

MENGHITUNG RUTE TERPENDEK MENGGUNAKAN ALGORITMA A* DENGAN FUNGSI *EUCLIDEAN DISTANCE*

Kiki Setiawan¹, Supriyadin², Imam Santoso³, Roy Buana⁴

Jurusan Teknik Informatika, Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika

Jalan Radin Inten II Duren Sawit Jakarta

Email: ki2djoaz@gmail.com, supri170845@gmail.com, imanofani@yahoo.com, roy_buana@gmail.com

ABSTRAKS

Banyak cara yang dapat dilakukan untuk mencari pencarian rute dalam suatu perjalanan menuju ke tempat tujuan, antara lain dengan menggunakan peta manual ataupun peta digital. Pencarian rute terpendek dari jalan Arus Jati menuju kampus STMIK Nusa Mandiri di jalan Kramat Raya dapat ditempuh dengan dua pilihan rute. Untuk dapat menyelesaikan permasalahan pencarian rute terpendek, dapat digunakan suatu algoritma A* (A Star). Algoritma A* merupakan salah satu algoritma path finding yang umum digunakan. Dalam menemukan rute yang disebutkan diatas dapat menggunakan perhitungan algoritma A* dan metode Heuristik. Hasil yang didapat yaitu jarak terpendek yang bisa dilalui yaitu dengan jarak tempuh 19,92 KM yaitu jalur II.

Kata Kunci : Rute Terpendek, Path Finding, Algoritma A*, Heuristik.

1. PENDAHULUAN

Di era globalisasi perkembangan teknologi informasi (TI) semakin berkembang pesat. Hampir seluruh orang di dunia, menggunakan teknologi informasi tersebut, untuk mempermudah dan memberikan kenyamanan untuk melakukan segala aktivitas dalam kehidupan sehari-hari. Seperti contohnya: Telepon seluler sebagai alat komunikasi, internet di gunakan sebagai alat untuk mencari informasi-informasi penting, dan sebagainya. Kemajuan teknologi informasi tidak bisa di hindarkan dari kehidupan manusia karena kemajuan teknologi informasi berjalan seiring dengan kemajuan ilmu pengetahuan.

Akan tetapi, perkembangan teknologi informasi ini, juga membawa dampak bagi manusia. Dalam pencarian tempat pun sekarang sudah memakai peta elektronik yang salah satunya menggunakan bernama *Google Map*. Peta digital yang terdapat pada perangkat *smartphone* maupun tablet ini lebih mendapatkan sorotan karena sangat berfungsi ketika kita dalam perjalanan. Bukankah *Google Map* akan sangat berguna jika kita ingin mengetahui letak suatu daerah/tempat ataupun jalan menuju tempat tersebut ketika kita sedang dalam perjalanan? Bahkan aplikasi Maps ini juga dapat memperhitungkan waktu yang kita perlukan untuk mencapai lokasi tersebut.

Pencarian rute sangat erat kaitannya dengan pencarian suatu lokasi dalam sebuah pemetaan. Ketika seseorang ingin menuju ke suatu lokasi, orang tersebut akan menganalisa rute mana yang di anggap tepat dengan memakan waktu relatif singkat, dalam hal ini adalah perjalanan dari rumah peneliti menuju lokasi kampus STMIK Nusa Mandiri. Dengan kepadatan lalu lintas yang dikarenakan jumlah kendaraan di kota Jakarta yang semakin tak terkendali, mencari rute yang paling terdekat adalah solusi tepat bagi seorang pekerja, mahasiswa atau siapapun untuk mencapai tujuan tepat waktu.

2. LANDASAN TEORI

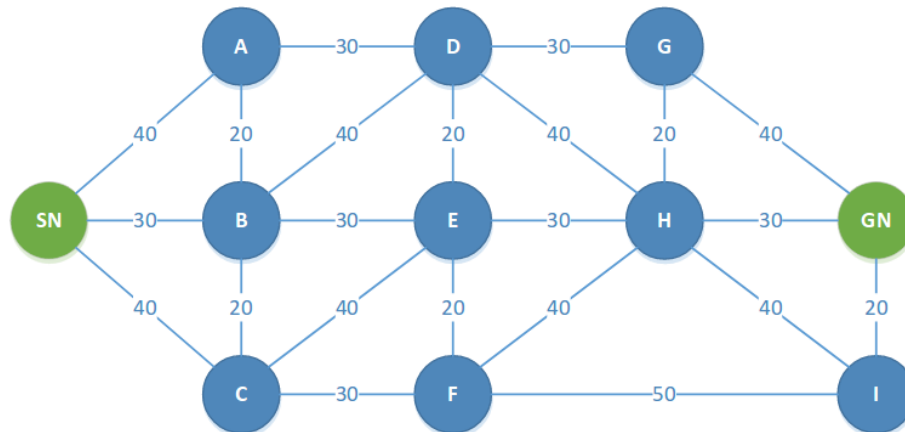
Pada tahun 1967 Bertram Raphael membuat perbaikan-perbaikan dramatis atas algoritma ini, tapi gagal menunjukkan keoptimalitasannya. Ia menyebut algoritma ini A2. Kemudian pada tahun 1968 Peter E. Hart memperkenalkan sebuah argumen yang membuktikan A2 optimal ketika menggunakan heuristik konsisten hanya dengan perubahan kecil. Pembuktiannya atas algoritma tersebut juga termasuk bagian yang menunjukkan bahwa algoritma A2 yang baru adalah algoritma terbaik. Maka dari itu, ia menamai algoritma baru ini dalam sintaksis *Kleene Star* untuk menjadi algoritma yang berawal dari A dan memasukkan semua nomor-nomor yang mungkin dari A*. Dalam sains komputer, A* (dibaca "A star") adalah algoritma komputer yang digunakan secara luas dalam mencari jalur (*pathfinding*) dan grafik melintang (*graph traversal*), proses plotting sebuah jalur melintang secara efisien antara titik-titik, disebut node (Suyanto, 2014).

2.1. Algoritma A*

Algoritma A Star atau A* adalah salah satu algoritma pencarian yang menganalisa input, mengevaluasi sejumlah jalur yang mungkin dilewati dan menghasilkan solusi. Algoritma A* adalah algoritma komputer yang digunakan secara luas dalam *graph traversal* dan penemuan jalur serta proses perencanaan jalur yang bisa dilewati secara efisien di sekitar titik-titik yang disebut *node* (Reddy, 2013).

Karakteristik yang menjelaskan algoritma A* adalah pengembangan dari “daftar tertutup” untuk merekam area yang dievaluasi. Daftar tertutup ini adalah sebuah daftar untuk merekam area berdekatan yang sudah dievaluasi, kemudian melakukan perhitungan jarak yang dikunjungi dari “titik awal” dengan jarak diperkirakan ke “titik tujuan” (Reddy, 2013).

Pada tahun 1964 Nils Nilsson menemukan heuristik berdasarkan pendekatan untuk menambah kecepatan pada Algoritma Dijkstra. Algoritma ini disebut dengan A1



Gambar 1. Graph Dijkstra

Pada gambar 1, suatu *graph* dengan *Node Awal* SN dan *Node Tujuan* GN terhubung dengan *node - node* lain oleh *edge* yang memiliki besaran yang berbeda. Jika ditelusuri, terdapat banyak kombinasi rute yang dapat dilalui untuk menuju *node* tujuan. Bisa dikatakan dari *graph* tersebut, setiap *node* akan memberikan solusi arah menuju *node* tujuan. A* merupakan bentuk yang paling dari *Best First Search* (BFS). A* mengevaluasi *node* dengan menggabungkan $g(n)$, yaitu *cost* untuk mencapai *node*, dan $h(n)$, yaitu *cost* yang diperlukan dari *node* untuk mencapai tujuan, dalam notasi matematika dituliskan sebagai:

$$f(n)=g(n)+h(n) \quad (1)$$

Dimana:

$f(n)$ = Biaya evaluasi

$g(n)$ = Biaya yang sudah dikeluarkan dari keadaan awal sampai keadaan n

$h(n)$ = Estimasi biaya untuk sampai pada suatu tujuan mulai dari n

2.2. Path Finding

Path Finding merupakan salah satu materi yang sangat penting didalam *Artificial Intelligence*. Path Finding biasanya digunakan untuk menyelesaikan masalah pada sebuah *graph*. Dalam matematika *graph* merupakan himpunan titik-titik atau biasa disebut dengan *node* yang terhubung oleh *edge*. *Edge* yang menghubungkan setiap *node* merupakan suatu *vektor* yang memiliki arah dan besaran tertentu. Untuk dapat menemukan jalan dari *Node Awal* menuju *Node Tujuan*, dilakukan penelusuran terhadap *graph* tersebut. Penelusuran biasanya dilakukan dengan mengikuti arah *edge* yang menghubungkan antar *node*.

2.3. Pencarian Heuristik

Pencarian Heuristik Kata Heuristic berasal dari sebuah kata kerja bahasa Yunani, *heuristic*, yang berarti “Mencari atau Menemukan”. Dalam dunia pemrograman, sebagian orang menggunakan kata *heuristic* sebagai lawan kata dari algoritma, dimana kata *heuristic* ini diartikan sebagai suatu proses yang mungkin dapat menyelesaikan suatu masalah tetapi tidak ada jaminan bahwa solusi yang dicari selalu dapat ditentukan.

Didalam mempelajari metode-metode pencarian ini, kata *heuristic* diartikan sebagai suatu fungsi yang memberikan suatu nilai berupa biaya perkiraan (estimasi) dari suatu solusi. Teknik pencarian *heuristic* (*heuristic searching*) merupakan suatu strategi untuk melakukan proses pencarian secara selektif dan dapat memandu proses pencarian yang memiliki kemungkinan sukses paling besar, namun dengan kemungkinan mengorbankan kelengkapan (*completeness*). Untuk menerapkan pencarian *heuristic* diperlukan suatu fungsi *heuristic*. Fungsi *heuristic* adalah aturan-aturan yang digunakan untuk mendapatkan solusi yang diinginkan.

Heuristik jarak ditambah biaya adalah penjumlahan dari dua fungsi:

- fungsi jalur biaya, di mana biayanya dihitung dari node awal hingga node saat ini (biasanya dinotasikan $g(x)$).
- dan “estimasi heuristik” yang dapat diterima (*admissible*) dari jarak menuju *goal* (biasanya dinotasikan $h(x)$).

2.4. Fungsi Euclidean Distance

Euclidean distance adalah perhitungan jarak dari 2 buah titik dalam *Euclidean space*. *Euclidean space* diperkenalkan oleh *Euclid*, seorang matematikawan dari Yunani sekitar tahun 300 B.C.E. untuk mempelajari hubungan antara sudut dan jarak. Euclidean ini berkaitan dengan *Teorema Pythagoras* dan biasanya diterapkan pada 1, 2 dan 3 dimensi. Tapi juga sederhana jika diterapkan pada dimensi yang lebih tinggi.

Pada 1 dimensi.

Semisal ingin menghitung jarak Euclidean 1 dimensi. Titip pertama adalah 4, titik

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2)$$

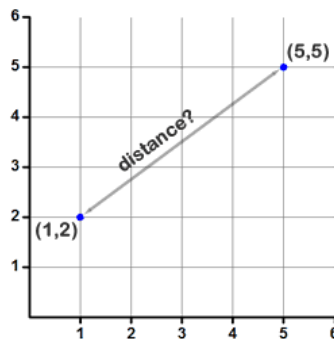
Rumus Euclidean

Sehingga dari Formula diatas kita dapat implementasi menjadi:

$$\text{Jarak} = \sqrt{(\text{Lat}_1 - \text{Lat}_2)^2 + (\text{Long}_1 - \text{Long}_2)^2} \quad (3)$$

Rumus Jarak Euclidean

kedua adalah -10. Caranya adalah kurangkan -10 dengan 4, sehingga menghasilkan -14. Cari nilai absolut dari nilai -14 dengan cara memangkatkannya sehingga mendapat nilai 196. Kemudian diakarkan sehingga mendapatkan nilai 14. Sehingga jarak euclidean dari 2 titik tersebut adalah 14. **Pada 2 dimensi.**



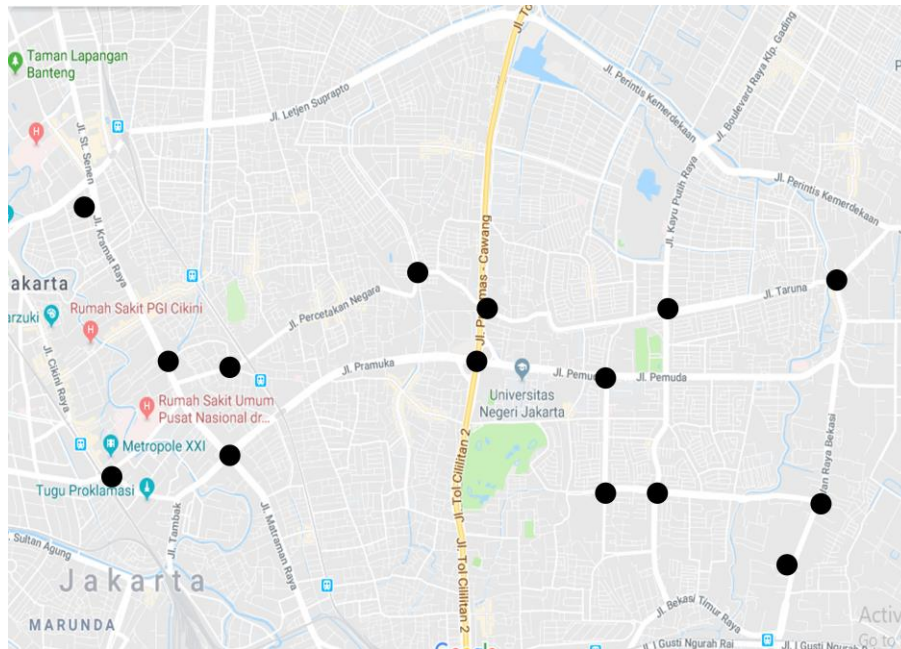
Gambar 2. Contoh 2 dimensi

Caranya hampir sama. Misalkan titik pertama mempunyai kordinat (1,2). Titik kedua ada di kordinat (5,5). Caranya adalah kurangkan setiap kordinat titik kedua dengan titik yang pertama. Yaitu, $(5-1, 5-2)$ sehingga menjadi (4,3). Kemudian pangkatkan masing-masing sehingga memperoleh (16,9). Kemudian tambahkan semuanya sehingga memperoleh nilai $16+9 = 25$. Hasil ini kemudian diakarkan menjadi 5. Sehingga jarak euclideannya adalah 5. Hasil perhitungan (*Jarak*) diatas masih dalam satuan *decimal degree* (sesuai dengan format longlat yang dipakai) sehingga untuk menyesuaikan perlu dikalikan dengan **111.319 km** (1 derajat bumi = 111.319 km).

3. PENDAHULUAN

3.1. Pengumpulan Data

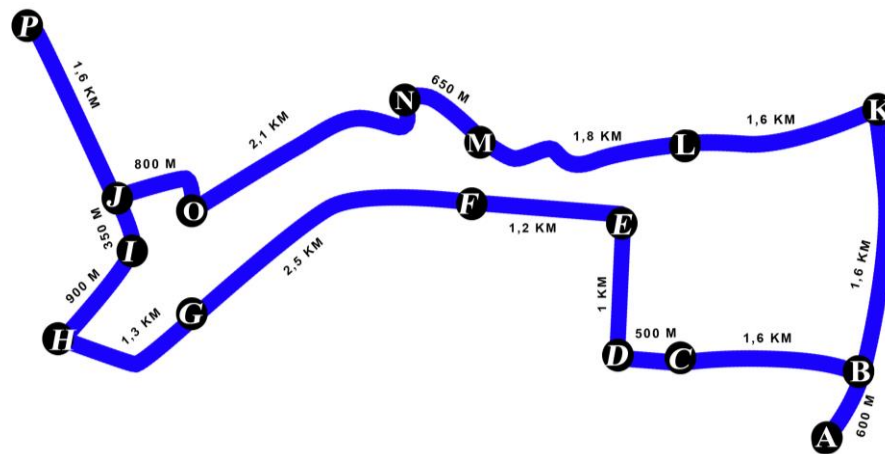
Seperti yang sudah peneliti sampaikan dipenjelasan sebelumnya, peneliti mengambil studi kasus jarak antara Jalan Arus Jati (Rumah) sampai dengan kampus STMIK Nusa Mandiri. Agar didapat hasil pengukuran yang lebih akurat, peneliti menggunakan software pendukung *Google Earth* untuk mengetahui jalur-jalur yang akan dilalui. Tujuh *node* dihasilkan dari proyeksi *Google Earth*, dimana pengambilan *nodenya* berdasarkan persimpangan jalan.



Gambar 3. Google earth

Data *google maps* tidak menjadi acuan pasti dalam pengujian karena daerah yang menjadi bahan penelitian termasuk daerah yang padat lalu lintasnya, dan masih dalam tahap pembangunan.

Tempat tersebut direpresentasikan sebagai *node*, sedangkan nama jalan direpresentasikan sebagai *path*. Panjang jalan didapatkan menggunakan *Distance Measurement Tools* dari *google maps* tersebut. Persimpangan-persimpangan yang ada di sepanjang jalan tersebut juga direpresentasikan sebagai *node*, dengan mempertimbangkan posisinya terhadap node lain, sehingga semua node dapat terhubung.



Gambar 4. Titik jalur

Data jalur – jalur yang dipilih, didapat dari *google maps*:

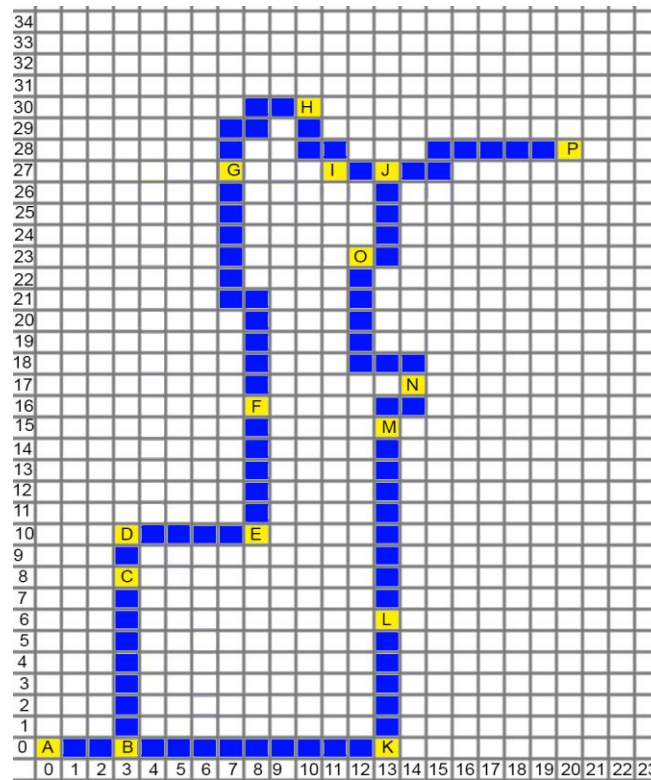
Jalur 1

- A. Arus Jati
- B. Jalan Raya Bekasi – Jatinegara Kaum
- C. Cipinang baru utara – Pegambiran
- D. Cipinang baru timur – Balai pustaka
- E. Balai Pustaka - Pemuda
- F. Pemuda - Pramuka

- G.Pramuka – Proklamasi
- H.Proklamasi – Diponegoro
- I. Diponegoro – Salemba
- J. Salemba – Kramat
- P. STMIK Nusa Mandiri

Jalur II

- A. Arus Jati
- B. Raya Bekasi – Jatinegara kaum
- K. Jatinegara Kaum – Jalan Taruna
- L. Jalan Taruna – Kayu Putih
- M. Kayu Putih – Haji Ten
- N. Haji Ten – Percetakan Negara
- O. Percetakan Negara – Paseban Raya
- J. Paseban Raya – Kramat
- P. STMIK Nusa Mandiri



Gambar 5. Titik indeks

Setiap Indeks mewakili jarak 200 meter.

- A. Arus Jati (0,0)
- B. Jalan Raya Bekasi – Jatinegara Kaum (0,3)
- C. Cipinang baru utara – Pegambiran (8,3)
- D. Cipinang baru timur – Balai pustaka (10,3)
- E. Balai Pustaka – Pemuda (10,8)
- F. Pemuda – Pramuka (16,8)
- G. Pramuka – Proklamasi (27,7)
- H. Proklamasi – Diponegoro (30,10)
- I. Diponegoro – Salemba (27,11)
- J. Salemba – Kramat (27,13)

- K. Jatinegara Kaum – Jalan Taruna (0,13)
- L. Jalan Taruna – Kayu Putih (6,13)
- M. Kayu Putih – Haji Ten (15,13)
- N. Haji Ten – Percetakan Negara (17,13)
- O. Percetakan Negara – Paseban Raya (23,12)
- J. Paseban Raya – Kramat (27,13)
- P. STMIK Nusa Mandiri (28,20)

3.2. Menghitung Heuristik

Langkah selanjutnya menghitung jarak dari dua titik indeks, Rumus jarak dua titik adalah sebagai berikut:

$$d(x, y) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (4)$$

Dengan menggunakan rumus di atas, maka perhitungan dari semua titik dapat dilihat sebagai berikut:

Tabel 1. Jalur/Rute 1

No	Titik Indeks Jalur/Rute 1	Hasil
1	A (0,0) ke B (0,3)	3
2	B (0,3) ke C (8,3)	8
3	C (8,3) ke D (10,3)	2
4	E (27,20) ke F (29,27)	7,2
5	D (10,3) ke E (10,8)	5
6	E (10,8) ke F (16,8)	6
7	F (16,8) ke G (27,7)	11
8	G (27,7) ke H (30,10)	4,2
9	H (30,10) ke I (27,11)	3,2
10	I (27,11) ke J (27,13)	2

Tabel 2. Jalur/Rute 2

No	Titik Indeks Jalur/Rute 2	Hasil
1	A (0,0) ke B (0,3)	3
2	B (0,3) ke K (0,13)	10
3	K (0,13) ke L (6,13)	6
4	L (6,13) ke M (15,13)	9
5	M (15,13) ke N (17,14)	2,2
6	N (17,14) ke O (23,12)	6,3
7	O (23,12) ke J (27,13)	4,1
8	J (27,13) ke P (28,20)	4,1

3.3. Langkah-langkah pencarian Alogritma A*.

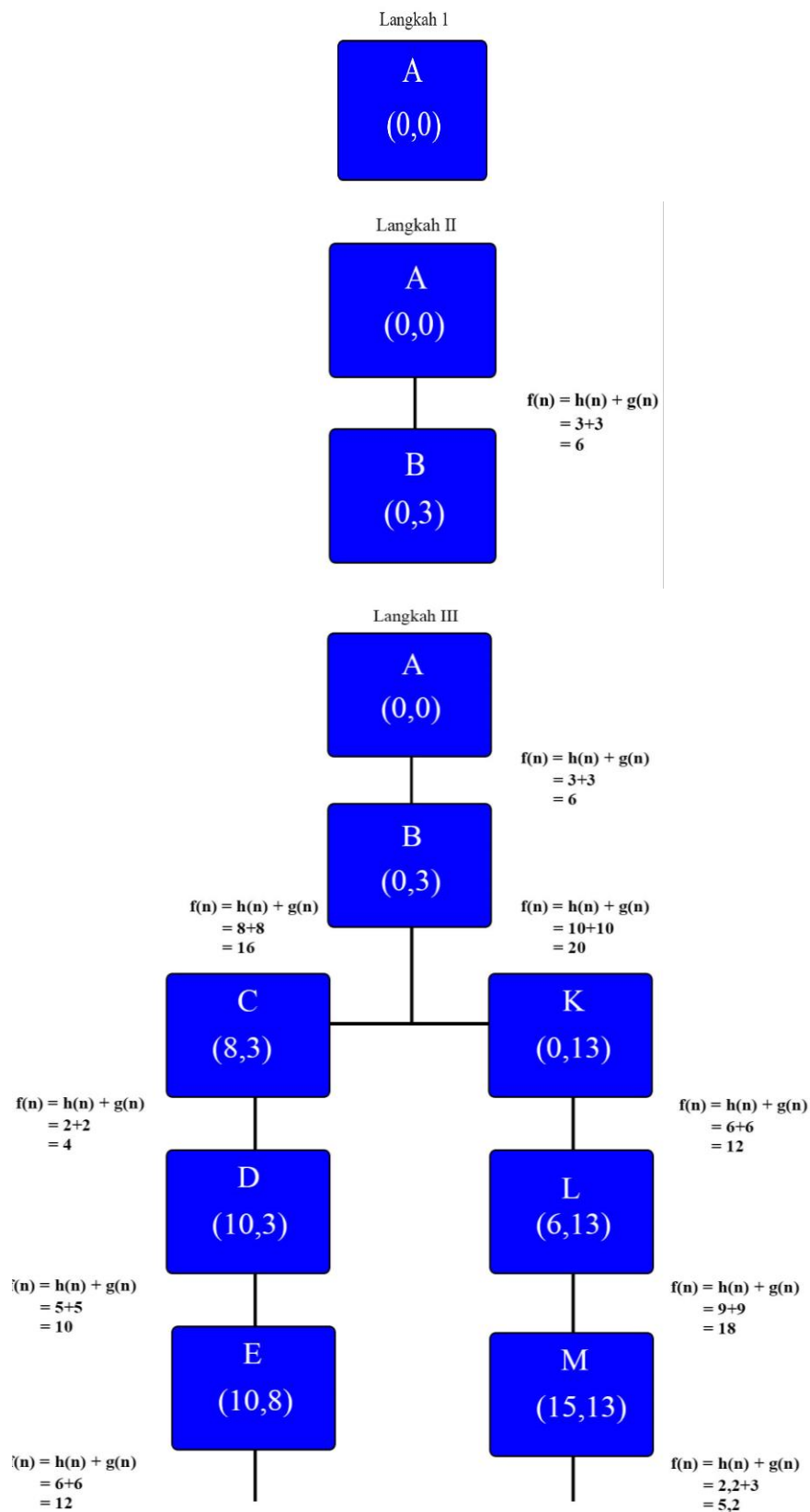
Setelah nilai heuristik dari masing-masing *node* didapat, maka langkah selanjutnya kita akan mencari $f(n)$ menggunakan algoritma A* dengan rumus:

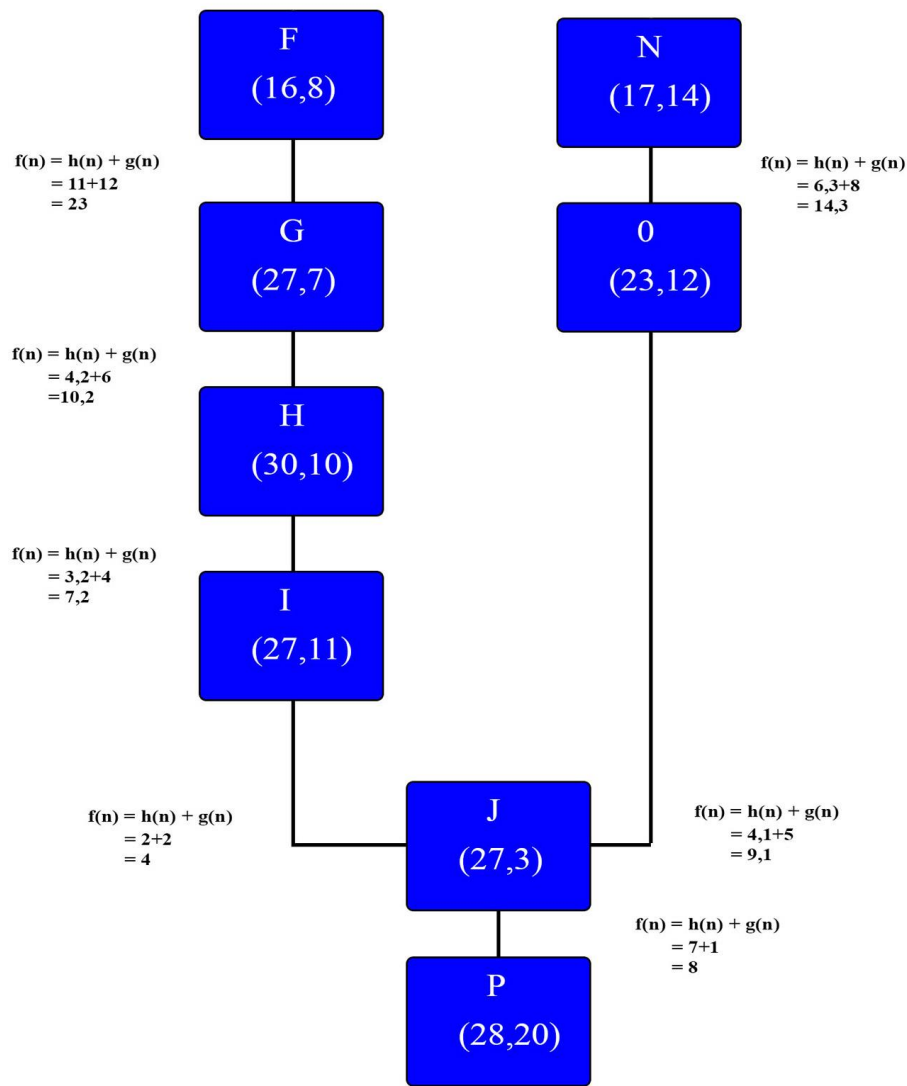
$$f(n) = h(n) + g(n)$$

dimana,

$h(n)$ = Nilai heuristik antar Koordinat

$g(n)$ = Jarak Koordinat ke titik tujuan





Gambar 6. Langkah-langkah pencarian

3.4. Perbandingan

Setelah menghitung heuristik serta melakukan langkah-langkah pencarian menggunakan algoritma A*, maka $f(n)$ total yang didapat dari:

Jalur I adalah 107,5

Karena satu titik koordinat mewakili 200 meter maka jarak sebenarnya dalam meter adalah :

$107,5 \times 200 = 21500$ Meter

Dalam satuan Kilo Meter menjadi = 21,5 KM

jalur yang dilalui adalah A-B-C-D-E-F-G-H-I-J-P.

Jalur II adalah 99,6

Karena satu titik koordinat mewakili 200 meter maka jarak sebenarnya dalam meter adalah :

$99,6 \times 200 = 19920$ Meter

Dalam satuan Kilo Meter menjadi = 19,92 KM

jalur yang dilalui adalah A-B-K-L-M-N-O-J-P.

4. KESIMPULAN

Berdasarkan dari hasil penelitian yang telah dilakukan dalam membahas pencarian rute terpendek Jalan Arus jati menuju Kampus STMIK Nusa Mandiri dengan menggunakan perhitungan algoritma A star (A*) dan metode Heuristik. Serta dengan dukungan pengumpulan data yang didapat dari berbagai sumber, termasuk dengan menggunakan aplikasi *Google Maps* dalam penentuan jalur yang akan dipilih oleh peneliti. Maka jarak terpendek yang bisa dilalui adalah jalur II, yaitu jalur A-B-K-L-M-N-O-J-P dengan jarak tempuh 19,92 KM. Hasil yang didapat dari penelitian di atas, hanya merupakan penghitungan jarak tempuh saja, dengan tidak mempertimbangkan bobot yang lebih dalam lagi.

Saran untuk penelitian selanjutnya, agar dicoba untuk melakukan penghitungan estimasi waktu tempuh atau lama perjalanan dengan mempertimbangkan beberapa unsur, antara lain tingkat kemacetan yang kemungkinan terjadi ataupun jumlah *traffic light* sepanjang perjalanan yang juga mungkin mempengaruhi lama/waktu perjalanan.

PUSTAKA

- Akbar Juang Saputra. *Penerapan Algoritma A* pada Google Map*. Jurnal Teknik Informatika Institut Teknologi Bandung.
- Aniket Chandak , Rutika Bodhale , Raveena Burad. 2016. *Optimal shortest path using HAS, A star and Dijkstra algorithm*. Journal Department of Computer Engineering, K.K.Wagh Institute of Engineering Education.& Research, Savitribai Phule Pune University.
- Eric A. Hansen, Shlomo Zilbersteinb. *LAO*: A heuristic search algorithm that finds solutions with loops*. Journal Computer Science Department, Mississippi State University, Mississippi State, MS 39762, USA.
- Karishma Talan, G R Bamnote. *Shortest Path Finding Using a Star Algorithm and Minimum weight Node First Principle*. Journal PG Student, Dept. of CSE, Prof. Ram Meghe Institute of Technology and Research Badnera, Maharashtra, India.
- Pramono Andy. 2015. Algoritma Pathfinding A* pada Game RPG Tanaman Higienis. *Jurnal Edukasi dan penelitian Informatika. (JEPIN) Vol. 1, No. 2*.
- Rakhmat Kurniawan. R., ST, M.Kom, Yusuf Ramadhan Nasution, M.Kom. *Penerapan Algoritma A* (A Star) Sebagai Solusi Pencarian Rute Terpendek Pada Maze*. Jurnal Program Studi Ilmu Komputer, Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara.
- Reddy, H., 2013. Path Finding-Dijkstra's and A* Algorithm's. [Online] Available at: <http://cs.indstate.edu/hgopireddy/newalg.html> [Diakses pada 25 Desember 2017].
- Suyanto, 2014. Artificial Intelligence: Searching, Reasoning, Planning, Learning. 2nd ed. Bandung: Informatika
- Veronica Mutiana, Fitria Amastini, Noviana Mutiara. *Optimasi Pencarian Jalur dengan Metode A-Star Studi Kasus: Area Gading Serpong, Tangerang*. Jurnal Teknik Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia.



Kiki Setiawan. Memperoleh gelar Sarjana Komputer (S.Kom) pada tahun 2011, dengan mengambil jurusan Teknik Informatika di Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika (STIKOM CKI). Saat ini sedang melanjutkan program Pasca Sarjana Magister Ilmu Komputer di STMIK Nusa Mandiri Jakarta, staff manajemen di STIKOM CKI, dan juga menjadi tenaga pengajar di kampus STIKOM CKI bidang Desain grafis.



Supriyadin. Memperoleh gelar Sarjana Komputer (S.Kom) pada tahun 2013, dengan mengambil jurusan Teknik Informatika di Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika (STIKOM CKI). Saat ini sedang melanjutkan program Pasca Sarjana Magister Ilmu Komputer di STMIK Nusa Mandiri Jakarta dan juga salah satu senior web developer di rumah123.com.



Imam Santoso. Memperoleh gelar Sarjana Komputer (S.Kom) pada tahun 2006, dengan mengambil jurusan Sistem Informasi di Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika (STIKOM CKI). Saat ini sedang melanjutkan program Pasca Sarjana Magister Ilmu Komputer di STMIK Nusa Mandiri Jakarta dan juga salah satu pengajar di STIKOM CKI.



Roy Buana. Memperoleh gelar Sarjana Komputer (S.Kom) pada tahun 2013, dengan mengambil jurusan Teknik Informatika di Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika (STIKOM CKI). Saat ini sedang melanjutkan program Pasca Sarjana Magister Ilmu Komputer di STMIK Nusa Mandiri Jakarta dan Pemilik CV. Mitra Buana Solusindo