

Implementasi Algoritma K-Nearest Neighbor dan Algoritma A-Star pada Sistem Navigasi Ruangan Berbasis *Wi-Fi Fingerprint Positioning*

Lanang Bagus Prakarsa*, Badie'ah.*, Sri Mulyono *

* Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Sultan Agung Semarang

Correspondence Author: lanangbagus@std.unissula.ac.id

Abstract

Area publik merupakan suatu area dimana lokasi tersebut menjadi pusat masyarakat umum untuk beraktifitas. Sebuah area publik biasanya mempunyai area yang luas dengan berbagai sub area yang terdapat di dalamnya. Contoh dari area public adalah mall, bandara, stasiun, museum, alun-alun, dan lain-lain. Pengunjung baru suatu public area biasanya kebingungan saat ingin menelusuri tempat-tempat di dalamnya. Akan sangat memudahkan jika suatu sistem navigasi dapat diterapkan pada fasilitas umum seperti ini. Area publik dapat berbentuk indoor maupun outdoor. Sistem navigasi berbasis satelit seperti GPS mempunyai keterbatasan jika diterapkan pada area indoor disebabkan oleh suatu kondisi yang disebut *line of sight* yaitu suatu kondisi dimana tidak memungkinkan terjadinya transmisi sinyal yang tidak terhalang antara perangkat dan satelit. *Wi-fi fingerprint positioning* merupakan salah satu metode alternatif dalam sistem penentuan posisi dengan memanfaatkan *wi-fi access point* sebagai transmitter. Jangkauan sinyal *wi-fi* yang cukup luas dan kebutuhan perangkat yang minimal dapat menjadi solusi optimal untuk membangun suatu sistem navigasi dalam ruangan. Prototipe sistem navigasi dalam ruangan dikembangkan di area lantai 3 gedung Fakultas Teknologi Industri Unissula sebagai studi kasus penelitian. Prototipe diimplementasikan dengan metode *fingerprint positioning* dan algoritma *k-NN* untuk dapat melakukan pemosisian, serta algoritma *A-star* untuk melakukan pencarian rute navigasi. Hasil pengujian menunjukkan bahwa prototipe dapat menentukan posisi pengguna serta memberikan rute navigasi dengan baik.

Keyword: *wi-fi fingerprint positioning*, sistem navigasi dalam ruangan, algoritma *k-NN*, algoritma *A-star*, android

1. PENDAHULUAN

Saat ini sistem navigasi berbasis GPS (*Global Positioning System*) telah mencakup hampir seluruh tempat di dunia dengan tingkat perincian lokasi yang sudah sangat detail dan akurasi yang tinggi. Pasinggi dkk., pada tahun 2018 dalam *literatur review*-nya mengungkapkan bahwa sistem navigasi berbasis satelit semacam GPS mampu memberikan tingkat akurasi yang tinggi untuk penggunaan di luar ruangan, namun akan menemukan tantangan yang signifikan untuk penggunaan di dalam ruangan. GPS tidak dirancang untuk penggunaan di area dalam ruangan karena kondisi ini tidak memungkinkan terjadinya transmisi sinyal yang tidak terhalang antara perangkat dengan satelit. Kondisi ini disebut dengan *line of sight*. [1]

Area publik merupakan suatu tempat yang menjadi pusat masyarakat umum dalam melakukan berbagai aktifitas. Area publik biasanya memiliki jangkauan area yang sangat luas dan memiliki sektor-sektor beragam di dalamnya baik *indoor* maupun *outdoor*. Seperti yang kita tahu bahwa area publik seperti *mall*, rumah sakit, bandara, stasiun kereta api, dan lain-lain merupakan area publik yang berbentuk *indoor*. Biasanya pengunjung area publik yang baru pertama kali datang ke tempat tersebut akan kebingungan mencari lokasi yang ingin dituju. Padahal sebenarnya di setiap area publik telah disediakan denah khusus untuk pengunjung, namun tidak jarang banyak pengunjung enggan ataupun kurang teliti dalam memerhatikannya.

Jika sistem navigasi dalam ruangan yang berbasis aplikasi *smartphone* dapat diterapkan pada area-area publik seperti ini, tentu akan banyak membantu orang-orang yang beraktifitas di dalamnya. Kepraktisan

dalam mengakses sistem yang terintegrasi pada aplikasi *smartphone* yang dimiliki oleh setiap pengunjung diharapkan mampu mengatasi masalah mereka ketika menelusuri tempat-tempat di dalam area publik.

Ada banyak teknik yang dapat diimplementasikan tentang sistem penentuan posisi. Namun jika menyangkut soal sistem penentuan posisi di dalam ruangan maka teknik *fingerprint positioning* merupakan salah satu yang sangat populer dikarenakan faktor kemudahannya serta kebutuhan perangkat yang sangat minimal. *Fingerprint positioning* merupakan salah satu teknik dalam penentuan lokasi. Mirip seperti teknologi *fingerprint* (pencocokan sidik jari), teknik ini mencocokkan fitur sinyal pada sebuah lokasi dengan data hasil pengukuran agar dapat mengetahui dimana posisi perangkat saat ini berada.

Terdapat dua fase dalam penggunaan teknik ini yaitu fase *online* dan fase *offline*. Pada fase *offline* dilakukan survei pada lokasi. Data yang diambil berupa koordinat/label disertai data hasil pengukuran kekuatan sinyal dari beberapa *transmitter* atau biasa disebut RSS (*Received Signal Strength*). Dan pada fase *online*, yaitu pada saat penentuan lokasi, RSS yang diterima dicocokkan dengan kumpulan data yang diperoleh pada fase *offline* [1].

Potensi pasar teknologi *indoor positioning* telah meningkat pesat dengan luasnya distribusi *wi-fi*. Presisi yang tinggi, konsumsi daya rendah, dan biaya rendah menjadikannya area yang sangat menarik dalam penelitian teknologi *indoor positioning*. Menurut standar IEEE 802.11, kartu jaringan nirkabel *wi-fi* dan *access point* memiliki fungsi mengukur intensitas sinyal frekuensi radio. Oleh karena itu, pengguna dapat memanfaatkan perangkat seluler seperti *smartphone*, laptop, PC tablet, dan lainnya untuk menjalankan *indoor positioning* dengan *wi-fi* dan algoritma tertentu.[2], [3]

Sampai saat ini, sebagian besar metode penentuan posisi dalam ruangan didasarkan pada teknologi *fingerprint matching* dikombinasikan dengan metode lain. Terdapat sebuah penelitian yang menjelaskan jika dibandingkan dengan sistem pemosisian lain, teknologi *wi-fi Fingerprint positioning* memiliki keuntungan dari biaya rendah dan presisi tinggi. Karena penyebaran yang luas dan penggunaan *wi-fi* di seluruh dunia, teknologi pemosisian sidik jari (*fingerprint positioning*) dapat diterapkan pada setiap skenario dalam ruangan di mana jaringan *wi-fi* berada dikerahkan tanpa perangkat keras tambahan, yang membuat biaya teknologi rendah. Teknologi ini menggunakan kekuatan sinyal *wi-fi* untuk memodelkan dan mengukur, tanpa harus mengidentifikasi lokasi yang tepat dari AP. Dalam lingkungan *indoor* yang kompleks, dalam kondisi biaya rendah, atribut ruang-waktu seperti *angle and time of arrival* dapat menghasilkan kesalahan besar; Namun, intensitas sinyal relatif stabil, sehingga membuat akurasi posisi metode ini lebih besar dari yang lain.[3]

Pada penelitian lain, diusulkan metode yang dikombinasikan dengan algoritma *k-NN* (*k-Nearest Neighbor*) dan teknologi *fingerprint positioning*; metode *triangular positioning* didasarkan pada model *classic path loss* dimana mereka menemukan bahwa akurasi metode *positioning* mereka meningkat 0,5 meter dibandingkan dengan sistem *Eka-hau real-time positioning*. [4]

Pada kasus yang sama, terdapat penelitian yang mengimplementasikan teknologi *Indoor Localization* dengan metode *Clustering Filtered k-Nearest Neighbors* dalam membangun sistem navigasi *indoor* yang beroperasi pada *smartphone* android untuk studi kasus pada gedung bertingkat. Pada hasil pengujian menunjukkan bahwa sistem ini memiliki akurasi pendeteksian posisi yang tergolong tinggi sebesar 88,953%. [5].

Penelitian yang ditulis oleh [6] mengusulkan implementasi metode *Iterative Deepening A** (IDA*) untuk melakukan pencarian rute dalam membangun aplikasi *Indoor Positioning System*. Hasil pengujian menunjukkan bahwa algoritma pencarian ini dapat menghitung jarak terpendek dari semua titik lokasi yang ada ke semua titik yang disediakan.[6]

Berdasarkan kajian tersebut, pada penelitian yang ditulis pada artikel ini bertujuan untuk menerapkan *fingerprint positioning* menggunakan algoritma K-NN. Metode kNN bekerja dengan cara mencari sejumlah *k* pola (diantara semua pola latih yang ada di semua kelas) yang terdekat dengan pola masukan, kemudian menentukan kelas keputusan berdasarkan jumlah pola terbanyak diantara *k* pola tersebut (*voting*).[7]

Selain itu pada penelitian ini juga mengimplementasikan teknik *pathfinding*. *Pathfinding* umumnya bertujuan untuk mencari rute terpendek di antara dua titik/posisi. [8] Algoritma yang dipilih pada penelitian ini untuk menerapkan teknik *pathfinding* adalah algoritma A*. A* adalah salah satu teknik terbaik dan populer yang digunakan dalam *path-finding* dan *graph travelsals*. Secara informal, algoritma pencarian A*, tidak seperti teknik traversal lainnya, algoritma ini memiliki "otak". Artinya adalah benar-benar algoritma pintar yang memisahkannya dari algoritma konvensional lainnya. Dengan fungsi *heuristic*-nya, kinerja algoritma A* dalam pengecekan *node* terbukti lebih efektif jika dibanding algoritma pencarian lainnya seperti *Dijkstra* dan *Floyd Warshall*. [9]. Adapun eksperimen penelitian ini dilakukan pada Gedung Fakultas Teknologi Industri pada Universitas Islam Sultan Agung Semarang

2. METODE PENELITIAN

3.1. Gambaran sistem

Sistem yang dirancang serta dibangun peneliti merupakan sebuah prototipe sistem penentuan posisi dan navigasi yang diharapkan mampu beroperasi di dalam suatu ruangan dengan baik. Sistem yang dibangun berlokasi pada lantai 3 gedung Fakultas Teknologi Industri Unissula sebagai studi kasus objek penelitian. Sistem memiliki dua fungsi utama yaitu :

- 1) Penentuan posisi
- 2) Navigasi

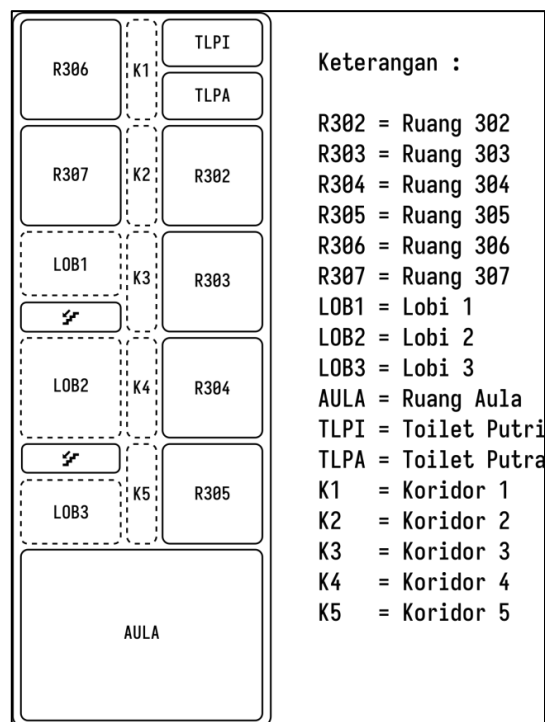
3.2. Metode pengembangan sistem

Prototyping dimulai dengan pengumpulan kebutuhan, melibatkan pengembang dan pengguna sistem untuk menentukan tujuan, fungsi, dan kebutuhan operasional sistem. Langkah-langkah dalam *prototyping* adalah sebagai berikut :

- 1) Pengumpulan kebutuhan
- 2) Proses desain yang cepat
- 3) Membangun prototipe
- 4) Evaluasi dan perbaikan

3.3. Survei lokasi penelitian

Survei lokasi pada objek penelitian yaitu area lantai 3 gedung Fakultas Teknologi Industri Unissula dilakukan untuk mengetahui bagaimana kondisi, medan, serta hal-hal yang dibutuhkan dalam menunjang perancangan dan pengembangan sistem. Setelah dilakukan survei, selanjutnya dilakukan pemetaan serta pembuatan denah yang mendefinisikan sektor-sektor yang berada di dalamnya. Hasilnya adalah seperti Gambar 1 di bawah ini.



Gambar 1. Pemetaan area lantai 3 gedung FTI Unissula

3.3. Komponen pemosisian

Terdapat dua komponen utama dalam sebuah sistem pemosisian yaitu komponen pemancar sinyal (*transmitter*) dan komponen penghitung.

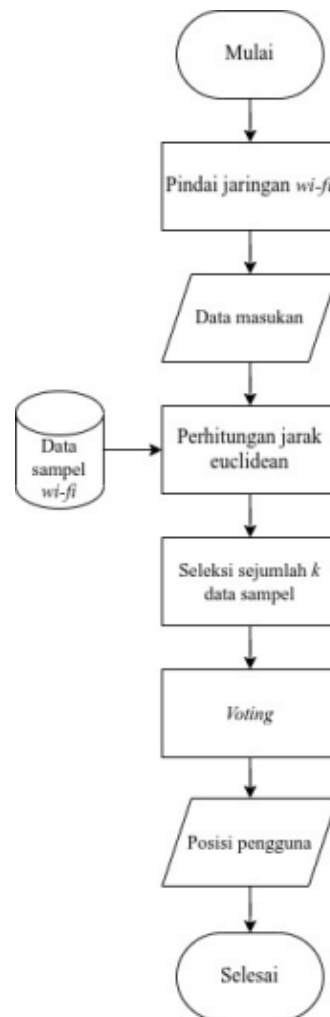
1) *Tansmitter*

Pada kasus pemosisian dalam ruangan, komponen *transmitter* dapat memanfaatkan teknologi telekomunikasi *wireless* seperti *infrared*, *bluetooth*, atau *wi-fi*. Namun untuk dapat menjangkau skala area yang luas, teknologi *wi-fi* merupakan pilihan yang lebih efektif. Perangkat *transmitter* dalam jaringan *wi-fi* biasa disebut dengan *access point*.

2) Komponen penghitung

Komponen penghitung yang digunakan adalah *smartphone* android. *Smartphone* android pada umumnya mampu menjalin sebuah interkoneksi melalui jaringan *wi-fi*, sehingga telah memenuhi syarat untuk dijadikan sebagai komponen penghitung. Sistem operasi android mempunyai kelas yang bernama “*WifiManager*”. Kelas ini menyediakan API (*Application Programming Interface*) untuk mengelola semua aspek konektivitas dengan *wi-fi*. [10]

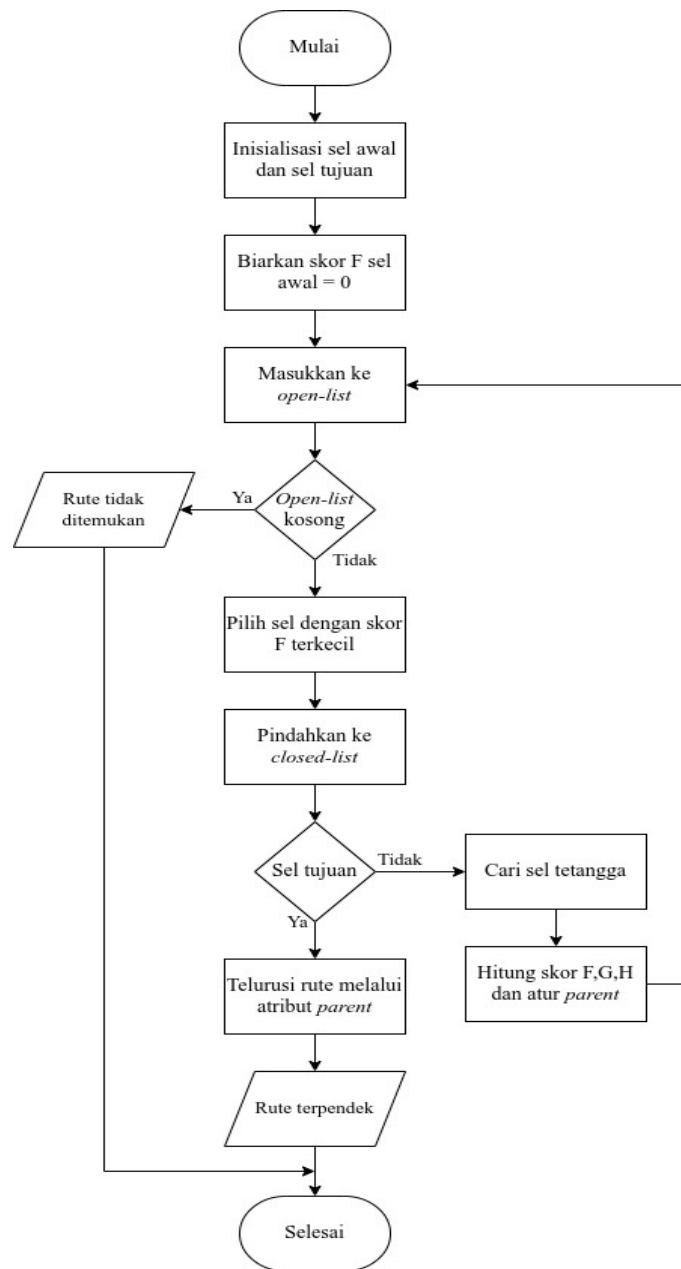
3.4. Pemposisian



Gambar 1 Flowchart pemposisian

Proses pemposisian dijelaskan lebih rinci pada *flowchart* yang ditampilkan pada Gambar 2. Diawali dengan dilakukannya pemindai jaringan *wi-fi* terlebih dahulu oleh aplikasi sistem melalui perangkat android. Pemindaian yang dilakukan akan menghasilkan informasi kekuatan sinyal dimana pengguna berada dan akan digunakan sebagai data masukan. Setelah data masukan didapatkan, maka selanjutnya adalah melakukan perhitungan jarak euclidean data sampel pada *database* terhadap data masukan. Setelah semua data sampel selesai dilakukan perhitungan, maka selanjutnya akan diseleksi sejumlah data sampel yang memiliki jarak euclidean terdekat sebanyak nilai dari variabel k . Hasil dari penyeleksian tersebut akan dilakukan *voting* berdasarkan label lokasi yang paling sering muncul. Hasil dari *voting* merupakan posisi dimana pengguna berada. Langkah terakhir adalah menampilkan posisi tersebut kepada pengguna.

2.8. Navigasi



Gambar 3 Flowchart diagram proses navigasi

Proses pencarian rute dijelaskan lebih rinci pada Gambar 3. Langkah pertama adalah melakukan inisialisasi posisi pengguna dan lokasi tujuan menjadi sel awal dan sel tujuan pada peta *grid layout* sel. Selanjutnya sel awal akan diberi skor 0 pada atribut F, dan dimasukkan ke dalam *open-list*.

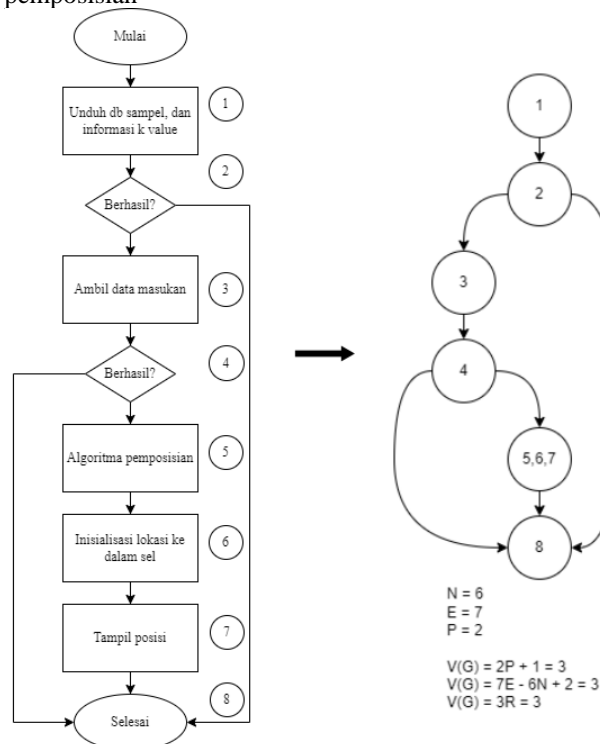
Sistem akan melakukan penelusuran rute selama *open-list* tidak kosong. Langkah awal penelusuran sel adalah memilih sel dengan skor F paling kecil yang ada pada *open-list*. Sel yang terpilih kemudian akan dikeluarkan dari *open-list* dan dipindahkan ke dalam *closed-list*. Selanjutnya akan dicari sel tetangga dari sel yang terpilih tersebut. Sel tetangga yang dapat dilewati kemudian akan dilakukan perhitungan skor F, G, H, dan mengatur atribut *parent*-nya dengan sel yang terpilih sebelumnya. Kemudian kumpulan sel tetangga tersebut dimasukkan ke *open-list*.

Penelusuran sel tersebut akan terus berlangsung dan akan berhenti hingga sel tujuan terpilih dan masuk ke dalam *closed-list*. Setelah sel tujuan ditemukan, maka akan dilakukan penelusuran rute melalui atribut *parent*. Penelusuran rute dilakukan dari sel tujuan kembali ke sel awal. Terakhir, hasil dari penelusuran rute tersebut akan ditampilkan kepada pengguna.

3. HASIL DAN ANALISA

4.1. Pengujian Sistem dengan Basis path Testing (White Box)

1. Pengujian pada fungsi pemosisian



Gambar 4 Perhitungan cyclomatic complexity pada flowchart pemosisian

Pada Gambar 4, digambarkan perhitungan *cyclomatic complexity*. Pada Gambar 4 dapat diketahui bahwa terdapat 3 buah jalur independen pada fungsi pemosisian. Rincian jalurnya adalah sebagai berikut :

- Jalur 1 = 1-2-8
- Jalur 2 = 1-2-3-4-8
- Jalur 3 = 1-2-3-4-5-6-7-8

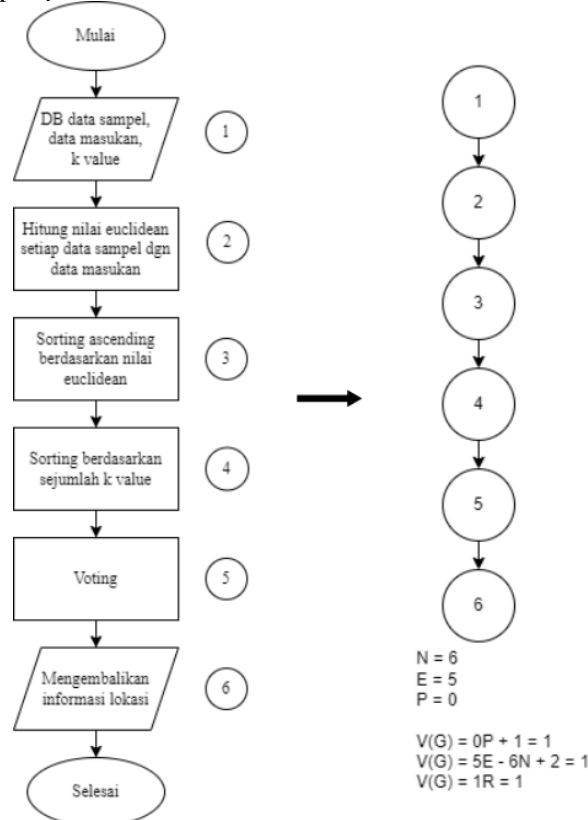
Tabel 1 merupakan *test case* yang diberikan ke program agar setiap jalur independen di atas dapat diketahui *output*-nya.

Tabel 1 Test case basis path testing pada fungsi pemosisian

Jalur 1	Test case	1. Menghapus permission internet pada program 2. Menonaktifkan koneksi internet pada device android
	Expected result	1. Muncul notifikasi tidak dapat mengunduh data sampel 2. Tidak dapat melakukan pemosisian
	Output	1. Muncul notifikasi tidak dapat mengunduh data sampel 2. Tidak dapat melakukan pemosisian
	Validation	<i>Valid</i>
Jalur 2	Test case	1. Mengubah kode program agar tidak melakukan scanning jaringan <i>wi-fi</i> 2. Menghapus permission <i>wi-fi</i> pada program 3. Menonaktifkan layanan <i>wi-fi</i> pada device android
	Expected result	1. Muncul notifikasi data RSS tidak lengkap 2. Tidak dapat melakukan pemosisian
	Output	1. Muncul notifikasi data RSS tidak lengkap 2. Tidak dapat melakukan pemosisian
	Validation	<i>Valid</i>

Jalur 3	Test case	1. Permission dan layanan <i>wi-fi</i> pada device android aktif 2. <i>Database</i> data sampel dapat diunduh 3. Data masukan RSS <i>Valid</i>
	Expected result	1. Berhasil mengambil data masukan 2. Pemposisian berhasil dan muncul lokasi pengguna
	Output	1. Berhasil mengambil data masukan 2. Pemposisian berhasil dan muncul lokasi pengguna
	Validation	<i>Valid</i>

2. Pengujian pada fungsi pemposisian



Gambar 5 Perhitungan cyclomatic complexity pada algoritma pemposisian k-NN

Pada perhitungan *cyclomatic complexity* yang ditampilkan pada Gambar 5, dapat diketahui hanya terdapat 1 buah jalur independen pada algoritma pemposisian *k-NN*. Rincian jalurnya adalah sebagai berikut.

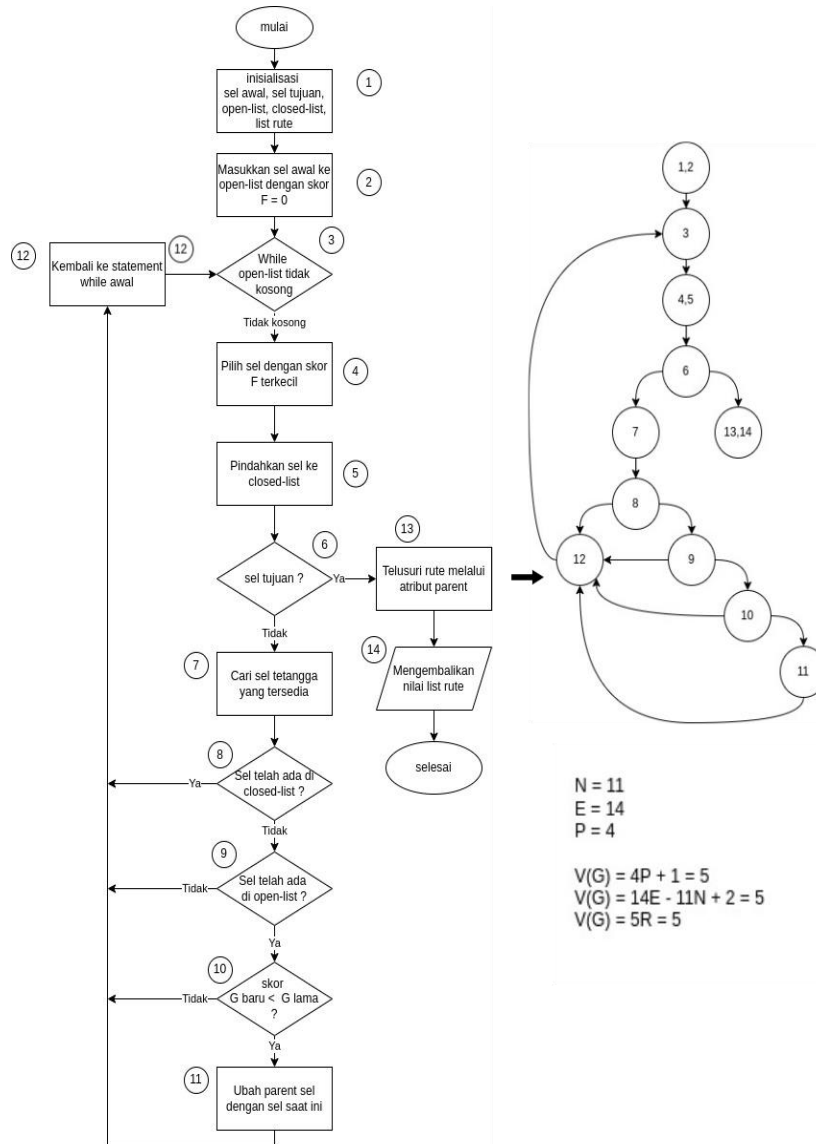
- Jalur 1 = 1-2-3-4-5-6-7-8

Tabel 2 merupakan *test case* yang diberikan ke program agar setiap jalur independen di atas dapat diketahui *output*-nya.

Tabel 2 Test case basis path testing pada algoritma k-NN

Jalur 1	Test case	1. Diberikan data masukan RSS yang <i>Valid</i> sebagai <i>input</i> 2. <i>Database</i> data sampel tersedia
	Expected result	Berhasil menentukan posisi pengguna
	Output	Berhasil menentukan posisi pengguna
	Validation	<i>Valid</i>

2. Pengujian whitebox pada algoritma pathfinding A*



Gambar 6 Perhitungan cyclomatic complexity pada algoritma pathfinding A*

Pada perhitungan *cyclomatic complexity* yang ditampilkan pada Gambar 6, diketahui bahwa terdapat 5 buah jalur independen pada algoritma *pathfinding A**. Rincian jalurnya adalah sebagai berikut.

- Jalur 1 = 1-2-3-4-5-6-13-14
- Jalur 2 = 1-2-3-4-5-6-7-8-12-3-4-5-6-13-14
- Jalur 3 = 1-2-3-4-5-6-7-8-9-12-3-4-5-6-13-14
- Jalur 4 = 1-2-3-4-5-6-7-8-9-10-12-3-4-5-6-13-14
- Jalur 5 = 1-2-3-4-5-6-7-8-9-10-11-12-3-4-5-6-13-14

Tabel 3 merupakan *test case* yang diberikan ke program agar setiap jalur independen di atas dapat diketahui *output*-nya.

Tabel 3 Test case basis path testing pada algoritma pathfinding A*

	Test case	Diberikan data sel awal dan sel tujuan yang sama
Jalur 1	Expected result	Muncul notifikasi posisi pengguna saat ini dan lokasi tujuan sama
	Output	Muncul notifikasi posisi pengguna saat ini dan lokasi tujuan

		sama
	<i>Validation</i>	<i>Valid</i>
Jalur 2	<i>Test case</i>	Diberikan data sel awal dan sel tujuan yang bersebelahan
	<i>Expected result</i>	Rute ditemukan dan <i>Valid</i>
	<i>Output</i>	Rute ditemukan dan <i>Valid</i>
	<i>Validation</i>	<i>Valid</i>
Jalur 3	<i>Test case</i>	Diberikan data sel awal dan tujuan yang disesuaikan
	<i>Expected result</i>	Rute ditemukan dan <i>Valid</i>
	<i>Output</i>	Rute ditemukan dan <i>Valid</i>
	<i>Validation</i>	<i>Valid</i>
Jalur 4	<i>Test case</i>	Diberikan data sel awal dan tujuan yang disesuaikan
	<i>Expected result</i>	Rute ditemukan dan <i>Valid</i>
	<i>Output</i>	Rute ditemukan dan <i>Valid</i>
	<i>Validation</i>	<i>Valid</i>
Jalur 5	<i>Test case</i>	Diberikan data sel awal dan tujuan yang disesuaikan
	<i>Expected result</i>	Rute ditemukan dan <i>Valid</i>
	<i>Output</i>	Rute ditemukan dan <i>Valid</i>
	<i>Validation</i>	<i>Valid</i>

3.4. Pengujian Akurasi Pemosisian dengan Nilai Variabel “k” Berbeda

Untuk mengetahui tingkat efektifitas kinerja dari algoritma k -NN dalam penentuan posisi pada sistem, maka perlu dilakukan pengujian dengan penggunaan variasi dari nilai variabel k . Hasil dari pengujian ditampilkan pada Tabel 4.

Tabel 4 Pengujian akurasi pemosisian dengan nilai variabel $k=3$

Variabel $k = 3$		
No	Lokasi	Hasil Pengujian
1	Ruang 302	Konsisten
2	Ruang 303	Konsisten
3	Ruang 304	Konsisten
4	Ruang 305	Konsisten
5	Ruang 306	Konsisten
6	Ruang 307	Konsisten
7	Aula	Konsisten
8	Lobi1	Tidak Konsisten
9	Lobi2	Tidak Konsisten
10	Lobi3	Konsisten
11	Toilet	Konsisten
12	Koridor1	Konsisten
13	Koridor2	Konsisten
14	Koridor3	Tidak Konsisten
15	Koridor4	Tidak Konsisten
16	Koridor5	Konsisten

Keterangan :

- Konsisten = Hasil pengujian benar dan hasil konsisten setelah beberapa kali pengujian
- Tidak Konsisten = Hasil pengujian tidak konsisten setelah beberapa kali pengujian

Hasil pengujian dengan nilai variabel $k=3$ yang ditampilkan pada Tabel 4 menghasilkan beberapa hasil yang tidak konsisten yaitu lokasi Lobi 1, Lobi 2, Koridor 3, dan Koridor 4. Diketahui bahwa keempat posisinya saling berdekatan serta tidak terhalang oleh dinding satu sama lain, sehingga ketiga lokasi ini memiliki beberapa data sampel *wi-fi* yang datanya mirip. Berbeda dengan lokasi lainnya yang terlindungi oleh dinding dan mengakibatkan variasi data yang sangat berbeda di setiap lokasinya.

Tabel 5 Pengujian akurasi pemposisian dengan nilai variable $k=4$

Variabel $k = 4$		
No	Lokasi	Hasil Pengujian
1	Ruang 302	Konsisten
2	Ruang 303	Konsisten
3	Ruang 304	Konsisten
4	Ruang 305	Konsisten
5	Ruang 306	Konsisten
6	Ruang 307	Konsisten
7	Aula	Konsisten
8	Lobi 1	Tidak Konsisten
9	Lobi2	Tidak Konsisten
10	Lobi3	Konsisten
11	Toilet	Konsisten
12	Koridor1	Konsisten
13	Koridor2	Konsisten
14	Koridor3	Konsisten
15	Koridor4	Tidak Konsisten
16	Koridor5	Konsisten

Hasil pengujian akurasi pemposisian dengan nilai variabel $k=4$ yang ditampilkan pada Tabel 5, masih menghasilkan beberapa output yang tidak konsisten. Ada 3 lokasi yang hasil pengujianjnya masih tidak konsisten yaitu Lobi 1, Lobi 2, dan Koridor 4. Ketiga lokasi tersebut kondisinya sama seperti pengujian sebelumnya yaitu posisinya yang saling berdekatan serta tidak terhalang oleh dinding.

Tabel 6 Pengujian akurasi pemposisian dengan nilai variabel $k=5$

Variabel $k = 5$		
No	Lokasi	Hasil Pengujian
1	Ruang 302	Konsisten
2	Ruang 303	Konsisten
3	Ruang 304	Konsisten
4	Ruang 305	Konsisten
5	Ruang 306	Konsisten
6	Ruang 307	Konsisten
7	Aula	Konsisten
8	Lobi 1	Tidak Konsisten
9	Lobi2	Konsisten
10	Lobi3	Konsisten
11	Toilet	Konsisten
12	Koridor1	Konsisten
13	Koridor2	Konsisten
14	Koridor3	Konsisten
15	Koridor4	Tidak Konsisten
16	Koridor5	Konsisten

Hasil pengujian dengan nilai variabel $k = 5$ yang ditampilkan pada Tabel 6 masih menghasilkan beberapa lokasi yang hasil pemposisiannya tidak konsisten. Beberapa lokasi tersebut yaitu Lobi 1 dan Koridor 4. Kedua lokasi ini masih berlokasi sama seperti pengujian sebelumnya, serta saling berdekatan dan tanpa ada sekat dinding.

Tabel 4.4 Pengujian akurasi pemosisian dengan nilai variabel $k=6$

Variabel $k = 6$		
No	Lokasi	Hasil Pengujian
1	Ruang 302	Konsisten
2	Ruang 303	Konsisten
3	Ruang 304	Konsisten
4	Ruang 305	Konsisten
5	Ruang 306	Konsisten
6	Ruang 307	Konsisten
7	Aula	Konsisten
8	Lobi1	Konsisten
9	Lobi2	Konsisten
10	Lobi3	Konsisten
11	Toilet	Konsisten
12	Koridor1	Konsisten
13	Koridor2	Konsisten
14	Koridor3	Konsisten
15	Koridor4	Konsisten
16	Koridor5	Konsisten

Hasil pengujian pemosisian dengan nilai variabel $k = 6$ sangat baik. Semua lokasi menghasilkan *output* yang tepat dan konsisten. Dengan hasil dari rangkaian pengujian pemosisian dengan variasi nilai variabel k di atas, maka dapat diambil kesimpulan bahwa menggunakan nilai variabel k yang cenderung besar akan menghasilkan hasil pemosisian yang semakin baik atau konsisten.

4. KESIMPULAN

Berdasarkan hasil pengkajian maka penulis dapat mengambil langkah kesimpulan antara lain :

1. Pada penelitian ini peneliti dapat membangun prototipe sistem navigasi dalam ruangan dengan mengimplementasikan metode wi-fi fingerprint positioning dengan algoritma k -NN untuk pemosisian, dan algoritma pathfinding A* untuk pencarian rute dengan hasil sistem dapat melakukan penentuan posisi dan melakukan pencarian rute dengan baik.
2. Dari hasil pengujian kinerja penentuan posisi dengan variasi nilai variabel k yaitu $k=3$, $k=4$, $k=5$, dan $k=6$, didapatkan nilai variabel $k=6$ yang memiliki hasil penentuan posisi yang konsisten. Serta dapat disimpulkan bahwa penggunaan nilai variabel k yang cenderung besar menghasilkan *output* yang lebih konsisten.
3. Dengan adanya sistem ini dapat memudahkan pengunjung baru suatu public area dalam menelusuri lokasi-lokasi yang berada dalam lingkup public area khususnya pada area dalam ruangan.

DAFTAR PUSTAKA

- [1] E. Suropto Pasinggi, S. Paelleng, and F. Boas Gallaran, "Literature Review: Arsitektur Sistem Penentuan Posisi di Dalam Ruangan," 2018.
- [2] Z. L. Deng, Y. P. Yu, and L. M. Xu, "Wireless Location and Navigation in Indoor and Outdoor," *Beijing University of Posts and Telecommunications Press: Beijing, China*, 2013.
- [3] S. Xia, Y. Liu, G. Yuan, M. Zhu, and Z. Wang, "Indoor fingerprint positioning based on Wi-Fi: An overview," *ISPRS International Journal of Geo-Information*, vol. 6, no. 5. MDPI AG, May 01, 2017. doi: 10.3390/ijgi6050135.
- [4] Y. B. Bai *et al.*, "A new method for improving Wi-Fi-based indoor positioning accuracy," *Journal of Location Based Services*, vol. 8, no. 3, pp. 135–147, Jul. 2014, doi: 10.1080/17489725.2014.977362.
- [5] A. R. Musthafa, R. V. H. Ginardi, and A. Arunanto, "Sistem Navigasi Indoor Menggunakan Sinyal Wi-fi dan Kompas Digital Berbasis Integrasi dengan Smartphone untuk Studi Kasus pada Gedung Bertingkat," *Jurnal Teknik ITS*, vol. 5, pp. A448–A452, 2016.
- [6] D. Yupiter Permana, A. Handojo, and J. Andjarwirawan, "Aplikasi Indoor Positioning System Menggunakan Android dan Wireless Local Area Network Dengan Metode Fuzzy Logic Indoor Positioning System."
- [7] Suyanto, *Machine Learning. "Tingkat Dasar dan Lanjut."* Informatika Bandung, 2018.
- [8] X. Cui and H. Shi, "A*-based Pathfinding in Modern Computer Games," 2011.

-
- [9] M. A. Cahyadi, M. A. B. P, Wi. Widhiarso, and Y. Yohannes, “PERBANDINGAN ALGORITMA A*, DIJKSTRA DAN FLOYD WARSHALL UNTUK MENENTUKAN JALUR TERPENDEK PADA PERMAINAN ‘BACTERIA DEFENSE,’” 2017.
- [10] Google Developers, “WifiManager Android API,”
<https://developer.android.com/reference/android/net/wifi/WifiManager>.