

Pencarian Rute Indoor Terpendek dalam Lingkungan Universitas Kristen Petra Surabaya menggunakan Algoritma D* Lite berbasis Android

Richard Hans Krisnajana, Agustinus Noertjahyana
Program Studi Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) - 8417658
E – mail: hanskrisnajana@ymail.com, agust@petra.ac.id

ABSTRAK

Pada zaman sekarang ini, sebuah bangunan yang besar dan terbagi menjadi beberapa lantai seringkali membingungkan pengunjung dalam mencari sebuah tempat / ruangan dan menghabiskan banyak waktu sia sia, salah satu contohnya adalah pada lingkungan Universitas Kristen Petra Surabaya.

Aplikasi yang akan dibuat dapat mempermudah baik mahasiswa, dosen, staff maupun pengunjung Universitas Kristen Petra Surabaya dalam menemukan rute terpendek menuju tujuan mereka. Untuk menjalankan aplikasi, diperlukan *server* berupa *database* untuk menyimpan node, data ruang, data wifi, dan data geomagnetis. Pengambilan data dilakukan secara manual dan diolah agar dapat digunakan sebagai acuan. Setelah proses pengambilan data, *user* dapat memasukkan *input* berupa tujuan mereka dan sistem akan melakukan proses kalkulasi rute terpendek serta menampilkan rute tersebut kepada *user*.

Berdasarkan hasil pengujian yang sudah dilakukan, aplikasi yang telah dibuat mampu memberikan rute terpendek dan menampilkannya kepada *user*. Selain itu aplikasi juga mampu memberikan kalkulasi rute terpendek dengan rata-rata waktu yang singkat, dan memberikan estimasi lokasi serta waktu yang tepat kepada *user* selama proses navigasi.

Kata Kunci: *Android, rute terpendek, D* Lite, wi-fi fingerprinting, geomagnetic matching*

ABSTRACT

In today's era, a large building divided into several floors often confuses visitors in looking for a place / room and spends a lot of time in vain, one example is the Petra Christian University Surabaya environment.

The application that will be made can make it easier for students, lecturers, staff and visitors to Petra Christian University Surabaya in finding the shortest route to their destination. To run the application, a server is needed in the form of a database to store nodes, space data, wi-fi data, and geomagnetic data. Data retrieval is done manually and processed so that it can be used as a reference. After the data retrieval process, the user can enter input in the form of their destination and the system will calculate the shortest route and display the route to the user.

Based on the results of the tests that have been carried out, the application that has been made is able to provide the shortest route and display it to the user. In addition, the application is also able to provide the shortest route calculation with a short average time,

and provide the user with an estimate of the exact location and time during the navigation process.

Keywords: *Android, rute terpendek, D* Lite, wi-fi fingerprinting, geomagnetic matching*

1. PENDAHULUAN

Gedung adalah bangunan tembok dan sebagainya yang berukuran besar sebagai tempat kegiatan, seperti perkantoran, pertemuan, perniagaan, pertunjukan, olahraga, dan sebagainya [2]. Bagian dalam dari gedung sendiri dapat dibagi menjadi beberapa lantai sesuai keinginan dari pemilik gedung tersebut dan setiap lantainya pun dapat dibagi lagi menjadi ruang olahraga, ruang bermain, ruang rapat, dan sebagainya.

Mencari sebuah tempat / ruangan pada gedung yang berskala besar dan kompleks dapat menghabiskan banyak waktu sia sia. Pendetang seringkali kesulitan dalam mencari tempat / ruangan yang diinginkan dan menghabiskan seperempat waktu mereka hanya untuk mencari tempat / ruangan tersebut [11]. Seperti yang diketahui sekarang ini, Universitas Kristen Petra Surabaya sendiri memiliki 4 gedung yang berskala besar, yaitu gedung T, W, P, dan Q. Setiap gedung tersebut memiliki luas, susunan lantai, dan ruang yang berbeda-beda sesuai dengan keperluannya masing-masing. Hal ini tentunya dapat membingungkan orang yang masih belum mengenal lingkungan Universitas Kristen Petra Surabaya dengan baik dalam mencari tujuan yang diinginkan tanpa adanya bantuan navigasi. Istilah navigasi bukan menjadi hal yang asing bagi kita. Aplikasi yang umum digunakan untuk navigasi diantaranya adalah *Google Maps, Waze, Sygic GPS*, dan lain sebagainya.

Pada penelitian sebelumnya telah dilakukan perancangan dan pembuatan aplikasi peta *online* Universitas Kristen Petra Surabaya dengan fitur pencarian rute terpendek menggunakan algoritma *A** [5]. Penelitian tersebut menggunakan sistem *node* dimana *administrator* akan menginputkan nilai dari *node-node* yang dapat dilewati sebagai rute perjalanan sehingga apabila *user* menginputkan tujuan yang diinginkan, maka aplikasi akan mencari rute terpendek dari *node-node* yang telah diinputkan sebelumnya oleh *administrator*. Kelemahan dari penelitian ini adalah hasil rute yang diberikan kepada *user* masih statis sehingga tidak dapat melacak *live location* dari *user*.

Penelitian lain yang dilakukan oleh [9] mengenai navigasi dalam ruangan dengan memanfaatkan *Bluetooth Low Energy (BLE) Beacon*. Penelitian ini memberikan hasil yang cukup tinggi dari sisi akurasi posisi *user* dan *BLE* sendiri merupakan salah satu metode yang efisien dalam hal energi. Akan tetapi metode *BLE* ini memiliki kekurangan yaitu bergantung pada kekuatan sinyal *Bluetooth* yang mudah terdistraksi oleh objek yang menghalangi seperti tembok sehingga akurasi lokasi yang diinginkan dapat berkurang [1].

Selain itu, metode *BLE* ini memerlukan *device* tambahan yaitu *BLE Beacon* sehingga memerlukan biaya yang lebih tinggi.

Selain itu juga ada penelitian mengenai navigasi dalam ruangan sebagai penunjuk arah dalam fasilitas transportasi publik oleh [4] dengan memanfaatkan *WiFi*, *GPS*, *Bluetooth*, kamera, dan sensor tekanan serta pengukuran inersia untuk *Pedestrian Dead Reckoning* dan *multi-level navigation*. Penelitian ini memberikan hasil yang cukup memuaskan dalam pengujian berskala besar dan dapat memberikan hasil yang akurat dengan memanfaatkan algoritma *A** serta *modified-Manhattan distance* untuk menghitung nilai heuristik.

Masalah yang diangkat dari skripsi ini adalah mempermudah perjalanan yang perlu dilakukan oleh mahasiswa, dosen, maupun pengunjung Universitas Kristen Petra Surabaya dalam menemukan tujuan yang diinginkan. Skripsi ini menggunakan satu sensor yaitu *magnetometer* serta *Wi-Fi* dalam *smartphone* untuk mendeteksi posisi *live* dari *user* termasuk apakah *user* berada di dalam ruangan dan di lantai berapa. Melalui data yang didapatkan dari sensor, peneliti dapat memperoleh data yang akurat mengenai posisi *user* dan selanjutnya data tersebut akan digunakan untuk kalkulasi rute terpendek menuju tujuan yang diinginkan menggunakan algoritma *D* Lite*. Algoritma *D* Lite* melakukan kalkulasi mulai dari *node* tujuan ke *node* awal sehingga perubahan *node* awal yang disebabkan oleh *update location* tidak perlu diulang lagi dari awal. Pergerakan *user* akan dideteksi dan diestimasi menggunakan dua sensor yaitu *gyroscope* dan *accelerometer* yang ada di dalam *smartphone*.

2. LANDASAN TEORI

2.1 Algoritma *D* Lite*

Algoritma *D* Lite* disajikan sebagai sarana untuk memecahkan masalah *pathfinding real-time* baik dalam lingkungan sebagian tidak diketahui maupun dinamis [6]. Algoritma *D* Lite* hanya membalikkan pencarian arah *Lifelong Planning A** (*LPA**), pencarian sekarang dilakukan dari *node* tujuan ke *node* awal, sehingga nilai *G* sekarang mewakili *cost* dari tujuan (*CSG*), bukan nilai *CSF* per-*node*. Jika nilai-nilai heuristik berubah maka *open list ordering* algoritma *D* Lite* akan di-*refresh* dan *open list* perlu diatur kembali dengan mempertimbangkan perubahan nilai heuristik. *D* Lite* menggunakan metode algoritma *D** untuk mencegah penataan ulang *open list* dengan menggunakan prioritas simpul batas bawah dengan yang digunakan di *LPA** [11]. Untuk menghapus kebutuhan penataan kembali *open list*, fungsi heuristik yang digunakan harus memenuhi rumus berikut:

$$h(s, s_{start}) = \begin{cases} 0, & s = s_{goal} \\ c(s', s) + h(s', s_{goal}), & \text{others} \end{cases} \quad (1)$$

Dimana $h(s, s')$ adalah nilai heuristik antara *node* s dan s' , sedangkan $c(s, s')$ adalah *cost* perpindahan dari *node* s ke s' . Fungsi heuristik harus diperkirakan dengan tepat dari *cost* antara dua *node*. Ketika *agent* berpindah dari *node* s ke s' , nilai heuristik akan menurun di sebagian besar $h(s, s')$ sehingga nilai F di *open list* akan juga menurun di sebagian besar $h(s, s')$.

2.2 Wireless Fidelity (Wi-Fi)

Wireless Fidelity (*Wi-Fi*) sendiri adalah teknologi jaringan nirkabel yang memungkinkan perangkat seperti komputer (laptop dan desktop), perangkat seluler (ponsel pintar dan perangkat yang dapat dikenakan), dan peralatan lain (printer dan kamera video) untuk berinteraksi dengan Internet. Hal ini memungkinkan perangkat-perangkat tersebut bertukar informasi satu sama lain, sehingga

menciptakan jaringan [7]. Cara yang digunakan oleh *Wi-Fi* untuk menunjukkan “keberadaannya” adalah dengan melakukan *broadcast* sinyal melalui *access point* (*AP*). Sinyal yang dikirimkan oleh *Wi-Fi AP* memiliki beberapa informasi mengenai *AP* tersebut diantaranya *SSID*, *MAC address* dari *AP*, *Received Signal Strength* (*RSS*) dan lain sebagainya [8].

Skripsi ini memanfaatkan sampel *Received Signal Strength Indicator* (*RSSI*) dari *Wi-Fi AP* yang akan dimasukkan ke dalam *location fingerprint database*. Akan tetapi sampel *RSSI* tidak dapat digunakan secara langsung karena terdapat distraksi seperti adanya gangguan sinyal sehingga sampel *RSSI* akan diproses dengan membandingkan sampel-sampel *RSSI* yang didapatkan dari sebuah lokasi tertentu.

$$Fingerprint_i = \{x_i, y_i, RSSI_i\} \quad (2)$$

Hasil dari perbandingan adalah nilai *average* dari *RSSI* yang merupakan nilai paling sesuai dan akan dijadikan acuan dalam *Wi-Fi location fingerprint* untuk lokasi tersebut dengan X sebagai penanda *longitude* dan Y sebagai penanda *latitude*.

2.3 Magnetometer

Magnetometer adalah sensor yang digunakan untuk mengukur magnetisasi dari bahan magnetik atau untuk mengukur kekuatan arah medan magnet pada titik dalam ruang atau sebagai kompas [3]. Sebagian besar *smartphone* sudah dilengkapi dengan sensor *magnetometer* di dalamnya.

Skripsi ini memanfaatkan sensor *magnetometer* pada *smartphone* untuk membaca medan magnet bumi melalui tiga arah, yaitu x sebagai utara, y sebagai timur, dan z tegak lurus pusat bumi. Penyebaran medan magnet dalam sebuah lingkungan *indoor* pada dasarnya bersifat stabil dan unik sehingga dapat digunakan sebagai acuan lokasi *indoor* tertentu. Maka dari itu, sampel medan magnet yang didapatkan dari sensor *magnetometer* dapat digunakan sebagai *geomagnetic fingerprint* dengan memenuhi persamaan sebagai berikut.

$$G_{total} = \sqrt{G_x^2 + G_y^2 + G_z^2} \quad (3)$$

Dengan G_{total} sebagai nilai total medan magnet serta G_x , G_y , dan G_z mewakili komponen-komponen intensitas vektor geomagnetik di utara, timur, dan tegak lurus pusat bumi [13].

2.4 Gyroscope

Gyroscope adalah alat sensor yang dipakai untuk melacak rotasi atau perputaran suatu perangkat berdasarkan gerakan. Dengan kata lain *gyroscope* juga disebut sebagai perangkat yang dipakai untuk mempertahankan orientasi dari sebuah sudut agar tetap stabil [3]. *Gyroscope* mampu mendeteksi gerakan gravitasi pengguna yang sedang melakukan perpindahan rotasi kepala atau pada saat berjalan.

Penggunaan *gyroscope* harus diawali dengan kalibrasi *gyro sensor* dengan tujuan untuk mendapatkan nilai faktor kalibrasi. Output yang dihasilkan *gyroscope* berupa kecepatan sudut dari arah 3 sumbu yaitu sumbu x yang menjadi sudut ϕ (kanan dan kiri), sumbu y yang nantinya akan menjadi sudut θ (atas bawah) dan sumbu z yang menjadi sudut ψ (depan dan belakang) [3]. Skripsi ini memanfaatkan sensor *gyroscope* pada *smartphone* untuk mendapatkan data orientasi *user* dimana data tersebut nantinya akan digunakan sebagai acuan arah hadap *user*.

2.5 Accelerometer

Accelerometer adalah *electromechanical device* yang mengukur gaya percepatan yang disebabkan oleh gerakan, gravitasi atau getaran. Gaya-gaya ini dapat bersifat statis seperti gaya gravitasi, gerakan indra dinamis, maupun getaran [10]. *Accelerometer* pada *smartphone* pada awalnya digunakan untuk mengukur akselerasi linear tetapi seiring dengan berkembangnya teknologi, *accelerometer* memiliki fungsi yang lebih canggih yaitu *motion input* dan *orientation sensing*. *Motion input* sudah menjadi hal yang umum karena *smartphone* sudah dilengkapi dengan fitur *touchscreen*, sedangkan *orientation sensing* dapat dirasakan saat kita memutar *smartphone* dari horizontal menjadi vertikal dan secara otomatis orientasi pada *smartphone* juga ikut berubah.

Skripsi ini memanfaatkan sensor *accelerometer* pada *smartphone* untuk mendapatkan data yang berkaitan dengan pergerakan *user* dari tiga sisi yaitu x, y, dan z. Data yang didapatkan berupa data mentah bernilai x, y, z yang dapat diolah dengan metode Kalman Filter untuk memperoleh data yang tepat [12].

2.6 Gerak Lurus Beraturan (GLB)

Gerak Lurus Beraturan (GLB) adalah gerak suatu benda pada lintasan yang lurus di mana pada setiap selang waktu yang sama, benda tersebut menempuh jarak yang sama (gerak suatu benda pada lintasan yang lurus dengan kelajuan tetap) [14]. Pada gerak lurus beraturan, benda menempuh jarak yang sama dalam selang waktu yang sama pula. Persamaan dari gerak lurus beraturan yaitu sebagai berikut.

$$s = v \cdot t \quad (4)$$

Dalam persamaan tersebut dapat dilihat bahwa jarak (s) ditetapkan sebagai perpindahan, (t) sebagai waktu, dan (v) sebagai kecepatan. Gerak lurus beraturan pada skripsi ini digunakan untuk menghitung estimasi waktu dan jarak.

2.7 Gerak Lurus Berubah Beraturan (GLBB)

Gerak Lurus Berubah Beraturan (GLBB) adalah benda dengan lintasan yang lurus dan kelajuannya mengalami perubahan yang sama setiap sekon [14]. Berdasarkan kelajuannya dapat dibedakan menjadi dua yaitu apabila kelajuan benda bertambah dengan nilai yang sama setiap sekonnya, maka disebut GLBB dipercepat sedangkan apabila kelajuan benda berkurang dengan nilai yang sama setiap sekonnya, maka disebut GLBB diperlambat. Persamaan GLBB menggunakan tanda positif (+) maupun negatif (-) tergantung pada dipercepat atau diperlambatnya gerak. Persamaan dari gerak lurus berubah beraturan adalah sebagai berikut.

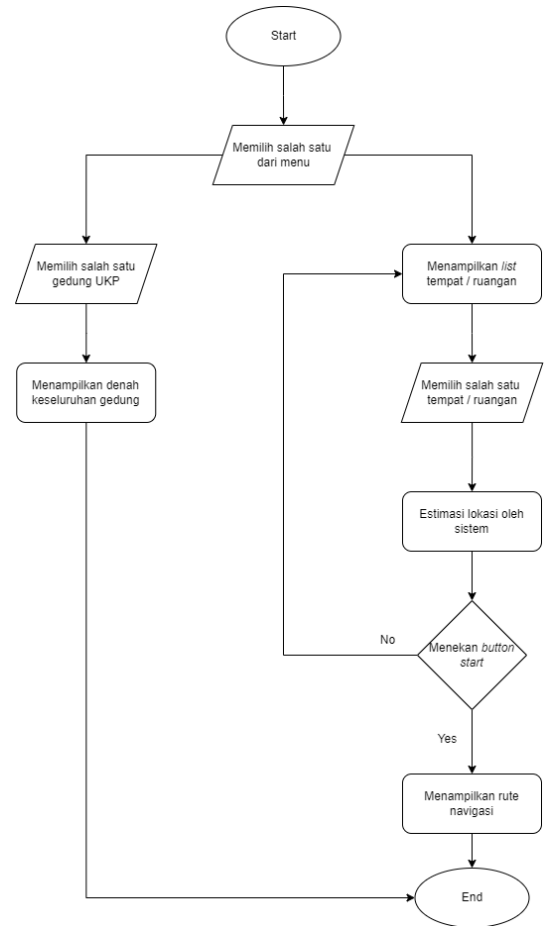
$$v_t^2 = v_0^2 + 2 \cdot a \cdot s \quad (5)$$

$$v_t = v_0 + a \cdot s \quad (6)$$

$$s = v_0 \cdot t + \frac{1}{2} \cdot a \cdot t^2 \quad (7)$$

Dalam persamaan tersebut dapat dilihat bahwa jarak (s) ditetapkan sebagai perpindahan, (t) sebagai waktu, (v) sebagai kecepatan, (v₀) sebagai kecepatan awal dan (a) sebagai percepatan. Gerak lurus berubah beraturan pada skripsi ini kurang lebih sama dengan GLB tetapi GLBB akan digunakan apabila user memiliki perbedaan kecepatan pada saat berjalan.

3. DESAIN SISTEM



Gambar 1. Flowchart Aplikasi

Tampilan denah ruang yang akan dibuat berupa denah ruang dari masing-masing gedung Universitas Kristen Petra Surabaya. Denah ruang tersebut akan diolah / digabungkan dengan *node-node* yang telah ditentukan sebelumnya. *Node-node* tersebut didapatkan dengan cara melakukan *mapping* secara manual. Fungsi daripada *node-node* tersebut adalah untuk menentukan posisi mana saja yang dapat dilalui sebagai rute oleh *user* serta dapat digunakan sebagai acuan untuk menentukan posisi *user*.

Untuk menentukan posisi awal *user*, sistem menggunakan metode *Wi-Fi Fingerprinting* dan *Geomagnetic Matching* dimana sistem akan memerlukan *database* tambahan yaitu *fingerprint location database* untuk masing-masing metode. Langkah-langkah yang dilalui untuk mendapatkan data terkait *fingerprint location database* yaitu sebagai berikut.

1. Mengambil data mengenai *node-node* yang telah diambil sebelumnya pada *database node*.
2. Pada setiap *node* akan dilakukan pengambilan data terkait *Received Signal Strength Indicator (RSSI)* dari *Wi-Fi Access Point* terdekat dengan *value* berupa *negative integer* dan *geomagnetic strength* dengan *value* berupa *micro-Tesla (μT)*. Data terkait RSSI akan dimasukkan ke dalam *wifi fingerprint database*, sedangkan data terkait *geomagnetic strength* akan dimasukkan ke dalam *geomagnetic fingerprint database*.

- Proses pengambilan data baik mengenai RSSI maupun *geomagnetic strength* dilakukan sebanyak 10 kali untuk setiap *node*. Hal ini bertujuan untuk meningkatkan ketepatan dari data yang diambil.
- Data yang diperoleh akan di-*filter* untuk dijadikan sebagai acuan untuk menentukan posisi awal *user*.

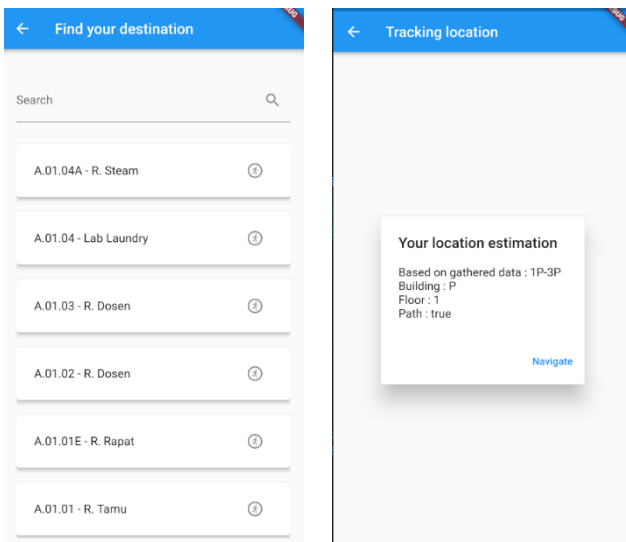
Alur kerja aplikasi ketika pertama kali dijalankan oleh *user* akan diawali dengan 2 pilihan *menu* dimana *user* dapat memilih antara mencari tempat tujuan yang diinginkan atau *explore* gedung Universitas Kristen Petra Surabaya. Apabila *user* memilih *menu explore* maka *user* harus memilih salah satu gedung Universitas Kristen Petra Surabaya. Aplikasi akan menampilkan denah keseluruhan dari gedung yang telah dipilih oleh *user* beserta tempat / ruangan yang ada di gedung tersebut. Sedangkan apabila *user* memilih *menu* mencari tujuan yang diinginkan, maka *user* harus memilih salah satu tempat / ruangan dan dilanjutkan dengan estimasi lokasi awal *user* oleh sistem. Setelah itu aplikasi akan melakukan kalkulasi rute dari posisi awal ke tujuan dan memberikan navigasi kepada *user*. *Flowchart* dari alur kerja aplikasi dapat dilihat pada Gambar 1.

4. PENGUJIAN SISTEM

4.1 Pengujian Kalkulasi Rute Terpendek

Pengujian dilakukan dengan tujuan untuk mengetahui kesesuaian *cost* antara lokasi awal dengan lokasi tujuan. Selain itu, pengujian juga dilakukan untuk memastikan kesamaan situasi yang terdapat pada aplikasi dengan keadaan sebenarnya.

Pada Gambar 2 merupakan halaman list tempat / ruangan yang dapat dituju oleh *user*. Setelah *user* memilih salah satu tempat yang dituju, maka *user* akan diarahkan ke halaman estimasi lokasi dimana sistem akan melakukan estimasi lokasi berdasarkan data yang dimiliki. Apabila *user* berada diluar lingkungan gedung Universitas Kristen Petra Surabaya maka sistem akan menampilkan *error* estimasi lokasi sebaliknya apabila sistem menemukan lokasi *user* maka akan ditampilkan seperti Gambar 2.

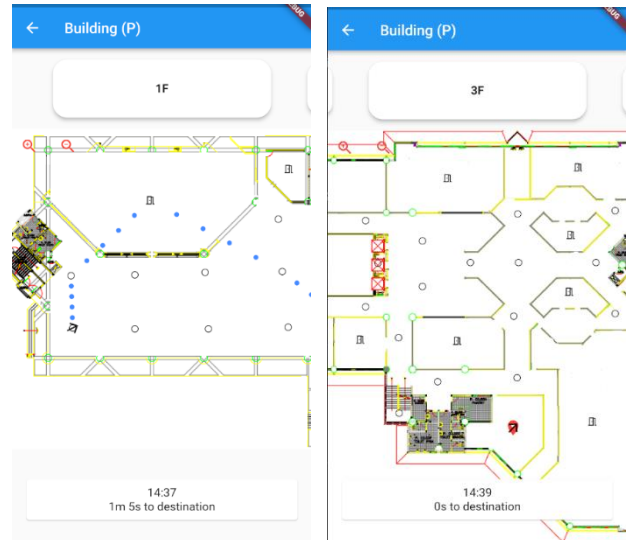


Gambar 2. Tampilan *list* ruangan dan estimasi lokasi

Setelah *user* menekan tombol *navigate*, maka *user* akan diarahkan ke halaman denah dimana sudah terdapat rute terpendek yang perlu dilalui oleh *user* untuk mencapai tujuan. Tampilan awal navigasi

adalah seperti pada Gambar 3. Dimana *user* akan diberi tanda panah dan titik-titik biru sebagai rute yang harus ditempuh *user*.

Apabila *user* mulai bergerak, maka *icon* panah yang menandakan *user* juga akan ikut bergerak dan titik-titik biru yang telah dilalui *user* hilang. Waktu estimasi juga akan berkurang sesuai posisi *user* dengan tujuan. Perubahan tampilan setelah pergerakan *user* dapat dilihat pada Gambar 3. Apabila *user* bergerak hingga pindah lantai, baik turun maupun naik, maka denah yang ditampilkan juga akan menyesuaikan dengan posisi *user*. Begitu pula dengan navigasi akan melanjutkan sampai tujuan yang diinginkan. Tampilan dapat dilihat pada Gambar 3. Apabila *user* telah bergerak hingga sampai pada tujuan yang diinginkan, maka titik-titik biru akan hilang semua dan estimasi waktu berubah menjadi 0s. Tujuan *user* ditandai dengan *marker* merah seperti pada Gambar 3.



Gambar 3. Tampilan navigasi

4.2 Proses terkait Pengujian D^* Lite

Dari hasil yang didapatkan melalui pengujian aplikasi, hal yang perlu dibahas adalah mengenai berapa lama rata-rata waktu algoritma D^* Lite mampu melakukan kalkulasi rute terpendek.

Tabel 1. Tabel pengujian algoritma D^* Lite yang dipotong

Starting point	End Point	Beda Lantai	Waktu	Jarak
Kantin P	Exit P ke Q	0	1,964 detik	± 45 meter
Lab PG	Puskom P2	0	1,833 detik	± 32 meter
TU Elektro	P.05.06	2	3,460 detik	± 80 meter
R. Pamer Interior	TU FSD	1	2,661 detik	± 28 meter
P.04.07	Lab Statistik	3	5,076 detik	± 70 meter

Berdasarkan Tabel 1, dapat diketahui bahwa waktu yang diperlukan oleh algoritma D^* Lite dalam menentukan rute terpendek berbeda-beda. Apabila perbedaan lantai bernilai 0 dengan rata-rata jarak 31,67 meter maka rata-rata waktu yang diperlukan adalah 1,651 detik, perbedaan 1 lantai dengan rata-rata

jarak 35,66 meter memerlukan rata-rata waktu 2,617 detik, perbedaan 2 lantai dengan rata-rata jarak 60 meter memerlukan rata-rata waktu 3,220 detik, dan perbedaan 3 lantai dengan rata-rata jarak 67,5 meter memerlukan rata-rata waktu 4,883 detik.

Apabila diamati dari hasil rata-rata waktu yang didapatkan, dapat diketahui bahwa jarak bukan satu-satunya faktor melainkan juga dipengaruhi oleh perbedaan lantai. Hal ini disebabkan karena algoritma *D* Lite* sendiri bekerja dengan cara mengecek *neighbor* dari *node* yang mungkin menjadi rute terpendek sehingga apabila *node* yang terpilih sebagai rute terpendek merupakan penghubung antar lantai (tangga / lift), maka akan memakan waktu yang lebih lama. Dapat disimpulkan bahwa waktu yang diperlukan oleh algoritma *D* Lite* dalam menentukan rute terpendek dipengaruhi oleh perbedaan lantai, tetapi tidak lepas dari jarak posisi tujuan / rute dengan jalur penghubung antar lantai yaitu tangga dan lift.

Selain rata-rata waktu, terdapat hal lain yang dapat dibahas yaitu mengenai *cost memory* yang diperlukan apabila peneliti mengkalkulasi semua rute terpendek yang memungkinkan terlebih dahulu lalu hasil rute terpendek tersebut disimpan ke dalam *Firebase Firestore*.

Tabel 2. Tabel cost memory

Jumlah node ruang	Memory yang diperlukan	Rata-rata jumlah node rute terpendek
1	25344 bytes	12
15 (1 lantai gedung P)	380160 bytes	180
132 (gedung P)	3345408 bytes	1851

Berdasarkan Tabel 2, dapat diketahui bahwa besar *memory* yang diperlukan dalam *Firebase Firestore* untuk menyimpan 1 rute terpendek dengan rata-rata jumlah *node* 12 buah yaitu 192 bytes. Sehingga besar *memory* yang diperlukan untuk menyimpan rute terpendek dari 1 ruangan ke 132 ruangan dalam lingkup gedung P yaitu 25344 bytes atau 25,34 kilobytes. Dalam lingkup gedung P, besar *memory* yang diperlukan untuk menyimpan semua rute terpendek yang memungkinkan dari 132 ruangan yaitu sebesar 3345408 bytes atau 3345,408 kilobytes. Dapat disimpulkan bahwa *cost memory* yang diperlukan dapat bervariasi sesuai kriteria gedung baik dari banyak ruangan, ukuran gedung, maupun jumlah *node* akan tetapi hasil pengujian menunjukkan bahwa *cost memory* yang diperlukan untuk menyimpan rute terpendek dari semua ruangan dalam suatu gedung relatif kecil.

Pembahasan selanjutnya yaitu mengenai pengujian aplikasi yang berkaitan dengan waktu tempuh beserta penggunaan sensor *accelerometer* dan *gyroscope*. Hasil pengujian dapat dilihat pada Tabel 3.

Tabel 3. Tabel pengujian pergerakan user

Starting Point	End Point	Beda Lantai	Waktu tempuh user	Accuracy
1P-1P	P.01.08	0	15 detik	100%
P.01.08	P.01.04A	0	30 detik	100%

P.01.04A	P.02.02	1	59 detik	97%
P.02.02	P.02.17	0	24 detik	100%
P.02.17	P.02.05A	0	29 detik	99%
P.02.05A	P.02.01	0	8 detik	100%
P.02.01	P.03.06	1	29 detik	100%
P.03.06	P.04.11	1	75 detik	95%
P.04.11	P.03.04	1	59 detik	100%
P.03.04	P.03.20	0	40 detik	100%

Berdasarkan Tabel 3, pengujian aplikasi dilakukan dengan memilih tempat / ruangan secara *random* dan melakukan pengujian dengan cara menempuh rute terpendek dari posisi awal hingga tujuan lalu menghitung waktu tempuh yang diperlukan. Hasil yang diperoleh dapat dilihat pada Tabel 3 dengan *parameter accuracy* sebagai acuan apakah sensor *accelerometer* serta *gyroscope* mampu memberikan data yang sesuai sehingga waktu tempuh *user* masuk ke dalam *range* waktu tempuh aplikasi yang tepat. Persamaan yang digunakan untuk menghitung nilai *accuracy* adalah sebagai berikut.

$$Accuracy = 100\% - \frac{(BB \text{ value range} - \text{waktu tempuh})}{\text{estimasi waktu tempuh aplikasi}} \times 100\% \quad (8)$$

$$Accuracy = 100\% - \frac{(\text{waktu tempuh} - BA \text{ value range})}{\text{estimasi waktu tempuh aplikasi}} \times 100\% \quad (9)$$

Penghitungan *accuracy* diikuti dengan adanya nilai *tolerance* sebesar 15% dari waktu tempuh *user*. Apabila waktu tempuh *user* dikurangi dengan nilai *tolerance* sebesar 15% masih masuk ke dalam *range* waktu tempuh aplikasi maka *accuracy* yang didapatkan sebesar 100%.

4.3 Hasil Wi-Fi Fingerprinting dan Geomagnetic Matching

Melalui pengujian aplikasi juga diperoleh data hasil pengujian estimasi lokasi awal *user* menggunakan *Wi-Fi Fingerprinting* dan *Geomagnetic Matching*. Data hasil pengujian dapat dilihat pada Tabel 4.

Tabel 4. Hasil pengujian estimasi lokasi

Lokasi user	Estimasi lokasi	Accuracy (meter)	Accuracy Tolerance
1P-2P	1P-3P	± 2 meter	± 2 meter
1P-12P	1P-14P	± 3 meter	
1P-4R	1P-4R	± 1 meter	
2P-22P	2P-20P	± 4 meter	
4P-1P	4P-1R	± 2.5 meter	
3P-25P	3P-28P	± 6 meter	
3P-15P	3P-9R	± 3 meter	
8P-1P	8P-1P	± 0.8 meter	

Berdasarkan Tabel 4, pengujian dilakukan dengan memilih tempat / ruangan secara *random* dan melakukan estimasi lokasi menggunakan aplikasi. Penghitungan *accuracy* ketepatan estimasi lokasi dihitung dalam satuan meter dengan nilai *tolerance* sebesar ± 2 meter. Rata-rata akurasi yang didapatkan melalui pengujian adalah ± 3.157 meter.

Pengujian mengenai perbandingan akurasi dapat dipengaruhi oleh beberapa faktor, salah satunya adalah adanya distraksi objek seperti pillar / tembok. Oleh karena itu, pengujian dilakukan dengan menggunakan acuan yang sama, yaitu jarak antara *user* dengan *Wi-Fi Access Point (AP)* pada penelitian ini dan jarak antara *user* dengan *BLE Beacon* pada penelitian terkait. Berdasarkan hasil pengujian, penentuan posisi *indoor user* dengan *Bluetooth Low Energy (BLE) Beacon* memiliki tingkat akurasi sebesar $\pm 4-5$ meter (Samuel et al., 2020) sedangkan *Wi-Fi Fingerprinting* dan *Geomagnetic Matching* memiliki akurasi sebesar ± 3.157 meter sehingga kenaikan akurasi yang diperoleh yaitu sebesar $\pm 0.843-1.843$ meter.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisa sistem, maka dapat disimpulkan bahwa semua perumusan masalah dapat dicapai. Berikut merupakan kesimpulan dari setiap perumusan masalah.

- Aplikasi yang telah dibuat mampu menjawab perumusan masalah yang ada.
- Berdasarkan hasil pengujian yang dilakukan, dapat diambil kesimpulan bahwa penggunaan algoritma *D* Lite* mampu menjadi solusi bagi *user* dalam memberikan rute terpendek yang diperlukan untuk mencapai tujuan.
- Terdapat beberapa faktor yang dapat mempengaruhi hasil pengujian sehingga perlu diperhatikan seperti pengukuran kecepatan *user* yang tidak menentu, posisi *smartphone* maupun bias pada sensor.
- Peta yang digunakan dalam aplikasi merupakan denah dari Unit Perencanaan Fisik Kampus (UPFK) Universitas Kristen Petra Surabaya yang diolah kembali oleh penulis sehingga lebih mudah dipahami serta membantu memfamiliarikan *user* dengan lingkungan Universitas Kristen Petra Surabaya.

5.2 Saran

Saran yang dapat diberikan untuk melakukan penyempurnaan dan pengembangan program lebih lanjut yaitu sebagai berikut.

- Menambahkan denah luar ruangan yang menghubungkan antar gedung Universitas Kristen Petra Surabaya sehingga terbentuk satu denah besar yang saling terhubung.
- Memperbaiki tampilan dan *flow* aplikasi agar lebih *user-friendly*.
- Meningkatkan akurasi metode untuk estimasi lokasi *user* baik dari sisi data maupun metode.

6. DAFTAR PUSTAKA

- [1] Aranda, F. J., Parralejo, F., Alvarez, F. J., Paredes, J. A. (2022). Performance Analysis of Fingerprinting Indoor Positioning Methods with BLE. *Expert Systems with Applications*, 202. <https://doi.org/10.1016/j.eswa.2022.117095>
- [2] Badan Pusat Pengembangan dan Pembinaan Bahasa. (n.d.). *Gedung* (Def. 1). Dalam Kamus Besar Bahasa Indonesia (KBBI) *Online*. Diakses 4 Januari 2022, melalui <https://kbbi.web.id>
- [3] Baktikominfo. (2019, 4 11). *Informasi tentang teknologi gyroscope, fungsi dan cara kerjanya yang wajib dibaca*. Diambil kembali dari Baktikominfo:https://www.baktikominfo.id/id/informasi/pe-ngetahuan/informasi_tentang_teknologi_gyroscope_fungsi_dan_cara_kerjanya_yang_wajib_dibaca-780
- [4] Czogalla, O. (2015). Smartphone based indoor navigation for guidance in public transport facilities. *IFAC PapersOnLine*, 48(10), 233–239. <https://doi.org/10.1016/j.ifacol.2015.08.137>
- [5] Gunawan, L. (2008). *Perancangan dan pembuatan aplikasi peta online Universitas Kristen Petra berbasis web dengan fitur pencarian rute terpendek menggunakan A**. (Skripsi Universitas Kristen Petra Surabaya, 2007). Diakses dari <https://dewey.petra.ac.id/catalog/digital/detail?id=11189>
- [6] Koenig, S. and Likhachev, M. (2002). *D* Lite*. *Proceedings of the National Conference on Artificial Intelligence*, pp. 476–483.
- [7] Kurniawan, A. (2021, 6 3). *Mengenal Fungsi WiFi beserta Jenis dan Cara Kerjanya*. Diambil kembali dari merdeka: <https://www.merdeka.com/jabar/mengenal-fungsi-wifi-beserta-jenis-dan-cara-kerjanya-klm.html>
- [8] Ninh, D. B., He, J., Trung, V. T., Huy, D. P. (2020). An Effective Random Statistical Method for Indoor Positioning System using WiFi Fingerprinting. *Future Generation Computer Systems*, 109. <https://doi.org/10.1016/j.future.2020.03.043>
- [9] Samuel, M., Nazeem, N., Sreevals, P., Ramachandran, R., & Careena, P. (2021). Smart Indoor Navigation and proximity advertising with Android application using BLE Technology. *Materials Today: Proceedings*, 43, 3799–3803. <https://doi.org/10.1016/j.matpr.2020.10.995>
- [10] Sharma, S. (2020, 7 2). *What Is Accelerometer? How to Use Accelerometer in Mobile Devices?*. Diambil kembali dari credencys: <https://www.credencys.com/blog/accelerometer/>
- [11] Stentz, A. (1995). The Focussed *D** Algorithm for Real-Time Planning. *Proceedings of the International Joint Conference on Artificial Intelligence*.
- [12] Tcherniak, D., Schwaab, M.. (2020). On a Method for Finding Position and Orientation of Accelerometers from their Signals. *Mechanical Systems and Signal Processing*, 140. <https://doi.org/10.1016/j.ymsp.2020.106662>
- [13] Wang, Q., Zhou, J. (2018). Simultaneous Localization and Mapping Method for Geomagnetic Aided Navigation. *Optik*, 171, 437–445. <https://doi.org/10.1016/j.ijleo.2018.06.069>
- [14] William, A. (2021, 9 8). *Pengertian Gerak Lurus Beraturan & Berubah Beraturan Serta Rumusnya*. Diambil kembali dari tirto.id: <https://tirto.id/pengertian-gerak-lurus-beraturan-berubah-beraturan-serta-rumusnya-gf65>