



SISTEM NAVIGASI AUGMENTED REALITY DENGAN PENCARIAN JALUR TERBAIK MENUJU LOKASI PUSTAKA (STUDI KASUS PADA UPT PERPUSTAKAAN UNILA)

Mardiana^{a,*}, Dikpride Despa^b, Meizano Ardhi Muhammad^c, Trisya Septiana^d, Tiara Ayu Lorenza^e

^{a,b,c,d,e}Jurusan Teknik Elektro, Universitas Lampung, Jl. Prof. Soemantri Brojonegoro, Bandar Lampung 35145

INFORMASI ARTIKEL

ABSTRAK

Riwayat artikel:

Diterima : 2 September 2022

Direvisi : 26 Oktober 2022

Diterbitkan : 24 Desember 2022

Kata kunci:

Perpustakaan

Augmented Reality (AR)

Kanban

Dijkstra

Best First Search (BFS)

Pencarian lokasi ruangan terutama pada gedung yang berukuran besar dan memiliki ruangan yang banyak tentunya memiliki kendala tersendiri. Penelitian ini bertujuan untuk mengembangkan sistem pencarian lokasi pada sebuah gedung dengan menggunakan teknologi Augmented Reality (AR). Sistem akan menampilkan informasi berupa jarak, waktu serta rute yang dilalui untuk sampai ke lokasi tujuan. Metoda pengembangan sistem menggunakan metode Kanban untuk mengontrol aliran aktifitas dalam menyelesaikan pekerjaan. Penelitian ini menggunakan Gedung dan ruangan Perpustakaan Unila sebagai objek lokasi dengan data sebanyak 10 titik tujuan. Jalur pencarian terbaik dicari menggunakan perbandingan antara tiga algoritma, yaitu Algoritma A Star (A*), Algoritma Dijkstra, dan Algoritma Best First Search (BFS). Hasil dari penelitian ini diperoleh sistem navigasi augmented reality perpustakaan dengan berdasarkan 4 *user story* yang ditentukan, 9 *backlog* dan 10 *task card*. Sistem berhasil memberikan informasi mengenai informasi rak buku tujuan berdasarkan nomor katalog dan terdapat juga informasi navigasi seperti jarak, waktu serta rute untuk sampai ke rak buku tujuan. Hasil juga menunjukkan bahwa Algoritma A Star (A*) memiliki total jalur yang tepat sebanyak 7 dengan persentase keakuratan 70%, Algoritma Dijkstra memiliki total jalur yang tepat sebanyak 9 dengan persentase keakuratan 90%, dan Algoritma Best First Search (BFS) memiliki total jalur yang tepat 0 dengan persentase keakuratan 0%. Dengan hasil tersebut maka Algoritma Dijkstra merupakan algoritma yang efisien dalam pencarian jalur terbaik.

1. Pendahuluan

Pencarian lokasi pada sebuah gedung yang belum pernah dikunjungi masih memiliki tantangan sendiri, biasanya masih menggunakan cara – cara sederhana salah satunya dengan bertanya kepada orang yang berada didalam gedung tersebut. Tetapi, tidak semua orang yang diberikan pertanyaan mengetahui lokasi yang sedang dicari. Masalah seperti ini biasanya terjadi pada gedung-gedung yang memiliki ukuran besar dan kompleks, seperti misalnya mall, gedung pemerintahan ataupun gedung Perpustakaan.

Teknologi yang dapat membantu memberikan informasi navigasi salah satunya adalah teknologi Augmented Reality (AR). Teknologi AR menciptakan lingkungan baru dengan menggabungkan interaktivitas lingkungan nyata dan virtual sehingga pengguna merasa bahwa lingkungan yang diciptakan adalah nyata. Penelitian terdahulu telah dilakukan menggunakan objek gedung perpustakaan Universitas Lampung (Unila) untuk navigasi ruangan menggunakan teknologi web AR [Ginanjari, 2018]. Cara kerja sistem ini, pengunjung dapat menggunakan

kamera untuk memindai marker sebagai penanda objek maka secara otomatis akan memberikan informasi berupa anak panah untuk menunjukkan arah yang akan dilalui. Tetapi pada penelitian tersebut belum ada informasi yang jelas mengenai perkiraan jarak yang ditempuh untuk sampai ke tujuan. Sehingga perlu dilakukan peningkatan terhadap sistem yaitu dengan memberikan informasi mengenai jarak yang ditempuh dan rute yang harus dilalui dengan lebih jelas. Dalam menentukan jalur terbaik dan rute jarak terpendek untuk sampai ke lokasi pada penelitian ini menggunakan beberapa algoritma perhitungan jarak, sehingga penelitian navigasi menggunakan AR marker tersebut perlu dilanjutkan [Siltanen, 2021].

Pathfinding terkadang disebut sebagai perencanaan jalur. Seringkali digunakan ketika mencari tahu kemana harus bergerak berdasarkan keadaan peta untuk menemukan nilai terbaik dalam perjalanan dari satu titik ke titik yang lain. Dalam pemecahan pathfinding akan dibutuhkan algoritma yang dapat dengan cepat memproses dan menghasilkan arah untuk mencapai suatu lokasi. [Owen, 1988]. Agar pathfinding dapat berjalan pada sebuah arena dapat menggunakan Navigation mesh (navmesh).

*Penulis korespondensi.

E-mail: mardiana@eng.unila.ac.id

Navmesh merupakan sebuah bidang yang terdiri dari poligon yang terhubung ke poligon lainnya untuk menemukan jalur [Dian, 2018]. Beberapa algoritma yang dapat digunakan untuk pencarian jalur adalah Algoritma A Star (A*), Algoritma Dijkstra dan Best First Search (BFS). Algoritma A Star (A*) adalah sebuah algoritma yang menerapkan suatu heuristik. Heuristik adalah nilai yang memberi nilai pada tiap simpul yang memandu Algoritma A Star (A*) mendapatkan solusi yang diinginkan. Algoritma ini meminimalkan jalur yang dilalui. Algoritma A Star (A*) memiliki lima komponen utama, yaitu: node awal, node goal, open list, closed list dan cost. Prinsip Algoritma A Star (A*) yaitu melintasi semua graf yang berhubungan dengan starting point, mengurutkan cost terkecil dengan memperhatikan cost (f) kedalam antrian graf yang dilalui. Jika pada titik tertentu segmen lintasan yang dilalui memiliki biaya yang lebih tinggi dari segmen lintasan yang lain yang sedang dihadapi, maka A Star (A*) akan meninggalkan lintasan dengan cost yang lebih tinggi. [Suprpto, 2017].

Algoritma Dijkstra merupakan salah satu algoritma yang efektif dalam memberikan lintasan terpendek dari suatu lokasi ke lokasi lainnya. Prinsip dari Algoritma Dijkstra memiliki iterasi untuk mencari titik yang jaraknya dari titik awal adalah paling pendek. Pada setiap iterasi, jarak titik memberikan jarak terpendek. Algoritma ini bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Misalkan titik menggambarkan gedung dan garis menggambarkan jalan, maka Algoritma Dijkstra melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik [Krisdiawan, 2018].

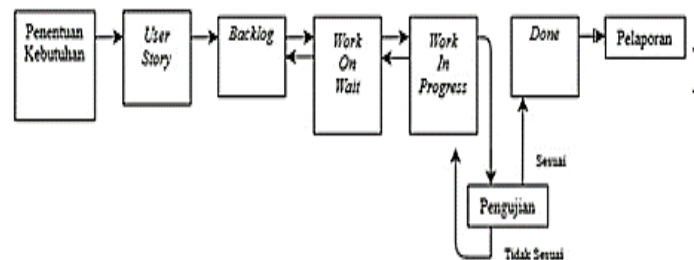
Pencarian terbaik pertama (Best First Search) merupakan suatu cara yang menggabungkan keuntungan atau kelebihan dari pencarian Breadth First dan Depth First. Pada setiap langkah proses pencarian terbaik pertama, memilih simpul-simpul (node) yang paling menjanjikan sesuatu. Hal ini dilakukan dengan menerapkan fungsi Heuristik yang memadai pada setiap simpul tersebut. Kemudian, mengembangkan simpul yang dipilih itu dengan menggunakan aturan-aturan tertentu untuk menghasilkan penggantinya. Jika salah satu simpul tersebut merupakan sebuah solusi, maka berhenti. Jika bukan, semua simpul baru itu ditambahkan ke himpunan simpul yang sejauh ini telah dibuat. [Hutahaen, 2017]

Penelitian ini bertujuan untuk membandingkan Algoritma A Star (A*), Algoritma Dijkstra dan Algoritma Best First Search (BFS) untuk mengetahui algoritma manakah yang lebih efisien untuk mencari jalur tercepat, serta menerapkan algoritma pencarian jalur tercepat tersebut pada sistem navigasi ruangan perpustakaan. Manfaat dari penelitian ini adalah pengguna perpustakaan dapat mempersingkat waktu pencarian dalam menemukan lokasi ruangan tempat pustaka yang hendak dicarinya.

2. Metodologi

Penelitian ini menggunakan metode Kanban yaitu sebuah metoda yang menggunakan kartu untuk mengontrol aliran aktifitas dalam menyelesaikan pekerjaan [Ahmad, 2016]. Dalam metode kanban terdapat beberapa tahapan seperti terlihat pada gambar 1. Tahapan penelitian dimulai dengan Penentuan Kebutuhan yang dilanjutkan dengan User Story. Tahap user story merupakan tahapan penentuan fitur dan kegiatan yang harus dilakukan dalam pembuatan aplikasi. Setelah penentuan user story maka dipecah menjadi beberapa *backlog* yang berfungsi sebagai *task card* kegiatan yang akan dikerjakan pada aplikasi. Selanjutnya, tahap Work In Progress (WIP) dan Work In Wait (WOW). Tahap WOW dan WIP adalah tahap pengembangan sistem, pada WIP dan WOW memiliki batas *task card* yang dapat

dikerjakan yaitu 5 *task* dalam sekali pengerjaan dan apabila terdapat slot pada WIP dan WOW maka diperbolehkan mengambil 1 *task card* sampai dengan slot terpenuhi. Setiap *task* yang berhasil dikerjakan pada tahap WIP akan langsung dilakukan pengujian. Pada tahap pengujian sistem, apabila *task card* telah sesuai dengan yang diharapkan maka akan masuk kedalam kategori done. Pengulangan kegiatan ini akan terus beriterasi sampai dengan sistem selesai. Tahapan terakhir adalah pelaporan.



Gambar 1. Alur Penelitian

3. Hasil dan pembahasan

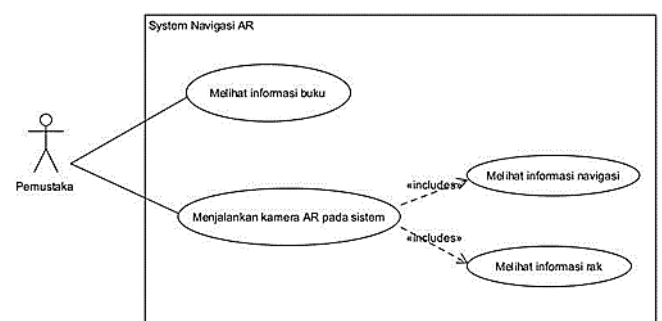
Penjelasan hasil penelitian dibagi dalam dua bagian yaitu : penentuan kebutuhan yang meliputi hasil dari tahapan Penentuan kebutuhan dan *User story* serta bagian Pengembangan yang meliputi hasil dari tahapan *Backlog*, WOW, WIP, Pengujian, Done dan Pelaporan.

3.1. Penentuan Kebutuhan

Tahapan penentuan kebutuhan merupakan tahap awal dalam metode kanban pada tahapan ini mencakup tentang design dari pekerjaan yang akan dilakukan.

Use Case Diagram

Use case diagram pada sistem navigasi augmented reality menggambarkan kondisi apa saja yang dapat pemustaka lakukan pada sistem (Hasan, 2022), yaitu pemustaka dapat mengakses kamera untuk mendapatkan navigasi berupa anak panah serta jarak dan waktu yang ditempuh menuju buku tersebut dan menjalankan navigasi untuk mengaktifkan kamera AR guna memindai marker yang sudah tersedia.

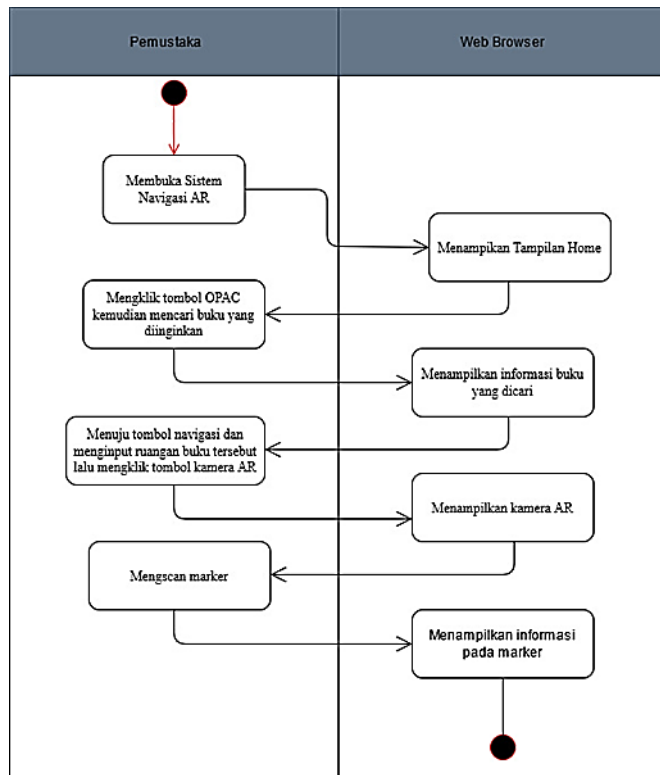


Gambar 2. Use Case Diagram

Activity Diagram

Activity diagram merupakan penggambaran sistem yang menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem. Pada gambar 3. merupakan activity diagram yang menampilkan proses pemustaka dalam mengakses sistem dimulai dari tahap awal sampai tahap akhir. Pemustaka akan mengakses sistem melalui browser kemudian, pemustaka

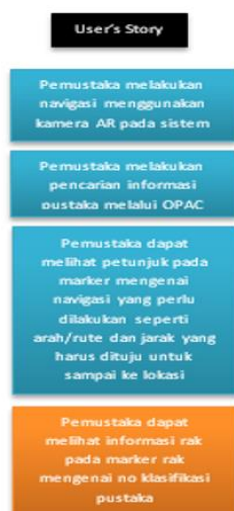
mencari informasi buku melalui OPAC setelah mendapatkan informasi buku pemustaka menjalankan navigasi AR untuk mengakses kamera AR. Sebelum mengakses kamera AR pemustaka menginput kode ruangan yang dituju ketika kamera AR telah terbuka barulah pemustaka dapat memindai marker yang sudah disediakan. Berikut merupakan gambaran activity diagram :



Gambar 3. Activity Diagram

User story

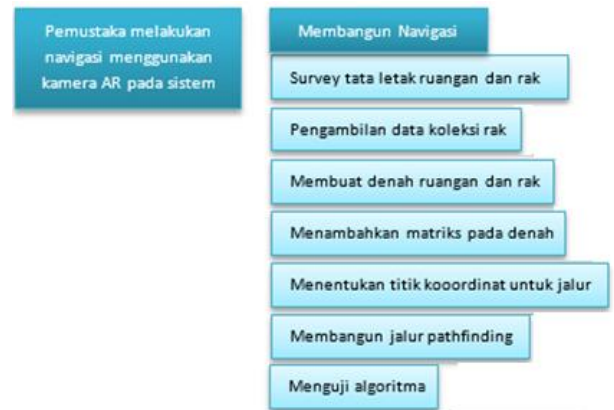
Sistem ini memiliki 4 *user story* seperti yang terlihat pada gambar 4 berikut. *User story* akan menjadi acuan dalam membuat daftar *backlog* untuk langkah-langkah menyelesaikan system.



Gambar 4. User story Sistem Navigasi Augmented Reality

3.2. Pengembangan

Pada tahap pengembangan, masing-masing *user story* dilakukan *breakdown* dan ditempatkan pada daftar *backlog*. Contoh pada gambar 5 menunjukkan *breakdown* pada *user story* pertama yang memiliki 7 *backlog*.



Gambar 5. User story pertama dan 7 Backlog

Daftar *backlog* dari *user story* yang pertama terdapat 7 *backlog* yaitu melakukan survey pada ruangan dan rak dipergustakaan universitas lampung kemudian meminta data klasifikasi rak. Dari semua bahan yang telah di dapat barulah langkah selanjutnya adalah membuat navigasi dimulai dengan membuat denah kemudian, menambahkan matriks sebanyak 100*100 untuk menentukan titik koordinat dalam pembuatan jalur pathfinding, kemudian menggunakan jalur tersebut sebagai media untuk mencari tau algoritma manakah yang terbaik. Untuk *breakdown user story* berikutnya masing-masing adalah : *user story* kedua memiliki 1 *backlog*, *user story* ketiga memiliki 1 *backlog*, dan *user story* keempat memiliki 1 *backlog*. Sehingga total 4 *user story* dalam sistem navigasi augmented reality ini sebanyak 10 *backlog*. Dalam pengembangan berikutnya, menghasilkan 10 *task card* pada tahap WOW dan WIP seperti yang terlihat pada gambar 6.berikut.



Gambar 6. WOW dan WIP Sistem Navigasi Augmented Reality

Berikut hasil dan pembahasan pada tahapan pengembangan berdasarkan user story dan tahapan metoda Kanban tersebut :

Penentuan Jalur

Langkah awal penentuan jalur adalah dengan memodelkan bentuk gedung dan posisi ruangan perpustakaan. Ruangan terdapat pada 3 lantai gedung dengan desain pemodelan dibuat berdasarkan bentuk gedung perpustakaan dan posisi letak lemari rak. Langkah selanjutnya untuk membuat jalur navigasi yaitu dengan memberikan matriks sebanyak 100*100 untuk menentukan titik kordinat letak ruangan dan rak buku pada

perpustakaan yang akan dijadikan node dalam graph. Setelah memberikan matriks kemudian menentukan titik koordinat yang akan diterapkan untuk membuat jalur navigasi. Titik koordinat meliputi ruangan, tangga, dan rak koleksi yang ada dilantai satu, lantai dua, dan lantai tiga. Informasi disimpan dalam variabel gedung dari *fungsi create3Dlayout()* dengan dimensi 100x100x8. Setiap titik gedung diberikan nama sesuai kode ruangan yang ada dengan koordinatnya masing-masing. Hasil pemetaan node dalam graph seperti yang dapat dilihat pada gambar 7. Graph diukur dengan menggunakan satuan meter (m). Graph tersebut digunakan sebagai media simulasi untuk pengujian.

Setelah menentukan titik koordinat selanjutnya membuat jalur navigasi yang akan dilalui dengan menggunakan *fungsi createWalkablePath*. Kemudian setiap jalur yang sudah disimpan akan digunakan dalam proses lebih lanjut untuk membandingkan algoritma pencari jalur terbaik. Setelah pembuatan jalur kemudian melakukan pengujian untuk menentukan algoritma yang efisien dari ketiga algoritma yang digunakan sesuai dengan jalur yang dilalui dari titik asal menuju titik tujuan. Dalam proses algoritma menggunakan *fungsi findlocation*. Hasil dari *findlocation* adalah waktu rute jalur terdekat berdasarkan algoritma yang digunakan.

Pencarian Jalur

Tahapan selanjutnya yaitu pencarian jalur menggunakan tiga algoritma dengan dua fungsi utama yaitu, *fungsi createWalkablePath* yang digunakan untuk membuat jalur secara keseluruhan dan *fungsi findLocation* yang digunakan untuk mencari jalur efisien dari satu titik asal ke titik akhir. Pengambilan sampel menggunakan 10 jalur. Untuk menentukan jalur tersebut efisien atau tidak dapat dinilai dari aspek panjang path (jarak) yang dilalui, waktu tempuh, dan banyak node yang dilalui. Node dalam graph seperti yang terlihat pada gambar 7. Pengujian mengambil titik asal PMP (node berwarna kuning) menuju rak 11_12 pada ruang RKC2 (node berwarna biru).

Hasil Pencarian Jalur

A. Algoritma A Star (A*)

Pada Algoritma A Star (A*) berdasarkan 10 jalur pengujian yang telah dibuat maka didapatkan 7 jalur yang memenuhi parameter yaitu jalur PMP menuju RC10_13L2, RC16_17L2, RC18_19L2, RC20_21L2, RC32_33L2, RC34_35L2, RC36_37L2 sedangkan 3 jalur lainnya tidak memenuhi parameter yaitu jalur PMP menuju RC5_6AL2, RC8_9AL2, dan RC11_12L2 dikarenakan jalur tersebut tidak melalui semua node yang telah ditentukan. Algoritma ini memperoleh tingkat keakuratan sebesar 70%. Hasil keakuratan tersebut diperoleh berdasarkan menggunakan matriks 100*100*8. Matriks tersebut dibuat berdasarkan perbandingan antara skala sebenarnya dengan skala pada gambar. Tabel 1 adalah hasil pencarian jalur menggunakan Algoritma A Star (A*) :

Tabel 1. Hasil Pencarian Jalur Algoritma A Star (A*)

Algoritma Astar					
NO	Asal	Tujuan	Jarak	Waktu	Rute yang dilewati
1	PMP	RC5_6AL2	53 M	75.71 D	PMP, TG1BD, TG2BD, L2T6, L2T7, L2T8, L2T5, RKC2, RC8_9AL2, RC5_6AL2
2	PMP	RC8_9AL2	51 M	72.86 D	PMP, TG1BD, TG2BD, L2T6, L2T7, L2T8, L2T5, RKC2, RC8_9AL2
3	PMP	RC10_13L2	55 M	78.57 D	PMP, TG1BD, TG2BD, L2T6, L2T7, L2T8, L2T5, RKC2, L2T12, L2T13, RC10_13L2
4	PMP	RC11_12L2	51 M	72.86 D	PMP, TG1BD, TG2BD, L2T6, L2T7, L2T8, L2T5, RKC2, RC11_12L2
5	PMP	RC16_17L2	58 M	82.86 D	PMP, TG1BD, TG2BD, L2T6, L2T7, L2T8, L2T5, RKC2, RC11_12L2, RC10_13L2, RC14_15L2, RC16_17L2
6	PMP	RC18_19L2	59 M	84.29 D	PMP, TG1BD, TG2BD, L2T6, L2T7, L2T8, L2T5, RKC2, RC11_12L2, RC10_13L2, RC14_15L2, RC18_19L2
7	PMP	RC20_21L2	61 M	87.14 D	PMP, TG1BD, TG2BD, L2T6, L2T7, L2T8, L2T5, RKC2, RC11_12L2, RC10_13L2, RC14_15L2, RC18_19L2, RC20_21L2
8	PMP	RC32_33L2	54 M	77.14 D	PMP, TG1BD, TG2BD, L2T6, L2T7, L2T8, L2T5, RKC2, L2T12, RC32_33L2
9	PMP	RC34_35L2	55 M	78.57 D	PMP, TG1BD, TG2BD, L2T6, L2T7, L2T8, L2T5, RKC2, L2T12, RC34_35L2
10	PMP	RC36_37L2	55 M	78.57 D	PMP, TG1BD, TG2BD, L2T6, L2T7, L2T8, L2T5, RKC2, L2T12, RC36_37L2
Total jalur yang tepat					7
Kesakuratan Algoritma					70%

B. Algoritma Dijkstra

Pada Algoritma Dijkstra berdasarkan 10 jalur pengujian yang telah dibuat maka didapatkan 9 jalur yang memenuhi parameter sedangkan 1 jalur lainnya tidak memenuhi parameter yang mana jalur tersebut adalah jalur PMP menuju RC10_13L2 dikarenakan jalur tersebut tidak melalui semua node yang telah ditentukan. Algoritma ini memperoleh tingkat keakuratan sebesar 90%. Hasil keakuratan tersebut diperoleh berdasarkan menggunakan matriks 100*100*8. Tabel 2. adalah hasil pencarian jalur menggunakan Algoritma Dijkstra :

Tabel 2. Hasil Pencarian Jalur Algoritma Dijkstra

Algoritma Dijkstra					
NO	Asal	Tujuan	Jarak	Waktu	Rute yang dilewati
1	PMP	RC5_6AL2	56 m	80 d	PMP, TG1BD, TG2BD, L2T6, L2T7, L2T8, L2T5, RKC2, L2T12, L2T13, RC11_12L2, RC5_6AL2
2	PMP	RC8_9AL2	54 m	77.14 d	PMP, TG1BD, TG2BD, L2T6, L2T7, L2T8, L2T5, RKC2, L2T12, L2T13, RC11_12L2, RC8_9AL2
3	PMP	RC10_13L2	55 m	78.57 d	PMP, TG1BD, TG2BD, L2T6, L2T7, L2T8, L2T5, RKC2, RC11_12L2, RC10_13L2
4	PMP	RC11_12L2	53 m	75.71 d	PMP, TG1BD, TG2BD, L2T6, L2T7, L2T8, L2T5, RKC2, L2T13, RC11_12L2
5	PMP	RC16_17L2	58 m	82.86 d	PMP, TG1BD, TG2BD, L2T6, L2T7, L2T8, L2T5, RKC2, L2T13, RC10_13L2, RC16_17L2
6	PMP	RC18_19L2	59 m	84.29 d	PMP, TG1BD, TG2BD, L2T6, L2T7, L2T8, L2T5, RKC2, L2T13, RC10_13L2, RC16_17L2, RC18_19L2
7	PMP	RC20_21L2	61 m	87.14 d	PMP, TG1BD, TG2BD, L2T6, L2T7, L2T8, L2T5, RKC2, L2T13, RC10_13L2, RC20_21L2
8	PMP	RC32_33L2	54 m	77.14 d	PMP, TG1BD, TG2BD, L2T6, L2T7, L2T8, L2T5, RKC2, L2T12, RC32_33L2
9	PMP	RC34_35L2	55 m	78.57 d	PMP, TG1BD, TG2BD, L2T6, L2T7, L2T8, L2T5, RKC2, L2T12, RC34_35L2
10	PMP	RC36_37L2	55 m	78.57 d	PMP, TG1BD, TG2BD, L2T6, L2T7, L2T8, L2T5, RKC2, L2T12, RC36_37L2
Total jalur yang tepat					9
Keakuratan Algoritma					90%

C. Algoritma Best First Search (BFS)

Pada algoritma best first search berdasarkan 10 jalur pengujian yang telah dibuat menggunakan matriks 100*100*8 tidak ada jalur yang memenuhi parameter dikarenakan jalur tersebut hanya melewati beberapa node saja untuk sampai ketujuan.

Berdasarkan penelitian untuk resolusi matriks 100*100*8 didapatkan hasil bahwa Algoritma A Star (A*) merupakan algoritma tercepat untuk menemukan lokasi tujuan dikarenakan waktu tempuh, waktu komputasi, dan jumlah node yang didapat pada saat pengujian lebih sedikit jika dibandingkan dengan

dengan Algoritma Dijkstra dan Best First Search, tetapi dalam pencarian jalur yang efisien menggunakan ketiga algoritma tersebut maka didapatkan hasil bahwa Algoritma Dijkstra merupakan algoritma yang efisien digunakan dalam pencarian jalur terbaik. Hal ini dikarenakan cara kerja Algoritma Dijkstra yaitu membaca terlebih dahulu semua jalur yang telah dibuat kemudian menentukan jalur mana yang memiliki bobot terkecil. Algoritma Dijkstra melakukan pencarian jarak terpendek dari titik asal ke titik terdekat dan seterusnya untuk mencapai lokasi tujuan berdasarkan beberapa parameter yaitu panjang path yang dilalui dan banyak node yang dilalui. Dengan cara kerja algoritma tersebut maka Algoritma Dijkstra memiliki tingkat keakuratan terbesar jika dibandingkan dengan algoritma lainnya yaitu sebesar 90%.

Tabel 3. adalah hasil pencarian jalur menggunakan algoritma best first search :

Tabel 3. Hasil Pencarian Jalur Algoritma BFS

Algoritma Best First Search					
NO	Asal	Tujuan	Jarak	Waktu	Rute yang dilewati
1	PMP	RC5_6AL2	612 M	874.29 D	PMP, TG3BD, L2T2, L2T9, TG3TL, TG2TL, L2T12, RC16_17L2, RC5_6AL2
2	PMP	RC8_9AL2	325 M	464.29 D	PMP, TG3BD, L2T2, L2T9, TG3TL, TG2TL, RC8_9AL2
3	PMP	RC10_13L2	324 M	462.86 D	PMP, TG3BD, L2T2, L2T9, TG3TL, TG2TL, RC10_13L2
4	PMP	RC11_12L2	325 M	464.29 D	PMP, TG3BD, L2T2, L2T9, TG3TL, TG2TL, RC11_12L2
5	PMP	RC16_17L2	596 M	851.43 D	PMP, TG3BD, L2T2, L2T9, TG3TL, TG2TL, L2T12, RC16_17L2
6	PMP	RC18_19L2	597 M	852.86 D	PMP, TG3BD, L2T2, L2T9, TG3TL, TG2TL, L2T12, RC18_19L2
7	PMP	RC20_21L2	72 M	102.86 D	PMP, L1T1, TG3BL, RC22_23L2, RC20_21L2
8	PMP	RC32_33L2	614 M	877.14 D	PMP, TG3BD, L2T2, L2T9, TG2TL, RC16_17L2, RC32_33L2
9	PMP	RC34_35L2	615 M	878.57 D	PMP, TG3BD, L2T2, L2T9, TG3TL, TG2TL, L2T12, RC16_17L2, RC34_35L2
10	PMP	RC36_37L2	213 M	304.29 D	PMP, TG3BD, L2T2, L2T9, RC36_37L2
Total jalur yang tepat			0		
Keakuratan Algoritma			0%		

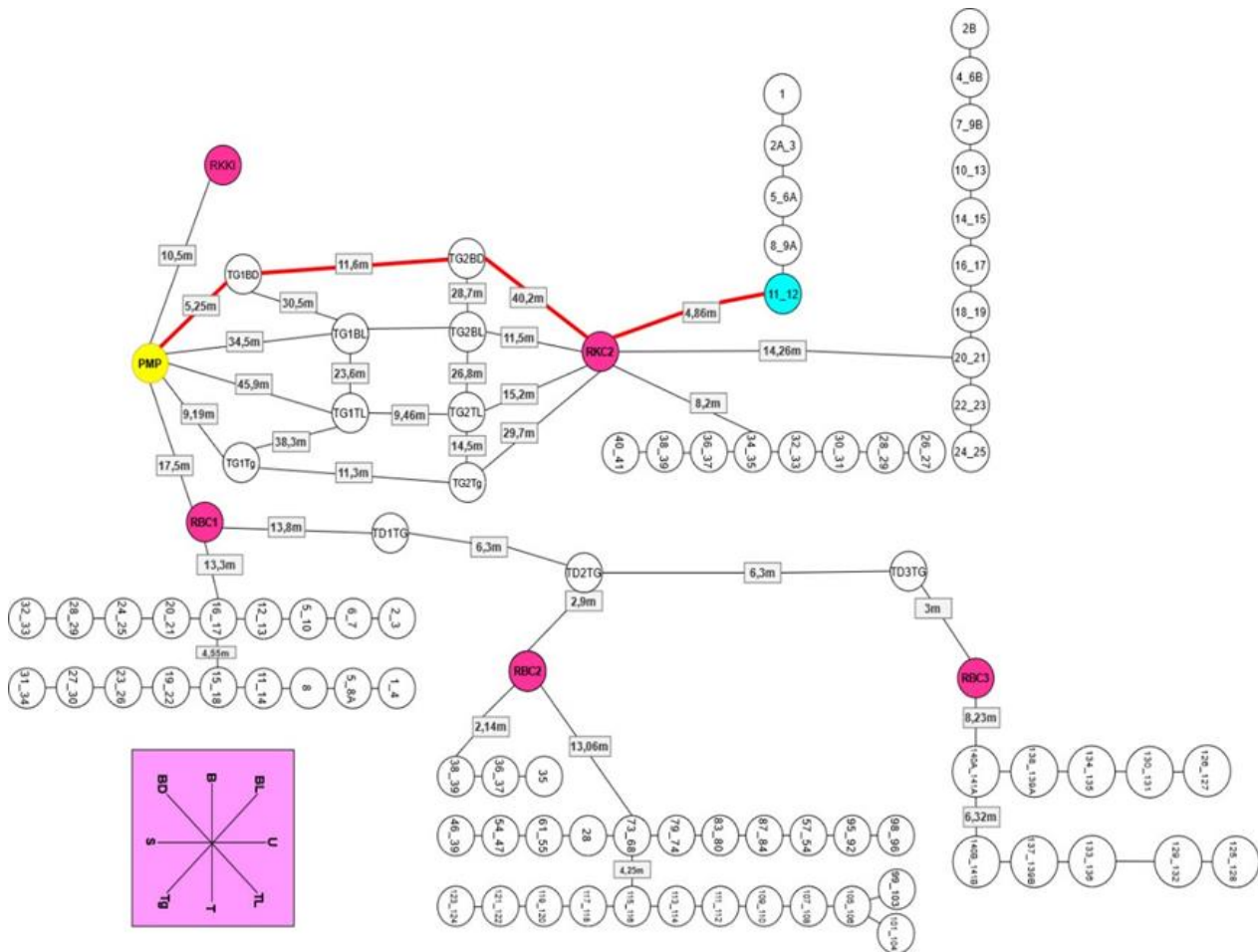
Dari pengujian menggunakan ketiga algoritma tersebut diperoleh hasil bahwa Algoritma Dijkstra pada kasus ini merupakan algoritma yang terbaik dengan tingkat keakuratan sebesar 90% , sedangkan tingkat keakuratan Algoritma A Star (A*) 70% dan Algoritma Best First Search 0%.

Setelah mendapatkan algoritma yang terbaik yaitu Algoritma Dijkstra, kemudian akan ditentukan rute yang tercepat. Simulasi digambarkan dengan pemustaka sedang berada pada node Pintu Masuk Perpustakaan (PMP) berwarna kuning menuju rak koleksi 11_12 ditandai dengan warna biru yang berada diruang koleksi cadangan lantai dua berwarna pink. Untuk ruangan yang berada pada graph ditandai dengan warna pink. Kemudian, jalur yang dilalui untuk mencapai node tujuan ditandai dengan garis yang berwarna merah. Perhitungan dilakukan dengan cara, jarak yang sudah ada pada antar node dijumlahkan kemudian menentukan jalur yang digunakan.

Pada pengujian Algoritma Dijkstra ini diperoleh jarak sejauh 54,55 m dan mendapatkan hasil jalur terpendek yang dilalui untuk sampai ke titik tujuan yaitu PMP – TG1BD – TG2BD – RKC2 –11_12. Graph dapat dilihat pada gambar 7, sedangkan table hasil perhitungan terdapat pada tabel 4.

Tabel 4. Perhitungan Algoritma Dijkstra

	PMP	TG1BD	TG1BL	TG1TL	TG1TG	TG2BD	TG2BL	TG2TL	TG2TG	RKC2	11_12
	0*	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
PMP	0*	5,25*	34,5	45,9	9,19	∞	∞	∞	∞	∞	∞
TG1BD	0*	5,25*	34,5	45,9	9,19*	16,85	∞	∞	∞	∞	∞
TG1TG	0*	5,25*	34,5	45,9	9,19*	16,85*	∞	∞	20,49	∞	∞
TG2BD	0*	5,25*	34,5	45,9	9,19*	16,85*	45,55	∞	20,49*	57,05	∞
TG2TG	0*	5,25*	34,5*	45,9	9,19*	16,85*	45,55	34,99	20,49*	49,69	∞
TG1BL	0*	5,25*	34,5*	45,9	9,19*	16,85*	45,55	34,99*	20,49*	49,69	∞
TG2TL	0*	5,25*	34,5*	45,9	9,19*	16,85*	45,55*	34,99*	20,49*	49,69	∞
TG2BL	0*	5,25*	34,5*	45,9*	9,19*	16,85*	45,55*	34,99*	20,49*	49,69	∞
TG1TL	0*	5,25*	34,5*	45,9*	9,19*	16,85*	45,55*	34,99*	20,49*	49,69*	∞
RKC2	0*	5,25*	34,5*	45,9*	9,19*	16,85*	45,55*	34,99*	20,49*	49,69*	54,55*
11_12	0*	5,25*	34,5*	45,9*	9,19*	16,85*	45,55*	34,99*	20,49*	49,69*	54,55*



Gambar 7. Graph Diagram Ruangan Perpustakaan

Sistem Navigasi AR

Tahapan selanjutnya pada pengembangan adalah proses pembuatan AR navigasi dan AR marker serta mengintegrasikan sistem navigasi dengan sistem perpustakaan yang berjalan yaitu OPAC (*Online Public Access Catalog*) [Opac, 2022]. Tahapan ini digunakan agar diperoleh tampilan navigasi AR ketika pemustaka menggunakan OPAC dan mengakses katalog buku yang dicari. Akan terdapat juga informasi lanjutan yaitu letak ruangan dan rak koleksi serta nomor koleksi pustaka yang terdapat pada perpustakaan Universitas Lampung. Berikut pada gambar 8 merupakan tampilan dari menu OPAC serta menu navigasi yang telah diintegrasikan.



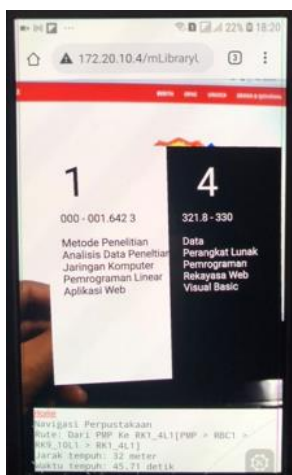
Gambar 8. Tampilan menu integrasi AR Navigasi dan OPAC

Setelah pemustaka mendapatkan no katalog buku yang dicari melalui menu OPAC (kiri), pemustaka dapat memindai navigasi AR untuk mendapatkan navigasi menuju keberadaan buku tersebut dengan memilih menu navigasi dan menginputkan kode rak tujuannya (kanan) kemudian mengklik tombol navigasi agar kamera AR dapat digunakan untuk memindai marker guna mendapatkan informasi navigasi seperti anak panah, jarak untuk sampai ke tujuan, waktu yang diperlukan, serta rute yang akan dilalui untuk sampai ke tujuan. Berikut merupakan tampilan ketika mendeteksi marker :



Gambar 9 Tampilan marker navigasi

Dari gambar tersebut terlihat bahwa sistem berhasil memberikan informasi mengenai informasi rak buku tujuan berdasarkan nomor katalog. Apabila pemustaka telah berhasil sampai ke rak buku tujuan dan pemustaka ingin memastikan apakah rak buku tersebut benar, maka pemustaka dapat mendeteksi marker lain yang terdapat pada rak buku untuk menyamakan nomor katalog yang dituju pada rak tersebut. Gambar 10 berikut merupakan tampilan pada marker rak yang terdeteksi oleh kamera AR :



Gambar 10 Tampilan marker rak

4. Kesimpulan

Sistem navigasi augmented reality perpustakaan telah berhasil dikembangkan sesuai dengan 4 user story yang ditentukan, 9 backlog serta 10 *task card*. Sistem berhasil memberikan informasi mengenai informasi rak buku tujuan berdasarkan nomor katalog. Pemustaka juga dapat memperoleh informasi pada marker navigasi seperti jarak, waktu serta rute untuk sampai ke tujuan. Berdasarkan hasil dari pengujian pada kasus ini, Algoritma Dijkstra merupakan algoritma terbaik yang digunakan dalam pencarian jalur efisien dengan tingkat keakuratan sebesar 90% sedangkan Algoritma A Star (A*) 70% dan Algoritma Best First Search 0%.

Daftar pustaka

- Dian,S., Permana,H., Bayu, K. Y., 2018. Comparative Analysis of Pathfinding Algorithms A *, Dijkstra , and BFS on Maze Runner Game,” Int. J. Inf. Syst. Technol., vol. 1, no. 2, pp. 1–8.
- GINANJAR, Mardiana, Muhammad, M. A, Sulistiono, W.E. 2018. Model 3 Dimensi Gedung Sebagai basis Untuk Membantu Navigasi Pengunjung Perpustakaan, Prosiding Seminar Nasional Inovasi Perpustakaan (SNIPer 2018), Bandar Lampung.
- Hasan, Y.A., Mardiana, G. F. Nama, 2022. Sistem Pendeteksi Kebocoran Tabung Gas Lpg Otomatis Berbasis Arduino Uno Menggunakan Metode Prototipe. Jurnal Informatika Dan Teknik Elektro Terapan, 10(3).
- Hutahaean, H. D. 2019. Penerapan Metode Best First Search pada Permainan Tic Tac Toe, IOCS, vol. 1, no. 1, pp. 10–15.
- Krisdiawan, R. A., Galih, E, C. 2018. Implementasi Algoritma Dijkstra pada Aplikasi Wisata Kuningan Berbasis Android,Nuansa Inform., vol. 12, no. 1, pp. 9–19.
- Suprpto, 2017. Perbandingan Performansi Pencarian Jarak Terpendek Menggunakan Algoritma A* dan Algoritma Dijkstra (Studi Kasus : Jalan Pariwisata Kabupaten Pati), Semarang.
- Owen,T. 1988. Artificial Intelligence by Patrick Henry Winston (second edition) Addison-Wesley Publishing Company, Massachusetts, USA, July 1984, student hardback edition)
- OPAC (*Online Public Access Catalog*) UPT Perpustakaan Unila, 2022. <https://opac.unila.ac.id>, akses 12 Maret 2022
- Siltanen, Sanni. 2012. Theory and applications of marker-based augmented reality. Espoo 2012. VTT Science Series 3. Retrieved from <http://www.vtt.fi/publications/index.jsptanen>, Theory and applications of marker-based augmented reality. 2012.
- Ahmad, M. O. 2016. Exploring Kanban in software engineering. University of Oulu