilan awal aplikasi	51
4.2.2. Menampilkan lokasi pengguna dan merubah map	52
4.2.3. Mencari rumah sakit terdekat	53
4.2.4. Hasil perhitungan algoritma <i>hubeny</i> , <i>vincenty</i> dan <i>ant colony</i>	54
4.2.5. Pengujian <i>white box</i>	55
BAB V PENUTUP	59
5.1. Kesimpulan	59
5.2. Saran	60
DAFTAR PUSTAKA	61

DAFTAR GAMBAR

Gambar 3. 1 Alur Penelitian.....	21
Gambar 3. 2. Skema Sistem	24
Gambar 4. 1. Perbandingan Efisiensi Waktu	45
Gambar 4. 2. Tampilan Awal Aplikasi	51
Gambar 4. 3. Lokasi Pengguna	52
Gambar 4. 4. Mencari Rumah Sakit Terdekat.....	53
Gambar 4. 5. Pengimplementasian Algoritma	54
Gambar 4. 6. Uji <i>Widget</i> 1	55
Gambar 4. 7. Uji <i>Widget</i> 2	56
Gambar 4. 8. Uji <i>Widget</i> 3	57
Gambar 4. 9. Uji <i>Widget</i> 4 Dan Bebarapa Teks.....	58

DAFTAR TABEL

Tabel 2. 1. Penelitian Terdahulu	14
Tabel 3. 1. Jadwal Penelitian.....	19
Tabel 3. 2. Data Penelitian	25
Tabel 4. 1. Data Lokasi Pengguna dan Rumah Sakit.....	27
Tabel 4. 2. Hasil Perhitungan Semua Lokasi Algoritma <i>Hubeny</i>	29
Tabel 4. 3. Iterasi Algoritma <i>Vincenty</i>	31
Tabel 4. 4. Hasil Perhitungan Dua Lokasi Algoritma <i>Vincenty</i>	31
Tabel 4. 5. Hasil Perhitungan Semua Lokasi Algoritma <i>Vincenty</i>	32
Tabel 4. 6. Jumlah Iterasi Yang Diperlukan	33
Tabel 4. 7. Nilai Feromon Awal.....	34
Tabel 4. 8. Visibilitas Tiap Jarak Antar Titik.....	34
Tabel 4. 9. Probabilitas Dari Titik 1.....	35
Tabel 4. 10. Probabilitas Dari Titik 8.....	36
Tabel 4. 11. Probabilitas Dari Titik 10.....	36
Tabel 4. 12. Probabilitas Dari Titik 11.....	37
Tabel 4. 13. Probabilitas Dari Titik 4.....	37
Tabel 4. 14. Probabilitas Dari Titik 3.....	38
Tabel 4. 15. Probabilitas Dari Titik 7.....	38
Tabel 4. 16. Probabilitas Dari Titik 5.....	38
Tabel 4. 17. Probabilitas Dari Titik 9.....	39
Tabel 4. 18. Hasil Dari Semua Semut Di Iterasi Pertama.....	40
Tabel 4. 19. Nilai Peromon Terbaru.....	41
Tabel 4. 20. Hasil Jarak Terpendek Pada 100 Iterasi.....	42
Tabel 4. 21. Perbedaan Persentase Jarak Algoritma <i>Hubeny</i> dan <i>Vincenty</i>	46
Tabel 4. 22. Perbedaan Persentase Jarak Algoritma <i>Hubeny</i> dan <i>Google Map</i>	47
Tabel 4. 23. Perbedaan Persentase Jarak Algoritma <i>Vincenty</i> dan <i>Google Map</i> ..	46
Tabel 4. 24. Perbedaan Persentase Algoritma <i>Hubeny</i> , Algoritma <i>Vincenty</i> dan <i>Google Map</i> pada hasil jarak terpendek pada Algoritma <i>Ant Colony</i>	49

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi informasi dan komunikasi (TIK) sangat pesat dalam beberapa tahun terakhir. Salah satu dampak positif dari perkembangan tersebut adalah munculnya aplikasi-aplikasi berbasis *android* yang sangat membantu masyarakat dalam melakukan berbagai aktivitas. Salah satu aplikasi mobile yang semakin populer saat ini adalah aplikasi navigasi Sistem Informasi Geografis ini memvisualisasikan data dan menganalisis temuan dengan menggunakan media peta yaitu peta digital. Sistem Informasi Geografis digunakan untuk membantu dalam hal pelayanan umum seperti melihat data rumah sakit karena sistem yang diterapkan ini sangatlah penting dalam sistem informasi (Ritonga et al., 2021).

Penelitian ini bertujuan untuk menentukan jarak terpendek pada Rumah Sakit di Lhokseumawe. Pemilihan rumah sakit sebagai objek penelitian dikarenakan rumah sakit merupakan salah satu lokasi yang sering dicari oleh masyarakat dan kecepatan dan ketepatan dalam menentukan jarak terpendek dapat sangat penting dalam situasi darurat.

Dalam penelitian ini, peneliti merancang aplikasi berbasis *android* yang dapat digunakan untuk menentukan jarak antara dua titik antar rumah sakit di Lhokseumawe menggunakan algoritma *Vincenty* dan algoritma *Hubeny*. Dalam pengujian aplikasi, akan dilakukan pengujian terhadap kecepatan dan ketepatan dari kedua algoritma tersebut yang kemudian diolah menggunakan algoritma *ant colony* dalam menentukan jarak terpendek antara dua rumah sakit.

Berdasarkan latar belakang diatas, peneliti tertarik untuk melakukan penelitian agar dapat memberikan suatu alternatif dalam menentukan jarak yang lebih efektif dan efisien dengan judul “**Perbandingan Algoritma *Vincenty* Dan *Hubeny* Dalam Menentukan Jarak Terpendek Pada Rumah Sakit Di Lhokseumawe Berbasis *Android***”. Dengan adanya penelitian ini diharapkan dapat memberikan pemahaman yang lebih baik tentang perbedaan kinerja antara

algoritma *Vincenty* dan algoritma *Hubeny* dalam menentukan jarak antara dua titik antar rumah sakit pada peta. Hasil dari penelitian ini dapat menjadi referensi bagi pengembang aplikasi navigasi untuk memilih algoritma yang tepat dalam menghitung jarak antara dua titik pada aplikasi ini.

1.2. Rumusan Masalah

Berdasarkan latar belakang diatas yang telah diuraikan maka dapat dirumuskan permasalahannya antara lain :

1. Bagaimana merancang dan membangun sebuah aplikasi “Pemetaan Rumah Sakit” sebagai sarana mempermudah masyarakat dalam mencari rumah sakit terdekat ?
2. Bagaimana cara implementasi algoritma *Vincenty*, algoritma *Hubeny* dan *ant colony* dalam menentukan jarak terpendek pada aplikasi berbasis *Android* ?
3. Apa perbedaan hasil perhitungan jarak terpendek antara algoritma *Vincenty* dan algoritma *Hubeny* pada rumah sakit di Lhokseumawe ?

1.3. Batasan Masalah

Melihat dari permasalahannya, maka penelitian ini dibuat beberapa asumsi dengan tujuan agar pembahasan menjadi lebih terarah serta membatasi permasalahan. Batasan masalah dari penelitian ini yaitu antara lain :

1. Aplikasi hanya bisa dijalankan pada sistem operasi *android* pada *smartphone*.
2. Aplikasi hanya bisa dijalankan secara online
3. Penelitian ini hanya membandingkan dua algoritma, yaitu algoritma *Vincenty* dan algoritma *Hubeny* dalam menentukan jarak terpendek pada rumah sakit di Lhokseumawe.
4. Penelitian ini hanya berfokus pada rumah sakit di sekitaran Lhokseumawe.
5. Penelitian ini hanya menggunakan data koordinat geografis untuk menghitung jarak terpendek antara dua titik.

1.4. Tujuan Penelitian

Berdasarkan rumusan masalah diatas didapat beberapa tujuan yang ingin dicapai dalam pembuatan aplikasi “Pemetaan Rumah Sakit” yaitu antara lain :

1. Untuk merancang dan membangun sebuah aplikasi “Pemetaan Rumah Sakit” sebagai sarana mempermudah masyarakat dalam mencari rumah sakit terdekat
2. Menerapkan algoritma *Vincenty*, algoritma *Hubeny* dan algoritma *ant colony* dalam menentukan jarak terpendek pada aplikasi berbasis *Android*
3. Menunjukkan seberapa besar perbedaan antara algoritma *Vincenty* dan algoritma *Hubeny* dalam perhitungan jarak terpendek antara pada rumah sakit di Lhokseumawe

1.5. Manfaat Penelitian

Manfaat dari penelitian mengenai aplikasi “Pemetaan Rumah Sakit” yaitu antara lain :

1. Mempermudah masyarakat khususnya disekitaran Kota Lhokseumawe dalam mencari Rumah Sakit terdekat disaat keadaan darurat maupun tidak darurat
2. Memperluas cakupan pengetahuan penulis sesuai dengan bidang teknologi komputer.
3. Untuk mengetahui sejauh mana ilmu akademik yang saya dapat di dalam kelas perkuliahan yang dapat saya terapkan ke dalam bentuk permasalahan yang sebenarnya.

1.6. Sistematika Penulisan

Sistematika penulisan berfungsi untuk lebih terarahnya penulisan laporan ini, maka sistematika penulisan laporan ini dibagi menjadi lima bab dan masing-masing bab menjadi sub-sub bab yang saling berhubungan. Penulisan masing-masing bab dapat dilihat sebagai berikut ini:

BAB I PENDAHULUAN

Pada bab ini peneliti membahas tentang latar belakang, definisi masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

BAB II TINJAUAN KEPUSTAKAAN

Bab ini menjelaskan landasan teori dan penelitian terdahulu yang digunakan dalam pengolahan masalah penelitian.

BAB III METODOLOGI PENELITIAN

Bab ini berisikan tentang tempat dan jadwal penelitian, teknik pengumpulan data, alat dan bahan, metode pengembangan sistem, prosedur alur penelitian, dan gambaran perencanaan singkat system UI dalam pembuatan game pertarungan bahasa pemograman dasar dan lanjut.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini peneliti akan menerapkan rancangan dan hasil dari metode yang dibuat pada penelitian.

BAB V KESIMPULAN DAN SARAN

Pada bab ini peneliti akan menjelaskan kesimpulan dan saran dari penelitian yang dilakukan.

BAB II

TINJAUAN PUSTAKA

2.1. Rumah sakit

Rumah sakit merupakan salah satu institusi pelayanan kesehatan yang kompleks, padat pakar, dan padat modal. Kompleksitas ini muncul karena pelayanan rumah sakit menyangkut berbagai fungsi pelayanan, pendidikan, dan penelitian, serta mencakup berbagai tindakan maupun jenis disiplin, agar rumah sakit mampu melaksanakan fungsi yang profesional baik dibidang teknis medis maupun administrasi kesehatan. Untuk menjaga dan meningkatkan mutu rumah sakit harus mempunyai suatu ukuran yang menjamin peningkatan mutu disemua tingkatan (Putra & Vadriasm, 2020).

Rumah sakit bertanggung jawab untuk melindungi informasi yang ada di dalam rekam Medis terhadap kemungkinan hilangnya keterangan ataupun pemalsuan data yang ada, Rekam Medis dipergunakan oleh orang yang tidak diberi izin. Rekam Medis diberi data yang cukup terperinci, sehingga dokter dapat mengetahui bagaimana pengobatan dan perawatan kepada pasien dan konsulen dapat pula memberikan pendapat yang tepat setelah memeriksanya atau dokter yang bersangkutan dapat memperkirakan kembali keadaan pasien yang akan datang dari prosedur yang telah dilaksanakan.

2.2. Hubeny

2.2.1. Pengertian algoritma *hubeny*

Algoritma *hubeny* adalah algoritma yang menganalisis jarak, dengan tujuan untuk mempertimbangkan kurva bumi, untuk menentukan dua jarak koordinasi dengan tujuan berapa besar selisih antara dua jarak yang ditempuh (Mauludi et al., 2021).

Algoritma *hubeny* biasanya digunakan untuk menganalisis antara dua titik yang ada di bumi, sehingga pada penelitian ini tidak bisa menggunakan rumus seperti *euclidean distance*. Rumus *euclidean distance* digunakan untuk mengukur dalam garis lurus, rumus euclidean berkaitan dengan teorema pythagoras dan

biasanya diterapkan pada 1, 2, dan 3 dimensi (Mauludi et al., 2021).

2.2.2. Rumus algoritma hubeny

Berdasarkan (Sekiguchi et al., 2021) diketahui rumus dari algoritma *hubeny* seperti yang penulis jabarkan dibawah :

$$s = \sqrt{\left(M \cdot \left(\frac{lat2 - lat1}{180}\right) \cdot \frac{\pi}{180}\right)^2 + \left(N \cdot \cos(P) \cdot \left(\frac{lon2 - lon1}{180}\right) \cdot \frac{\pi}{180}\right)^2} \dots\dots\dots (1)$$

$$M = \frac{6334834}{\sqrt{(1 - 0,006674 \cdot \sin(P)^2)^3}} \dots\dots\dots (2)$$

$$N = \frac{6377397}{\sqrt{1 - 0,006674 \cdot \sin(P)^2}} \dots\dots\dots (3)$$

$$P = (lat1 + lat2) \cdot \frac{\pi}{180 \cdot 2} \dots\dots\dots (4)$$

dimana,

s = jarak antara kedua titik dalam meter

lat1 =latitude lokasi awal

lon1 = longtitude lokasi awal

lat2 = latitude lokasi kedua

lon2 = longtitude lokasi kedua

2.3. *Vincenty*

2.3.1. Pengertian algoritma *vincenty*

Menurut (Nugraha & Hajar, 2023) formula *vincenty* merupakan salah satu teknik perhitungan matematis yang dapat digunakan untuk menentukan besaran *azimuth geodetic*. Teknik tersebut diklaim lebih teliti dibandingkan teknik perhitungan seperti konsep segitiga bola.

2.3.2. Rumus algoritma *vincenty*

Berdasarkan (Windarni & Setiawan, 2022) diketahui rumus dari algoritma *vincenty* seperti yang penulis jabarkan dibawah :

$$f = 1/298.257223563 \dots\dots\dots (1)$$

$$U_1 = \text{atan}((1 - f) \cdot \tan((\text{lat1} \cdot \pi)/180)) \dots\dots\dots (2)$$

$$U_2 = \text{atan}((1 - f) \cdot \tan((\text{lat2} \cdot \pi)/180)) \dots\dots\dots (3)$$

$$L = ((\text{lon2} - \text{lon1}) \cdot \pi)/180 \dots\dots\dots (4)$$

$$\lambda = L \text{ (nilai pertama)} \dots\dots\dots (5)$$

Lakukan iterasi hingga konvergen dengan ketentuan epsilon, biasanya dalam praktek digunakan nilai epsilon (λ) = 10^{-12}

$$\sin \sigma = \sqrt{(\cos(U_2) \cdot \sin(\lambda))^2 + (\cos(U_1) \cdot \sin(U_2) - \sin(U_1) \cdot \cos(U_2) \cdot \cos(\lambda))^2} \dots (6)$$

$$\cos \sigma = \sin(U_1) \cdot \sin(U_2) + \cos(U_1) \cdot \cos(U_2) \cdot \cos(\lambda) \dots\dots\dots (7)$$

$$\sigma = \arctan(\sin \sigma / \cos \sigma) \dots\dots\dots (8)$$

$$\sin \alpha = (\cos(U_1) \cdot \cos(U_2) \cdot \sin(\lambda)) / \sin \sigma \dots\dots\dots (9)$$

$$\cos^2 \alpha = 1 - \sin^2 \alpha \dots\dots\dots (10)$$

$$\cos 2\alpha_m = \cos \sigma - ((2 \cdot \sin(U_1) \cdot \sin(U_2)) / \cos \sigma) \dots\dots\dots (11)$$

$$C = f/16 \cdot \cos^2 \alpha \cdot (4 + f \cdot (4 - 3 \cdot \cos^2 \alpha)) \dots\dots\dots (12)$$

$$\lambda P = \lambda \dots\dots\dots (13)$$

$$\lambda = L + (1 - C) \cdot f \cdot \sin \alpha \cdot (\sigma + C \cdot \sin \sigma \cdot (\cos 2\alpha_m + C \cdot \cos \sigma \cdot (-1 + 2 \cdot \cos^2 \alpha_m))) \dots\dots\dots (14)$$

setelah di iterasi maka selanjutnya

$$u^2 = \cos^2 \alpha \cdot (a^2 - b^2) / b^2 \dots\dots\dots (15)$$

$$A = 1 + (u^2/16384) \cdot (4096 + u^2 \cdot (-768 + u^2 \cdot (320 - 175 \cdot u^2))) \dots\dots\dots (16)$$

$$B = (u^2/1024) \cdot (256 + u^2 \cdot (-128 + u^2 \cdot (74 - 47 \cdot u^2))) \dots\dots\dots (17)$$

$$\Delta \sigma = B \cdot \sin \sigma \cdot (\cos 2\alpha_m + B/4 \cdot (\cos \sigma \cdot (-1 + 2 \cdot \cos^2 \alpha_m) - B/6 \cdot \cos 2\alpha_m \cdot (-3 + 4 \cdot \sin^2 \sigma) \cdot (-3 + 4 \cdot \cos^2 \alpha_m))) \dots (18)$$

$$s = b \cdot A \cdot (\sigma - \Delta \sigma) \dots\dots\dots (19)$$

dimana,

a = 6378137 (jari – jari khatulistiwa bumi dalam meter)

b = 6356752.314245 (jari – jari kutub bumi dalam meter)

lat1 = latitude lokasi awal

lon1 = longtitude lokasi awal

lat2 = latitude lokasi kedua

lon2 = longtitude lokasi kedua

s = jarak antara kedua titik dalam meter

2.4. *Ant Colony*

2.4.1. Pengertian algoritma *ant colony*

Ant colony Optimization (ACO) merupakan salah satu metode penyelesaian masalah optimasi berdasarkan prinsip komunikasi koloni semut dalam mencari makanan. Pada ACO, prinsip dasarnya yaitu setiap semut akan meninggalkan suatu zat khusus yang disebut dengan feromon. Feromon tersebutlah yang akan menjadi acuan pada semut lain dalam melakukan pencarian. Apabila semakin pendek jalur kunjungan, maka semakin sedikit penguapan yang terjadi dan semakin tinggi pula jejak feromon dalam jalur tersebut. Semut cenderung bergerak mengikuti jalur yang memiliki jejak feromon yang tinggi (Nurharyanto & Perdana, 2021).

2.4.2. Rumus algoritma *ant colony*

Menurut (Nugroho & Permadi, 2020) pada algoritma *ant colony* diperlukan beberapa variabel dan langkah-langkah untuk menentukan jarak terpendek.

Langkah 1

1. Tentukan jarak dari *node i* ke *node j* atau matriks d_{ij} .
2. Inisialisasikan nilai parameter yang diperlukan seperti jumlah semut, α , β , ρ , Q dan jumlah iterasi yang diperlukan.

m = jumlah semut

α = Nilai pengendali intensitas feromon ($\alpha > 0$)

β = Nilai pengendali intensitas visibilitas ($\beta > 0$)

ρ = Tetapan penguapan jejak semut. Nilai ρ ditentukan oleh pengguna.

Q = Tetapan siklus semut. Nilai Q ditentukan oleh pengguna

Untuk nilai parameter α sebaiknya diberi nilai $0 \leq \alpha \leq \beta$ hal ini dimaksudkan untuk menghindari akumulasi feromon yang tidak terbatas pada sisi tersebut. Karena jumlah feromon yang ditinggalkan tidak mungkin bertambah kuat tetapi akan bertambah kurang. Untuk nilai parameter β sebaiknya tidak diberi nilai 0 karena jika diberi nilai 0 maka hasil yang dicapai tidak maksimum. Tidak optimal

disini berarti suatu kondisi dimana panjang perjalanan yang dicapai tidak minimum.

3. Tentukan nilai feromon awal atau τ_{ij} .
4. Cari nilai visibilitas atau η_{ij} yang didapat dari $1/d_{ij}$

Langkah 2

1. Probabilitas kunjungan semut dari titik i ke titik j digunakan persamaan berikut :

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} ([\tau_{il}]^\alpha \cdot [\eta_{il}]^\beta)} & \text{jika } j \in N_i^k \\ 0 & \text{lainnya} \end{cases} \dots\dots\dots (1)$$

dimana,

τ_{ij} = nilai feromon

ij = dari 1 titik awal ke 1 titik akhir

ik = dari 1 titik awal ke semua titik akhir

α = Nilai pengendali intensitas feromon ($\alpha > 0$)

β = Nilai pengendali intensitas visibilitas ($\beta > 0$)

N_i^k = titik yang belum dikunjungi semut

2. Tentukan bilangan random r, dimana $0 < r < 1$
3. Temukan bilangan yang lebih besar dari hasil persamaan diatas dan lakukan kembali persamaan diatas sampai semua matriks jarak semua titik menjadi 0.
4. Maka akan didapati jalur dan jarak yang dilalui semut tersebut.
5. Lakukan langkah 2 ini sebanyak jumlah semut.

Langkah 3

1. Untuk memperbarui nilai intensitas feromon yang telah terbentuk pada suatu lintasan gunakan persamaan dibawah :

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \Delta\tau_{ij} \dots\dots\dots (2)$$

$$\Delta\tau_{ij} = \sum_k^m \Delta\tau_{ij}^k \dots\dots\dots (3)$$

$$\Delta\tau_{ij}^k = \frac{Q}{L_k} \dots\dots\dots (4)$$

dimana,

ρ = Tetapan penguapan jejak semut. Nilai ρ ditentukan oleh pengguna.

m = jumlah semut

Lk = panjang tur semut

Q = Tetapan siklus semut. Nilai Q ditentukan oleh pengguna

2. Setelah peromon diperbarui, lakukan langkah 2 dan langkah 3 ini sebanyak iterasi yang diperlukan.
3. Carilah nilai jarak terpendek dari setiap tur (mencari jalan) yang dilakukan oleh semut pada setiap iterasi untuk mencari jarak terbaik.

2.5. *Android*

Android adalah sistem operasi berbasis Linux bagi telepon seluler seperti telepon pintar dan komputer tablet. *Android* juga menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk berbagai macam piranti gerak. Awalnya, *Google Inc.* membeli *Android Inc.*, pendatang baru yang membuat piranti lunak untuk ponsel. kemudian dalam pengembangan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan piranti Jurnal Teknologi dan Sistem Informasi (JTSl), Vol. 2, No. 1, 104 - 111 106 keras, piranti lunak, dan telekomunikasi, termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, dan *Nvidia* (Sulastio et al., 2021).

Android menawarkan pendekatan yang menyeluruh dalam pengembangan aplikasi. Artinya, satu aplikasi *android* yang dibangun dapat berjalan di berbagai perangkat yang menggunakan system operasi *android* baik itu *smartphone*, *smartwatch*, *tablet*, dan perangkat lainnya. Perkembangan teknologi android yang bergitu pesat juga tidak dapat dilepas dari peran *AOSP(Android Open Source Project)* yang bertanggung jawab dalam pengembangan sistem operasi android dan dipimpin langsung oleh *google* (Irvansyah et al., 2020).

Menurut (Frialdo et al., 2023) *Android* adalah sistem operasi telepon pintar atau *tablet* yang memiliki banyak fitur untuk mempermudah kehidupan manusia dan merupakan sistem operasi berbasis *Linux* yang terus berkembang dan

dimodifikasi untuk perangkat seluler.

Android dapat diartikan sebagai sistem operasi untuk telepon seluler seperti telepon pintar. Sistem operasi *Android* menggunakan bahasa pemrograman *Java* dan bersifat *open source* sehingga dapat dikembangkan oleh pihak ketiga (Ilhadi et al., 2022).

2.6. Bahasa Pemrograman

Bahasa Pemrograman merupakan notasi - notasi yang digunakan untuk menulis sebuah program di komputer. Bahasa Pemrograman menjadi sebuah alat komunikasi yang menghubungkan manusia dengan perangkat komputer (R.H. Zer et al., 2022).

Bahasa Pemrograman yang digunakan untuk memberitahu komputer apa yang harus dilakukan. Terdapat berbagai jenis bahasa pemrograman, seperti *C++*, *Java*, *Python*, *dart* dan lainnya, yang digunakan untuk menulis perangkat lunak dan aplikasi yang berbeda. Setiap bahasa pemrograman memiliki sintaks dan aturan yang berbeda-beda dan digunakan untuk tujuan yang berbeda-beda pula.

Menurut (Rochmawati, 2023) Bahasa pemrograman adalah instruksi standar untuk menginstruksikan komputer untuk mencapai fungsi yang diinginkan. Bahasa pemrograman ini adalah seperangkat aturan sintaksis dan semantik yang digunakan untuk mendefinisikan program komputer/laptop.

2.7. Dart

Bahasa *Dart* merupakan salah satu bahasa pemrograman oleh *Google* yang merupakan bahasa general-purpose yang dapat digunakan untuk mengembangkan berbagai platform termasuk *web*, *mobile server*, dan *IOT*. Bahasa ini juga merupakan bahasa standar yang digunakan dari *flutter* (Hanif & Sinambela, 2021).

Dart merupakan salah satu bahasa pemrograman yang dibuat oleh *google* yang dapat digunakan untuk berbagai platform termasuk *Web*, *Mobile Server*, dan *IOT*. *Dart* bersifat *open source* sehingga *dart* dapat dikembangkan dan dipelajari oleh siapapun (Siematauw, 2023).

2.8. Framework

Framework adalah kerangka kerja (sebuah struktur atau rangkaian kerja) untuk membangun aplikasi perangkat lunak. Kerangka kerja ini berisi kumpulan alat dan perpustakaan (*library*) yang sudah tersedia dan terorganisir dengan baik, yang memudahkan pengembang untuk membangun aplikasi dengan cepat.

Framework diterjemahkan sebagai ‘kerangka kerja’. Secara garis besar *framework* adalah menyediakan kerangka kerja untuk membangun suatu sistem sehingga pembangun sistem tidak harus merancang sistem dari awal. Dalam konsep web *framework*, *framework* menyediakan berbagai fungsi, sintaksis, *library*, *extension*, dan *template* siap pakai untuk mempercepat pekerjaan membangun website. Banyak jenis kerangka kerja ditemukan termasuk: *Django*, *Laravel* dan juga *Flutter* (Sinaga & Samsudin, 2021).

2.9. Flutter

Flutter merupakan perangkat UI seluler dari *Google* untuk membuat antarmuka yang bagus untuk aplikasi *mobile* yaitu *android* dan *iOS*. Eksperimen yang dilakukan menggunakan *flutter* sangat cepat dan mudah seperti membuat *UI*, menambahkan fitur-fitur yang diperlukan dan memperbaiki *bug*. Implementasi kode pada *flutter* menggunakan *widget*. *Widget* tersebut dapat berupa komponen visual atau hanya sekedar penampung bagi *widget* yang lainnya. Bahasa pemrograman yang digunakan pada *framework flutter* adalah *dart*. *Dart* merupakan bahasa pemrograman yang dikembangkan oleh *Google* untuk kebutuhan umum. *Dart* dapat digunakan pada beberapa platform diantaranya *flutter*, *web* dan *server*. Sintak yang dimiliki oleh *dart* hampir sama dengan bahasa pemrograman *C++*, *C#*, *java* dan *javascript* (Ernawati et al., 2021).

Menurut (Tjandra & Chandra, 2020) *Flutter* adalah SDK untuk pengembangan aplikasi *mobile* dengan kinerja tinggi, aplikasi untuk *iOS* dan *Android*, dari satu *codebase* (basis kode) yang di buat oleh *Google* dengan lisensi *open source*. Tujuannya adalah memungkinkan pengembang untuk menghadirkan aplikasi berkinerja tinggi yang terasa alami pada platform yang berbeda.

Untuk membuat aplikasi *Flutter*, diperlukan untuk mengerti bahasa *Dart*. *Dart* merupakan bahasa pemrograman yang dibuat oleh *Google* untuk

menggantikan Javascript. *Dart* menggunakan *static typing* yang berarti sebelum memakai variabel, variabel perlu didefinisikan terlebih dahulu. *Dart* bisa berjalan pada semua perangkat juga, pada web *Dart* memakai *dart2js* yang artinya *Dart* diubah ke *Javascript* agar bisa dimengerti browser. Pada perangkat desktop *Dart* memakai *dart2aot* yang mengubah *Dart* menjadi bahasa mesin. Pada perangkat *mobile* *Dart* memakai *Flutter*. Syntax pada bahasa *Dart* sangat mudah dipelajari, karena *Dart* memiliki kemiripan dengan sintaks - sintaks pemrograman lain, seperti Javascript dan Java.

2.10. White Box Testing

Pada pengujian white box ini merupakan pengujian yang dilakukan untuk menguji dan menganalisis kode program bila mana terjadi kesalahan atau tidak. Pengujian *white box* ini dilakukan dengan melihat pure kode tanpa melihat tampilan interface dari halaman aplikasi. Kelebihan dari penggunaan metode *white box testing* adalah dapat memperlihatkan galat pada kode yang dibuat dengan menghapus baris yang tidak diperlukan serta maksimalnya cakupan pengujian aplikasi saat uji coba sebuah skenario (Setiawan et al., 2022).

2.11. Visual Studio Code

Visual Studio Code adalah editor kode sumber yang ringan namun kuat yang berjalan di desktop dan tersedia untuk *Windows*, *macOS*, dan *Linux*. Muncul dengan dukungan built-in untuk *JavaScript*, *TypeScript* dan *Node.js* dan memiliki ekosistem ekstensi yang kaya untuk bahasa lain (seperti *C ++*, *C #*, *Java*, *Python*, *PHP*, *Go*) dan runtime (seperti *.NET* dan *Unity*) (Microsoft, 2020).

2.12. Penulisan Terdahulu

Dalam penelitian ini, peneliti juga meninjau beberapa penelitian sebelumnya untuk perbandingan dan referensi. Selanjutnya, hipotesis kesamaan dengan penelitian ini dihindari. Oleh karena itu, dalam kajian dokumen ini, peneliti mencantumkan hasil penelitian terdahulu sebagai berikut:

Tabel 2. 1. Penelitian Terdahulu

No	Peneliti	Tahun	Penelitian	Hasil
1.	Kindi Mauludi, Aji Gautama Putrada dan Maman Abdurohman	2021	Analisis Kinerja metode <i>Hubeny</i> Formula dalam Navigasi <i>Drone</i> <i>Otonomus</i> Menggunakan <i>GPS</i>	Algoritma <i>hubeny</i> dapat digunakan untuk melakukan analisis apakah <i>drone</i> telah terbang sesuai dengan jalur yang telah ditentukan sebelumnya dan membantu pengguna <i>drone</i> mencapai titik yang diinginkan dengan mudah tanpa harus menerbangkan <i>drone</i> dengan <i>remote control</i> .
2.	Ika Rahayu Wulansari, Andri Suprayogi dan Arief Laila Nugraha	2015	Pembuatan Aplikasi Sebaran Lokasi Fasilitas Kesehatan Penerima BPJS Kesehatan Di Kota Semarang Berbasis <i>Android</i>	Penggunaan algoritma <i>Vincenty</i> sangat berguna dalam membantu masyarakat berdasarkan hasil dari kuesioner yang telah didapatkan dari responden, maka dapat diambil kesimpulan bahwa sebanyak 84% responden mengaku sangat puas dan terbantu dengan adanya aplikasi <i>BPJS</i> <i>Health Maps</i> ini
3.	Arief Kelik Nugroho, Ipung Permadi	2020	Implementasi Jalur Terpendek Menggunakan Ant Colony Optimization	Dari hasil pencarian yang dilakukan maka dapat disimpulkan dalam menentukan nilai optimal menggunakan algoritma <i>ACO</i> dengan variabel dapat dilakukan dengan

Tabel 2. 1. Penelitian Terdahulu (Lanjutan)

				menentukan probabilitas terbesar dan nilai pheromone terkecil berdasarkan 8 tahapan perhitungan algoritma <i>ACO</i> , yakni identifikasi d_{ij} , inisialisasi parameter awal, menentukan jumlah semut, membuat tabulist, menghitung probabilitas dan pheromone, lakukan iterasi, kemudian lakukan update pheromone dan cetak hasil rute terbaik
4.	Mutammimul Ula , Rayhan Rahul Mutuahmi, Sayed Fachrurrazi, Reyhan Achmad Rizal dan Ilham Sahputra	2023	Aplikasi Sistem Informasi Geografis Pencarian Rumah Sakit dan Puskesmas dengan Algoritma Djiktra Berbasis Android	Hasil dari penelitian ini sistem yang bangun dapat menampilkan lokasi rumah sakit dan puskesmas dan sistem ini diharapkan dapat digunakan oleh seluruh elemen masyarakat dan dinas. Kemudian penelitian ini dapat melakukan pendataan rumah sakit dan puskesmas berupa sebuah Aplikasi Sistem Informasi Geografis Rumah Sakit dan Puskesmas Berbasis Android. Keberhasilan dalam menggunakan aplikasi berdasarkan pengujian dari program yang telah dipresentasi tingkat keberhasilan dari masing-masing rute yang dituju dapat langsung ditampilkan.

Tabel 2. 1. Penelitian Terdahulu (Lanjutan)

				<p>Kemudian dapat menampilkan nilai total 1.23 dari rumah sakit cut meutia bukit rata dengan kota lokasi tujuan rumah sakit Dr Hanafiah hasan. Hasil penelitian kedua kota awal dari rumah sakit umum bunda dengan rumah sakit melati dengan jarak dan adanya pemberitahuan menu aplikasi setiap jalan yang dilewati. Tindak lanjut dari penelitian adalah dengan memberikan sebuah model algoritma untuk dapat menghitung efesiensi jarak dan adanya akurasi dari masing-masing model tersebut untuk melihat nilai efektif</p>
5.	Heri Syahputra Ritonga, Deci Irmayani dan Ramadhani Pane	2021	Sistem Informasi Geografis (Gis) Pada Rumah Sakit Di Kabupaten Labuhanbatu Berbasis Web	<p><i>Webgis</i> rumah sakit di kabupaten Labuhanbatu memberikan kemudahan bagi masyarakat yang mencari lokasi rumah sakit. Dengan adanya peta digital ini masyarakat dapat mengetahui rumah sakit yang akan dicari sesuai informasi. Dapat mengetahui nama dokter, informasi poli klinik dan titik koordinat rumah sakit dan</p>

Tabel 2. 1. Penelitian Terdahulu (Lanjutan)

				masyarakat dapat mengetahui jadwal dokter. Dengan sistem informasi geografis diharapkan dapat mengurangi kesalahan informasi mengenai data rumah sakit. Dan sebagai suatu sistem sudah di uji dengan <i>black box</i> agar tidak ada bug atau tombol operasi yang tidak berfungsi.
6.	Ridwan Halim	2023	Perbandingan Algoritma <i>Vincenty</i> Dan <i>Hubeny</i> Dalam Menentukan Jarak Terpendek Pada Rumah Sakit Di Lhokseumawe Berbasis Android	Penggunaan algoritma <i>Vincenty</i> dan dan <i>Hubeny</i> yang kemudian hasil dari kedua algoritma tersebut diolah lagi dengan algoritma <i>Ant Colony</i> menghasilkan hasil yang berbeda. Dimana algoritma <i>Vincenty</i> lebih akurat dibandingkan algoritma <i>Hubeny</i> . Hal ini dikarenakan algoritma <i>Vincenty</i> menggunakan iterasi untuk menghasilkan data yang lebih akurat. Hasil dari keduanya tidak berbeda terlalu jauh yang mana hasil dari algoritma <i>Hubeny</i> masih efektif dalam menentukan jarak antar rumah sakit.

Dari penelitian terdahulu yang peneliti paparkan, peneliti menggunakan data lokasi (koordinat) rumah sakit di Kota Lhokseumawe yang berupa data latitude dan longitude dan diolah menggunakan algoritma *Vincenty* dan *Hubeny* untuk menentukan jarak antar lokasi rumah sakit dimana terdapat perbedaan hasil dari kedua algoritma tersebut. Kemudian hasil dari kedua algoritma tersebut diolah lagi menggunakan algoritma *Ant Colony* untuk membandingkan jarak terpendek dari algoritma *Vincenty* dan *Hubeny*. Setelah mendapatkan hasil yang berupa jarak terpendek dari setiap lokasi rumah sakit, diberikan jalurnya oleh sistem berdasarkan jalan yang tersedia pada maps dan lokasi dari pengguna.

BAB III METODOLOGI PENELITIAN

3.1. Tempat dan Jadwal Penelitian

Dalam pelaksanaan penelitian akan dilakukan pada laboratorium jurusan sistem informasi, pencarian data dilapangan dan studi literatur dari perpustakaan, website dan media lain Fakultas Teknik Universitas Malikussaleh di Universitas Malikussaleh dengan jadwal sebagai berikut:

Tabel 3. 1. Jadwal Penelitian

Kegiatan	Tahun 2023																			
	Maret				April				Mei				Juni				Juli			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Penyusunan Proposal																				
Seminar																				
Revisi Proposal																				
Pengumpulan Data																				
Perancangan Sistem																				
Pembuatan Program																				
Pengujian Program																				
Perbaikan Program																				
Penyusunan Laporan TGA																				
Sidang																				
Revisi Laporan TGA																				

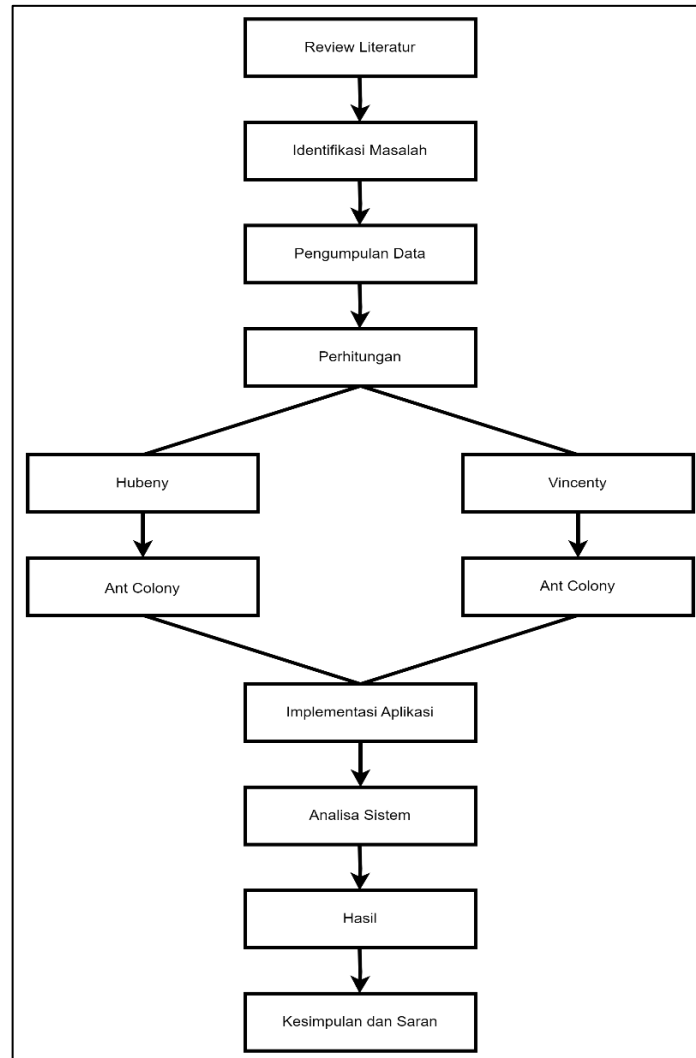
3.2. Metode Penelitian

Metode penelitian yang digunakan adalah deskriptif dan kualitatif. Penelitian deskriptif adalah suatu metode untuk menemukan gambaran, situasi, hal dengan cara mendeskripsikannya sedetail mungkin berdasarkan fakta-fakta yang ada dimana peneliti mengambil data lokasi rumah sakit berupa latitude dan longitude.

Adapun metode dan algoritma yang digunakan dalam penelitian ini berfokus pada 3 metode dan algoritma :

1. Algoritma *Vincenty*
2. Algoritma *Hubeny*
3. Algoritma *Ant Colony*

Untuk pendekatan kualitatif sendiri bertujuan untuk menjabarkan data analisis secara naratif. Pendekatan kualitatif lebih berfokus pada sesuatu yang tidak bisa diukur oleh hitam putih kebenaran, sehingga pada penelitian kualitatif peneliti mengorek data sedalam-dalamnya atas hal-hal tertentu. Pada metode ini dalam membuat kesimpulan, proses pengumpulan dan analisis data dibuat berdasarkan pada konsep perhitungan algoritma *Vincenty*, *Hubeny* dan *Ant Colony*.



Gambar 3. 1 Alur Penelitian

Berikut ini penjelasan mengenai tahapan-tahapan alur penelitian diatas yaitu:

1. Review Literatur

Pada tahapan ini peneliti meninjau penelitian terdahulu sebagai acuan awal gambaran penelitian. Referensi yang didapat diantaranya website, perpustakaan Universitas Malikussaleh dan media lainnya.

2. Identifikasi Masalah

Setelah mendapat tema dan judul untuk penelitian, dilakukan identifikasi masalah pada penelitian yang peneliti teliti dimana terdapat perbedaan hasil perhitungan antara algoritma *Vincenty* dan *Hubeny* dalam

menentukan jarak antara dua lokasi (dua titik). Yang kemudian kedua hasil tersebut digunakan untuk mencari jarak terpendek menggunakan algoritma *Ant Colony*.

3. Pengumpulan Data

Data yang dikumpulkan berupa koordinat dari rumah sakit yang ada di kota Lhokseumawe yang terdiri dari data latitude dan longitude yang didapat dari *Google Maps* berdasarkan data nama rumah sakit yang didapat dari Dinas Kesehatan Kota Lhokseumawe.

4. Perhitungan

Setelah data lokasi rumah sakit didapat, maka dilakukan perhitungan dengan rumus dari algoritma *Vincenty* dan *Hubeny*. Setelah hasil didapat, dilakukan perhitungan kembali dengan algoritma *Ant Colony*.

5. Implementasi Aplikasi

Jika perhitungan berhasil dilakukan, perhitungan rumus menggunakan algoritma *Vincenty*, *Hubeny* dan *Ant Colony* diimplementasikan pada aplikasi android yang dibangun menggunakan *framework Flutter*.

6. Analisa Sistem

Analisa Sistem dilakukan untuk mengetahui kelemahan dari aplikasi yang mengimplementasikan ketiga algoritma tersebut dan melakukan perbaikan yang diperlukan agar aplikasi tersebut dapat berjalan dengan baik.

7. Hasil

Hasil didapat setelah sistem dianalisa dan data telah diolah menjadi keluaran yang diperlukan untuk penelitian. Hasil dari penelitian ini sendiri berupa perbandingan jarak terpendek antara gabungan algoritma *Vincenty – Ant Colony* dan *Hubeny – Ant Colony* dan juga pemberian jalur berdasarkan jalan yang tersedia pada maps dan lokasi pengguna.

8. Kesimpulan dan Saran

Pada tahapan ini, didapati perbedaan jarak antara gabungan algoritma *Vincenty – Ant Colony* dan *Hubeny - Ant Colony* dalam mencari jarak terpendek. Sehingga dari informasi tersebut dapat dijadikan sebagai acuan dalam memberikan saran dan perbaikan untuk penelitian selanjutnya.

3.3. Teknik Pengumpulan Data

Pengumpulan data dilakukan untuk mendapatkan data yang dibutuhkan untuk penelitian. Tergantung pada kebutuhan pertanyaan penelitian, berikut ini adalah teknik pengumpulan data dalam penelitian.

a. Metode penelitian lapangan (*field Research*)

Observasi langsung terhadap subjek penelitian atau data melalui Google Maps dan meninjau langsung Rumah Sakit

b. Studi pustaka (*Library Research*)

Carilah referensi melalui penelitian atau buku-buku pendukung yang berhubungan dengan judul sebagai landasan atau teori. Referensi diambil dari perpustakaan Universitas Malikussaleh, website dan media lain.

3.4. Kebutuhan Sistem

Bagian kebutuhan sistem merupakan tahapan penting dalam pengembangan penelitian. Pada bagian ini akan menjelaskan berbagai kebutuhan sistem yang diperlukan untuk memenuhi tujuan penelitian ini dan dibagi antara kebutuhan perangkat keras dan kebutuhan perangkat lunak.

1. Kebutuhan perangkat keras (*hardware*)

Adapun kebutuhan perangkat keras yang peneliti gunakan dalam penelitian ini adalah sebuah laptop HP Pavilion Gaming 15 Dk1064tx yang berspesifikasi :

- a. *Processor* : Intel Core I5-10300H
- b. *Memmmory* : 16 GB
- c. *Storage* : 512 GB SSD
- d. *Graphics* : Nvidia RTX 2060 max-q
- e. *Os* : Windows 11 home

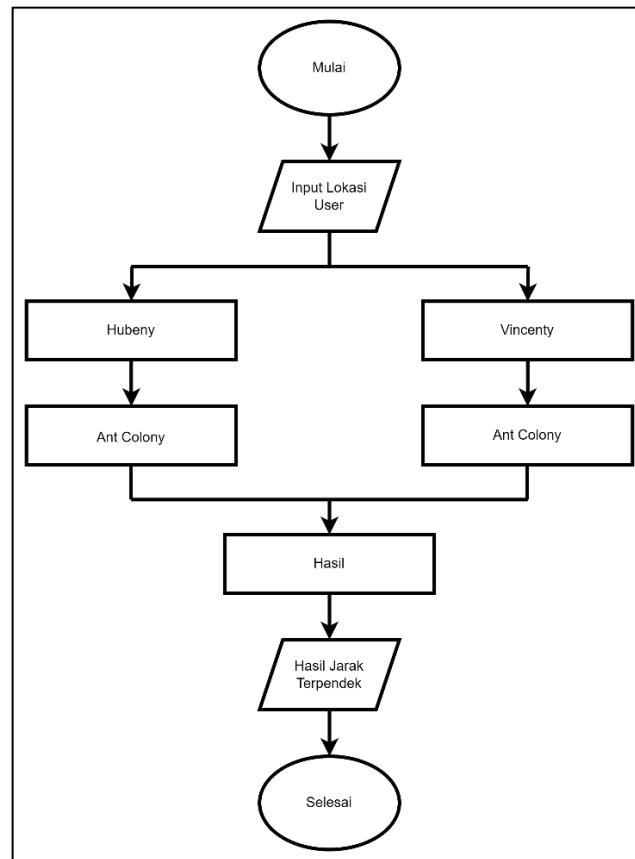
2. Kebutuhan perangkat lunak (*software*)

Adapun kebutuhan perangkat lunak untuk memaksimalkan dan meningkatkan efisiensi hasil dari penelitian ini antara lain:

- a. *Visual Studio Code*
- b. *Microsoft Word*
- c. *Microsoft Excel*

- d. *Mendeley*
- e. *Harzing's Publish and Perish*

3.5. Skema Sistem



Gambar 3. 2. Skema Sistem

Skema sistem pada penelitian secara singkat diilustrasikan pada gambar diatas yang penjelasannya sebagai berikut:

1. Mulai

User memulai sistem, kemudian memilih lokasi user yang dapat dilakukan dengan mengaktifkan lokasi user pada aplikasi. Data berupa latitude dan longitude lokasi user.

2. *Vincenty - Ant Colony*

Data lokasi user diolah bersamaan dengan data lokasi rumah sakit yang berupa data latitude dan longitude menggunakan algoritma *Vincenty*

untuk mencari jarak antara user dan lokasi rumah sakit. Setelah hasil didapatkan data diolah menggunakan algoritma *Ant Colony* untuk menentukan jarak terpendek antar titik.

3. *Hubeny - Ant Colony*

Data lokasi user diolah bersamaan dengan data lokasi rumah sakit yang berupa data latitude dan longitude menggunakan algoritma *Hubeny* untuk mencari jarak antara user dan lokasi rumah sakit. Setelah hasil didapatkan data diolah menggunakan algoritma *Ant Colony* untuk menentukan jarak terpendek antar titik.

4. Hasil

Setelah data latitude dan longitude diolah, maka didapati perbedaan jarak terpendek gabungan algoritma *Vincenty - Ant Colony* dan *Hubeny - Ant Colony*.

5. Selesai

Dari hasil yang tertera akan diberikan jalurnya oleh sistem berdasarkan jalan yang tersedia pada maps. Setelah itu sistem ini berakhir.

3.6. Data Penelitian

Adapun data dalam penelitian ini yaitu data dari rumah sakit yang ada di Kota Lhokseumawe beserta koordinatnya dalam bentuk latitude dan longitude yaitu sebagai berikut :

Tabel 3. 2. Data Penelitian

No	Rumah Sakit	Latitude	Longitude
1.	Rumah Sakit Arun	5.2163871	97.0487424
2.	Rumah Sakit Palang Merah Indonesia	5.1833251	97.1499713
3.	Rumah Sakit Tentara Kesrem	5.1824636	97.1502247
4.	Rumah Sakit Bunda	5.1833024	97.1456549
5.	Rumah Sakit Bunga Melati	5.1753787	97.1465562
6.	Rumah Sakit Ibu Anak ABBY	5.1828459	97.1473555

7.	Rumah Sakit Metro Medical Center	5.1743256	97.1294514
8.	Rumah Sakit Umum Sakinah	5.1880415	97.1442397
9.	Rumah Sakit Umum Yayasan Kasih Ibu	5.1822983	97.1378816
10.	Rumah Sakit Azzuhra	5.1830042	97.1406023

Sumber: Dinas Kesehatan Kota Lhokseumawe

BAB IV

HASIL DAN PEMBAHASAN

4.1. Implementasi Algoritma

Algoritma yang digunakan diimplementasikan dalam bahasa dart dan penjelasan singkat hasil perhitungan dari data penelitian yang sudah dijabarkan.

Tabel 4. 1. Data Lokasi Pengguna dan Rumah Sakit

No	Rumah Sakit	Latitude	Longitude
1.	Lokasi Pengguna (Disesuaikan dimana lokasi pengguna menggunakan aplikasi tersebut)	5.205555392595985	97.10709307522131
2.	Rumah Sakit Arun	5.2163871	97.0487424
3.	Rumah Sakit Palang Merah Indonesia	5.1833251	97.1499713
4.	Rumah Sakit Tentara Kesrem	5.1824636	97.1502247
5.	Rumah Sakit Bunda	5.1833024	97.1456549
6.	Rumah Sakit Bunga Melati	5.1753787	97.1465562
7.	Rumah Sakit Ibu Anak ABBY	5.1828459	97.1473555
8.	Rumah Sakit Metro Medical Center	5.1743256	97.1294514
9.	Rumah Sakit Umum Sakinah	5.1880415	97.1442397
10.	Rumah Sakit Umum Yayasan Kasih Ibu	5.1822983	97.1378816
11.	Rumah Sakit Azzuhra	5.1830042	97.1406023

Sumber: Dinas Kesehatan Kota Lhokseumawe

4.1.1. Perhitungan algoritma *hubeny*

Dalam melakukan perhitungan *hubeny*, diperlukan minimal titik 2 koordinat untuk menghasilkan jarak antar keduanya, juga nilai M dan N dimana keduanya merupakan jarak jari – jari khatulistiwa dan jari – jari kutub bumi dalam meter.

Dimana,

Koordinat 1 = Rumah Sakit Arun

Koordinat 2 = Rumah Sakit Palang Merah Indonesia

lat1 = 5.2163871

lat2 = 5.1833251

lon1 = 97.0487424

lon2 = 97. 0487424

$$P = (\text{lat2} + \text{lat1}) \frac{\pi}{180 * 2}$$

$$M = \frac{6334834}{\sqrt{(1 - 0,006674 * \sin(0.0907546095749115))^2}^3}$$

$$N = \frac{6377397}{\sqrt{1 - 0,006674 * \sin(0.0907546095749115)^2}}$$

s = 11801.8615627249 meter

Maka jarak dari koordinat 1 ke koordinat 2 pada algoritma *hubeny* adalah 11801.8615627249 meter. Untuk lebih lengkap dari hasil perhitungan untuk setiap titik koordinat dirangkum pada tabel dibawah

Tabel 4. 2. Hasil Perhitungan Semua Lokasi Algoritma Hubeny

Lokasi	1	2	3	4	5	6	7	8	9	10	11
1	0.000	6581.969	5344.380	5413.446	4925.119	5493.691	5114.016	4240.457	4544.021	4265.196	4466.183
2	6581.969	0.000	11801.862	11858.367	11348.693	11752.826	11543.427	10083.420	11040.226	10575.737	10831.204
3	5344.380	11801.862	0.000	99.314	478.499	956.744	294.774	2482.872	821.988	1344.998	1039.201
4	5413.446	11858.367	99.314	0.000	515.003	882.662	320.861	2472.396	905.858	1368.411	1068.360
5	4925.119	11348.693	478.499	515.003	0.000	881.824	195.160	2052.248	546.995	868.829	561.073
6	5493.691	11752.826	956.744	882.662	881.824	0.000	830.409	1899.740	1423.514	1228.873	1070.779
7	5114.016	11543.427	294.774	320.861	195.160	830.409	0.000	2197.012	670.329	1051.969	748.827
8	4240.457	10083.420	2482.872	2472.396	2052.248	1899.740	2197.012	0.000	2233.287	1284.720	1564.895
9	4544.021	11040.226	821.988	905.858	546.995	1423.514	670.329	2233.287	0.000	948.710	687.621
10	4265.196	10575.737	1344.998	1368.411	868.829	1228.873	1051.969	1284.720	948.710	0.000	311.538
11	4466.183	10831.204	1039.201	1068.360	561.073	1070.779	748.827	1564.895	687.621	311.538	0.000

Setiap lokasi diinisialkan dengan angka dengan rincian sebagai berikut :

- 1 = Lokasi Pengguna
- 2 = Rumah Sakit Arun
- 3 = Rumah Sakit Palang Merah Indonesia
- 4 = Rumah Sakit Tentara Kesrem
- 5 = Rumah Sakit Bunda
- 6 = Rumah Sakit Bunga Melati
- 7 = Rumah Sakit Ibu Anak ABBY
- 8 = Rumah Sakit Umum Metro Medical Center
- 9 = Rumah Sakit Umum Sakinah
- 10 = Rumah Sakit Yayasan Kasih Ibu
- 11 = Rumah Sakit Azzuhra

4.1.2. Perhitungan algoritma *vincenty*

Dalam melakukan perhitungan *vincenty*, diperlukan minimal titik 2 koordinat untuk menghasilkan jarak antar keduanya, juga nilai M dan N dimana keduanya merupakan jarak jari – jari khatulistiwa dan jari – jari kutub bumi dalam meter.

Dimana,

$a = 6378137$ (jari – jari khatulistiwa bumi dalam meter)

$b = 6356752.314245$ (jari – jari kutub bumi dalam meter)

Koordinat 1 = Rumah Sakit Arun

Koordinat 2 = Rumah Sakit Palang Merah Indonesia

$lat1 = 5.2163871$

$lat2 = 5.1833251$

$lon1 = 97.0487424$

$lon2 = 97.0487424$

$f = 1/298.257223563$

$L = 6356752.314245(lon2-lon1) \frac{\pi}{180}$

$u1 = \text{atan}((1-0.003352810665)\tan(lat1 * \frac{\pi}{180}))$

$u2 = \text{atan}((1-0.003352810665)\tan(lat2 * \frac{\pi}{180}))$

$\sin U1 = \sin(0.090739555162)$

$\sin U2 = \sin(0.090164417488)$

$\cos U1 = \cos(0.090739555162)$

$\cos U2 = \cos(0.090164417488)$

Lakukan iterasi hingga konvergen dengan ketentuan epsilon, biasanya dalam praktek digunakan nilai epsilon (λ) = 10^{-12} sesuai formula yang digunakan dibawah

$$\begin{aligned} \sin \Sigma &= \sqrt{(\cos(U2) * \sin(\lambda))^2 + (\cos(U1) * \sin(U2) - \sin(U1) * \cos(U2) * \cos(\lambda))^2} \\ \cos \Sigma &= \sin(U1). \sin(u2) + \cos(U1). \cos(U2). \cos(\lambda) \\ \Sigma &= \text{atan2}(\sin \Sigma, \cos \Sigma) \\ \sin \alpha &= \cos(U1). \cos(U2). \sin(\lambda) / \sin \Sigma \\ \cos^2 \alpha &= 1 - \sin^2 \alpha \\ \cos 2\alpha_m &= \cos \Sigma - 2. \sin(U1). \sin(U2) / \cos \Sigma \\ C &= f / 16. \cos^2 \alpha. (4 + f. (4 - 3. \cos^2 \alpha)) \\ \lambda P &= \lambda \\ \lambda &= L + (1 - C). f. \sin \alpha. (\Sigma + C. \sin \Sigma. (\cos 2\alpha_m + C. \cos \Sigma. (-1 + 2 * \cos^2 2\alpha_m))) \end{aligned}$$

Tabel 4. 3. Iterasi Algoritma Vincenty

sin λ	0.001766776684	0.001772651932	0.001772671470	0.001772671534
cos λ	0.999998439249	0.999998428851	0.999998428817	0.999998428817
sin Σ	0.001851165373	0.001856727923	0.001856746424	0.001856746485
cos Σ	0.999998286592	0.999998276279	0.999998276245	0.999998276245
Σ	0.001851166430	0.001856728990	0.001856747491	0.001856747552
sin α	0.946625811828	0.946928309488	0.946929310883	0.946929314213
cos ² α	0.103899572382	0.103326776691	0.103324880191	0.103324873885
cos 2α _m	0.842939106809	0.842068433302	0.842065534509	0.842065524870
C	0.000087358138	0.000086876659	0.000086875065	0.000086875060
λP	0.001766777603	0.001772652860	0.001772672398	0.001772672463
λ	0.001772652860	0.001772672398	0.001772672463	0.001772672463
kondisi	0.000005875257	0.00000019538	0.000000000065	0.000000000000
iterasi	1	2	3	4

Setelah nilai didapat saat nilai $\lambda - \lambda P$ berada dibawah 10^{-12} maka:

$$u^2 = \cos^2 \alpha. (a^2 - b^2) / b^2$$

$$A = 1 + u^2 / 16384. (4096 + u^2. (-768 + u^2. (320 - 175. u^2)))$$

$$B = u^2 / 1024. (256 + u^2. (-128 + u^2. (74 - 47. u^2)))$$

$$\Delta \Sigma = B. \sin \Sigma. (\cos 2\alpha_m + B / 4 (\cos \Sigma. (-1 + 2. \cos^2 2\alpha_m) - B / 6. \cos 2\alpha_m. (-3 + 4. \sin^2 \Sigma). (-3 + 4. \cos^2 2\alpha_m)))$$

$$s = b. A (\Sigma - \Delta \Sigma)$$

Tabel 4. 4. Hasil Perhitungan Dua Lokasi Algoritma Vincenty

u ²	0.000696357652
A	1.000174066689
B	0.000174028823
ΔΣ	0.000000272100
s	11803.208812760400

Maka jarak dari koordinat 1 ke koordinat 2 pada algoritma *vincenty* adalah 11803.209 meter. Untuk lebih lengkap dari hasil perhitungan untuk setiap titik koordinat dirangkum pada tabel dibawah

Tabel 4. 5. Hasil Perhitungan Semua Lokasi Algoritma Vincenty

Lokasi	1	2	3	4	5	6	7	8	9	10	11
1	0.000	6578. 861	5351. 729	5420. 921	4932. 816	5502. 501	5121. 650	4250. 998	4550. 938	4273. 826	4474. 405
2	6578. 861	0.000	11803 .209	11859 .720	11349 .987	11754 .155	11544 .743	10084 .547	11041 .490	10576 .937	10832 .436
3	5351. 729	11803 .209	0.000	99.32 4	478.5 54	956.8 39	294.8 08	2483. 153	822.0 77	1345. 154	1039. 322
4	5420. 921	11859 .720	99.32 4	0.000	515.0 63	882.7 51	320.8 98	2472. 676	905.9 55	1368. 570	1068. 484
5	4932. 816	11349 .987	478.5 54	515.0 63	0.000	881.9 08	195.1 82	2052. 476	547.0 48	868.9 29	561.1 38
6	5502. 501	11754 .155	956.8 39	882.7 51	881.9 08	0.000	830.4 89	1899. 961	1423. 652	1229. 006	1070. 889
7	5121. 650	11544 .743	294.8 08	320.8 98	195.1 82	830.4 89	0.000	2197. 259	670.3 97	1052. 091	748.9 14
8	4250. 998	10084 .547	2483. 153	2472. 676	2052. 476	1899. 961	2197. 259	0.000	2233. 526	1284. 856	1565. 064
9	4550. 938	11041 .490	822.0 77	905.9 55	547.0 48	1423. 652	670.3 97	2233. 526	0.000	948.8 12	687.6 92
10	4273. 826	10576 .937	1345. 154	1368. 570	868.9 29	1229. 006	1052. 091	1284. 856	948.8 12	0.000	311.5 74
11	4474. 405	10832 .436	1039. 322	1068. 484	561.1 38	1070. 889	748.9 14	1565. 064	687.6 92	311.5 74	0.000

Setiap lokasi diinisialkan dengan angka dengan rincian sebagai berikut :

- 1 = Lokasi Pengguna
- 2 = Rumah Sakit Arun
- 3 = Rumah Sakit Palang Merah Indonesia
- 4 = Rumah Sakit Tentara Kesrem
- 5 = Rumah Sakit Bunda
- 6 = Rumah Sakit Bunga Melati
- 7 = Rumah Sakit Ibu Anak ABBY

8 = Rumah Sakit Umum Metro Medical Center

9 = Rumah Sakit Umum Sakinah

10 = Rumah Sakit Yayasan Kasih Ibu

11 = Rumah Sakit Azzuhra

Penulis juga merangkum seberapa banyak iterasi yang diperlukan dalam menyelesaikan perjalanan pada suatu koordinat ke koordinat lainnya.

Tabel 4. 6. Jumlah Iterasi Yang Diperlukan

Iterasi	1	2	3	4	5	6	7	8	9	10	11
1	0	4	4	4	4	4	4	4	4	4	4
2	4	0	4	4	4	4	4	4	4	4	4
3	4	4	0	3	4	4	4	4	4	4	4
4	4	4	3	0	4	4	4	4	4	4	4
5	4	4	4	4	0	3	4	4	3	4	4
6	4	4	4	4	3	0	3	4	4	4	4
7	4	4	4	4	4	3	0	4	4	4	4
8	4	4	4	4	4	4	4	0	4	4	4
9	4	4	4	4	3	4	4	4	0	4	4
10	4	4	4	4	4	4	4	4	4	0	4
11	4	4	4	4	4	4	4	4	4	4	0

4.1.3. Perhitungan algoritma *ant colony*

Dalam melakukan perhitungan *ant colony* diperlukan data jarak antar titik koordinat, dalam hal ini kita akan memakai data hasil dari algoritma *hubeny*. Untuk parameter lainnya yaitu jumlah semut (m), α , β , ρ dan Q .

Langkah 1

1. Tentukan jarak dari *node i* ke *node j* atau matriks d_{ij} .
2. Inisialisasikan nilai parameter yang diperlukan seperti jumlah semut, α , β , ρ , Q dan jumlah iterasi yang diperlukan.

$$m = 25$$

$$\alpha = 1$$

$$\beta = 2$$

$$\rho = 0.5$$

$$Q = 1$$

3. Tentukan nilai feromon awal atau τ_{ij} .
4. Cari nilai visibilitas atau η_{ij} yang didapat dari $1/d_{ij}$

Tabel 4. 7. Nilai Feromon Awal

τ_{ij}	1	2	3	4	5	6	7	8	9	10	11
1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1

Tabel 4. 8. Visibilitas Tiap Jarak Antar Titik

visibilitas	1	2	3	4	5	6	7	8	9	10	11
1	0.000 000	0.000 152	0.000 187	0.000 184	0.000 203	0.000 182	0.000 195	0.000 235	0.000 220	0.000 234	0.000 224
2	0.000 152	0.000 000	0.000 085	0.000 084	0.000 088	0.000 085	0.000 087	0.000 099	0.000 091	0.000 095	0.000 092
3	0.000 187	0.000 085	0.000 000	0.010 069	0.002 090	0.001 045	0.003 392	0.000 403	0.001 217	0.000 743	0.000 962
4	0.000 184	0.000 084	0.010 069	0.000 000	0.001 942	0.001 133	0.003 117	0.000 404	0.001 104	0.000 731	0.000 936
5	0.000 203	0.000 088	0.002 090	0.001 942	0.000 000	0.001 134	0.005 124	0.000 487	0.001 828	0.001 151	0.001 782
6	0.000 182	0.000 085	0.001 045	0.001 133	0.001 134	0.000 000	0.001 204	0.000 526	0.000 702	0.000 814	0.000 934
7	0.000 195	0.000 087	0.003 392	0.003 117	0.005 124	0.001 204	0.000 000	0.000 455	0.001 492	0.000 951	0.001 335

8	0.000 235	0.000 099	0.000 403	0.000 404	0.000 487	0.000 526	0.000 455	0.000 000	0.000 448	0.000 778	0.000 639
9	0.000 220	0.000 091	0.001 217	0.001 104	0.001 828	0.000 702	0.001 492	0.000 448	0.000 000	0.001 054	0.001 454
10	0.000 234	0.000 095	0.000 743	0.000 731	0.001 151	0.000 814	0.000 951	0.000 778	0.001 054	0.000 000	0.003 210
11	0.000 224	0.000 092	0.000 962	0.000 936	0.001 782	0.000 934	0.001 335	0.000 639	0.001 454	0.003 210	0.000 000

Langkah 2

1. Probabilitas kunjungan semut dari titik i ke titik j digunakan persamaan berikut :

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} ([\tau_{il}]^\alpha \cdot [\eta_{il}]^\beta)} & \text{jika } j \in N_i^k \\ 0 & \text{lainnya} \end{cases}$$

2. Tentukan bilangan random r, dimana $0 < r < 1$
3. Temukan bilangan yang lebih besar dari hasil persamaan diatas dan lakukan kembali persamaan diatas sampai semua matriks jarak semua titik menjadi 0.
4. Maka akan didapati jalur dan jarak yang dilalui semut tersebut.
5. Lakukan langkah 2 ini sebanyak jumlah semut.

Tabel 4. 9. Probabilitas Dari Titik 1

Titik	$[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta$	$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} ([\tau_{il}]^\alpha \cdot [\eta_{il}]^\beta)} \\ 0 \end{cases}$	probabilitas kumulatif
2	0.000000023110	0.055996	0.055996
3	0.000000034923	0.084619	0.140616
4	0.000000034037	0.082473	0.223089
5	0.000000041106	0.099602	0.322690
6	0.000000033035	0.080045	0.402736
7	0.000000038131	0.092393	0.495128
8	0.000000055349	0.134112	0.629240
9	0.000000048294	0.117019	0.746259
10	0.000000054760	0.132685	0.878944

11	0.000000049960	0.121056	1.000000
----	----------------	----------	----------

total = 0.000000412705

$r = 0.580427479785714$

Karena titik 8 > r, maka titik selanjutnya adalah titik 8.

Tabel 4. 10. Probabilitas Dari Titik 8

Titik	$[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta$	$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} ([\tau_{ik}]^\alpha \cdot [\eta_{ik}]^\beta)} & \text{jika } j \in N_i^k \\ 0 & \text{lainnya} \end{cases}$	probabilitas kumulatif
2	0.000000009835	0.004329	0.004329
3	0.000000162215	0.071396	0.075725
4	0.000000163593	0.072002	0.147727
5	0.000000237433	0.104501	0.252228
6	0.000000277084	0.121953	0.374181
7	0.000000207174	0.091183	0.465365
9	0.000000200498	0.088245	0.553610
10	0.000000605875	0.266664	0.820274
11	0.000000408347	0.179726	1.000000

Total = 0.000002272055

$r = 0.672465598426017$

Karena titik 10 > r, maka titik selanjutnya adalah 10.

Tabel 4. 11. Probabilitas Dari Titik 10

Titik	$[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta$	$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} ([\tau_{ik}]^\alpha \cdot [\eta_{ik}]^\beta)} & \text{jika } j \in N_i^k \\ 0 & \text{lainnya} \end{cases}$	probabilitas kumulatif
2	0.000000008941	0.000581	0.000581
3	0.000000552785	0.035894	0.036474
4	0.000000534031	0.034676	0.071150
5	0.000001324744	0.086018	0.157168
6	0.000000662195	0.042998	0.200166
7	0.000000903637	0.058675	0.258841
9	0.000001111048	0.072143	0.330984
11	0.000010303323	0.669016	1.000000

Total = 0.000015400704

$r = 0.531135100205301$

Karena titik 11 > r, maka titik selanjutnya adalah 11.

Tabel 4. 12. Probabilitas Dari Titik 11

Titik	$[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta$	$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} ([\tau_{ik}]^\alpha \cdot [\eta_{ik}]^\beta)} & \text{jika } j \in N_i^k \\ 0 & \text{lainnya} \end{cases}$	probabilitas kumulatif
2	0.00000008524	0.000874	0.000874
3	0.000000925978	0.094897	0.095771
4	0.000000876122	0.089788	0.185559
5	0.000003176591	0.325547	0.511106
6	0.000000872169	0.089383	0.600489
7	0.000001783351	0.182764	0.783252
9	0.000002114958	0.216748	1.000000

Total = 0.000009757693

$r = 0.072465598426017$

Karena titik 4 > r, maka titik selanjutnya adalah 4.

Tabel 4. 13. Probabilitas Dari Titik 4

Titik	$[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta$	$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} ([\tau_{ik}]^\alpha \cdot [\eta_{ik}]^\beta)} & \text{jika } j \in N_i^k \\ 0 & \text{lainnya} \end{cases}$	probabilitas kumulatif
2	0.00000007111	0.000061	0.000061
3	0.000101386518	0.863750	0.863811
5	0.000003770333	0.032121	0.895932
6	0.000001283544	0.010935	0.906867
7	0.000009713307	0.082751	0.989618
9	0.000001218652	0.010382	1.000000

Total = 0.000117379465

$r = 0.624239525825483$

Karena titik 3 > r, maka titik selanjutnya adalah 3.

Tabel 4. 14. Probabilitas Dari Titik 3

Titik	$[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta$	$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} ([\tau_{ik}]^\alpha \cdot [\eta_{ik}]^\beta)} & \text{jika } j \in N_i^k \\ 0 & \text{lainnya} \end{cases}$	probabilitas kumulatif
2	0.000000007180	0.000389	0.000389
5	0.000004367557	0.236650	0.237039
6	0.000001092468	0.059194	0.296233
7	0.000011508542	0.623574	0.919807
9	0.000001480026	0.080193	1.000000

Total = 0.000018455771

$r = 0.784915115683033$

Karena titik $7 > r$, maka titik selanjutnya adalah 7.

Tabel 4. 15. Probabilitas Dari Titik 7

Titik	$[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta$	$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} ([\tau_{ik}]^\alpha \cdot [\eta_{ik}]^\beta)} & \text{jika } j \in N_i^k \\ 0 & \text{lainnya} \end{cases}$	probabilitas kumulatif
2	0.000000007505	0.000251	0.000251
5	0.000026255449	0.876977	0.877227
6	0.000001450158	0.048438	0.925665
9	0.000002225480	0.074335	1.000000

Total = 0.000029938591

$r = 0.142698531456782$

Karena titik $5 > r$, maka titik selanjutnya adalah 5.

Tabel 4. 16. Probabilitas Dari Titik 5

Titik	$[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta$	$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} ([\tau_{ik}]^\alpha \cdot [\eta_{ik}]^\beta)} & \text{jika } j \in N_i^k \\ 0 & \text{lainnya} \end{cases}$	probabilitas kumulatif
2	0.000000007764	0.001675	0.001675
6	0.000001285987	0.277394	0.279069
9	0.000003342206	0.720931	1.000000

Total = 0.000004635957

$r = 0.982452369566423$

Karena titik $9 > r$, maka titik selanjutnya adalah 9.

Tabel 4. 17. Probabilitas Dari Titik 9

Titik	$[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta$	$p_{ij}^k(t) = \begin{cases} [\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta & \text{jika } j \in N_i^k \\ \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} ([\tau_{il}]^\alpha \cdot [\eta_{il}]^\beta)} & \\ 0 & \end{cases}$	probabilitas kumulatif
2	0.000000008204	0.016353	0.016353
6	0.000000493488	0.983647	1.000000

Total = 0.000000501692

$r = 0.752632458962175$

Karena titik 6 > r, maka titik selanjutnya adalah 6.

Setelah semut 1 menempuh jalur secara acak untuk sampai kesemua titik, maka didapati bahwa jalur yang ditempuh oleh semut 1 yaitu, 1-6-3-4-7-5-9-10-11-8-2 dan jarak yang ditempuhnya adalah 20529.696532 meter. Titik diatas adalah inisial dari titik koordinat yang sudah dipaparkan diatas dengan rincian sebagai berikut :

- 1 = Lokasi Pengguna
- 2 = Rumah Sakit Arun
- 3 = Rumah Sakit Palang Merah Indonesia
- 4 = Rumah Sakit Tentara Kesrem
- 5 = Rumah Sakit Bunda
- 6 = Rumah Sakit Bunga Melati
- 7 = Rumah Sakit Ibu Anak ABBY
- 8 = Rumah Sakit Umum Metro Medical Center
- 9 = Rumah Sakit Umum Sakinah
- 10 = Rumah Sakit Yayasan Kasih Ibu
- 11 = Rumah Sakit Azzuhra

Langkah 3

1. Untuk memperbarui nilai intensitas feromon yang telah terbentuk pada suatu lintasan gunakan persamaan dibawah :

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \Delta\tau_{ij}$$

$$\Delta\tau_{ij} = \sum_k^m \Delta\tau_{ij}^k$$

$$= \frac{Q}{L_k}$$

Dari langkah 2 yang sudah dilakukan didapati pula jalur, jarak dan $\Delta\tau_{ij}^k$ dari 25 semut.

Tabel 4. 18. Hasil Dari Semua Semut Di Iterasi Pertama

Semut	Jalur	Jarak (L_k)	$\Delta\tau_{ij}^k$
1	1-6-3-4-7-5-9-10-11-8-2	20529.696532	0.000049
2	1-4-3-7-5-6-10-8-11-9-2	22697.920142	0.000044
3	1-10-11-4-3-5-7-6-9-8-2	20997.020260	0.000048
4	1-11-10-3-4-7-5-9-6-8-2	20699.624450	0.000048
5	1-6-8-10-11-9-7-3-4-5-2	22613.836548	0.000044
6	1-9-5-7-3-4-8-10-11-6-2	22579.128198	0.000044
7	1-11-10-8-6-7-3-4-5-9-2	21296.825336	0.000047
8	1-11-10-9-3-4-7-5-6-8-2	20036.633865	0.000050
9	1-4-3-7-5-9-11-10-8-6-2	22493.205312	0.000044
10	1-10-11-9-5-7-4-3-6-2-8	29228.087063	0.000034
11	1-11-3-4-7-5-6-9-10-8-2	20750.792811	0.000048
12	1-10-11-5-7-3-4-6-9-8-2	20358.255799	0.000049
13	1-5-7-4-3-11-9-10-6-8-2	21435.355299	0.000047
14	1-4-3-7-5-11-10-8-9-6-2	23576.719558	0.000042
15	1-11-5-7-4-3-9-6-10-8-2	20492.993981	0.000049
16	1-9-7-5-11-10-8-3-4-6-2	22791.122823	0.000044
17	1-5-7-3-4-9-11-10-8-6-2	22364.020013	0.000045
18	1-11-10-3-4-7-5-8-9-6-2	24207.862930	0.000041
19	1-5-7-4-3-9-11-10-6-8-2	20580.961106	0.000049
20	1-8-10-11-4-3-7-5-9-6-2	21227.921955	0.000047
21	1-5-7-3-4-6-11-10-8-9-2	22344.921859	0.000045
22	1-4-3-7-5-9-11-10-6-8-2	20767.930182	0.000048
23	1-4-3-7-10-11-5-9-6-8-2	21692.838130	0.000046
24	1-3-4-7-5-9-11-10-6-8-2	20724.831087	0.000048
25	1-11-9-7-3-4-5-6-10-8-2	20219.946018	0.000049

dimana

$$Q = 1$$

$$\rho = 0.5$$

Kemudian dari parameter yang tersedia lakukan pembaruan feromon (τ_{ij}).

Tabel 4. 19. Nilai Peromon Terbaru

τ_{ij}	1	2	3	4	5	6	7	8	9	10	11
1	0.500 000	0.500 000	0.500 048	0.500 225	0.500 185	0.500 093	0.500 000	0.500 047	0.500 088	0.500 131	0.500 333
2	0.500 000	0.500 000	0.500 000	0.500 000	0.500 044	0.500 342	0.500 000	0.500 662	0.500 136	0.500 000	0.500 000
3	0.500 048	0.500 000	0.500 000	0.501 150	0.500 048	0.500 083	0.500 596	0.500 044	0.500 147	0.500 090	0.500 095
4	0.500 225	0.500 000	0.501 150	0.500 000	0.500 141	0.500 138	0.500 463	0.500 044	0.500 045	0.500 000	0.500 095
5	0.500 185	0.500 044	0.500 048	0.500 141	0.500 000	0.500 192	0.500 963	0.500 041	0.500 457	0.500 000	0.500 230
6	0.500 093	0.500 342	0.500 083	0.500 138	0.500 192	0.500 000	0.500 095	0.500 516	0.500 419	0.500 334	0.500 089
7	0.500 000	0.500 000	0.500 596	0.500 463	0.500 963	0.500 095	0.500 000	0.500 000	0.500 138	0.500 046	0.500 000
8	0.500 047	0.500 662	0.500 044	0.500 044	0.500 041	0.500 516	0.500 000	0.500 000	0.500 225	0.500 593	0.500 093
9	0.500 088	0.500 136	0.500 147	0.500 045	0.500 457	0.500 419	0.500 138	0.500 225	0.500 000	0.500 193	0.500 453
10	0.500 131	0.500 000	0.500 090	0.500 000	0.500 000	0.500 334	0.500 046	0.500 593	0.500 193	0.500 000	0.500 913
11	0.500 333	0.500 000	0.500 095	0.500 095	0.500 230	0.500 089	0.500 000	0.500 093	0.500 453	0.500 913	0.500 000

Gunakan feromon (τ_{ij}) yang baru dan ulangi langkah 2 dan 3 sebanyak 100 kali. Pada iterasi pertama diatas diketahui jalur terpendek dari 10 semut adalah pada semut 8 yaitu, 1-8-10-9-4-3-5-6-7-2 dengan jarak 27912.888929 meter. Untuk lebih lengkap dari hasil perhitungan jalur terpendek dan jarak terpendek pada setiap iterasi dirangkum pada tabel dibawah.

Tabel 4. 20. Hasil Jarak Terpendek Pada 100 Iterasi

Iterasi	Semut	Jalur Terbaik	Jarak Terbaik
1	7	1-9-5-7-4-3-6-11-10-8-2	19420.111106
2	9	1-10-11-7-3-4-9-5-6-8-2	20045.814800
3	18	1-5-6-4-3-7-9-11-10-8-2	20128.623468
4	21	1-9-3-4-7-5-11-10-6-8-2	20072.575089
5	3	1-9-5-4-3-7-6-10-11-8-2	20025.812740
6	7	1-10-11-9-5-3-4-7-6-8-2	19531.915439
7	21	1-10-11-9-5-4-3-7-6-8-2	19542.333927
8	3	1-10-11-5-7-4-3-9-6-8-2	19990.131863
9	23	1-10-11-9-3-4-5-7-6-8-2	19717.715844
10	18	1-5-7-3-4-6-9-11-10-8-2	20195.144978
11	9	1-10-11-9-5-7-3-4-6-8-2	19274.741553
12	1	1-9-5-7-3-4-6-11-10-8-2	19319.943551
13	23	1-10-11-9-5-7-4-3-6-8-2	19374.909107
14	3	1-10-11-9-5-7-4-3-6-8-2	19374.909107
15	13	1-10-11-6-4-3-7-5-9-8-2	19991.440389
16	8	1-8-6-4-3-7-5-9-11-10-2	19744.259095
17	4	1-10-11-5-7-3-4-9-6-8-2	20047.915640
18	11	1-10-11-9-5-7-4-3-6-8-2	19374.909107
19	6	1-8-10-11-6-4-3-7-5-9-2	19976.884674
20	19	1-10-11-9-3-4-7-5-6-8-2	19574.985254
21	21	1-9-5-7-4-3-6-11-10-8-2	19420.111106
22	21	1-10-11-9-5-7-3-4-6-8-2	19274.741553
23	13	1-8-6-3-4-7-5-9-11-10-2	19844.426649
24	19	1-10-11-9-5-7-3-4-6-8-2	19274.741553
25	20	1-10-11-9-5-7-3-4-6-8-2	19274.741553
26	25	1-8-6-4-3-7-5-9-11-10-2	19744.259095
27	23	1-10-11-5-9-3-4-7-6-8-2	19748.860380
28	2	1-10-11-5-7-4-3-9-6-8-2	19990.131863
29	3	1-10-11-5-7-4-3-9-6-8-2	19990.131863
30	14	1-10-11-9-5-7-4-3-6-8-2	19374.909107
31	16	1-10-11-5-7-4-3-9-6-8-2	19990.131863
32	1	1-8-10-11-6-4-3-7-5-9-2	19976.884674
33	23	1-8-10-11-6-4-3-7-5-9-2	19976.884674
34	10	1-10-11-9-5-7-3-4-6-8-2	19274.741553
35	1	1-8-10-11-6-4-3-7-5-9-2	19976.884674
36	22	1-10-11-5-7-4-3-9-6-8-2	19990.131863
37	1	1-10-11-9-5-7-3-4-6-8-2	19274.741553
38	15	1-10-11-9-5-7-3-4-6-8-2	19274.741553
39	14	1-10-11-5-9-4-3-7-6-8-2	19806.644157

40	9	1-8-6-4-3-7-5-9-11-10-2	19744.259095
41	6	1-10-11-5-7-4-3-9-6-8-2	19990.131863
42	1	1-8-10-11-5-7-3-4-6-9-2	20343.700085
43	17	1-8-6-4-3-7-5-9-11-10-2	19744.259095
44	13	1-10-11-9-5-7-3-4-6-8-2	19274.741553
45	9	1-10-11-6-4-3-7-5-9-8-2	19991.440389
46	17	1-10-11-9-5-7-3-4-6-8-2	19274.741553
47	23	1-10-11-9-5-7-3-4-6-8-2	19274.741553
48	12	1-8-6-4-3-7-5-9-11-10-2	19744.259095
49	9	1-10-11-9-5-7-3-4-6-8-2	19274.741553
50	7	1-8-10-11-9-5-7-3-4-6-2	20306.322987
51	24	1-10-11-9-5-7-4-3-6-8-2	19374.909107
52	7	1-10-11-9-5-7-3-4-6-8-2	19274.741553
53	19	1-10-11-5-7-4-3-9-6-8-2	19990.131863
54	4	1-10-11-5-7-4-3-9-6-8-2	19990.131863
55	7	1-8-10-11-5-7-3-4-6-9-2	20343.700085
56	16	1-8-10-11-9-5-7-3-4-6-2	20306.322987
57	12	1-8-10-11-6-4-3-7-5-9-2	19976.884674
58	5	1-8-10-11-5-7-3-4-6-9-2	20343.700085
59	17	1-10-11-5-7-4-3-9-6-8-2	19990.131863
60	11	1-10-11-5-7-4-3-9-6-8-2	19990.131863
61	8	1-8-10-11-6-4-3-7-5-9-2	19976.884674
62	7	1-10-11-5-7-4-3-9-6-8-2	19990.131863
63	18	1-8-10-11-6-4-3-7-5-9-2	19976.884674
64	12	1-10-9-11-5-7-3-4-6-8-2	19925.999737
65	12	1-10-11-5-7-4-3-9-6-8-2	19990.131863
66	4	1-8-10-11-5-7-3-4-6-9-2	20343.700085
67	18	1-10-11-6-4-3-7-5-9-8-2	19991.440389
68	15	1-10-11-5-7-3-4-9-6-8-2	20047.915640
69	3	1-10-11-9-5-7-3-4-6-8-2	19274.741553
70	8	1-8-6-4-3-7-5-9-11-10-2	19744.259095
71	10	1-10-11-5-7-3-4-9-6-8-2	20047.915640
72	12	1-10-11-5-7-3-4-9-6-8-2	20047.915640
73	5	1-8-10-11-9-5-7-3-4-6-2	20306.322987
74	16	1-8-10-11-6-4-3-7-5-9-2	19976.884674
75	3	1-10-11-5-7-3-4-9-6-8-2	20047.915640
76	22	1-10-11-5-7-3-4-9-6-8-2	20047.915640
77	1	1-8-10-11-5-7-3-4-6-9-2	20343.700085
78	3	1-8-10-11-5-7-3-4-6-9-2	20343.700085
79	4	1-8-10-11-5-7-3-4-6-9-2	20343.700085
80	19	1-10-11-5-7-3-4-9-6-8-2	20047.915640

81	13	1-10-11-5-7-3-4-9-6-8-2	20047.915640
82	2	1-8-10-11-5-7-3-4-6-9-2	20343.700085
83	1	1-8-10-11-5-7-3-4-6-9-2	20343.700085
84	1	1-8-10-11-5-7-3-4-6-9-2	20343.700085
85	1	1-8-10-11-5-7-3-4-6-9-2	20343.700085
86	15	1-8-10-11-6-4-3-7-5-9-2	19976.884674
87	13	1-10-11-5-7-3-4-9-6-8-2	20047.915640
88	16	1-10-11-6-4-3-7-5-9-8-2	19991.440389
89	2	1-8-10-11-5-7-3-4-6-9-2	20343.700085
90	22	1-8-10-11-6-4-3-7-5-9-2	19976.884674
91	1	1-8-10-11-5-7-3-4-6-9-2	20343.700085
92	1	1-8-10-11-5-7-3-4-6-9-2	20343.700085
93	11	1-10-11-6-4-3-7-5-9-8-2	19991.440389
94	15	1-10-11-6-4-3-7-5-9-8-2	19991.440389
95	2	1-8-10-11-5-7-3-4-6-9-2	20343.700085
96	1	1-8-10-11-5-7-3-4-6-9-2	20343.700085
97	1	1-8-10-11-5-7-3-4-6-9-2	20343.700085
98	1	1-8-10-11-5-7-3-4-6-9-2	20343.700085
99	2	1-8-10-11-5-7-3-4-6-9-2	20343.700085
100	1	1-8-10-11-5-7-3-4-6-9-2	20343.700085

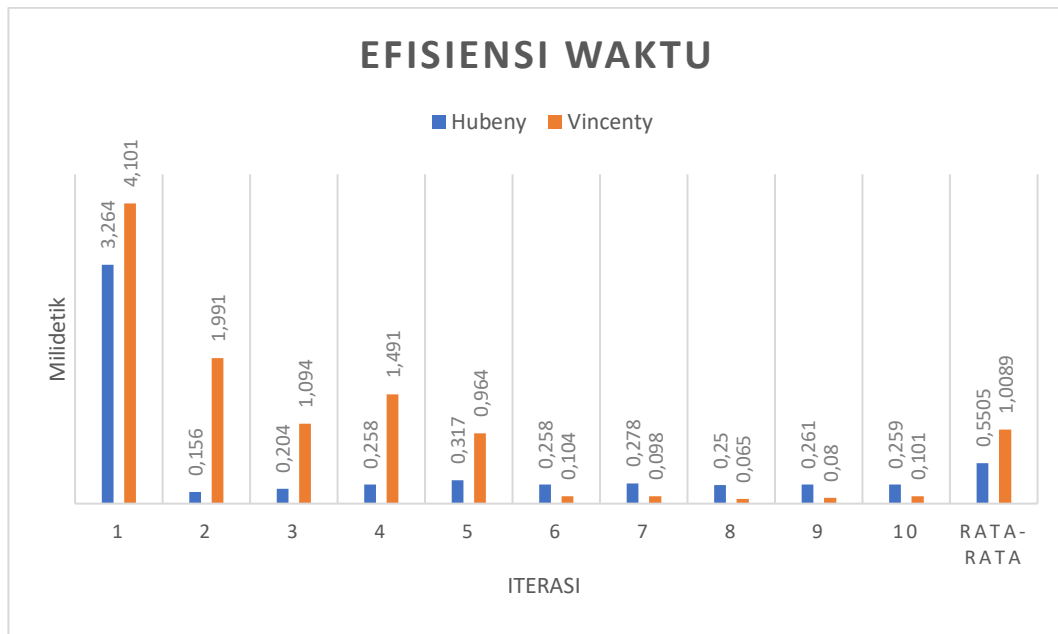
Dari 100 iterasi diatas didapati bahwa jalur terpendek yang dilalui adalah 1-10-11-9-5-7-3-4-6-8-2 dengan jarak 19274.741553 meter. Perlu dicatat bahwa hasil iterasi 1 pada tabel diatas berbeda dengan hasil iterasi 1 yang dipaparkan oleh penulis. Hal ini karena setiap kali terjadi perpindahan jarak pada setiap semut terdapat angka acak (*random*) yang selalu berubah saat program dijalankan ulang.

4.1.4. Analisa perbandingan algoritma *vincenty* dan *hubeny*

Perbandingan dilakukan dalam dua hal yaitu uji efisiensi waktu dan perbedaan persentase jarak antara kedua algoritma untuk menemukan beberapa poin perbedaan yang menjadi fokus penelitian ini.

1. Uji efisiensi waktu

Pada uji ini kedua algoritma di jalankan berulang (dilakukan iterasi) sebanyak 10 kali untuk mengetahui seberapa cepat algoritma dalam mengeksekusi kode yang dijalankan.



Gambar 4. 1. Perbandingan Efisiensi Waktu

Dari chart diatas dapat disimpulkan bahwa program algoritma *hubeny* lebih efisien dibandingkan algoritma *vincenty* dalam kecepatan pengeksesuan program dengan rata-rata perbedaan dalam waktu yaitu algoritma *hubeny* lebih cepat 0.4584 milidetik dibandingkan algoritma *vincenty*. Hal ini dikarenakan formula dari algoritma *hubeny* tidak memerlukan iterasi seperti formula dari algoritma *vincenty* untuk menghasilkan output jarak. Dari chart diatas juga disimpulkan bahwa setiap iterasi yang dilakukan semakin cepat pula kedua algoritma tersebut dieksekusi yang disebabkan program dalam bahasa dart menyimpan proses pengeksesuan program tersebut sehingga program tidak perlu dicari ulang pada file program.

2. Uji perbedaan persentase jarak

Dalam uji ini hasil dari kedua algoritma dan juga jarak dari hasil perhitungan dengan *api Google Map* dibandingkan untuk mencari tahu berapa persen perbedaan tiap jarak antar titik koordinat dan rata – rata perbedaan persentase dari semua hasil.

Tabel 4. 21. Perbedaan Persentase Jarak Algoritma *Hubeny* dan *Vincenty*

Persen (%)	1	2	3	4	5	6	7	8	9	10	11
1	0.0000 00%	0.0104 39%	0.0102 92%	0.0102 82%	0.0102 58%	0.0101 52%	0.0102 65%	0.0098 88%	0.0103 17%	0.0101 56%	0.0102 02%
2	0.0104 39%	0.0000 00%	0.0103 91%	0.0103 87%	0.0103 83%	0.0103 45%	0.0103 84%	0.0102 88%	0.0104 03%	0.0103 63%	0.0103 73%
3	0.0102 92%	0.0103 91%	0.0000 00%	0.0096 52%	0.0104 92%	0.0097 23%	0.0104 63%	0.0103 49%	0.0101 24%	0.0104 86%	0.0104 91%
4	0.0102 82%	0.0103 87%	0.0096 52%	0.0000 00%	0.0104 63%	0.0097 74%	0.0104 77%	0.0103 75%	0.0100 68%	0.0104 93%	0.0104 90%
5	0.0102 58%	0.0103 83%	0.0104 92%	0.0104 63%	0.0000 00%	0.0095 91%	0.0104 32%	0.0102 82%	0.0096 54%	0.0104 78%	0.0104 89%
6	0.0101 52%	0.0103 45%	0.0097 23%	0.0097 74%	0.0095 91%	0.0000 00%	0.0095 90%	0.0104 96%	0.0096 09%	0.0101 41%	0.0099 28%
7	0.0102 65%	0.0103 84%	0.0104 63%	0.0104 77%	0.0104 32%	0.0095 90%	0.0000 00%	0.0103 28%	0.0098 21%	0.0104 90%	0.0104 92%
8	0.0098 88%	0.0102 88%	0.0103 49%	0.0103 75%	0.0102 82%	0.0104 96%	0.0103 28%	0.0000 00%	0.0100 72%	0.0100 65%	0.0101 52%
9	0.0103 17%	0.0104 03%	0.0101 24%	0.0100 68%	0.0096 54%	0.0096 09%	0.0098 21%	0.0100 72%	0.0000 00%	0.0100 83%	0.0098 93%
10	0.0101 56%	0.0103 63%	0.0104 86%	0.0104 93%	0.0104 78%	0.0101 41%	0.0104 90%	0.0100 65%	0.0100 83%	0.0000 00%	0.0104 36%
11	0.0102 02%	0.0103 73%	0.0104 91%	0.0104 90%	0.0104 89%	0.0099 28%	0.0104 92%	0.0101 52%	0.0098 93%	0.0104 36%	0.0000 00%

Dari tabel diatas dapat disimpulkan bahwa perbedaan jarak antara algoritma *hubeny* dan *vincenty* berada dibawah 1% dengan rata-rata perbedaannya 0.009289% dimana hasil jarak algoritma *hubeny* sedikit lebih pendek daripada hasil jarak algoritma *vincenty*.

Tabel 4. 22. Perbedaan Persentase Jarak Algoritma *Hubeny* dan *Google Map*

Perse n (%)	1	2	3	4	5	6	7	8	9	10	11
1	0,0000 00%	13,626 916%	26,656 002%	26,011 962%	26,174 071%	20,595 429%	26,091 441%	2,7327 66%	34,317 715%	26,231 795%	24,757 009%
2	15,087 465%	0,0000 00%	21,430 115%	21,211 296%	21,221 655%	19,791 584%	21,225 987%	16,630 759%	24,654 737%	21,764 749%	20,771 619%
3	30,031 668%	22,569 332%	0,0000 00%	68,371 351%	26,383 998%	1,3664 46%	41,972 904%	30,934 753%	37,156 473%	25,853 727%	26,087 270%
4	29,403 964%	22,351 442%	68,371 351%	0,0000 00%	50,951 523%	11,733 557%	64,662 538%	31,814 131%	46,994 522%	25,790 307%	40,843 181%
5	29,849 092%	22,401 310%	26,383 998%	46,520 352%	0,0000 00%	28,014 379%	52,399 562%	31,108 921%	31,368 185%	27,415 360%	25,783 174%
6	26,835 868%	22,238 063%	1,2646 57%	11,645 202%	26,940 857%	0,0000 00%	20,152 924%	45,220 264%	28,966 325%	27,112 604%	43,790 915%
7	29,645 225%	22,385 908%	41,972 904%	60,965 431%	55,135 219%	20,229 626%	0,0000 00%	31,917 189%	38,726 574%	27,198 562%	25,265 909%
8	12,754 964%	19,111 907%	34,557 311%	35,361 579%	35,463 357%	44,564 890%	35,909 209%	0,0000 00%	34,832 198%	43,947 331%	35,969 494%
9	35,682 422%	24,829 159%	37,156 473%	44,151 441%	31,368 185%	29,598 672%	38,726 574%	31,092 237%	0,0000 00%	34,974 912%	26,770 650%
10	28,274 721%	21,961 030%	39,549 960%	39,315 888%	46,034 898%	30,650 028%	43,381 013%	32,595 636%	48,911 360%	0,0000 00%	64,395 258%
11	32,079 121%	23,620 958%	31,765 398%	31,205 915%	38,275 115%	32,992 298%	35,277 823%	38,268 025%	26,770 650%	58,626 628%	0,0000 00%

Dari tabel diatas dapat disimpulkan bahwa perbedaan jarak antara algoritma *hubeny* dan *google map* berada dibawah 30% dengan rata-rata perbedaannya 0. 28,712882% dimana hasil jarak algoritma *hubeny* sangat jauh berbeda dari pada hasil pengukuran dari *google map*. Hal ini dikarenakan algoritma perhitungan *google map* yang kompleks sehingga didapatkan hasil yang akurat pada jarak sebenarnya melalui jalur yang dilalui. Sementara algoritma *hubeny* hanya mencari jarak berdasarkan garis lurus antar titik.

Tabel 4. 23. Perbedaan Persentase Jarak Algoritma *Vincenty* dan *Google Map*

Perse n (%)	1	2	3	4	5	6	7	8	9	10	11
1	0,0000 00%	13,617 898%	26,648 453%	26,004 354%	26,166 498%	20,587 367%	26,083 853%	2,7231 47%	34,310 938%	26,224 303%	24,749 332%
2	15,078 600%	0,0000 00%	21,421 951%	21,203 112%	21,213 474%	19,783 285%	21,217 806%	16,622 181%	24,646 898%	21,756 641%	20,763 401%
3	30,024 466%	22,561 286%	0,0000 00%	68,368 298%	26,376 273%	1,3568 55%	41,966 831%	30,927 605%	37,150 110%	25,845 951%	26,079 514%
4	29,396 704%	22,343 376%	68,368 298%	0,0000 00%	50,946 390%	11,724 929%	64,658 835%	31,807 056%	46,989 185%	25,782 520%	40,836 975%
5	29,841 896%	22,393 252%	26,376 273%	46,514 756%	0,0000 00%	28,007 475%	52,394 596%	31,101 837%	31,361 558%	27,407 754%	25,775 388%
6	26,828 439%	22,230 017%	1,2550 56%	11,636 566%	26,933 849%	0,0000 00%	20,145 266%	45,214 514%	28,959 499%	27,105 212%	43,785 334%
7	29,638 002%	22,377 847%	41,966 831%	60,961 341%	55,130 539%	20,221 976%	0,0000 00%	31,910 157%	38,720 556%	27,190 924%	25,258 067%
8	12,746 337%	19,103 584%	34,550 538%	35,354 872%	35,456 721%	44,559 070%	35,902 589%	0,0000 00%	34,825 633%	43,941 689%	35,962 993%
9	35,675 785%	24,821 338%	37,150 110%	44,145 818%	31,361 558%	29,591 907%	38,720 556%	31,085 296%	0,0000 00%	34,968 355%	26,763 405%
10	28,267 436%	21,952 942%	39,543 620%	39,309 519%	46,029 243%	30,642 995%	43,375 073%	32,588 851%	48,906 209%	0,0000 00%	64,391 542%
11	32,072 192%	23,613 035%	31,758 238%	31,198 698%	38,268 639%	32,985 645%	35,271 031%	38,261 758%	26,763 405%	58,622 310%	0,0000 00%

Dari tabel diatas dapat disimpulkan bahwa perbedaan jarak antara algoritma *vincenty* dan *google map* berada dibawah 30% dengan rata-rata perbedaannya 0. 28,706531% dimana hasil jarak algoritma *vincenty* sangat jauh berbeda dari pada hasil pengukuran dari *google map*. Hal ini dikarenakan algoritma perhitungan *google map* yang kompleks sehingga didapatkan hasil yang akurat pada jarak sebenarnya melalui jalur yang dilalui. Sementara algoritma *vincenty* hanya mencari jarak berdasarkan garis lurus antar titik.

Dari kedua uji diatas penulis menyimpulkan bahwa algoritma *vincenty* lebih akurat dikarenakan terdapat iterasi yang dilakukan. Namun memiliki efisiensi waktu sedikit lebih lama dari pada algoritma *hubeny*. Kedua algoritma diatas masih layak untuk digunakan dengan efisiensi dan persentase perbedaan jarak yang tidak terlalu besar. Untuk dibandingkan dengan jarak yang sebenarnya melalui *google map* kedua algoritma memiliki perbedaan sebesar hampir 30% yang mana jika kedua titik terlalu jauh maka akan terjadi penyimpangan jarak melebihi 30%.

4.1.5. Analisa perbandingan hasil menggunakan algoritma *ant colony*

Dari ketiga hasil diatas kemudian diolah dengan algoritma *ant colony* menghasil perbedaan yang disajikan pada tabel dibawah.

Tabel 4. 24. Perbedaan Persentase Algoritma *Hubeny*, Algoritma *Vincenty* dan *Google Map* pada hasil jarak terpendek pada Algoritma *Ant Colony*

	Algoritma Hubeny	Algoritma Vincenty	Google Map
Jalur Terbaik	1-10-11-9-5-7-3-4-6-8-2	1-10-11-9-5-7-3-4-6-8-2	1-11-9-5-7-3-4-6-10-8-2
Jarak Terbaik	19274,741553	19276,712964	25972

Dari hasil diatas didapati bahwa algoritma *hubeny* dan *vincenty* memiliki jalur yang sama dengan jarak yang berbeda. Sementara pada *google map* memiliki jalur dan dan jarak yang berbeda dari kedua algoritma diatas.

Dimana jalur terbaik algoritma *hubeny* dan algoritma *vincenty* adalah :

Lokasi pengguna - Rumah Sakit Yayasan Kasih Ibu - Rumah Sakit Azzuhra - Rumah Sakit Umum Sakinah - Rumah Sakit Bunda - Rumah Sakit Ibu Anak ABBY - Rumah Sakit Palang Merah Indonesia - Rumah Sakit Tentara Kesrem - Rumah Sakit Bunga Melati - Rumah Sakit Umum Metro Medical Center - Rumah Sakit Arun

Dengan jarak terbaik algoritma *hubeny* : 19274,741553 meter dan algoritma *vincenty* : 19276,712964 meter, yang mana selisih keduanya hanya sebesar 1,971411096 meter dan rata – rata keduanya 19275,727258 meter

Untuk hasil dari *google map* sendiri yaitu:

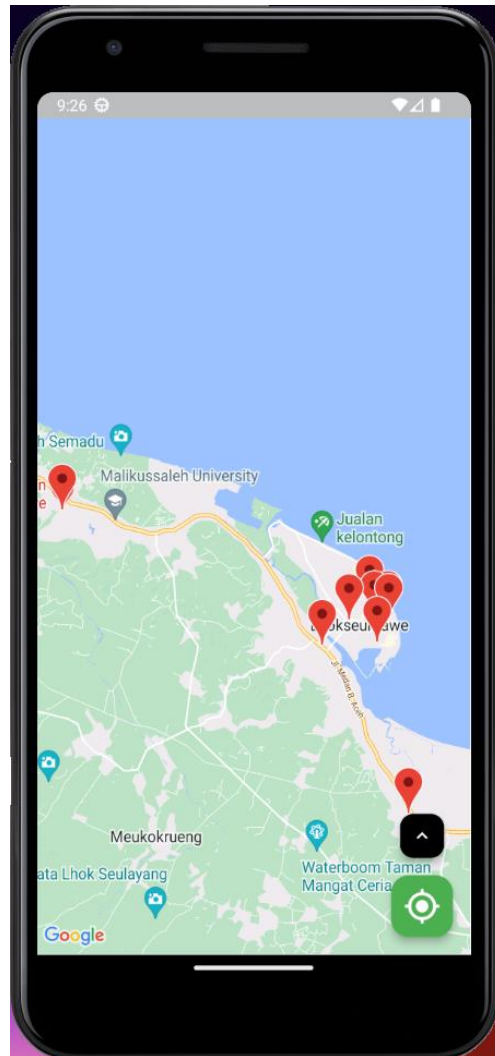
Lokasi pengguna - Rumah Sakit Azzuhra - Rumah Sakit Umum Sakinah - Rumah Sakit Bunda - Rumah Sakit Ibu Anak ABBY - Rumah Sakit Palang Merah Indonesia - Rumah Sakit Tentara Kesrem - Rumah Sakit Bunga Melati - Rumah Sakit Yayasan Kasih Ibu - Rumah Sakit Umum Metro Medical Center - Rumah Sakit Arun

Dengan jarak terbaik : 25972 meter, yang mana perbedaan dengan rata- rata kedua algoritma sebelumnya 6696,272742 meter.

4.2. Implementasi Aplikasi

Ketiga formula algoritma tersebut diimplementasikan dalam bentuk aplikasi *android* yang digunakan untuk mencari rumah sakit terdekat berdasarkan lokasi pengguna. Dalam merancang aplikasi tersebut, penulis menggunakan *framework flutter* dan *library dart* yaitu *google maps*.

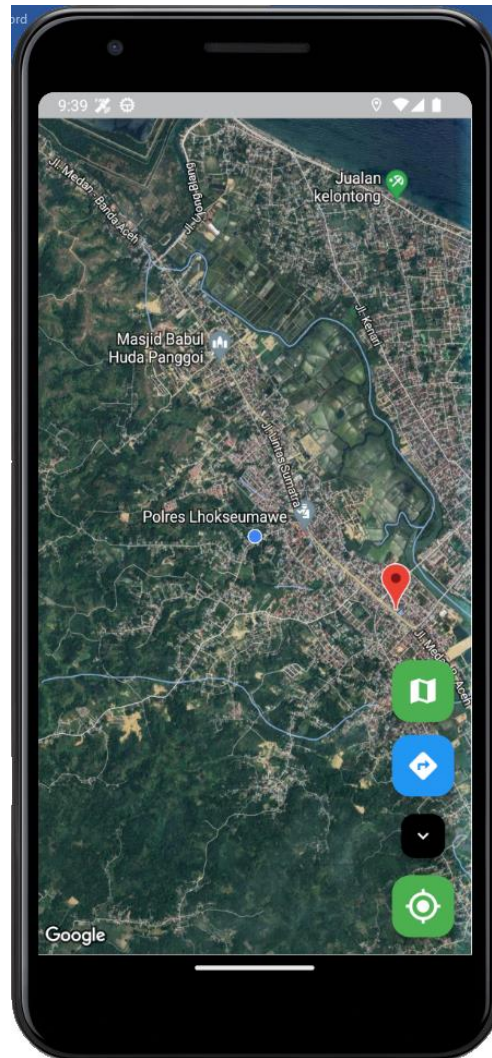
4.2.1. Tampilan awal aplikasi



Gambar 4. 2. Tampilan Awal Aplikasi

Pada halaman ini menampilkan map yang berfokus pada kota Lhokseumawe dan mark titik koordinat rumah sakit yang ada di Lhokseumawe. Terdapat tombol *location* untuk langsung menuju ke lokasi pengguna dan tombol *expand_more* untuk menampilkan tombol lain.

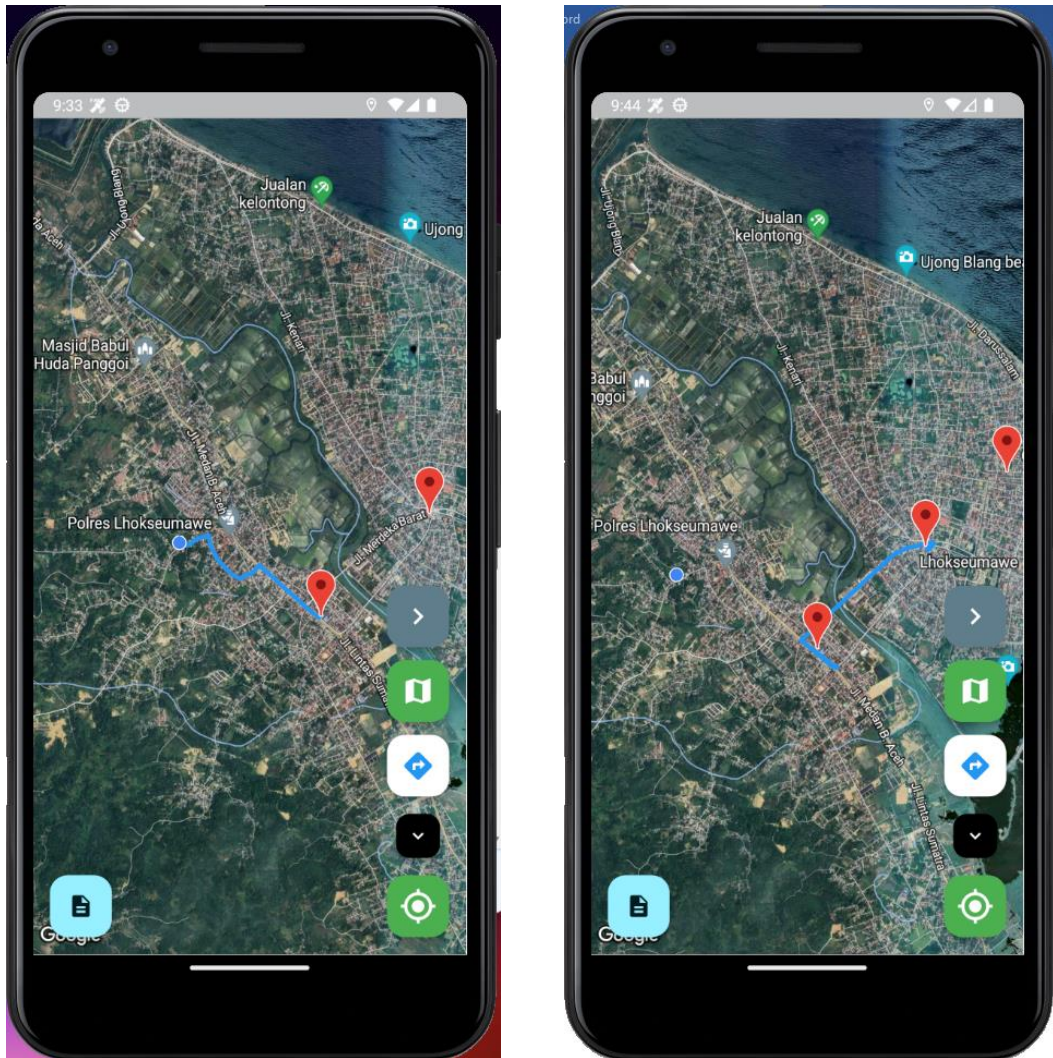
4.2.2. Menampilkan lokasi pengguna dan merubah map



Gambar 4. 3. Lokasi Pengguna

Setelah tombol *location* dan *expand_more* ditekan, maka akan memunculkan lokasi pengguna dan tombol lain yaitu tombol *direction* untuk mencari rumah sakit terdekat dan tombol *maps* untuk mengganti mode map.

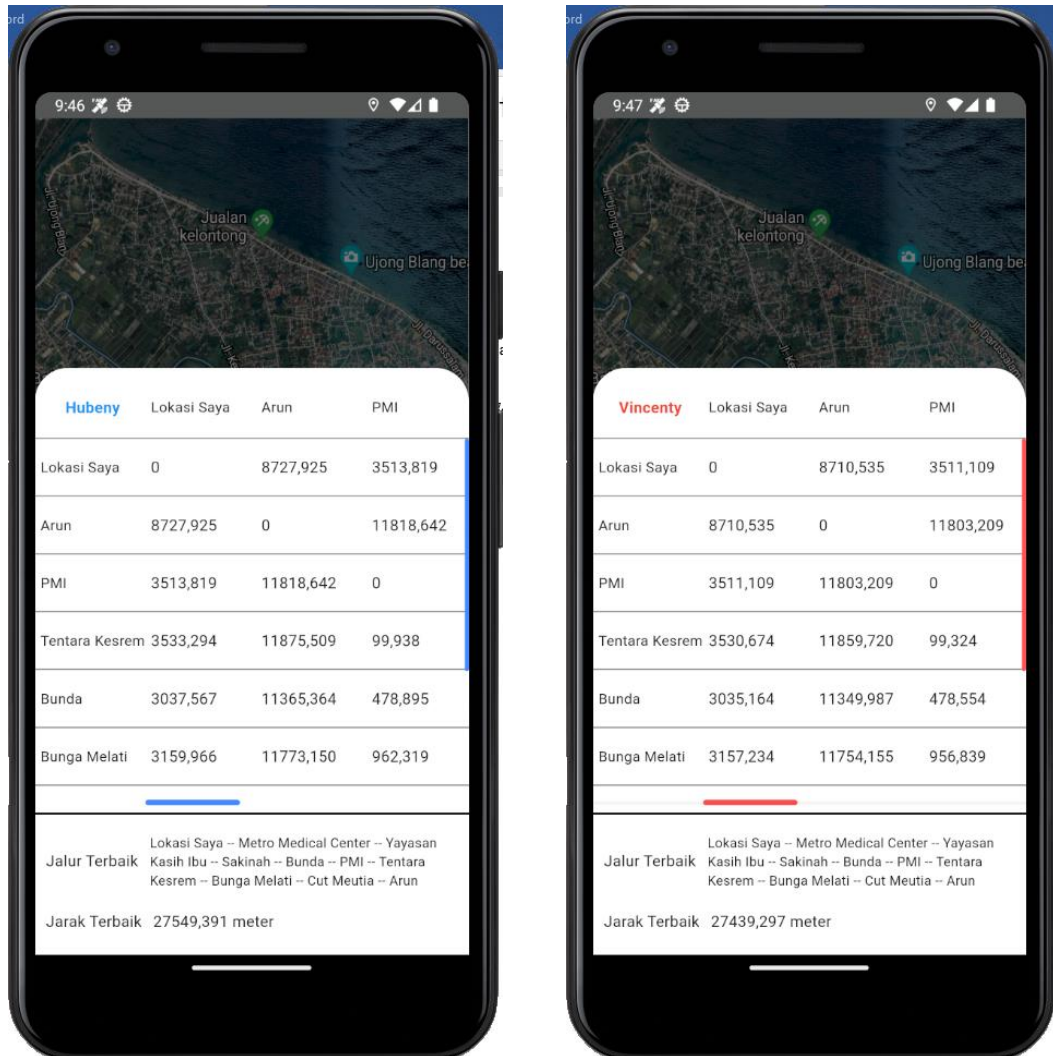
4.2.3. Mencari rumah sakit terdekat



Gambar 4. 4. Mencari Rumah Sakit Terdekat

Menekan tombol *direction* akan mencari rumah sakit terdekat dari pengguna dan memunculkan tombol *next* untuk rumah sakit berikutnya jika rumah sakit awal penuh dan tombol *calculation* untuk melihat hasil perhitungan algoritmanya.

4.2.4. Hasil perhitungan algoritma *hubeny*, *vincenty* dan *ant colony*



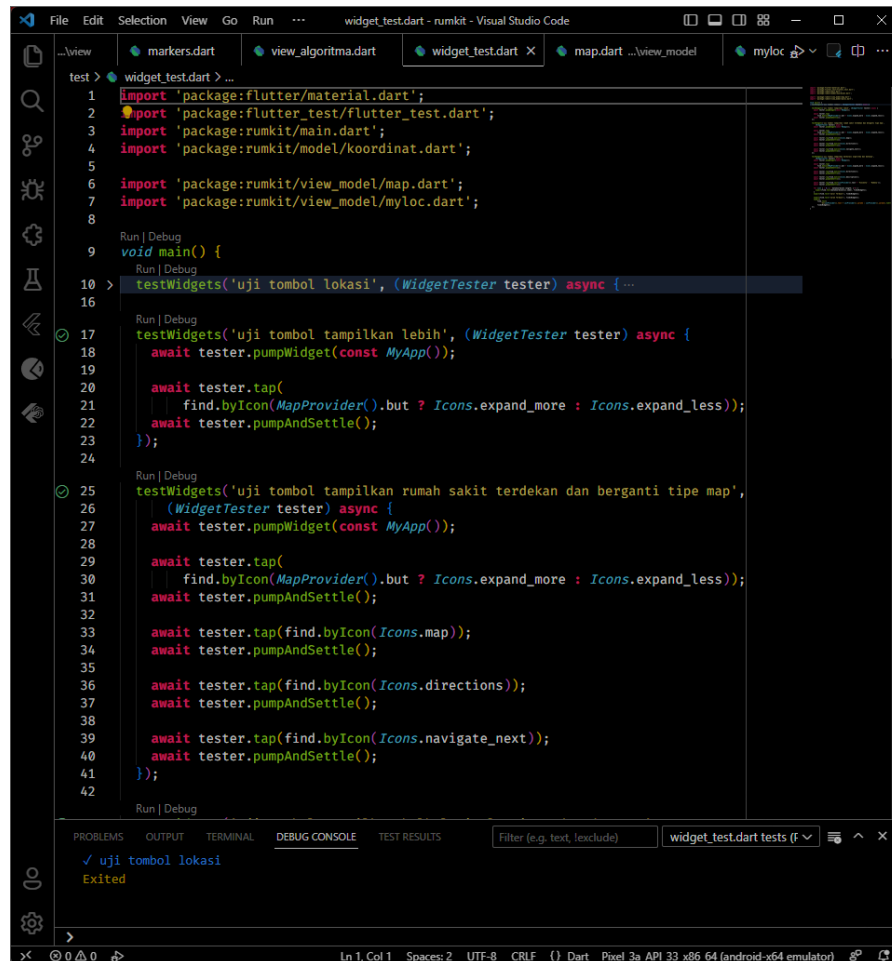
Gambar 4. 5. Pengimplementasian Algoritma

Menampilkan hasil perhitungan dari algoritma *hubeny* dan *vincenty* yang kemudian diolah oleh algoritma *ant colony* untuk mencari jalur terbaik dan jarak terbaik.

4.2.5. Pengujian *white box*

Pengujian *white box* dilakukan untuk mengetahui kode program yang gagal dijalankan tanpa melihat tampilan aplikasi. Uji dilakukan pada tombol yang ada dan juga beberapa teks yang ditampilkan pada aplikasi.

1. Uji tombol *my_location*



```

test > widget_test.dart > ...
1 import 'package:flutter/material.dart';
2 import 'package:flutter_test/flutter_test.dart';
3 import 'package:rumkit/main.dart';
4 import 'package:rumkit/model/koordinat.dart';
5
6 import 'package:rumkit/view_model/map.dart';
7 import 'package:rumkit/view_model/myloc.dart';
8
9 void main() {
10 testWidgets('uji tombol lokasi', (WidgetTester tester) async {
11
12
13
14
15
16
17 testWidgets('uji tombol tampilkan lebih', (WidgetTester tester) async {
18   await tester.pumpWidget(const MyApp());
19
20   await tester.tap(
21     find.byIcon(MapProvider().but ? Icons.expand_more : Icons.expand_less));
22   await tester.pumpAndSettle();
23 });
24
25 testWidgets('uji tombol tampilkan rumah sakit terdekat dan berganti tipe map',
26   (WidgetTester tester) async {
27   await tester.pumpWidget(const MyApp());
28
29   await tester.tap(
30     find.byIcon(MapProvider().but ? Icons.expand_more : Icons.expand_less));
31   await tester.pumpAndSettle();
32
33   await tester.tap(find.byIcon(Icons.map));
34   await tester.pumpAndSettle();
35
36   await tester.tap(find.byIcon(Icons.directions));
37   await tester.pumpAndSettle();
38
39   await tester.tap(find.byIcon(Icons.navigate_next));
40   await tester.pumpAndSettle();
41 });
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE TEST RESULTS Filter (e.g. text, !exclude) widget_test.dart tests (f) ^ x

✓ uji tombol lokasi
Exited

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF () Dart Pixel_3a_API_33_x86_64 (android-x64 emulator)

Gambar 4. 6. Uji Widget 1

Tombol *my_location* ditekan untuk memastikan berfungsi atau terjadi error saat menampilkan lokasi pengguna.

2. Uji tombol *expand_more*

```

test > widget_test.dart > ...
4 import 'package:runkit/model/koordinat.dart';
5
6 import 'package:runkit/view_model/map.dart';
7 import 'package:runkit/view_model/myloc.dart';
8
9 void main() {
10   Run | Debug
11   testWidgets('uji tombol lokasi', (WidgetTester tester) async {
12     Run | Debug
13     await tester.pumpWidget(const MyApp());
14     await tester.tap(find.byIcon(Icons.my_location_outlined));
15     await tester.pump();
16   });
17
18   Run | Debug
19   testWidgets('uji tombol tampilkan lebih', (WidgetTester tester) async {
20     Run | Debug
21     await tester.pumpWidget(const MyApp());
22     await tester.tap(
23       find.byIcon(MapProvider().but ? Icons.expand_more : Icons.expand_less));
24     await tester.pumpAndSettle();
25   });
26
27   Run | Debug
28   testWidgets('uji tombol tampilkan rumah sakit terdekat dan berganti tipe map',
29     (WidgetTester tester) async {
30     Run | Debug
31     await tester.pumpWidget(const MyApp());
32     await tester.tap(
33       find.byIcon(MapProvider().but ? Icons.expand_more : Icons.expand_less));
34     await tester.pumpAndSettle();
35     await tester.tap(find.byIcon(Icons.map));
36     await tester.pumpAndSettle();
37     await tester.tap(find.byIcon(Icons.directions));
38     await tester.pumpAndSettle();
39     await tester.tap(find.byIcon(Icons.navigate_next));
40     await tester.pumpAndSettle();
41   });
42 }
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE TEST RESULTS Filter (e.g. text, lexclude) widget_test.dart tests (f) ^ X

✓ uji tombol tampilkan lebih
Exited

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF {} Dart Pixel_3a_API_33_x86_64 (android-x64 emulator)

Gambar 4. 7. Uji Widget 2

Tombol *expand_more* ditekan untuk memastikan berfungsi atau terjadi error saat menampilkan tombol *directions* dan *map*.

3. Uji tombol *directions*, *map* dan *next*

```

17 testWidgets('uji tombol tampilkan lebih', (WidgetTester tester) async {
18   await tester.pumpWidget(const MyApp());
19
20   await tester.tap(
21     find.byIcon(MapProvider().but ? Icons.expand_more : Icons.expand_less));
22   await tester.pumpAndSettle();
23 });
24
25 Run | Debug
26 testWidgets('uji tombol tampilkan rumah sakit terdekat dan berganti tipe map',
27   (WidgetTester tester) async {
28     await tester.pumpWidget(const MyApp());
29
30     await tester.tap(
31       find.byIcon(MapProvider().but ? Icons.expand_more : Icons.expand_less));
32     await tester.pumpAndSettle();
33
34     await tester.tap(find.byIcon(Icons.map));
35     await tester.pumpAndSettle();
36
37     await tester.tap(find.byIcon(Icons.directions));
38     await tester.pumpAndSettle();
39
40     await tester.tap(find.byIcon(Icons.navigate_next));
41     await tester.pumpAndSettle();
42 });
43
44 Run | Debug
45 testWidgets('uji tombol tampilkan kalkulasi algoritma dan datanya',
46   (WidgetTester tester) async {
47     await tester.pumpWidget(const MyApp());
48
49     await tester.tap(
50       find.byIcon(MapProvider().but ? Icons.expand_more : Icons.expand_less));
51     await tester.pumpAndSettle();
52
53     await tester.tap(find.byIcon(Icons.directions));
54     await tester.pumpAndSettle();
55 });

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE TEST RESULTS Filter (e.g. text, lexclude) widget_test.dart tests (f)

✓ uji tombol tampilkan rumah sakit terdekat dan berganti tipe map
 Exited

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF {} Dart Pixel_3a_API_33_x86_64 (android-x64 emulator)

Gambar 4. 8. Uji Widget 3

Tombol *directions*, *map* dan *next* ditekan untuk memastikan berfungsi atau terjadi error saat menampilkan jarak rumah sakit terdekat, berganti tipe map dan rumah sakit berikutnya.

4. Uji tombol *description* dan datanya

```

35
36 await tester.tap(find.byIcon(Icons.directions));
37 await tester.pumpAndSettle();
38
39 await tester.tap(find.byIcon(Icons.navigate_next));
40 await tester.pumpAndSettle();
41 });
42
43 Run | Debug
44 testWidgets('uji tombol tampilkan kalkulasi algoritma dan datanya',
45   (WidgetTester tester) async {
46     await tester.pumpWidget(const MyApp());
47
48     await tester.tap(
49       find.byIcon(MapProvider()).but ? Icons.expand_more : Icons.expand_less);
50     await tester.pumpAndSettle();
51
52     await tester.tap(find.byIcon(Icons.directions));
53     await tester.pumpAndSettle();
54
55     await tester.tap(find.byIcon(Icons.description));
56     await tester.pumpAndSettle();
57
58     await tester.tap(find.text(LocProvider().text ? 'Vincenty' : 'Hubeny'));
59     await tester.pumpAndSettle();
60
61     for (int i = 0; i < datakoordinat.length; i++) {
62       expect(find.text(datakoordinat[i].name), findsWidgets);
63     }
64     expect(find.text('Jalur Terbaik'), findsWidgets);
65     expect(find.text('Jarak Terbaik'), findsWidgets);
66     expect(
67       find.text(
68         '${(LocProvider().text ? LocProvider().jarak2 : LocProvider().jarak1).toString()}
69     );
70   });
71 }
72

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE TEST RESULTS Filter (e.g. text, exclude) widget_test.dart tests (f) widget_test.dart tests (f) Exited

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF Dart Pixel_3a_API_33_x86_64 (android-x64 emulator)

Gambar 4. 9. Uji Widget 4 Dan Beberapa Teks

Tombol *description* ditekan untuk memastikan berfungsi atau terjadi error saat menampilkan hasil perhitungan algoritma yang sudah diimplementasikan.

BAB V

PENUTUP

5.1. Kesimpulan

Setelah dilakukan penelitian, ada beberapa kesimpulan yang dapat penulis jawab dari rumusan masalah, yaitu sebagai berikut:

1. Merancang dan membangun sebuah aplikasi “Pemetaan Rumah Sakit” menggunakan *framework flutter* sangat relevan. Hal itu dikenakan tersedianya library google map yang memiliki berbagai macam fitur guna memudahkan merancang aplikasi “Pemetaan Rumah Sakit” sudah sangat cukup.
2. Pengimplementasi algoritma *hubeny*, algoritma *vincenty* dan algoritma *ant colony* pada aplikasi “Pemetaan Rumah Sakit” bisa diimplementasikan dengan baik dengan cara membuat file yang berisi formula dari ketiga algoritma tersebut yang kemudian diintegrasikan pada tombol *direction* untuk mencari jarak terpendek untuk lokasi pengguna dan semua titik rumah sakit .
3. Perbedaan jarak antara algoritma *hubeny* dan *vincenty* berada dibawah 1% dengan rata-rata perbedaannya 0.009289%. Untuk dibandingkan dengan jarak yang sebenarnya melalui google map kedua algoritma memiliki perbedaan sebesar hampir 30% yang mana jika kedua titik terlalu jauh maka akan terjadi penyimpangan jarak melebihi 30%. Kedua algoritma diatas masih layak untuk digunakan dengan catatan hanya bisa diaplikasikan pada daerah seukuran kota madya atau ibukota provinsi. Program algoritma *hubeny* lebih efisien dibandingkan algoritma *vincenty* dalam kecepatan pengeksekusian program dengan rata-rata perbedaan dalam waktu yaitu algoritma *hubeny* lebih cepat 0.4584 milidetik dibandingkan algoritma *vincenty*. Dari hasil pengujian, penulis menyimpulkan bahwa algoritma *vincenty* lebih akurat dikarenakan terdapat iterasi yang dilakukan. Namun memiliki efisiensi waktu sedikit lebih lama dari pada algoritma *hubeny*.
4. Perbedaan jarak terbaik algoritma *hubeny*, algoritma *vincenty* dan *google map* pada algoritma *ant colony* adalah algoritma *hubeny* : 19274,741553 meter dan algoritma *vincenty* : 19276,712964 meter, yang mana selisih keduanya hanya

sebesar 1,971411096 meter dan rata – rata keduanya 19275,727258 meter dan *google map* dengan jarak terbaik : 25972 meter, yang mana perbedaan dengan rata- rata kedua algoritma sebelumnya 6696,272742 meter.

5.2. Saran

Pada penelitian yang dilakukan penulis terdapat kekurangan dan saran perbaikan untuk penelitian selanjutnya. Diantara adalah sebagai berikut:

1. Aplikasi yang penulis buat masih memiliki beberapa kekurangan seperti tidak adanya pelacakan lokasi pengguna disaat sedang dalam perjalanan, keadaan lalu lintas dan lain sebagainya.
2. Diharapkan untuk peneliti selanjutnya dapat menambahkan algoritma lain untuk dibandingkan dengan algoritma *ant colony* untuk mengetahui seberapa jauh perbedaan hasil yang didapatkan dari perbandingan tersebut dan juga untuk menambahkan objek lain seperti puskesmas dan fasilitas kesehatan lainnya.

DAFTAR PUSTAKA

- Ernawati, S., Wati, R., & Maulana, I. (2021). Penerapan model fountain untuk pengembangan aplikasi text recognition dan text to speech berbasis android menggunakan flutter. *Prosiding Snast*.
<https://journal.akprind.ac.id/index.php/snast/article/view/3405>
- Frialdo, D., Helmina, A., Oktaviani Melianti, E., Jalinus, N., Abdullah, R., Sarjana, P., Negeri Padang, U., Hamka, J., Tawar Bar, A., & Barat, S. (2023). Rancang Bangun Media Pembelajaran Sistem Operasi Jaringan Materi Instalasi Debian Berbasis Android. *Journal on Education*, 05(02).
- Hanif, I. F., & Sinambela, G. M. (2021). Pembuatan Aplikasi E-Tatib Berbasis Android Menggunakan Bahasa Pemrograman Dart. *Jurnal Teknologi Dan Terapan Bisnis (JTTB)*, 4(1).
- Ilhadi, V., Rahma, E., & Fachrurrazi, S. (2022). Online Sales Application Typical Cake Aceh Based On Android. *Jurnal Mantik*.
<http://iocscience.org/ejournal/index.php/mantik/article/view/2047>
- Irvansyah, F., Setiawansyah, S., & Muhaqiqin, M. (2020). APLIKASI PEMESANAN JASA CUKUR RAMBUT BERBASIS ANDROID. *Jurnal Ilmiah Infrastruktur Teknologi Informasi*, 1(1).
<https://doi.org/10.33365/jiiti.v1i1.253>
- Mauludi, K., Gautama Putrada, A., & Abdurohman, M. (2021). Analisis Kinerja metode Hubeny Formula dalam Navigasi Drone Otonomus Menggunakan GPS. *E-Proceeding of Engineering*, 8(2).
- Microsoft. (2020). *Visual studio code*. Microsoft Redmond, WA, USA.

- Nugraha, Y. K., & Hajar, A. (2023). Pemanfaatan Informasi Geospasial Dasar (IGD) untuk Analisis Penyimpangan Arah Kiblat Bangunan Masjid secara Masal. *Jurnal Teknik: Media Pengembangan Ilmu Dan Aplikasi Teknik*, 21(2). <https://doi.org/10.55893/jt.vol21no2.473>
- Nugroho, A. K., & Permadi, I. (2020). IMPLEMENTASI JALUR PENDEK MENGGUNAKAN ANT COLONY OPTIMIZATION. *Dinamika Rekayasa*, 16(1). <https://doi.org/10.20884/1.dr.2020.16.1.294>
- Nurharyanto, & Perdana, S. (2021). Menentukan Rute Distribusi Di PT Sinar Harapan Plastik Dengan Metode Algoritma Ant Colony Optimization. *Ikrath Teknologi*, 5(1).
- Putra, D. M., & Vadriasmis, D. (2020). Analisis Penerapan Sistem Informasi Manajemen Rumah Sakit (SIMRS) di TPPRJ Menggunakan Metode UTAUT di RSD TK. III Dr. Reksodiwiryo Padang. *Administration & Health Information of Journal*, 1(1).
- R.H. Zer, P. P. P. A. N. W. F. I., Hayadi, B. H., & Damanik, A. R. (2022). PENDEKATAN MACHINE LEARNING MENGGUNAKAN ALGORITMA C4.5 BERBASIS PSO DALAM ANALISA PEMAHAMAN PEMROGRAMAN WEBSITE. *Jurnal Informatika Dan Teknik Elektro Terapan*, 10(3). <https://doi.org/10.23960/jitet.v10i3.2700>
- Ritonga, H. S., Irmayani, D., & Pane, R. (2021). SISTEM INFORMASI GEOGRAFIS (GIS) PADA RUMAH SAKIT DI KABUPATEN LABUHANBATU BERBASIS WEB. *JURTEKSI (Jurnal Teknologi Dan Sistem Informasi)*, 7(2). <https://doi.org/10.33330/jurtekxi.v7i2.1089>
- Rochmawati, F. D. (2023). PERKEMBANGAN BAHASA PEMROGRAMAN KOMPUTER DI AMERIKA SERIKAT TAHUN 1955-1995. digilib.uinkhas.ac.id.

<http://digilib.uinkhas.ac.id/17639/1/Fltria%20Dewi%20%20Rochmawati.pdf>

Sekiguchi, R., Abe, K., Shogo, S., Kumano, M., Asakura, D., Okabe, R., Kariya, T., & Kawakatsu, M. (2021). Phased Human Activity Recognition based on GPS. *UbiComp/ISWC 2021 - Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers*. <https://doi.org/10.1145/3460418.3479382>

Setiawan, A., Ananda, M. R. D., Alvario, R., & ... (2022). Pengujian Sistem Informasi Aplikasi Perpustakaan Berbasis Web Di SMAN 1 Gunung Sindur Dengan White Box Testing. ... *Sacra: Jurnal Sains ...*, 2(1).

Siematauw, R. R. (2023). Aplikasi Arsip Surat Mobile Menggunakan Framework Flutter (Studi Kasus: PT. BPD MalukuMalut). *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, 10(1).
<https://doi.org/https://doi.org/10.35957/jatisi.v10i1.2577>

Sinaga, G. R. U., & Samsudin, S. (2021). Implementasi Framework Laravel dalam Sistem Reservasi pada Restoran Cindelas Kota Medan. *Jurnal Janitra Informatika Dan Sistem Informasi*, 1(2).
<https://doi.org/10.25008/janitra.v1i2.131>

Sulastio, B. S., Anggono, H., & Putra, A. D. (2021). Sistem informasi geografis untuk menentukan lokasi rawan macet di jam kerja pada Kota Bandarlampung pada berbasis android. *Jurnal Teknologi Dan Sistem Informasi (JTISI)*, 2(1).

Tjandra, S., & Chandra, G. S. (2020). Pemanfaatan Flutter dan Electron Framework pada Aplikasi Inventori dan Pengaturan Pengiriman Barang. *Journal of Information System, Graphics, Hospitality and Technology*, 2(02).
<https://doi.org/10.37823/insight.v2i02.109>

Windarni, V. A., & Setiawan, A. (2022). COMPARATIVE ANALYSIS OF VINCENTY AND GEODESIC METHOD APPROACHES IN MEASURING THE DISTANCE BETWEEN SUBDISTRICT OFFICES IN SALATIGA CITY. *BAREKENG: Jurnal Ilmu Matematika Dan Terapan*, 16(4). <https://doi.org/10.30598/barekengvol16iss4pp1207-1220>