

Pencarian Rute Terpendek di dalam Mall menggunakan Lifelong Planning A* pada Android

Igor Maha Putra¹, Agustinus Noertjahyana², Justinus Andjarwirawan³

Program Studi Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra

Jl. Siwalankerto 121 – 131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) – 8417658

E-Mail: igorohnanda@gmail.com¹, agust@petra.ac.id², justin@petra.ac.id³

ABSTRAK

Dari zaman dulu hingga sekarang, tentu orang masih suka melakukan berpergian, terkadang dalam berpergian kadang kita mendatangi toko pusat perbelanjaan atau yang lebih kita kenal dengan sebutan *mall*. Tentu di *mall* kita bisa menikmati banyak sekali fasilitas-fasilitas yang bisa kita manfaatkan di dalam *mall*, yaitu seperti stan makanan dan minuman, toko bangunan, toko baju, dan masih banyak lagi yang bisa kita nikmati tentunya. Selain kita tentu ada subjek-subjek lain yaitu antara lain turis dan pengemudi ojek daring. Dimana tentunya untuk subjek-subjek seperti mereka pun kita tidak bisa menjamin mereka mengerti letak-letak toko maupun fasilitas yang mereka ingin kunjungi. Aplikasi yang dibuat akan membantu semua kalangan yang mendatangi suatu *mall* tertentu. Untuk melakukannya, diperlukan sebuah *server* yang dapat menyimpan data peta dan toko-toko yang dimiliki oleh *mall*. Setelah proses pengambilan data, dilakukan penerimaan masukkan dari pengguna tentang toko / fasilitas yang ingin dikunjungi, kemudian setelah mendapatkan masukkan dari pengguna, maka akan dilakukan proses kalkulasi data yang kemudian akan ditunjukkan rute yang telah dikalkulasi oleh aplikasi dan kemudian hasilnya akan ditampilkan pada aplikasi yang telah dibuat. Berdasarkan hasil pengujian yang sudah dilakukan, program yang dibuat berhasil untuk menyimpan, mengkalkulasi dan menampilkan rute mana yang harus dilalui oleh pengguna.

Kata Kunci: Android, rute terpendek, *Lifelong Planning A**

ABSTRACT

Since past time until now, people still loves to travel and sometimes when they travel they went to a shopping center store or what we known as a mall. Inside the mall there're a lot of facilities that wes can use/enjoy like food and beverages stands, utilities shops, clothing lines, and many more. Apart from us of course there're another subjects that visits the mall including tourists and online taxi drivers. Of course, even them we can't guarantee that they understand the location of the shops or facilities that they want to visit. The application will help all people who came to a particular mall. To do this, we need a server that can store the maps data and shops inside each mall. After the data collection process, the application will receive an input from the user about a shop / a facility that they want to visit, then after received an input from the user, a data calculation process will be carried out which will be shown the route that has been calculated by the application and then the results will be displayed on the application that has been made. Based on the results of the tests that have been done, the program is quite successful in storing, calculation and displaying which route the user must take.

Keywords: Android, *shortest route*, *Lifelong Planning A**

1. PENDAHULUAN

Mall merupakan salah satu tempat yang tidak asing lagi didengar. Mall atau yang mungkin disebut juga sebagai pusat perbelanjaan merupakan tempat yang memiliki banyak stan di dalamnya. Stan yang berada di dalam mall pun beragam ada stan yang menjual baju, menjual makanan, menjual barang elektronik, dan lain sebagainya. Kemudian karena banyak sekali stan yang berada di dalam mall membuat orang-orang pendatang, *traveler*, dan para *ojek online* kebingungan untuk mencari tempat-tempat yang dituju. Berkunjung ke mall tentunya merupakan aktivitas yang mungkin sering dilakukan oleh beberapa orang yang kita kenal, kegiatan yang dilakukan pun bermacam-macam ada yang pergi untuk mencari *furniture*, belanja bulanan, ada yang sekedar jalan-jalan, dan lain sebagainya. Orang-orang yang berada di dalam mall tidak semuanya merupakan orang yang sering mengunjungi mall tersebut bisa juga seorang *traveller* yang sedang berkunjung atau *ojek online* yang mendapatkan pesanan untuk salah satu stan yang ada di dalam mall tersebut. Tentunya dengan jaman yang sudah maju ini kita pasti pernah mendengar atau mengetahui aplikasi untuk navigasi, salah satu yang paling populer adalah *Google Maps*. Rata-rata dari kita mungkin sudah menggunakan *Google Maps* untuk menjadi aplikasi navigasi pada saat kita berpergian menggunakan kendaraan pribadi, tetapi *Google Maps* hanya menampilkan tampilan 2 dimensi (2D) sehingga tidak bisa mengukur secara tepat untuk rute yang efektif karena hanya menghitung koordinat *longitude* dan *latitude* dari *data* yang dimiliki dan tidak menghitung lantai lokasi *user* dengan *stand* yang ingin dikunjungi. Adanya *Indoor Location Positioning* bisa membantu banyak orang dalam menemukan stan yang mereka ingin kunjungi [4], dengan sistem yang ada sekarang sebenarnya kurang efisien karena hanya menggunakan GPS untuk menentukan suatu lokasi seseorang dan tidak mungkin bisa menentukan *user* berada di lantai berapa pada sebuah mall [5]. Sebenarnya cukup sulit untuk merealisasikan *Indoor Positioning* di dalam sebuah mall karena *developer* tentunya tidak memiliki peta / *blueprint* mall-mall yang ada. Masalah yang diangkat dari penelitian ini adalah mempermudah dan mempersingkat perjalanan yang dilakukan oleh pengunjung mall, *traveler*, dan *ojek online* dalam menemukan tempat yang mereka ingin tuju. Penelitian ini berfokus pada 2 hal yaitu untuk menghasilkan rute terpendek yang paling optimal dan juga dengan posisi *user* yang paling akurat agar bisa menghasilkan jawaban yang paling baik.

2. DASAR TEORI

2.1 *Lifelong Planning A**

*Lifelong Planning A** merupakan sebuah perkembangan dari salah satu Search Algorithm yaitu *A** [3]. *Lifelong Planning A** merupakan *Search Algorithm* yang bisa beradaptasi dengan input yang berbeda, seperti salah satu contohnya di *Google Maps* ketika kita berada di jalan dan ingin *Add Stop* maka *Google Maps* akan

langsung melakukan *re-route* dengan menambahkan lokasi yang harus dituju terlebih dahulu kemudian lokasi selanjutnya. Perbedaan dengan A^* , *Lifelong Planning A** merupakan *Incremental Heuristic Search* yang berarti *path-finding* yang bisa melakukan adaptasi dengan perubahan *input* yang dilakukan oleh *user*, berbeda dengan A^* , dimana A^* cenderung lebih kaku dalam melakukan *path-finding* sehingga jika terjadi perubahan *input* oleh *user* maka A^* akan langsung *recalculate* rute yang akan menjadi *output* untuk *user*.

Persamaan *Priority Queue* pada *Lifelong Planning A**

$$k(n) = \begin{bmatrix} k_1(n) \\ k_2(n) \end{bmatrix} = \begin{bmatrix} \min(g(n), rhs(n)) + h(n, goal) \\ \min(g(n), rhs(n)) \end{bmatrix} \quad (1)$$

Keterangan:

g = kalkulasi *g-value* sebelumnya (jarak awal) seperti di dalam A^*

rhs = *node* yang memiliki relasi dengan *g-value*

h = fungsi *heuristic* yang memperkirakan *cheapest cost* dari n menuju *goal*

k_1 = dianggap seperti f_1 pada A^* yang berarti *initial node*

k_2 = dianggap seperti f_2 pada A^* yang berarti *node changes*

2.2 Gyroscope

Gyroscope pada mulanya merupakan sebuah alat untuk mengukur orientasi dan kecepatan sudut [2]. *Gyroscope* sendiri sudah banyak digunakan di zaman sekarang seperti untuk menjaga arah pada saat melakukan pertambangan, selain itu *gyroscope* juga digunakan untuk kapal selam dalam mempertahankan posisinya. *Gyroscope* yang digunakan di dalam *smartphone Android* adalah *Microelectromechanical Systems Gyroscope* atau lebih dikenal dengan *MEMS Gyroscope*.

MEMS Gyroscope merupakan *sensor orientation* yang ada pada *smartphone Android* dimana prinsip yang digunakan adalah *Foucault Pendulum*. *Foucault Pendulum* merupakan sebuah alat untuk mendemonstrasikan rotasi bumi, *MEMS Gyroscope* biasanya digunakan untuk *image stabilization* pada saat melakukan pengambilan gambar atau mengambil sebuah *video*. *MEMS Gyroscope* menggunakan prinsip *Angular Velocity Sensing* untuk menemukan arah berjalan *user*.

2.3 Tri-Axis Accelerometer

Accelerometer merupakan alat untuk mengukur percepatan. *Accelerometer* sudah banyak digunakan di beberapa industri dan keperluan sains. Seperti *High Sensitive Accelerometers* digunakan sebagai sistem navigasi untuk pesawat terbang dan rudal. Untuk keperluan sains *Accelerometer* digunakan untuk *volcanology* dimana *accelerometer* digunakan untuk melakukan *monitoring magma* pada gunung berapi. Pada bidang barang elektronik *accelerometer* digunakan di dalam *smartphone Android* yang gunanya untuk melakukan *adjustment* atau penyesuaian orientasi pada sebuah *smartphone Android*, selain itu *accelerometer* digunakan juga sebagai *motion input*, terutama *Tri-Axis Accelerometer* karena digunakan untuk *dualshock 3* sehingga bisa membuat permainan balapan lebih realistis karena bisa mendeteksi pergerakan dari *dualshock 3* tersebut [2].

Tri-Axis Accelerometer pada *smartphone android* untuk membantu *pedometers* yang fungsinya untuk mengukur berapa langkah *user* telah berjalan dan berapa jarak yang sudah *user* tempu dari posisi awal *user* berdiri.

2.4 Gerak Lurus Beraturan (GLB)

Gerak Lurus Beraturan atau yang biasa disingkat GLB merupakan materi di dalam mata pelajaran Fisika. GLB adalah gerak lurus yang memiliki kecepatan tetapi tidak memiliki percepatan. Jadi nilai percepatan pada *object* tidak diperhatikan atau dianggap nol ($a = 0$). GLB pada penelitian ini digunakan untuk menjadi patokan jarak terpendek dan waktu tercepat dari posisi *user* ke lokasi.

Persamaan GLB:

$$s = v \cdot t \quad (2)$$

Keterangan:

s = jarak / *space* (m)

v = kecepatan / *velocity* (m/s)

t = waktu / *time* (s)

2.5 Gerak Lurus Berubah Beraturan (GLBB)

Gerak Lurus Berubah Beraturan atau yang biasa disingkat GLBB juga merupakan materi di dalam mata pelajaran Fisika. GLBB adalah gerak lurus yang memperhatikan percepatan, percepatan dalam GLBB dibagi menjadi 2, jika nilai $a(+)$ pada GLBB disebut **percepatan** sedangkan nilai $a(-)$ disebut **perlambatan**. GLBB akan digunakan jika *user* memiliki kecepatan rata-rata yang berbeda dengan patokan yang telah dihitung oleh GLB.

Persamaan GLBB:

$$s = v_0 \cdot t \pm \frac{1}{2} \cdot a \cdot t^2 \quad (3)$$

$$v_t = v_0 + a \cdot t \quad (4)$$

Keterangan dalam rumus:

s = jarak / *space* (m)

v_0 = kecepatan awal (m/s)

v_t = kecepatan akhir (m/s)

t = waktu / *time* (s)

a = percepatan / *acceleration* (m/s^2)

3. DESAIN SISTEM

Selama mendesain sebuah sistem, dilakukan analisa terhadap aplikasi yang sejenis. Aplikasi tersebut dianalisa untuk melihat fitur-fitur yang sudah ada, dan untuk mencari tahu fitur yang bisa ditambah dan digunakan dalam pembuatan aplikasi pencarian rute terpendek di dalam mall.

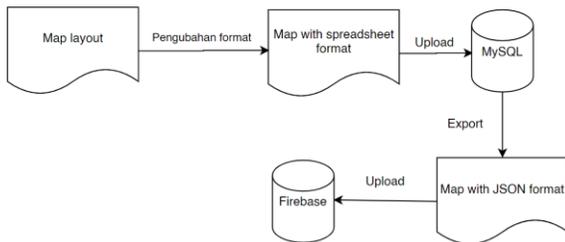
Dalam dunia yang serba canggih sekarang tentunya kita dituntut untuk melakukan segala sesuatu dengan lebih cepat dan lebih efisien. Salah satunya adalah untuk mencari jalan tercepat untuk sampai ke suatu tujuan. Karena di zaman sekarang ini mungkin orang masih kurang efektif dalam menentukan rute yang digunakan. Hal tersebut bisa terjadi karena 2 hal yaitu yang pertama karena orang tidak seberapa mengerti daerah / tempat yang dikunjungi dan yang kedua yaitu orang tersebut tidak terlalu memikirkan jarak dan waktu yang dilalui / ditempuh untuk mencapai tujuannya.

Sebenarnya sudah terdapat aplikasi yang bisa menunjukkan / mengkalkulasi rute terpendek, tetapi aplikasi itu hanya bisa menunjukkan rute untuk perjalanan menggunakan mobil, sedangkan untuk kalkulasi rute terpendek di mall hanya ditampilkan secara 2 dimensi (2D), yang dimaksud adalah aplikasi navigasi tidak bisa mengetahui *user* berada di lantai berapa, hal ini mengakibatkan jika *user* ingin mengunjungi *stand* yang berada di lantai lainnya maka rute yang diberikan hanyalah sebuah garis lurus sehingga *user* tidak mengetahui rute sebenarnya yang seharusnya dilewati, sehingga bisa mengakibatkan salah jalan yang membuat *user* tertunda untuk mencapai *stand* yang ingin dikunjungi.

Aplikasi *MyRoute* akan menampilkan informasi berupa peta di dalam mall secara keseluruhan, selain itu *MyRoute* akan menunjukkan lokasi *user* secara *live* sehingga *user* tidak perlu melakukan *input* untuk lokasi awal di *smartphone* nya. Dengan menggunakan aplikasi *MyRoute* ini *user* bisa mendapatkan kalkulasi rute yang efektif karena aplikasi *MyRoute* ini akan mengkalkulasi lokasi *user* dengan *stand* tujuan *user* sehingga bisa mendapatkan rute yang paling efektif.

Sebenarnya banyak sekali jalan yang bisa dilalui oleh orang-orang, karena mungkin terlalu banyak jalan yang bisa dilalui orang semakin bingung untuk menentukan jalan yang paling efektif dan cepat. Seringkali juga orang salah jalan yang mengakibatkan harus memutar tempat tersebut untuk mendapatkan jalan yang seharusnya dilalui. Untuk menuju lokasi/*stand* yang memiliki perbedaan lantai maka akan dilakukan kalkulasi terlebih dahulu menggunakan prinsip *Nearest Neighbor* agar *user* bisa mendapatkan eskalator terdekat yang harus digunakan untuk mencapai lantai yang ingin dituju.

Untuk mengetahui bagaimana sebuah sistem bekerja, dibutuhkan sebuah desain struktur dari sistem tersebut. Hal ini bertujuan agar *user* bisa memahami proses yang terjadi.



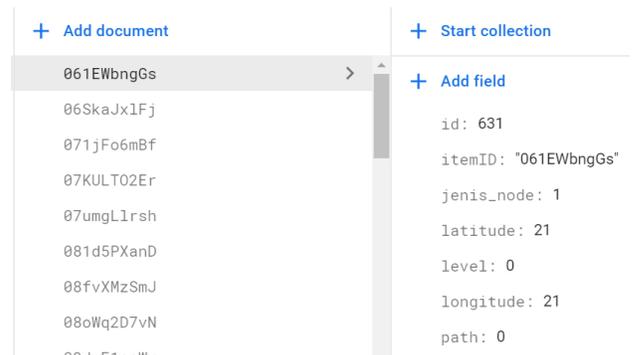
Gambar 1. Diagram perubahan data dari Map Layout menjadi data di dalam Firebase

Berikut penjelasan sesuai dengan Gambar 1. *Layout / blueprint* peta mall dilakukan pengolahan data terlebih dahulu yaitu, pertama melakukan *mapping* mengenai peta mall yang dimiliki. Kedua setelah melakukan *mapping* yang dilakukan adalah menentukan perbandingan skala 1 petak di dalam aplikasi dengan dunia yang asli nya. Ketiga akan dilakukan *formatting* untuk dimasukkan ke dalam *spreadsheet* dengan keterangan – keterangan yang sudah menjadi kesepakatan yang bisa dicermati di tabel 1 agar tidak membingungkan *user*. Keempat melakukan *input* ke dalam *database MySQL* dengan *field* seperti di Gambar 3 dengan menggunakan bahasa pemrograman *Python* dan *library pandas* untuk memudahkan melakukan *input* di *database*. Kelima yaitu melakukan pengecek-an mengenai data yang telah di *input*

dengan menggunakan *Python* juga agar tidak terjadi kesalahan pada saat nanti memproses data yang ada. Langkah selanjutnya yaitu langkah ke-enam dimana hasil *input* yang dimasukkan ke dalam *database MySQL* akan di-*export* dengan *format* JSON supaya bisa dimasukkan ke dalam *Firebase* dengan *Collection*, *Document*, dan *field* sesuai dengan Gambar 2.

Tabel 1. Mapping keterangan peta

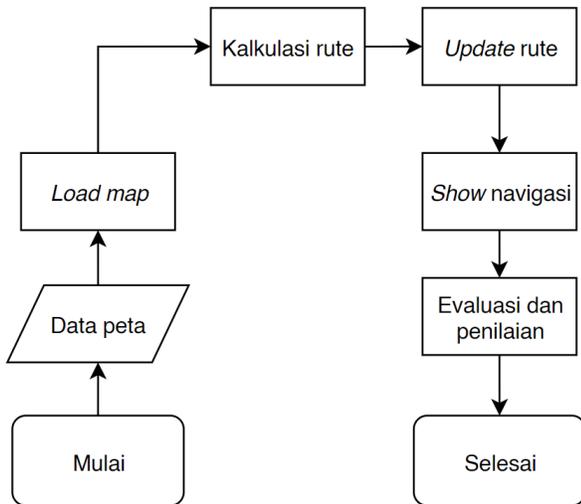
Jenis Node	Keterangan	Path	Warna (Hexadecimal)
0	Obstacle	0	#000000
1	Path	0	#FFFFFF
1	Path (route)	1	#00A2E8
1	Path (user)	2	#00A2E8 + tanda O
1	Path (end)	3	#00A2E8 + tanda bintang
2	Toko	0	#C8BFE7
3	Toko (dinding)	0	#C3C3C3
4	Eskalator Naik	0	#FFF200
5	Ujung Eskalator	0	#EAF040
6	Eskalator Turun	0	#32C07A
7	Toilet	0	#8F6A9E



Gambar 2. Tabel/collection skripsi_node yang berada di Firebase Firestore yang akan diakses oleh Mobile Apps

Options	id	jenis_node	longitude	latitude	level	path	itemID
<input type="checkbox"/> Edit Copy Delete	1	0	0	0	0	0	KV9IH1dYvI
<input type="checkbox"/> Edit Copy Delete	2	0	0	1	0	0	u9IRDQpEPn
<input type="checkbox"/> Edit Copy Delete	3	0	0	2	0	0	QAYwg2lhmk
<input type="checkbox"/> Edit Copy Delete	4	0	0	3	0	0	wZKDBoUGn1
<input type="checkbox"/> Edit Copy Delete	5	0	0	4	0	0	Vr5xNYBauQ
<input type="checkbox"/> Edit Copy Delete	6	0	0	5	0	0	yliGQxPWFN
<input type="checkbox"/> Edit Copy Delete	7	0	0	6	0	0	xBT1Cre5EK
<input type="checkbox"/> Edit Copy Delete	8	0	0	7	0	0	i5pvP8Ay2f
<input type="checkbox"/> Edit Copy Delete	9	0	0	8	0	0	gW7dH9pyCF
<input type="checkbox"/> Edit Copy Delete	10	0	0	9	0	0	iTRDDOhSIV
<input type="checkbox"/> Edit Copy Delete	11	0	0	10	0	0	sfuKRoSF4e
<input type="checkbox"/> Edit Copy Delete	12	0	0	11	0	0	WFGiUnRt4
<input type="checkbox"/> Edit Copy Delete	13	0	0	12	0	0	h809xatjJo

Gambar 3. Tabel skripsi_node yang berada di MySQL



Gambar 4. Arsitektur Sistem

Berikut penjelasan yang lebih detail mengenai bagaimana cara aplikasi bekerja sesuai dengan gambar 4. Pertama kali ketika aplikasi dijalankan, aplikasi akan mengambil data *layout / blueprint* yang ada di *database* dan menampilkannya kepada *user*. Data dari *database* merupakan data yang sudah didapatkan melalui proses *re-create* peta. Apabila *user* ingin mengunjungi suatu *stand/store* tertentu, *user* bisa langsung memilih *stand/store* apa yang sudah tersedia oleh aplikasi jika pemilihan sudah selesai, maka data hasilnya akan ditampilkan ke dalam aplikasi. Setelah itu, data yang telah dipilih dari *database* akan diproses kembali untuk melakukan kalkulasi pencarian rute terpendek. Setelah selesai melakukan kalkulasi, maka data akan ditampilkan dalam bentuk navigasi kepada *user* kemudian *user* bisa langsung melakukan perjalanan sesuai dengan rute yang telah diberikan oleh aplikasi.

4. HASIL PENGUJIAN

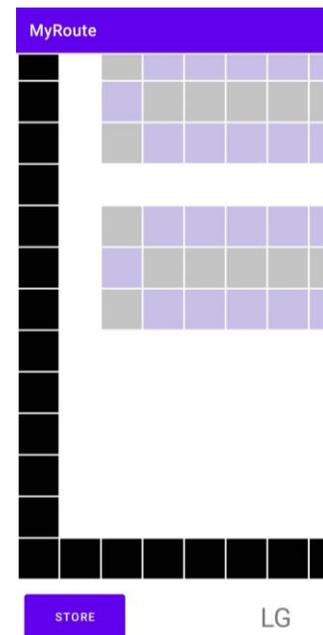
Pengujian ini dilakukan dengan tujuan mendapatkan hasil yang diinginkan. Pengujian dilakukan sebanyak 2 kali yang pertama dengan menggunakan Huawei P30 Pro dengan spesifikasi *chipset* Kirin 980, Octa-core (2x2.6 GHz Cortex-A76 & 2x1.92 GHz Cortex-A76 & 4x1.8 GHz Cortex-A55), dengan menggunakan Android versi 10 dan dengan sensor-sensor yang cukup baru yaitu *gyro*, *accelerometer*, *compass*, dan *barometer*.

Smartphone kedua android Samsung Galaxy S21 Ultra 5G dengan spesifikasi CPU Exynos 2100, Octa-core (1x2.9 GHz Cortex-X1 & 3x2.80 GHz Cortex-A78 & 4x2.2 GHz Cortex-A55). Pengujian dilakukan dengan menggunakan Android versi 11 dengan One UI version 3.1. Dengan sensor *gyro*, *accelerometer*, *compass*, dan *barometer* yang paling baru.

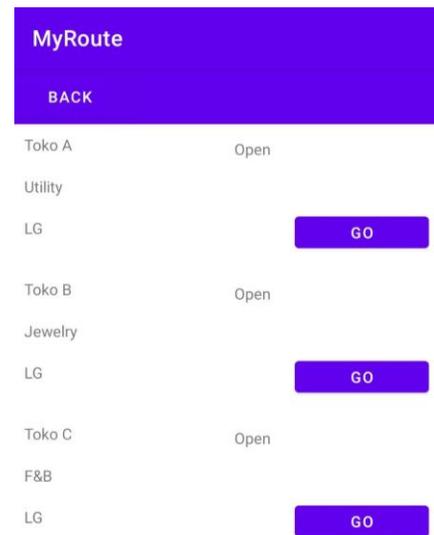
Pengujian dilakukan 2 kali dengan 2 smartphone yang berbeda bertujuan untuk menguji apakah aplikasi bisa tetap digunakan dengan smartphone yang teknologinya berbeda 2 tahun.

Bisa dilihat di gambar 5 dimana aplikasi sudah menentukan *user* berada di *point* mana, dan di gambar 6 ada *list* lokasi/*stand* apa saja yang bisa dikunjungi oleh *user*. Ketika *user* sudah memilih

store yang ingin dikunjungi maka akan langsung ditampilkan oleh aplikasi seperti di gambar 7.



Gambar 5. Tampilan home aplikasi

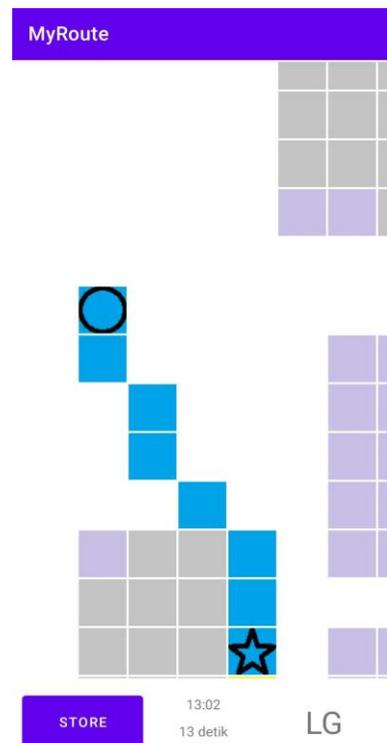


Gambar 6. Tampilan List Store Aplikasi

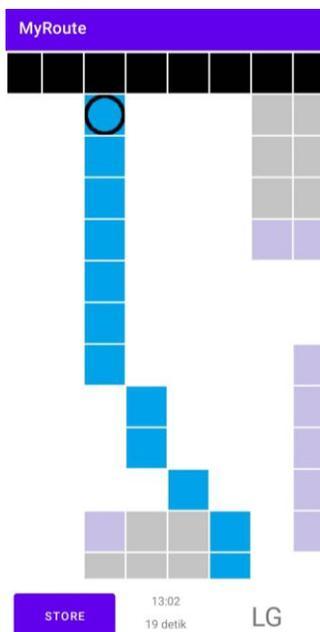
Tabel 2. Perhitungan waktu tempuh menggunakan Lifelong Planning A*

Point A	Point B	Waktu Tempuh
Toko A	Toko H	30 detik
Toko H	Toko L	54 detik
Toko J	Toko D	65 detik

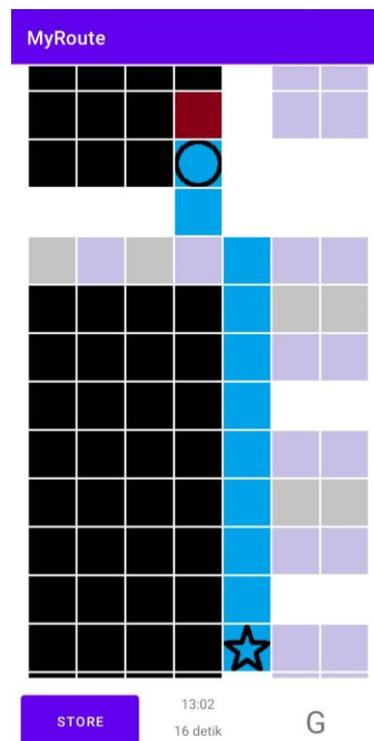
Dari Tabel 2 bisa dilihat waktu tempuh yang dihasilkan dengan menggunakan algoritma *Lifelong Planning A** dengan rumus Fisika GLB dan GLBB. Dimana ada konstanta yang sudah ditetapkan yaitu tiap 1 petak yang dilalui memiliki skala 1:1,8 m yang berarti 1 petak yang berada di aplikasi memiliki ukuran 1,8me x 1,8m. Kecepatan yang digunakan adalah kecepatan rata-rata manusia berjalan yaitu sekitar 1,4 m/s. Dan juga untuk rumus GLBB diikuti oleh bagaimana *behaviour user* dalam berjalan, tentu tidak semua *user* berjalan dengan kecepatan 1,4 m/s, karena 1,4 m/s merupakan kecepatan rata-rata orang berjalan di luar negeri, sedangkan untuk orang-orang di Indonesia kalau lagi santai berjalannya hanya sekitar 0,6 m/s – 0,8 m/s, selain *behaviour user* dalam berjalan tantangan yang dialami dalam membuat aplikasi ini pun juga ada faktor *external* yaitu keramaian lokasi dimana *user* berada, tentu aplikasi ini masih belum bisa untuk mendeteksi seberapa banyak kerumunan masyarakat yang berada di lokasi tersebut. Bisa dilihat di gambar 8 mengenai bagaimana *user* telah berjalan maka aplikasi juga akan mengikuti langkah yang diambil oleh *user*. Kemudian di gambar 9 aplikasi bisa mengidentifikasi bahwa *user* telah menaiki sebuah eskalator, sehingga aplikasi akan langsung berganti peta untuk ditampilkan karena sudah berada di lantai yang berbeda. Ketika *user* sudah berada di lantai yang berbeda, maka aplikasi akan melakukan *refresh* data peta karena telah menaiki/menuruni sebuah eskalator, sehingga *user* bisa melihat sekarang berada di posisi ujung eskalator naik/turun dan bisa melanjutkan perjalanan ke lokasi tujuan. Juga aplikasi akan menampilkan waktu tempuh yang baru kepada *user* dimana waktu tempuh *user* berbeda-beda di setiap lantai *user* itu berada. Dan yang terakhir seperti ditampilkan di gambar 10 saat *user* sudah mencapai lokasi/*stand* yang ingin dikunjungi maka waktu yang diperkirakan *user* akan sampai dan waktu tempuh *user* yang berada di aplikasi akan diselesaikan.



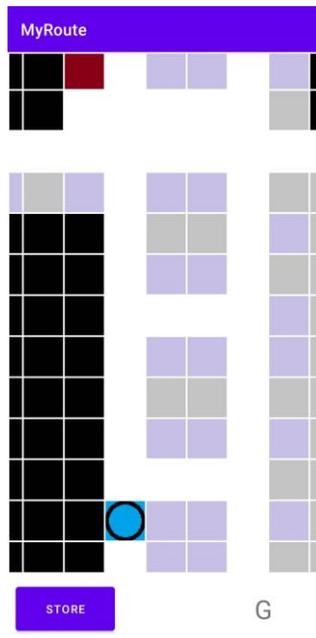
Gambar 8. Ketika *user* melakukan perjalanan dengan rute yang sudah disediakan



Gambar 7. Navigasi / rute yang akan ditampilkan kepada *user* untuk lantai *Lower Ground*



Gambar 9. Ketika *user* sudah menaiki eskalator yang menuju ke lantai *Ground* akan melakukan *update* pada label di aplikasi



Gambar 10. Ketika *user* sudah mencapai lokasi maka waktu perkiraan sampai dengan waktu estimasi perjalanan sudah tidak dimunculkan lagi

Setelah tabel 2 yang menunjukkan bagaimana waktu tempuh yang dikalkulasi oleh aplikasi dengan *actual* waktu tempuh yang dialami oleh *user*, selanjutnya akan melihat bagaimana dengan kenaikan persentase untuk mendeteksi apakah *user* menaiki suatu eskalator.

Tabel 3. Hasil pengujian

<i>Point A</i>	<i>Point B</i>	<i>Accuracy</i>	<i>Beda level</i>
Toko A	Toko H	100%	1
Toko H	Toko L	100%	0
Toko J	Toko D	100%	1

Berdasarkan Tabel 3 *accuracy* berarti apakah aplikasi mencapai *target* yang diprediksi oleh *user* baikpun dalam *actual* dan ekspektasi seharusnya. Kemudian dengan hasil yang bisa dilihat bahwa dengan menggunakan *Gyroscope* dan *Tri-Axis Accelerometer* bisa mendapatkan *accuracy* untuk *actual* yang telah dikalkulasi oleh aplikasi. Hasilnya memiliki nilai *tolerance* sebesar 15%, dimana nilai *tolerance* itu diambil dari *behaviour user* dalam berjalan dan seberapa ramai *mall* yang dikunjungi.

Dibandingkan dengan *Map Matching* dan *Magnetic Field Sensor* yang memiliki 97,5% [1] untuk nilai akurasi apakah *user* berpindah lantai, tapi dengan menggunakan *Gyroscope*, *Tri-Axis Accelerometer* dan peta yang telah ada, akurasi bisa naik 2,5% yang sehingga mencapai hasil 100%.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian sistem, dapat disimpulkan bahwa seluruh rancangan sistem yang telah dirancang berhasil diterapkan. Namun, masih ada beberapa *feature* yang belum bisa digunakan secara maksimal dikarenakan oleh bentuk *mall* yang bermacam-macam, juga selain itu ada faktor lain yaitu perbedaan jenis *smartphone* Android yang memiliki banyak sekali perbedaan karena berbeda *brand* bisa menggunakan *hardware* yang berbeda sehingga akurasi sensor yang dimiliki oleh tiap *smartphone* Android bisa berbeda.

Data yang dihasilkan pun sebenarnya sudah dilakukan *mapping* peta yang memudahkan pembuatan aplikasi dan mempermudah *user* untuk melihat peta yang sudah ditampilkan di aplikasi.

5.2 Saran

Berdasarkan hasil pengujian sistem, dapat diberikan saran yaitu:

- Menambahkan peta yang lebih besar dan bervariasi dari beberapa *mall*.
- Memperbaiki tampilan data agar lebih dinamis saat menerima peta berbeda bentuk.
- Memperbaiki penyajian data yang masih kurang sempurna pada aplikasi.

6. DAFTAR PUSTAKA

- [1] Ashraf, I., Hur, S., Shafiq, M., & Park, Y. 2019. Floor identification using magnetic field data with smartphone sensors. *Sensors (Switzerland)*, 19(11). <https://doi.org/10.3390/s19112538>
- [2] Hsu, H. H., Peng, W. J., Shih, T. K., Pai, T. W., & Man, K. L. 2014. Smartphone indoor localization with accelerometer and gyroscope. *Proceedings - 2014 International Conference on Network-Based Information Systems, NBI S 2014*, 465–469. <https://doi.org/10.1109/NBI S.2014.72>
- [3] Li, J., Tinka, A., Kiesel, S., Durham, J. W., Kumar, T. K. S., & Koenig, S. 2020. *Lifelong Multi-Agent Path Finding in Large-Scale Warehouses*. <http://arxiv.org/abs/2005.07371>
- [4] Heo, J. Y., & Kim, K. J. 2017. Development of a scale to measure the quality of mobile location-based services. *Service Business*, 11(1), 141–159. <https://doi.org/10.1007/s11628-016-0305-6>
- [5] Renaudin, V., Ortiz, M., Perul, J., Torres-Sospedra, J., Jimenez, A. R., Perez-Navarro, A., Martin Mendoza-Silva, G., Seco, F., Landau, Y., Marbel, R., Ben-Moshe, B., Zheng, X., Ye, F., Kuang, J., Li, Y., Niu, X., Landa, V., Hacohen, S., Shvalb, N., ... Park, Y. 2019. Evaluating Indoor Positioning Systems in a Shopping Mall: The Lessons Learned from the IPIN 2018 Competition. *IEEE Access*, 7, 148594–148628. <https://doi.org/10.1109/ACCESS.2019.2944389>