Penerapan Vektor Euclidean dalam Navigasi Google Maps dan Aplikasi Peta Digital

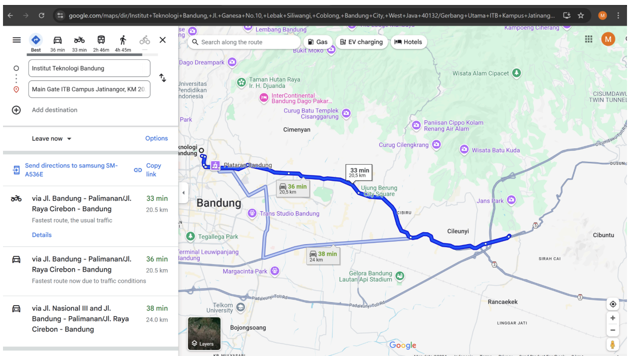
Muhammad Timur Kanigara 13523055
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13523055@std.stei.itb.ac.id, timurkanigara21@gmail.com

***Abstract*—Makalah ini bertujuan untuk membahas penerapan vektor Euclidean dalam navigasi digital, khususnya dalam sistem peta digital seperti Google Maps. Dengan meningkatnya kebutuhan akan navigasi yang akurat dan efisien, pemahaman tentang vektor Euclidean menjadi sangat penting. Metode yang digunakan dalam penelitian ini meliputi analisis teori vektor Euclidean, perhitungan jarak antar titik, serta implementasi simulasi untuk menghitung jarak antara dua lokasi. Hasil yang diharapkan adalah pemahaman yang lebih baik tentang bagaimana vektor Euclidean dapat diterapkan dalam navigasi digital dan penghitungan jarak yang lebih efisien.**

***Keywords*—Vektor Euclidean, Navigasi, Peta Digital, Google Maps.**

I. PENDAHULUAN

Sistem navigasi telah menjadi bagian penting dalam kehidupan sehari-hari, membantu individu menemukan rute tercepat dan paling efisien untuk bepergian. Aplikasi seperti Google Maps dan Waze menggunakan prinsip-prinsip matematika untuk mengelola data lokasi, mengoptimalkan rute, dan memperkirakan waktu tempuh. Dengan kemajuan teknologi, sistem navigasi digital kini dapat memberikan informasi yang lebih akurat dan *real-time*, menjadikannya sebagai alat yang sangat berguna bagi pengguna.



Gambar 1.1 Contoh rute dan jarak ITB Ganesha dan ITB Jatinangor
https://www.google.com/maps/dir/Institut+Teknologi+Bandung,+Jl.+Ganesa+No.10,+Lebak+Siliwangi,+Coblong,+Bandung+City,+West+Java+40132/Gerbang+Utama+ITB+Kampus+Jatinangor,+Jalan+Raya+Jatinangor,+Sayang,+Sumedang+Regency,+West+Java/@-6.911928,107.6681347,13z/data=!4m13!4m12!1m5!1m1!1s0x2e68e65767c9b183:0x2478e3dcdce37961!2m2!1d107.6101912!2d-6.8903617!1m5!1m1!1s0x2e68c4a6151db7ed:0x63524d5fe8a4034d!2m2!1d107.76834!2d-6.933871?entry=ttu&g_ep=EgoyMDI0MTIxMS4wIKXMDSoASAFQAw%3D%3D

Salah satu konsep matematika yang mendasari sistem ini adalah vektor Euclidean. Vektor Euclidean digunakan untuk merepresentasikan posisi dan arah dalam sistem koordinat, memungkinkan perhitungan jarak yang presisi dan optimasi jalur. Dalam konteks sebuah navigasi, vektor ini sangat penting untuk menghitung jarak antar titik dan menentukan arah suatu perjalanan. Dengan menggunakan rumus jarak Euclidean, sistem navigasi dapat memberikan estimasi yang akurat tentang jarak yang harus ditempuh pengguna.

Vektor Euclidean juga berfungsi sebagai dasar untuk algoritma pencarian rute yang kompleks. Algoritma seperti Dijkstra dan A* sudah memanfaatkan konsep-konsep vektor untuk menghitung jalur terpendek dan paling efisien. Dengan memanfaatkan data *real-time*, sistem navigasi dapat memberikan rute yang optimal beserta dengan kondisi lalu lintas saat itu.

Makalah ini membahas dasar teori vektor Euclidean beserta penerapannya dalam sistem navigasi digital, yang disertai dengan simulasi perhitungan dan hasil implementasi sederhana. Selain itu, makalah ini juga menyoroti kelebihan dan keterbatasan metode Euclidean beserta potensinya untuk pengembangan lebih lanjut dalam navigasi berbasis graf.

Dalam era digital ini, penting untuk memahami bagaimana vektor Euclidean berkontribusi pada pengembangan teknologi navigasi yang lebih canggih. Dengan pemahaman yang baik terkait konsep ini, kita dapat mengembangkan sebuah sistem navigasi yang lebih efisien dan responsif terhadap kebutuhan pengguna.

Salah satu keunggulan utama sistem berbasis vektor Euclidean adalah kemampuannya dalam menyederhanakan data spasial yang kompleks. Data geografis yang sangat beragam, seperti informasi koordinat, dapat diolah dan direpresentasikan menjadi lebih sederhana melalui vektor. Dengan begitu, sistem navigasi dapat mengelola dan menganalisis data dalam waktu singkat tanpa kehilangan akurasinya.

Penerapan vektor Euclidean juga tidak hanya terbatas pada penghitungan jarak. Konsep ini juga sering digunakan dalam pemetaan ulang sebuah data dari berbagai sistem koordinat yang berbeda-beda. Proses transformasi ini penting untuk menjamin konsistensi dan akurasi dari pengintegrasian data yang didapat dari berbagai sumber, seperti peta digital atau sensor GPS.

Selain itu, vektor Euclidean dapat membuka peluang untuk pengembangan lebih lanjut di bidang berbasis kecerdasan buatan. Dengan menggunakan konsep ini, algoritma pembelajaran suatu emsin dapat dilatih untuk mengenali pola dan memprediksi rute yang optimal berdasarkan perilaku pengguna dan kondisi lingkungan yang ada. Hal ini tidak hanya dapat meningkatkan efisiensi, tetapi juga dapat menciptakan pengalaman pengguna yang lebih personal.

II. Latar Belakang Teoritis

A. Vektor Euclidean

Vektor Euclidean adalah objek matematis yang memiliki arah dan panjang. Dalam ruang dua dimensi, vektor dapat dinyatakan sebagai:

$$v = (x, y)$$

atau jika vektor tersebut berdimensi n, maka vektor dapat dinyatakan sebagai

$$v = (v_1, v_2, v_3, ..., v_n)$$

Setiap vektor memiliki komponen yang menunjukkan posisi dalam sistem koordinat. Misalnya, vektor:

$$v = (3, 4)$$

menunjukkan bahwa vektor tersebut memiliki komponen 3 pada sumbu x dan 4 pada sumbu y.

Operasi Dasar

Penjumlahan:

Penjumlahan dua vektor dilakukan dengan menjumlahkan komponen-komponennya. Misalnya, jika:

$$v_1 = (x_1, y_1)$$

dan

$$v_2 = (x_2, y_2)$$

maka,

$$v_1 + v_2 = (x_1 + x_2, y_1 + y_2)$$

Pengurangan:

Pengurangan dua vektor dilakukan dengan mengurangkan komponen-komponennya. Misalnya, jika:

$$v_1 = (x_1, y_1)$$

dan

$$v_2 = (x_2, y_2)$$

maka,

$$v_1 + v_2 = (x_1 - x_2, y_1 - y_2)$$

Perkalian Skalar:

Perkalian suatu vektor dengan suatu nilai skalar dilakukan dengan mengurangkan mengalikan nilai skalar tersebut ke masing-masing komponennya. Misalnya, ada sebuah skalar k dan vektor **v**, maka:

$$kv = (kv_1, kv_2, ..., kv_n)$$

Vektor juga dapat digunakan untuk merepresentasikan perpindahan posisi dalam ruang dua atau tiga dimensi, Dengan kemampuannya tersebut, vektor menjadi suatu alat utama dalam navigasi yang membutuhkan presisi tinggi.

B. Jarak Euclidean

Jarak antara dua titik (x_1, y_1) dan (x_2, y_2) dapat dihitung dengan rumus:

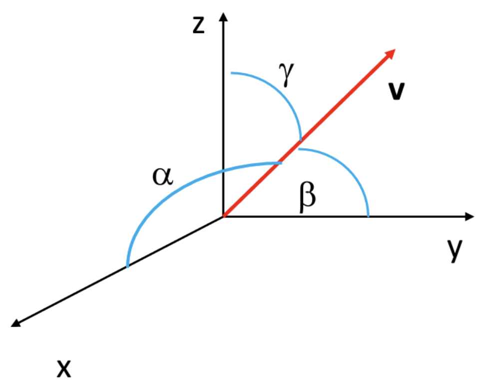
$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Jika kedua vektor berdimensi n, maka jarak antara dua vektor $u = (u_1, u_2, ..., u_n)$ dan $v = (v_1, v_2, ..., v_n)$ dapat dihitung dengan rumus:

$$d = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + ... + (u_n - v_n)^2}$$

Rumus ini menjadi sebuah dasar dalam perhitungan suatu jarak pada sistem navigasi. Jarak ini sering juga disebut dengan jarak garis lurus atau “as-the-crow-flies distance”, yang berarti jarak terpendek dalam memperhitungkan hambatan fisik

C. Arah Vektor



Gambar 2.1. Arah Vektor
https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-15-Ruang-vektor-umum-Bagian1-2023.pdf

Misalkan vektor $v = (v_1, v_1, v_3)$ adalah sebuah vektor berdimensi 3 maka arah **v** adalah:

$$\cos \alpha = \frac{v_1}{|v|}$$

$$\cos \beta = \frac{v_2}{|v|}$$

$$\cos\gamma = \frac{v_3}{|v|}$$

Perhitungan sudut ini berguna dalam menentukan arah sebuah belokan atau perubahan orientasi pada sistem navigasi berbasis graf.

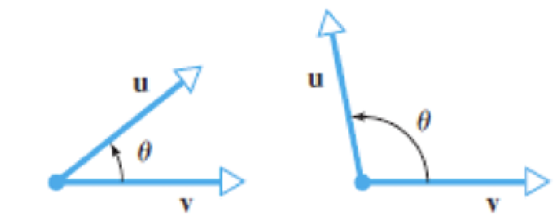
D. Dot Product

Jika **u** dan **v** adalah vektor tidak nol di R^2 atau R^3, maka perkalian titik, atau disebut juga dengan *Euclidean inner product*, **u** dan **v** adalah:

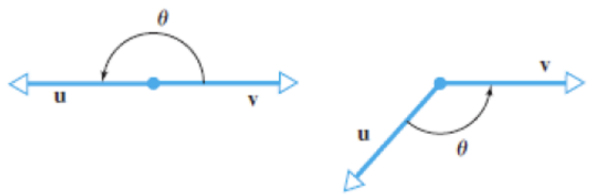
$$u \,.\, v = |u||v| \cos\theta$$

dengan θ adalah sudut yang dibentuk oleh **u** dan **v.** Jika **u** = 0 atau **v** = 0, maka **u . v** = 0

Contoh bentuk sudut **u** dan **v**:



Gambar 2.2 Contoh bentuk sudut dua vektor (bagian 1) https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-15-Ruang-vektor-umum-Bagian1-2023.pdf



Gambar 2.2 Contoh bentuk sudut dua vektor (bagian 2) https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-15-Ruang-vektor-umum-Bagian1-2023.pdf

Sudut antara dua vektor dapat dihitung menggunakan dot product, jika **u** dan **v** adalah dua vektor, maka sudut θ antara keduanya dapat dihitung dengan rumus:

$$\cos\theta = \frac{u.v}{|u||v|}.$$

III. Aplikasi Vektor Euclidean dalam Sistem Navigasi

A. Sistem Navigasi GPS

Penggunaan Vektor: GPS menggunakan vektor Euclidean untuk menghitung jarak antara satelit dengan penerima yang ada di permukaan Bumi. Setiap satelit akan mengirimkan sinyal yang memungkinkan penerima untuk menentukan posisinya dengan menghitung jarak ke beberapa satelit.

Perhitungan Jarak: Jarak dihitung menggunakan rumus Euclidean yang memungkinan sistem untuk menentukan lokasi pengguna dengan akurasi yang tinggi.

B. Google Maps

Model Navigasi Digital : Google Maps menggunakan latitude dan longitude untuk merepresentasikan lokasi yang ada di permukaan Bumi. Setiap lokasi bisa diubah menjadi sebuah vektor Euclidean untuk perhitungan lebih lanjut.

Perhitungan Jarak : Ketika seorang pengguna meminta arah, Google Maps menghitung jarak antara titik awal dan titik tujuan menggunakan rumus jarak Euclidean. Hal ini sering menjadi langkah pertama sebelum menerapkan algoritma dari perutean yang lebih kompleks.

C. OpenStreetMap

Data Representasi : OSM menggunakan vektor Euclidean untuk merepresentasikan fitur-fitur geografis seperti jalanan, sungai, dan batas-batas dari suatu wilayah. Setiap fitur diwakili oleh serangkaian titik (node) yang membentuk jalur vektor.

Routing Algorithm : OSM menerapkan algoritma perutean yang memanfaatkan vektor Euclidean untuk menemukan rute terpendek antara dua titik yang ada.

D. Rute Terpendek

Algoritma seperti Dijkstra dan A* digunakan untuk mencari rute terpendek antara dua titik. Jarak Euclidean sering dijadikan heuristik dalam algoritma ini untuk dapat mempercepat proses pencarian. Contohnya, dalam sebuah algoritma A*, jarak Euclidean digunakan untuk memperkirakan jarak tersisa untuk sampai ke tujuan, memungkinkan sistem agar bisa fokus pada jalur yang lebih baik.

E. Hubungan dengan Heuristik

Penggunaan Jarak Euclidean dalam algoritma pencarian rute seperti A* didukung oleh sifatnya yaitu heuristik yang konsisten dan admissible. Hal ini berarti perkiraan jarak dengan menggunakan Euclidean tidak pernah melebihi jarak sebenarnya, sehingga membantu algoritma untuk lebih efisien dalam menemukan solusi optimal.

F. Robotika dan Kendaraan Otonom

Dalam robotika dan kendaraan otonom, vektor Euclidean digunakan untuk navigasi secara lokal, seperti menghindari rintangan yang ada atau mengikuti suatu jalur tertentu. Sistem ini memanfaatkan sensor untuk mendeteksi lingkungan sekitar dan menggunakan vektor untuk merencanakan rute berdasarkan data yang diperoleh.

IV. Implementasi dan Simulasi

A. Simulasi Studi Kasus

Disini kita akan menentukan beberapa jarak dua vektor yang berbeda-beda.

ITB Genesha dan Gedung Sate:

1. ITB Ganesha : 6.890903°LS, 107.6104°BT
2. Gedung Sate :6.902639°LS, 107.6192°BT

ITB Ganesha dan ITB Jatinangor :

1. ITB Ganesha : 6.890903°LS, 107.6104°BT
2. ITB Jatinangor : 6.92809°LS,107.7704°BT

ITB Ganesha dan ITB Cirebon :

1. ITB Ganesha : 6.890903°LS, 107.6104°BT
2. ITB Cirebon : 6.6649° LS, 108.4160° BT

ITB Ganesha dan Universitas Indonesia :

1. ITB Ganesha : 6.890903°LS, 107.6104°BT
2. UI :6.360063°LS, 106.826538°BT

Dengan menggunakan rumus Euclidean yaitu:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

maka dapat ditentukan jarak Euclideannya.

B. Implementasi dalam Python

```python
import math

def euclidean_distance(point1, point2):
    x1, y1 = point1
    x2, y2 = point2
    scaling_factor = 111  # rata-rata jarak 1 derajat latitude/longitude dalam kilometer
    return math.sqrt((x2 - x1)**2 + (y2 - y1)**2) * scaling_factor

#misalkan kita ingin mencari jarak antara ITB Ganesha dan Gedung Sate
itb_gane = (-6.8903617, 107.6101912)
gedung_sate = (-6.902484, 107.618621)
itb_nangor = (-6.92809,107.7704)
itb_cirebon = (-6.6649, 108.4160)
ui = (-6.360063, 106.826538)

print("Jarak antara ITB Ganesha dengan beberapa tempat:")
print("")
print("ITB Ganesha dengan Gedung Sate")
print("ITB Ganesha : -6.8903617, 107.6101912")
print("Gedung Sate : -6.902484, 107.618621")
print(f"Jarak ITB Ganesha dengan Gedung Sate yaitu: {euclidean_distance(itb_gane, gedung_sate):.5f} km")
print("")
print("ITB Ganesha dengan ITB Jatinangor")
print("ITB Ganesha : -6.8903617, 107.6101912")
print("ITB Jatinangor : -6.92809,107.7704")
print(f"Jarak ITB Ganesha dengan ITB Jatinangor yaitu: {euclidean_distance(itb_gane, itb_nangor):.5f} km")
print("")
print("ITB Ganesha dengan ITB Cirebon")
print("ITB Ganesha : -6.8903617, 107.6101912")
print("ITB Cirebon : -6.6649, 108.4160")
print(f"Jarak ITB Ganesha dengan ITB Cirebon yaitu: {euclidean_distance(itb_gane, itb_cirebon):.5f} km")
print("")
print("ITB Ganesha dengan UI")
print("ITB Ganesha : -6.8903617, 107.6101912")
print("UI : -6.360063, 106.826538")
print(f"Jarak ITB Ganesha dengan UI yaitu: {euclidean_distance(itb_gane, ui):.5f} km")
```

C. Visualisasi dan Analisis

Dengan memvisualisasikan lokasi-lokasi ini pada peta dua dimensi, kita dapat melihat hubungan antara titik-titik tersebut. Visualisasi ini berguna untuk menverifikasi keakuratan perhitungan. Untuk pengembangan lebih lanjut, algoritma dapat diperluas ke model tiga dimensi untuk aplikasi yang lebih kompleks.

V. Hasil dan Pembahasan

A. Analisis Hasil Perhitungan

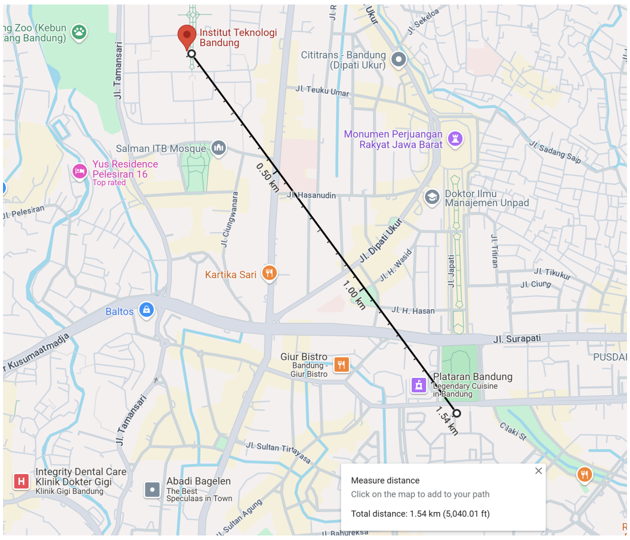
```
PS C:\COOLYEAH\SMT 3\ORKOM\Prak 3 Orkom> python euclidean.py
Jarak antara ITB Ganesha dengan beberapa tempat:

ITB Ganesha dengan Gedung Sate
ITB Ganesha : -6.8903617, 107.6101912
Gedung Sate : -6.902484, 107.618621
Jarak ITB Ganesha dengan Gedung Sate yaitu: 1.63894 km

ITB Ganesha dengan ITB Jatinangor
ITB Ganesha : -6.8903617, 107.6101912
ITB Jatinangor : -6.92809,107.7704
Jarak ITB Ganesha dengan ITB Jatinangor yaitu: 18.26963 km

ITB Ganesha dengan ITB Cirebon
ITB Ganesha : -6.8903617, 107.6101912
ITB Cirebon : -6.6649, 108.4160
Jarak ITB Ganesha dengan ITB Cirebon yaitu: 92.87993 km

ITB Ganesha dengan UI
ITB Ganesha : -6.8903617, 107.6101912
UI : -6.360063, 106.826538
Jarak ITB Ganesha dengan UI yaitu: 105.03023 km
```

Gambar 5.1. Hasil menjalankan program implementasi dalam python

Hasil simulasi menunjukkan bahwa jarak antara ITB Ganesha dengan Gedung Sate adalah sekitar 1.63 km, jarak antara ITB Ganesha dengan ITB Jatinangor adalah 18.27 km, jarak antara ITB Ganesha dengan ITB Cirebon adalah 92.88 km, dan jarak antara ITB Ganesha dengan Universitas Indonesia adalah 105.83 km. Jika dibandingkan dengan pengukuran jarak menggunakan Google Maps, jarak antara ITB Ganesha dengan Gedung Sate adalah sekitar 1.54 km, jarak antara ITB Ganesha dengan ITB Jatinangor adalah 18.09 km, jarak antara ITB Ganesha dengan ITB Cirebon adalah 92.43 km, dan jarak antara ITB Ganesha dengan Universitas Indonesia adalah 104.73 km. Perhitungan ini menggunakan data koordinat GPS yang sederhana dengan faktor koreksi yang sesuai untuk menyesuaikan jarak latitude/longitude ke kilometer.

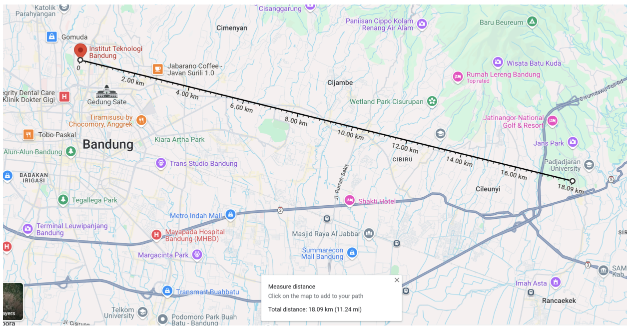
Jika dibandingkan dengan perhitungan manual, hasil algoritma implementasi dengan python ini memberikan nilai hasil yang hampir sama, menunjukkan akurasi metode yang diunakan untuk kasus sederhana. Adanya perbedaan dimungkinkan karena perbedaan pembulatan koordinat atau *scaling factor* yaitu rata-rata jarak 1 derajat latitude/longitude dalam kilometer yang bernilai 111 km untuk 1 derajat. Hasil ini juga konsisten dengan nilai jarak yang diperoleh dari peta digital seperti Google Maps, yang menggunakan pendekatan serupa hal ini juga dalam perhitungan jaraknya.

Berikut adalah jarak dari tempat-tempat tersebut berdasarkan Google Maps:



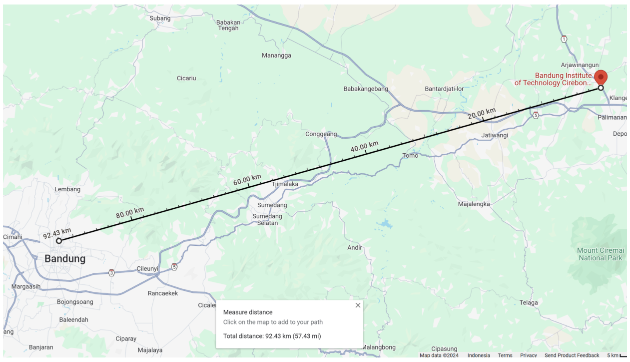
Gambar 5.2 Jarak ITB Ganesha ke Gedung Sate berdasarkan Google Maps

https://www.google.com/maps/place/Institut+Teknologi+Bandung/@-6.8971086,107.6181792,15.93z/data=!4m6!3m5!1s0x2e68e65767c9b183:0x2478e3dcdce37961!8m2!3d-6.8903617!4d107.6101912!16s%2Fm%2F06wc97_?entry=ttu&g_ep=EgoyMDI0MTIxMS4wIKXMDSoASAFQAw%3D%3D



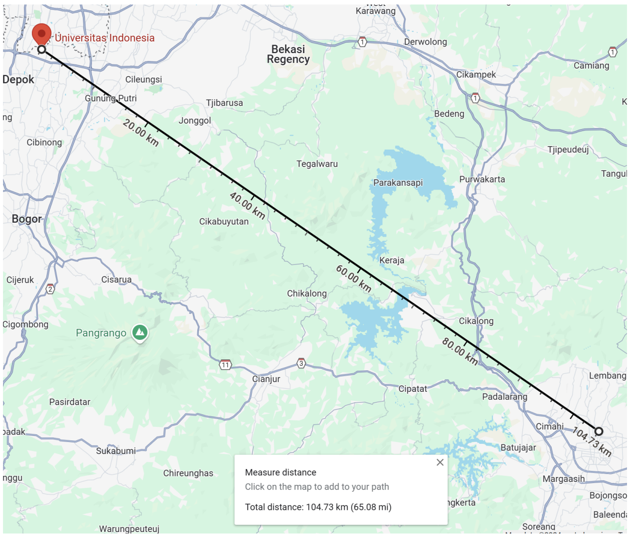
Gambar 5.3 Jarak ITB Ganesha ke ITB Jatinangor berdasarkan Google Maps

https://www.google.com/maps/place/Institut+Teknologi+Bandung/@-6.9163262,107.6913975,13.1z/data=!4m6!3m5!1s0x2e68e65767c9b183:0x2478e3dcdce37961!8m2!3d-6.8903617!4d107.6101912!16s%2Fm%2F06wc97_?entry=ttu&g_ep=EgoyMDI0MTIxMS4wIKXMDSoASAFQAw%3D%3D



Gambar 5.4 Jarak ITB Ganesha ke ITB Cirebon berdasarkan Google Maps

https://www.google.com/maps/place/Bandung+Institute+of+Technology+Cirebon+Campus/@-6.7713869,107.9531504,10.83z/data=!4m6!3m5!1s0x2e6edf8c624e0eeb:0x750d66e2866d640b!8m2!3d-6.664872!4d108.416003!16s%2Fg%2F11gxs0hbhy?entry=ttu&g_ep=EgoyMDI0MTIxMS4wIKXMDSoASAFQAw%3D%3D



Gambar 5.5 Jarak ITB Ganesha ke Universitas Indonesia berdasarkan Google Maps

https://www.google.com/maps/place/Universitas+Indonesia/@-6.8930304,107.597889,13.1z/data=!4m6!3m5!1s0x2e69ec1a804e8b85:0xd7bf80e1977cea07!8m2!3d-6.3606229!4d106.8272343!16zL20vMDF6bHJ2?entry=ttu&g_ep=EgoyMDI0MTIxMS4wIKXMDSoASAFQAw%3D%3D

B. Keterbatasan Metode

1. Tidak memperhitungkan medan atau penghalang fisik dalam jarak Euclidean.
2. Keakuratan koordinat GPS yang digunakan dapat memengaruhi hasil hasil perhitungan.
3. Algoritma ini tidak mempertimbangkan faktor lalu lintas atau kondisi jalan.

C. Keuntungan Metode Euclidean

1. Sederhana dan mudah diimplementasikan.
2. Efisien dalam menghitung jarak lurus dalam ruang dua dimensi.

3. Dapat digunakan sebagai dasar heuristik untuk algoritma pencarian rute yang lebih kompleks seperti A*

VI. Kesimpulan

Vektor Euclidean memiliki peran penting dalam navigasi digital. Dari GPS hingga ke peta digital seperti Google Maps, konsep ini memberikan dasar yang kuat untuk menghitung jarak dan rute secara efisien. Namun, pendekatan ini memiliki keterbatasan, seperti tidak memperhitungkan medan atau hambatan secara fisik. Dalam kasus tertentu, metode ini perlu dilengkapi dengan algoritma lain yang lebih kompleks untuk mempertimbangkan variabel tambahan seperti lalu lintas dan kondisi jalanan.

Integrasi dari vektor Euclidean dengan teknologi *real-time* dan kecerdasan buatan (AI) memberikan potensi besar terhadap aplikasi navigasi yang lebih canggih di masa depan. Misalnya, sistem kendaraan otonom dan robotika dapat memanfaatkan kembali kombinasi algoritma ini untuk menghasilkan navigasi yang aman dan efisien. Selain itu, metode ini dapat menjadi dasar sebuah pengembangan untuk algoritma berbasis graf yang lebih kompleks sehingga dapat memungkikan perhitungan rute yang lebih opitmal.

VII. Acknowledgment

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa yang telah memampukan penulis untuk menyelesaikan makalah ini dengan baik. Selanjutnya penulis juga mengucapkan terima kasih kepada orang tua penulis serta rekan-rekan di sekitar penulis yang terus memberikan dukungan dan nasihat sehingga menjadi motivasi bagi penulis dalam menyelesaikan kewajiban dalam perkuliahan di ITB.

Penulis juga mengucapkan terima kasih kepada Bapak Ir. Rila Mandala, M.Eng., Ph.D. dan Bapak Dr. Ir. Rinaldi Munir selaku dosen mata kuliah Aljabar Geometri yang telah memberikan tugas ini serta membekali penulis dengan materi yang berkaitan dengan Aljabar Geometri yang digunakan dan dimanfaatkan dalam penulisan makalah ini. Tidak lupa penulis juga berterima kasih kepada pihak-pihak yang secara langsung maupun tidak telah membantu dalam merampungkan makalah ini.

References

[1] J. Vince, *Geometric Algebra for Computer Graphics.*

[2] H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985.

[3] Google Maps API Documentation. [Online]. Available: https://developers.google.com/maps/documentation?hl=id. diakses pada 30 Desember 2024

[4] R. Munir, *Vektor di Ruang Euclidean (Bagian 1)*, Bahan Kuliah IF2123 Aljabar Linier dan Geometri, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, 2024. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2024-2025/Algeo-11-Vektor-di-Ruang-Euclidean-Bag1-2024.pdf . diakses pada 30 Desember 2024.

[5] Google Maps. [Online]. Available: https://www.google.com/maps . diakses pada 30 Desember 2024.

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 31 Desember 2024

Muhammad Timur Kanigara 13523055