IMPLEMENTASI SISTEM PARKIR CERDAS DI UNIVERSITAS TELKOM. SUBSISTEM : APLIKASI MOBILE

Implementation Of Smart Parking System In Telkom University. Subsystem : Mobile Application

Annis Waziroh[1], Agus Virgono, Andrew B. Osmond[3]
Prodi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom, anniswaz@gmail.com[1]
Prodi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom,
avirgono@telkomuniversity[2]
Prodi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom, abo.course@gmail.com[3]

Abstrak

Pertumbuhan penduduk yang sangat pesat akan berimbas terhadap peningkatan jumlah kendaraan. Mengingat akan banyaknya jumlah kendaraan yang ada, terutama kendaraan beroda empat yang membutuhkan *space* yang luas jika akan diparkirkan. Hal ini akan menimbulkan kendala yaitu kesulitan menemukan lokasi parkir jika pengemudi mobil belum mengetahui apakah ada lokasi parkir yang kosong atau tidak. Masalah tersebut dapat teratasi apabila *user* mengetahui kondisi lahan parkir yang akan dituju. Dengan permasalahan di atas maka penulis mempunyai ide untuk membuat aplikasi mengenai *smart parking system* berbasis *mobile*.

Aplikasi ini dibuat sebagai aplikasi *mobile* dengan *platform* android dan dimaksimalkan dengan menambahkan AI (*Artificial Intelligence*). Pengemudi dapat melakukan booking para area parkir tujuan dan jika kuota pada area parkir yang dituju penuh, maka pengemudi akan diberikan area parkir alternatif yang terdekat dengan area parkir tujuan. Pencarian area parkir alternatif akan menggunakan metode DWA*. Metode DWA* dipengaruhi oleh nilai heuristik (h(n)) dan nilai dinamis (w(n)). Nilai heuristik (h(n)) dan nilai dinamis (w(n)) yang tidak tepat akan membuat pencarian menjadi salah arah. Aplikasi Smart Parking System ini dikembangkan untuk membantu pengemudi mengetahui informasi mengenai lokasi parkir, dengan nilai akurasi 92.592%, rata-rata waktu eksekusi 0.49 msekon dengan penghematan memori.

1. Pendahuluan

Tansportasi merupakan salah satu elemen yang sangat penting bagi kebutuhan kehidupan manusia. Saat ini sebagian masyarakat pada umumnya menggunakan kendaraan milik sendiri (pribadi). Berdasarkan data yang dirilis oleh bps.go.id terjadi kenaikan mobil penumpang dengan 10.54 juta unit dengan kenaikan sebesar 11% dari tahun sebelumnya yaitu 9.524 juta unit Karena perkembangan yang sangat pesat, terutama kendaraan pribadi beroda empat, tentunya hal ini akan menimbulkan kendala untuk tempat parkir. Kebanyakan dari sistem parkir yang ada hanya mencatat waktu masuk dan waktu keluar dari mobil serta menghitung lamanya suatu mobil berada ditempat parkir. Pada kasus seperti ini pengemudi akan dirugikan karena waktu akan terbuang untuk mencari lokasi parkir yang kosong. Salah satu cara untuk menghindari permasalahan tersebut yaitu apabila pengemudi telah mengetahui informasi keadaan tempat parkir pada tempat yang akan dituju. Melihat permasalahan yang telah dipaparkan maka penulis mempunyai ide untuk membuat "*Implementasi Sistem Parkir Cerdas di Universitas Telkom. Subsistem : Aplikasi Mobile*".

Aplikasi ini akan memberikan informasi mengenai lokasi parkir. Data mengenai bagaimana keadaan lokasi parkir apakah dalam keadaan kosong ataupun penuh akan didapatkan dengan memberikan kamera pada lokasi parkir. Data yang didapatkan akan diolah sedemikan rupa sampai dapat digunakan pada aplikasi *mobile*. Aplikasi akan memberikan lokasi parkir yang diinginkan *user* berdasarkan pilihan *user*, kuota dan kecepatan *user* mengakses aplikasi. Apabila

lokasi parkir penuh maka akan diberikan alternatif parkir lain yang paling dekat dengan parkir tujuan awal. Penentuan solusi alternatif parkir menggunakan metode penyelesaian *Shortest Path Solution* dengan memilih salah satu varian dari A* yaitu *Dynamic Weighting A* (DWA*)*. Diharapkan aplikasi ini dapat membantu pengemudi dan dapat diterapkan pada kehidupan sehari-hari. Tujuan dari pembuatan Tugas Akhir ini adalah untuk memastikan bahwa *user* mendapatkan informasi lokasi parkir dengan menggunakan aplikasi *smart parking system*, Mengimplementasikan *Dynamic Weighting A** (DWA*) dalam menyelesaikan masalah dalam menemukan solusi alternatif parkir dari lokasi parkir tujuan dan Mengetahui performansi dari *Dynamic Weighting A** (DWA*) dalam mendapatkan *Shortest Path Problem* dari sisi output sistem, waktu dan memory

2. Dasar Teori

2.1. Smart Parking System

Smart system secara harfiah berarti sistem pintar, dalam hal ini pintar memiliki arti mampu melakukan sesuatu dengan baik, teratur, dan rapi sesuai dengan aturan atau etika yang berlaku, serta mampu menyerap informasi dengan baik dan cepat sebagai hasil dari pembelajaran. *Smart parking system* merupakan bagian atau *part of smart city* yang mengkhususkan pada tata kelola lokasi parkir sehingga lebih teratur dan efisien [1]. Diharapkan dengan adanya *Smart Parking System* dapat membantu pengemudi untuk mengurangi waktu yang hilang karena harus mencari-cari tempat parkir yang kosong.

2.2. Kecerdasan Buatan

Para ilmuan memiliki dua cara pandang yang berbeda mengenai kecerdasan buatan (*Artificial Intelligent*). Pertama yaitu kecerdasan buatan merupakan bidang ilmu yang terfokus pada proses berfikir. Kedua yaitu kecerdasan buatan yang berfokus pada bidang ilmu tingkah laku. Kedua cara pandang ini saling terikat karena tingkah laku selalu diawali dengan proses berfikir.[2]

Kecerdasan buatan adalah salah satu bagian ilmu komputer yang membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik manusia. Agar komputer dapat melakukan hal tersebut, maka komputer harus diberi bakal pengetahuan, dan mempunyai kemampuan untuk menalar. Untuk itu, pada kecerdasan buatan diberikan beberapa metode untuk membekali komputer agar bisa menjadi mesin yang pintar. [3]

2.3. Pencarian atau Searching

Beberapa permasalahan pencarian merupakan sebuah permasalahan yang mencari rute dimulai dari titik awal (*initial state*) menuju titik tujuan (*goal state*) dan rute yang dihasilkan tersebut merupakan rute dengan jarak terpendek. Untuk mengukur performansi metode pencarian, terdapat empat kriteria yang dapat digunakan, yaitu:

1. *Completeness*
   Apakah metode tersebut menjamin penemuan solusi jika solusinya memang ada?
2. *Time complexity*
   Berapa lama waktu yang diperlukan
3. *Space complexity*
   Berapa banyak memori yang diperlukan
4. *Optimality*
   apakah metode tersebut menjamin menemukan solusi yang terbaik? [3]

2.3.1 Metode Pencarian Buta

Sebuah metode pencarian yang tidak memiliki informasi awal yang digunakan dalam proses pencarian dikenal dengan istilah pencarian buta atau *blind/un-informed search*. Metode-

metode pencarian ini memiliki karakteristik sendiri dalam pemecahan masalah. Ada beberapa keterbasan dari metode pencarian buta dalam memecahkan permasalahan yang ada, hal itu berhubungan dengan penggunaan waktu yang relatif besar dalam pemecahan masalahnya. Salah satu contoh pencarian buta adalah metode *Uniform Cost Search* (UCS).[3]

2.3.2 Metode Pencarian Heuristik

Fungsi heuristik dapat dipecahkan dengan menggunakan *euclidean. Euclidean* adalah fungsi heuristik yang diperoleh berdasarkan jarak langsung bebas hambatan seperti untuk mendapatkan panjang garis diagonal pada segitiga. Tapi sebelum mendapatkan hasil kedua titik harus direpresentasikan ke dalam koordinat dua dimensi (x, y) , sehingga dapat dihitung melalui rumus : [4]

$h_{1,2} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

$h_{1,2}$ = nilai heuristik antara simpul1 dan 2
x_1 = koordinat x titik 1
x_2 = koordinat x titik 2
y_1 = kordinat y titik 1
y_2 = kordinat y titik 2

2.4. Dynamic Weighting A* (DWA*)

Selain metode pencarian buta, terdapat beberapa pencarian lain yang dapat digunakan untuk Salah satu metode pencarian heuristik adalah *Dynamic Weighting* A*, asumsi untuk melakukan pencarian ke segala arah pada awal iterasi dan ketika *goal state* sudah dekat baru pencarian difokuskan kearah *goal state* merupakan karakteristik dari algoritma ini. Untuk itu digunakan pembobotan yang dinamis terhadap fungsi heuristik

$$f(n) = g(n) + w(n) * h(n)$$

Persamaan di atas merupakan fungsi heuristik dari algoritma DWA*, dengan g(n) merupakan jarak sebenarnya antara dua *state*, h(n) merupakan jarak perkiraan antara dua *state* (jarak garis lurus), dan w(n) merupakan bobot dinamis (w >=1). Nilai w(n) diperoleh dari perbandingan antara kedua jarak tersebut, dengan besar w(n) yang semakin mengecil ketika state semakin mendekati *Goal State*. Nilai dinamis dari algoritma yang lebih besar dari satu (w >= 1) dan semakin mengecil mendekati satu ketika semakin mendekati goal state. Dalam notasi matematika dapat dituliskan sebagai berikut:

$$W(n) = \frac{g(n)}{h(n)}$$

Nilai w(n) diperoleh dari rasio antara jarak sebenarnya dengan jarak perkiraan antara dua node. Bobot dinamis ini nilainya akan semakin kecil ketika state mendekati *Goal*, pengurangan nilai bobot dinamis dilakukan berdasarkan iterasi yang dilakukan selama proses mencari *Goal*. [3]

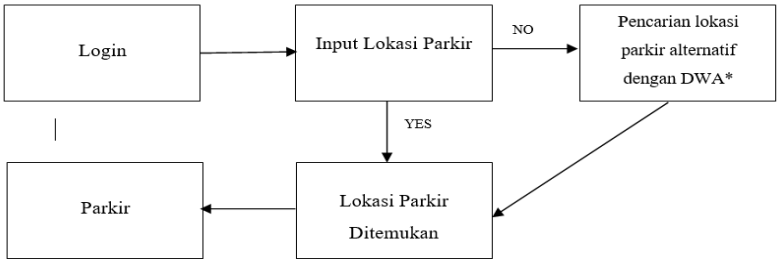
3. Perancangan Sistem

Sistem yang akan dibangun adalah sebuah aplikasi dari *Smart Parking System*, yaitu sebuah aplikasi yang menampilkan informasi mengenai keadaan lokasi parkir. Jika parkir tujuan penuh, maka sistem akan memberikan lokasi parkir alternatif yang terdekat (terpendek) dari parkir tujuan. User dapat melakukan aktivitas '*booking'* pada lokasi parkir tujuan dengan waktu yang telah ditentukan oleh sistem. Apabila *user* tidak sampai di lokasi parkir tujuan sesuai dengan waktu yang telah ditentukan makan secara otomatis lokasi parkir yang telah di *booking*

akan dilepas dan diberikan kepada *user* lain yang akan melakukan aktivitas parkir.

Fitur lain yang akan diberikan pada aplikasi ini yaitu fitur topup. Dengan fitur topup, user dapat mengisi saldo sebelum menggunakan layanan parkir. Saldo akan otomatis berkurang setelah *user* melakukan parkir sesuai dengan waktu yang telah *user* gunakan tanpa harus melakukan pembayaran secara manual setelah parkir. Sistem akan menerima inputan dari *server* berupa keadaan di lokasi parkir dan sistem juga akan memberikan *output* kepada *server* jika ada kendaraan yang akan melakukan aktivitas parkir. *Output* yang akan diberikan berisi informasi mengenai waktu dan status.

Untuk metoda penyelesaian masalah pencarian lokasi parkir alternatif akan menggunakan metode pencarian heuristik yaitu *Dynamic Weighting* A* (DWA*). Dengan menggunakan algoritma inilah lokasi parkir alternatif



Gambar 3. 1 Diagram blog sistem

3.4.2 Fungsi Heuristik

Algoritma *Dynamic Weighting* A* merupakan salah satu metode pencarian yang menggunakan fungsi heuristik. Sesuai dengan notasi matematis yang telah dijelaskan di bab sebelumnya, nilai dari DWA* (f(n)) terdiri dari penjumlahan antara g(n) yang merupakan jarak sebenarnya dengan hasil perkalian *weight* atau bobot dinamis dan h(n) yang merupakan jarak perkiraan. Nilai g(n) atau jarak sebenarnya diperoleh melalui proses perhitungan antara dua node atau dua koordinat yang terhubung melalui lintasan yang ada pada denah area parkir. Sedangkan nilai h(n) atau jarak perkiraan merupakan hasil proses perhitungan panjang garis lulus antara dua koordinat yang ada di denah area parkir.

3.4.3 Proses Pencarian Alternatif Parkir Terdekat

Lokasi parkir terdekat diperoleh dengan cara melakukan iterasi dengan metode DWA* terhadap setiap koordinat pada area parkir. Koordinat tempat parkir yang telah diproses melalui fungsi DWA* dan menghasilkan jarak terpendek akan dipilih menjadi solusi, yaitu tempat parkir yang memiliki jarak terdekat dengan parkir tujuan awal. Untuk rute terpendek diperoleh dari output fungsi DWA*. Setiap *node* hanya memiliki satu *parent* dan setiap simpul merupakan jarak yang terdekat yang terpilih dari tiap suksesornya. Melalui lintasan yang didapatkan dengan menggunakan koordinat, maka akan diperoleh alternatif parkir terdekat dari parkir tujuan awal.

3.4.4 Proses Penghitungan Waktu Proses DWA*

Untuk mengetahui performansi dari metode pencarian heuristik DWA*, dibutuhkan penghitungan waktu yang digunakan untuk menjalankan fungsi DWA* sampai fungsi tersebut selesai dan memberikan hasil. Proses penghitungan waktu dimulai dari saat fungsi DWA* dipanggil, dan diakhiri ketika rute terpendek menuju koordinat *goal* ditemukan. Hasil penghitungan ini akan ditampilkan pada sistem dalam satuan milisecond.

3.5 Teknis Pembangunan Sistem

Untuk menemukan alternatif parkir terdekat dari parkir tujuan, dilakukan melalui tahapan sebagai berikut:

1. Pembacahaan denah area parker
2. Menemukan lokasi parkir alternatif menggunakan metode DWA*

3.5.1 Pembacaan Denah Area Parkir

Untuk mendefinisikan ruang masalah pencarian, dibutuhkan sebuah denah area parkir yang merupakan informasi dasar dalam proses pencarian lokasi parkir. Sebelum menggunakan metode pencarian DWA*, dibutuhkan ruang masalah yang jelas. Dalam hal ini denah area parkir memiliki peranan sebagai ruang masalah. Melalui denah area parkir ini dapat diketahui setiap komponen pada area parkir, yaitu koordinat masing-masing area parkir, lintasan pada area parkir dan lokasi parkir yang tersedia (kuota). Informasi jarak antara satu lokasi dengan lokasi yang lain dapat diketahui dari koordinat masing-masing lokasi.

3.5.2 Menemukan Lokasi Parkir Alternatif Menggunakan Metode DWA*

Lokasi parkir alternatif terdekat diperoleh dengan menggunakan DWA* terhadap semoa *node* atau lokasi parkir yang tersedia. Lokasi parkir yang memili nilai terkecil atau berada pada posisi terdekat akan terpilih sebagai solusi. Namun dalam proses untuk menemukan jarak terdekat tersebut, tiap lokasi parkir akan diproses menggunakan metode DWA* yang akan menghasilkan *shortest path solution* untuk menemukan lokasi parkir alternatif tersebut. Untuk menemukan *goal*, digunakan dua array yaitu *Open* dan *Closed* yang akan diisi oleh *node-node* sesuai dengan posisinya. *Node Start* yang merupakan *node* pertama dimasukkan ke *Open*, namun karena hanya ada satu *node* maka *node Start* langsung dipindahkan ke *Closed*. Setelah itu semua suksesor dari *node Start* tadi dibangkitkan, lalu dimasukkan dalam node *Open*. Lakukan penghitungan biaya untuk semua suksesor tersebut menggunakan fungsi heuristik, *node* dengan biaya terkecil akan terpilih sebagai *Best Node* dan akan dipindahkan dari Open ke *Closed*. Sama seperti sebelumnya, semua suksesor dari *Best Node* dibangkitkan dan dimasukkan ke *Open*.

Untuk suksesor yang sudah berada di Open dilakukan pengecekan lebih lanjut untuk menentukan parentnya harus diubah atau tidak, jika ternyata parent lain memberikan hasil yang lebih kecil maka parent harus diubah dan nilai f dan g harus diupdate. Proses pengecekan suksesor tersebut diiterasi sampai *Goal* ditemukan atau Open habis. Dalam metode DWA*, nilai fungsi heuristik dipengaruhi oleh nilai w atau bobot dinamis yang diberikan. Nilai dinamis ini dikalikan dengan jarak heuristik, penentuan nilai dinamis ini harus tepat karena nilai dinamis yang salah akan membuat algoritma ini menjadi tidak optimal dan bahkan malah mencari ke arah yang salah. Dalam implementasinya nilai dinamis diperoleh dari rasio jarak sebenarnya dan jarak perkiraan. Hal ini sesuai dengan kondisi area parkir yang cenderung lintasannya berbentuk menyiku (belokan), dengan adanya nilai dinamis ini menjadikan DWA* memiliki kemampuan untuk memilih node atau state yang lebih potensial tanpa melakukan kesalahan arah. Dengan kata lain nilai dinamis ini juga dapat mempengaruhi kelengkungan pencarian, sehingga terhindar dari pencarian atau penelusuran yang menyimpang.

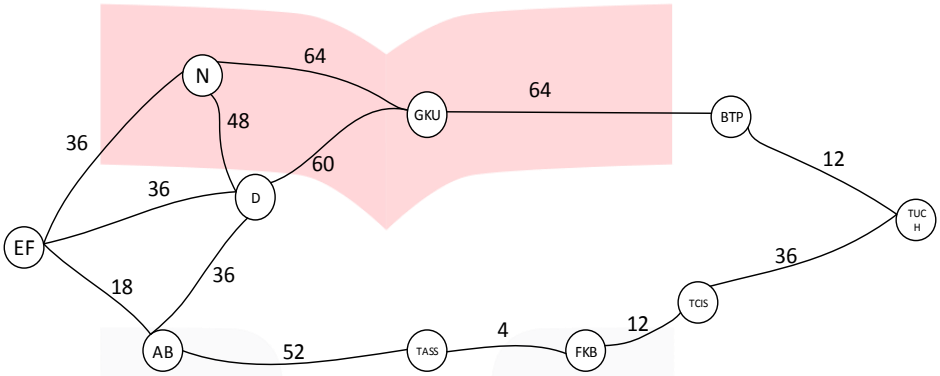
4. Pembahasan

Untuk mengetahui performansi sistem yang telah dirancang, maka perlu dilakukan pengujian terhadap sistem. Tujuan dari pengujian ini adalah sebagai berikut:

1. Mengetahui performansi sistem berdasarkan parameter akurasi yang didapatkan dari outputan sistem.
2. Mengetahui performansi dari DWA* dari sisi waktu operasi.
3. Mengetahui apakah sistem yang menggunakan DWA* ini dapat diimplementasikan pada *Smart Parking System* secara efektif.

4.1.2 Pencarian Parkir Alternatif

Setelah aplikasi menerima permintaan *user* untuk memberikan area parkir yang *user* minta, tetapi pada area tujuan *user* ternyata tidak ada slot yang tersedia maka proses pencarian parkir alternatif menggunakan DWA* mulai dijalankan seperti berikut:



Gambar 4. 2 Rute area parkir

Keterangan : parkir tujuan user adalah EF

4.2.1 Pengujian Sistem Berdasarkan *Output*

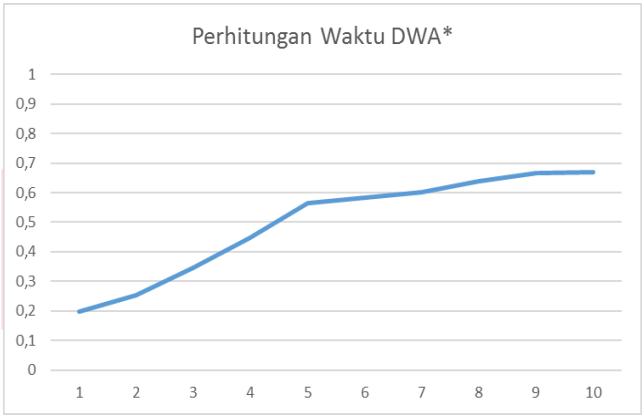
Pengujuian ini bertujuan untuk mengetahui sejauh mana kebenaran output yang dihasilkan oleh sistem alternatif parkir terdekat dengan menggunakan pencarian DWA* yang diimplementasikan ke dalam aplikasi. Pengujian ini membandingkan hasil metode pencarian yang diimplementasikan pada aplikasi dengan metode pencarian *Uniform Cost Search* (UCS) yang menggunakan nilai sebenarnya (g(n)) dalam proses pencarian. Skenario pada pengujian ini yaitu parkir tujuan adalah area gedung TUCH, EF dan EF. Kondisi area parkir tujuan tersebut tidak memiliki kuota kosong, sehingga akan dicari area alternatif parkir tujuan yang terdekat dengan area parkir tujuan.

Kuota pada *node* (area) yang terhubung dengan *node* tujuan awal akan dikondisikan selalu penuh sampai hanya satu saja *node* yang tersisa pada status *Closed*. Pada tabel berikut merupakan perbandingan hasil pengujian dari hasil pengujian tiga skenario dan menghasilkan sebanyak 27 data, maka metode UCS menghasilkan 6 *output* yang dinyatakan tidak sesuai. Sedangkan dengan menggunakan metode DWA* menghasilkan 2 *output* yang dinyatakan tidak sesuai. Sehingga nilai kebenaran dari metode UCS adalah 77.7% dengan nilai error 22.23% dan nilai kebenaran dari metode DWA* adalah 92.592% dengan nilai error 7.408%

4.2.2 Pengujian Sistem Berdasarkan Waktu Eksekusi

4.2.2.1 Pengujian Menggunakan *DWA**

Pengujian sistem terhadap waktu eksekusi dapat dilihat pada gambar di bawah. Pada gambar tersebut dapat dilihat waktu yang dibutuhkan oleh algoritma DWA* untuk menemukan alternatif parkir terdekat dari lokasi parkir tujuan. Waktu yang dibutuhkan untuk mencari lokasi parkir alternatif ke-10 dengan menggunakan DWA* hanya sebesar 0.67 milisekon. Waktu rata-rata DWA* dari iterasi 1- iterasi 10 adalah 0.49 milisekon.

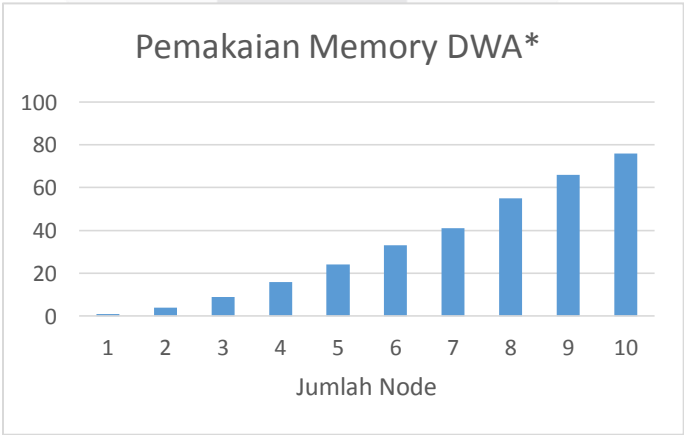


*Gambar 4. 1 Perhitungan waktu DWA**

Dengan waktu eksekusi yang cukup singkat ini, dapat diketahui bahwa DWA* dapat digunakan dalam *Smart Parking System* untuk menemukan lokasi alternatif parkir terdekat.

4.2.2 Analisis Jumlah Memori yang Digunakan

Dalam proses, DWA* memiliki dua buah array yaitu *Open* dan *Closed*. Jumlah *node* yang disimpan pada masing-maasing array merupakan jumlah *node* yang ada dalam kondisi *Open* dan *Closed*. Banyaknya memori yang digunakan juga dipengaruhi oleh banyaknya *node* yang merupakan percabangan (*branch*). Semakin banyak area (*node*) parkir maka akan semakin banyak memory yang digunakan. Dapat diketahui bahwa semalkim banyak jumlah node yang disimpan, maka semakin banyak memory yang digunakan.



*Gambar 4. 2 Pemakaian memory DWA**

5. Kesimpulan dan Saran

5.1 Kesimpulan

Dari hasil pengujian dan analisa yang telah dilakukan pada sistem, dapat diambil beberapa kesimpulan sebagai berikut :

1. Dengan menggunakan aplikasi *Smart Parking System* yang memiliki fungsionalitas yang dapat membantu pengemudi untuk mendapatkan informasi lokasi parkir dan menemukan alternatif parkir terdekat (jika terdapat parkir yang kosong) dengan menggunakan metode pencarian DWA*.
2. Nilai heuristik (h(n)) dan nilai dinamis (w(n)) sangat berpengaruh dalam menentukan *goal state* yang terbaik. Nilai heuristik yang tidak tepat akan membuat pencarian menjadi salah arah. Pada Aplikasi Smart Parking Sistem, DWA* menggunakan h(n) dan w(n) dalam proses perhitungannya untuk menentukan solusi area alternatif terdekat. Setelelah dilakukan pengujian didapatkan data output tidak sesuai 2 dari 27 data. Sehingga menghasilkan nilai error 7.408% dengan nilai kebenaran 92.592%.
3. Pengujian waktu eksekusi pencarian parkir alternatif menggunakan DWA* dengan waktu ekseskusi untuk 10 area parkir membutuhkan waktu rata-rata 0.49 milisekon dengan penggunaan memory yang lebih kecil. Waktu ekseskusi dan penggunaan memory akan bertambah seiring dengan pertambahan jumlah node pada lokasi parkir.

5.2 Saran

Untuk pengembangan lebih lanjut penulis memberikan beberapa saran, antara lain:

1. Untuk kedepannya dapat dibangun *Smart Parkir System* yang lebih utuh. Seperti dengan menambahkan *security* identitas kendaraan dan sistem keamanan aplikasi.
2. Dapat dikembangkan di semua *Operating System.*

Daftar Pustaka

[1] Pratama, I. P. (2014). *Smart City beserta Cloud Computing.* Bandung: Informatika.

[2] Kusumadewi. (2003). *Artificial Intelligence (Teknik dan Aplikasinya).* Yogyakarta: Graha Ilmu.

[3] Suyanto. (2007). *Artificial Intelligence (Searching, Reasoning dan Learning).* Bandung: Informatika Bandung.

[4] Russel, S., & Peter, N. (2003). *Artificial Intelligence.* New Jersey: Prentice Hall.

[5] F. Bacchus. & F. Kabanza (2000). *Using Temporal Logics to Express Search Control Knowledge for Planning, Artificial Intelligence.*