

## MENENTUKAN RUTE TERPENDEK DENGAN MEMANFAATKAN METODE HEURISTIK BERBASIS ALGORITMA A\*

Mohammad Taufiq<sup>✉</sup>, Amin Suyitno, Dwijanto

Jurusan Matematika, FMIPA, Universitas Negeri Semarang, Indonesia  
Gedung D7 Lt.1, Kampus Sekaran Gunung pati, Semarang 50299

### Info Artikel

*Sejarah Artikel:*

Diterima 28 Agustus 2015  
Disetujui 23 Maret 2019  
Dipublikasikan 1 April 2019

*Keywords:*

Algorithm A\*  
Goods Distribution  
Shortest Route

### Abstrak

Penelitian ini mengkaji tentang sebuah permasalahan untuk mencari solusi terkait penentuan rute terpendek yang harus ditempuh. Tujuan dari penelitian ini adalah untuk mengetahui dasar-dasar algoritma A\* yang berguna untuk meneliti rute pendistribusian bahan baku berupa kacang tanah dari PT. Dua Kelinci ke beberapa daerah di Kabupaten Pati. Penelitian ini menggunakan bantuan program MATLAB. Data yang dibutuhkan berupa tempat-tempat produksi kacang tanah yang diperoleh dari kantor PT. Dua Kelinci. Selanjutnya, peneliti melakukan pencarian jarak dengan menggunakan bantuan Google Maps. Data tersebut dianalisis dengan menggunakan penghitungan manual yang kemudian hasilnya dibandingkan dengan hasil penghitungan yang menggunakan program MATLAB. Dari kedua cara penghitungan data tersebut, jarak terpendek pendistribusian kacang tanah di Kabupaten Pati adalah 218,4 km. Rute nya adalah sebagai berikut: PT. Dua Kelinci – Kecamatan Margorejo – Kecamatan Pati – Kecamatan Tayu – Kecamatan Tlogowungu – Kecamatan Gembong – Kecamatan Tambakkromo – Kecamatan Gunung Wungkal – Kecamatan Trangkil – Kecamatan Puncakwangi – PT. Dua Kelinci. Analisis penghitungan dengan cara manual maupun MATLAB menghasilkan jarak tempuh dan rute yang sama untuk mencapai semua titik dengan jarak minimal.

### Abstract

This research studies a problem solution related to the determination of the shortest route to pass. It aims to find out the basics of algorithm A\* due to the analysis of the shortest route to distribute peanuts materials from Dua Kelinci Company to the certain subdistricts in Pati Regency. This analysis uses the help of MATLAB program. The collected data is about the places of peanuts production. It was taken from the office of Dua Kelinci Company. After collecting the data, the mileage between the company and the distribution places was taken using Google Maps application. The data is analysed by using manual calculation. The result of it is then compared with the result of MATLAB calculation. The analysis using both ways of calculation shows that the shortest mileage to distribute peanuts materials in Pati Regency is 218,4 km. The route is described as follows: Dua Kelinci Company – Margorejo Subdistrict – Pati Subdistrict – Tayu Subdistrict – Tlogowungu Subdistrict – Gembong Subdistrict – Tambakkromo Subdistrict – Gunung Wungkal Subdistrict – Trangkil Subdistrict – Puncakwangi Subdistrict – Dua Kelinci Company. Both manual and MATLAB calculation result the same mileage and route to reach all points with the minimum distance.

<sup>✉</sup>Alamat korespondensi:

Mohammadtaufiq737@gmail.com

## PENDAHULUAN

Matematika adalah cabang ilmu pengetahuan yang sangat penting dan sangat berperan dalam perkembangan dunia. Matematika dibandingkan dengan disiplin-disiplin ilmu yang lain mempunyai karakteristik tersendiri. Pentingnya matematika tidak lepas dari perannya dalam segala jenis dimensi kehidupan. Selain itu, matematika juga seringkali dibutuhkan untuk menunjang eksistensi ilmu-ilmu lain seperti fisika, kimia, astronomi, biologi, ekonomi dan lain sebagainya. Tidak heran mengapa matematika dijuluki "*mathematic is a queen, but also a servant*", matematika sebagai ratu ilmu, tetapi juga sekaligus pelayan untuk ilmu-ilmu lain (Bell, 1952). Matematika dikatakan sebagai ratu ilmu karena matematika dapat tumbuh dan berkembang untuk dirinya sendiri sebagai suatu ilmu tanpa adanya bantuan dari ilmu lain.

Persoalan lintasan terpendek yaitu menemukan lintasan terpendek antara dua atau beberapa simpul lebih yang berhubungan. Persoalan mencari lintasan terpendek di dalam graf merupakan salah satu persoalan optimasi. Persoalan ini biasanya direpresentasikan dalam bentuk graf. Graf yang digunakan dalam pencarian lintasan terpendek atau *shortest path* adalah graf berbobot (*weighted graph*), yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot. Bobot pada sisi graf dapat menyatakan jarak antar kota, waktu pengiriman pesan, ongkos pembangunan, dan sebagainya.

Asumsi yang kita gunakan di sini adalah bahwa semua bobot bernilai positif. Kata "terpendek" tidak selalu diartikan secara fisik sebagai panjang minimum, sebab kata "terpendek" berbeda-beda maknanya bergantung pada tipikal persoalan yang akan diselesaikan. Namun, secara umum "terpendek" berarti meminimalisasi bobot pada suatu lintasan di dalam graf. Misalkan titik pada graf dapat mewakili kota, sedangkan sisi menyatakan jalan yang menghubungkan dua buah kota. Bobot sisi pada graf dapat menyatakan jarak antara dua buah kota atau rata-rata waktu tempuh antara dua buah kota. Apabila terdapat lebih dari satu lintasan dari kota A ke kota B, maka persoalan lintasan terpendek di sini adalah menentukan jarak terpendek atau waktu tersingkat dari kota A ke kota B.

Misalkan titik pada graf dapat mewakili terminal komputer atau titik komunikasi dalam suatu jaringan, sedangkan sisi menyatakan saluran komunikasi yang menghubungkan dua buah terminal. Bobot pada graf dapat menyatakan biaya pemakaian saluran komunikasi antara dua buah terminal, jarak antara dua buah terminal, atau waktu pengiriman pesan (message) antara dua buah terminal. Persoalan lintasan terpendek di sini adalah menentukan jalur komunikasi terpendek antara dua buah terminal komputer. Lintasan terpendek akan menghemat waktu pengiriman pesan dan biaya komunikasi.

Ada beberapa macam persoalan lintasan terpendek, antara lain: (a) Lintasan terpendek antara dua buah titik tertentu (*a pair shortest path*). (b) Lintasan terpendek antara semua pasangan titik (*all pairs shortest path*). (c) Lintasan terpendek dari titik tertentu ke semua titik yang lain (*single-source shortest path*). (d) Lintasan terpendek antara dua buah titik yang melalui beberapa titik tertentu (*intermediate shortest path*).

Permasalahan penentuan rute biasanya merupakan permasalahan *NP-Hard Problem* dimana penyelesaian dengan metode *exact* seringkali akan memakan waktu yang cukup lama untuk menyelesaikannya. Karena sebab inilah banyak para ahli yang merancang penyelesaian suatu problem dengan menggunakan metode heuristik. Metode Heuristik adalah teknik yang dirancang untuk memecahkan masalah yang mengabaikan apakah solusi dapat dibuktikan benar, tapi yang biasanya menghasilkan solusi yang baik atau memecahkan masalah yang lebih sederhana yang mengandung atau memotong dengan pemecahan masalah yang lebih kompleks. Metode Heuristik ini bertujuan untuk mendapatkan performa komputasi atau penyederhanaan konseptual, berpotensi pada biaya keakuratan atau presisi. Metode heuristik ada dua jenis yakni metode heuristik sederhana dan *metaheuristik*. Metode heuristik contohnya adalah *A star*, *cheapest insertion*, *Priciest Insertion*, *Nearest insertion*, *Farthest Insertion*, *Nearest addition* dan *Clarke and Wright Saving Method*.

Algoritma A\* adalah salah satu algoritma pencarian yang cukup populer di kalangan pemrogram. Algoritma ini memberikan solusi yang cukup bagus dalam pembuatan perangkat lunak. Namun tidak banyak yang sudah mengaplikasikan algoritma A\* untuk permasalahan kehidupan sehari-hari, contohnya seperti permasalahan pemasaran hasil suatu *product*. Apakah algoritma A\* juga bagus dalam menyelesaikan masalah dalam kehidupan sehari-hari ataukah hanya untuk menyelesaikan masalah dalam pembuatan perangkat lunak saja.

PT. Dua Kelinci merupakan salah satu perusahaan yang bergerak dalam bidang penyedia makanan terkemuka di Indonesia. Produk perusahaan ini terkenal dengan merek Dua Kelinci. Perjalanan perusahaan ini dimulai sejak tahun 1972 di Surabaya. Berawal dari usaha re-packing kacang garing yang berlabel "Sari Gurih" yang berlogo gambar dua kelinci. Dengan berkembangnya bisnis, pada tanggal 15 Juli 1985, didirikanlah PT. Dua Kelinci yang kini menjelma menjadi produsen kacang terkemuka di Indonesia dengan menerapkan sistem manajemen kualitas produk berstandar internasional. Usaha re-packing kacang ini yang didirikan oleh Bapak Ho Sie Ak dan Ibu Lauw Bie Giok di Pati, Jawa Tengah merupakan cikal bakal tumbuhnya industri kacang garing besar di Indonesia. Selain dipasarkan di

Indonesia, perusahaan ini juga menjual produknya di mancanegara seperti, Eropa, China, Hong Kong, Thailand, Arab Saudi, Filipina, Singapur, Malaysia, Brunei Darussalam, Kanada dan Amerika Serikat. Pemasok bahan utama dari perusahaan ini adalah petani-petani kacang tanah dari desa di kabupaten Pati.

Lahirnya teori graf pertama kali dikenalkan oleh *Leonard Euler* seorang matematikawan berkebangsaan *Swiss* pada tahun 1736 melalui tulisan *Euler* yang berisi tentang upaya pemecahan masalah jembatan *Konigsberg* yang sangat terkenal di Eropa. Secara Formal definisi graf adalah sebuah graf  $G$  yang berisikan dua himpunan yaitu himpunan berhingga tak kosong  $V(G)$  dari obyek-obyek yang disebut titik dan himpunan berhingga (mungkin kosong)  $E(G)$  merupakan pasangan tak berurutan dari titik-titik di  $V(G)$  (Budayasa, 2007: 1).

Jenis-jenis graf berdasarkan ada tidaknya gelung atau sisi ganda dibedakan menjadi dua (1) Graf sederhana yaitu graf yang tidak mengandung sisi ganda dan gelung. (2) Graf tak sederhana yaitu graf yang mengandung sisi ganda atau gelung.

Jenis-jenis graf berdasarkan orientasi arah pada sisi juga dibedakan menjadi dua (1) Graf berarah yaitu graf yang setiap sisinya diberikan orientasi arah. (2) Graf tak berarah yaitu graf yang tidak mempunyai orientasi arah.

Ada beberapa contoh graf yang termasuk graf khusus. Seperti graf lintasan, graf sikel, graf *star*, graf *double star*, graf *sun* dan lain lain (Muttaqien, Mulyono, Suyitno, 2013: 86). Misalkan  $G$  adalah sebuah graf. Sebuah jalan (*walk*) di  $G$  adalah sebuah barisan berhingga (tak kosong)  $W = (v_0, e_1, v_1, e_2, v_2, \dots, e_k, v_k)$  yang suku-sukunya bergantian sedemikian hingga  $v_{i-k}$  dan  $v_i$  adalah titik-titik akhir sisi  $e_i$  untuk  $1 \leq i \leq k$ . Dikatakan  $W$  adalah sebuah jalan dari titik  $v_0$  ke titik  $v_k$ , atau jalan  $(v_0, v_k)$ . Titik  $v_0$  dan titik  $v_k$  berturut-turut disebut titik awal dan titik akhir  $W$ . Sedangkan titik-titik  $v_1, v_2, \dots, v_{k-1}$  disebut titik internal  $W$ , dan  $k$  disebut panjang jalan  $W$ . Panjang jalan  $W$  adalah banyaknya sisi dalam  $W$ . Sebuah jalan  $W$  dengan panjang positif disebut tertutup, jika titik awal dan titik akhir dari  $W$  identik (sama) (Budayasa, 2007:6).

Sebuah titik  $G$ , mungkin saja muncul lebih dari satu kali dalam jalan  $W$ , begitu juga dengan sebuah sisi  $G$ , boleh muncul lebih dari satu kali pada jalan  $W$ . Jika semua sisi  $e_1, e_2, e_3, \dots, e_k$  dalam jalan  $W$  berbeda, maka  $W$  disebut sebuah jejak (*Trail*) (Budayasa, 2007:6).

Jika semua titik  $k_k k k k k k k k k k k k$  dalam jejak  $W$  berbeda, maka  $W$  disebut lintasan (*Path*) (Budayasa, 2007:6).

Jejak yang berawal dan berakhir pada titik yang sama disebut sirkuit (Budayasa, 2007:6). Sebuah sirkuit dikatakan sirkuit sederhana (*Simple Circuit*) jika sirkuit tersebut tidak memuat atau melewati sisi

yang sama dua kali (setiap sisi yang dilewati hanya sekali). Sebuah sirkuit dikatakan sirkuit dasar (*Elementary Circuit*) jika sirkuit tersebut tidak memuat atau melewati titik yang sama dua kali (setiap titik yang dilewati hanya satu kali, titik awal dan titik akhir boleh sama).

Graf berlabel/berbobot adalah graf yang setiap sisinya mempunyai nilai/bobot berupa bilangan non negatif.

Heuristik adalah sebuah teknik yang mengembangkan efisiensi dalam proses pencarian, namun dengan kemungkinan mengorbankan kelengkapan (*completeness*). Fungsi heuristik digunakan untuk mengavaluasi keadaan-keadaan problema individual dan menentukan seberapa jauh hal tersebut dapat digunakan untuk mendapatkan solusi yang diinginkan.

A\* diperkenalkan oleh Peter Hart, Nils Nilsson dan Bertram Raphael pertama kali pada tahun 1968 dengan menggunakan *heuristic*. Algoritma A\* adalah algoritma yang menggabungkan algoritma *Dijkstra* dan algoritma *greedy best first search*. Algoritma *Dijkstra* sendiri dinamai sesuai dengan nama penemunya yaitu Edsger Dijkstra. Algoritma *Dijkstra* menggunakan prinsip *greedy*, dimana pada setiap langkah dipilih sisi dengan bobot minimum yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih, hal ini seperti yang dikemukakan oleh Pogas, Somantri dan Satoto.

Selain menghitung biaya yang diperlukan untuk berjalan dari titik satu ke titik lainnya, algoritma A\* juga menggunakan fungsi heuristik untuk memprioritaskan pemeriksaan titik-titik pada arah yang benar, sehingga algoritma A\* mempunyai efisiensi waktu yang baik dengan tidak mengorbankan penghitungan biaya sebenarnya.

Algoritma ini merupakan algoritma *Best First Search* yang menggabungkan *Uniform Cost Search* dan *Greedy Best-First Search*. Dimana Harga yang dipertimbangkan  $f(n)$  didapat dari harga sesungguhnya  $g(n)$  ditambah dengan harga perkiraan  $h(n)$  (Harianja, 2013).

Beberapa strategi *best first search* berbeda hanya pada bentuk fungsi evaluasi yang digunakan. Pada strategi *best first search*, titik yang terpilih untuk dikembangkan adalah titik dengan nilai fungsi evaluasi terendah dan ketika dua lintasan mengarah pada titik yang sama, titik dengan nilai paling besar akan dibuang. Secara matematis, nilai fungsi evaluasi heuristik sebuah titik pada algoritma A\* diberikan oleh:

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

keterangan,

$g(n)$  =biaya dari titik awal (*start node*) ke titik n.

$h(n)$  =estimasi biaya dari titik n ke titik tujuan (*goal node*)

Matlab merupakan bahasa pemrograman yang hadir dengan fungsi dan karakteristik yang berbeda dengan bahasa pemrograman lain yang sudah ada lebih dahulu seperti Delphi, Basic maupun C++. Matlab merupakan bahasa pemrograman level tinggi yang dikhususkan untuk kebutuhan komputasi teknis, visualisasi dan pemrograman seperti komputasi matematik, analisis data, pengembangan algoritma, simulasi dan pemodelan dan grafik-grafik perhitungan.

Matlab hadir dengan membawa warna yang berbeda. Hal ini karena Matlab membawa keistimewaan dalam fungsi-fungsi matematika, fisika, statistik, dan visualisasi. Matlab dikembangkan oleh MathWorks, yang pada awalnya dibuat untuk memberikan kemudahan mengakses data matrik pada proyek Linpack dan Eispack. Saat ini Matlab memiliki ratusan fungsi yang dapat digunakan sebagai problem solver mulai dari simple sampai masalah-masalah yang kompleks dari berbagai disiplin ilmu (Firmansyah, 2007).

**METODE**

Metode penelitian merupakan suatu cara yang digunakan dalam penelitian sehingga pelaksanaan penelitian dapat dipertanggungjawabkan secara ilmiah. Dengan metode penelitian, data yang diperoleh semakin lengkap untuk memecahkan masalah yang dihadapi. Pada penelitian ini yang penulis gunakan adalah metode studi kasus.

Untuk melakukan penelitian harus memperhatikan prosedur dan langkah-langkah yang akan dilakukan untuk memulai penelitian sehingga dapat terarah dan terlaksana dengan baik dalam hal pelaporan penelitian. Langkah-langkah yang dilakukan dalam metode penelitian ini yaitu identifikasi masalah, studi pustaka, pengumpulan data, analisis data, pembuatan program, pengujian program dan penarikan kesimpulan.

**HASIL DAN PEMBAHASAN**

**Titik**

Titik dalam penelitian ini adalah 9 kecamatan dikota pati yang merupakan penyuplai bahan baku kacang tanah di PT. Dua Kelinci, berdasarkan data yang diperoleh dari PT. Dua Kelinci. Kecamatan yang menjadi penyuplai bahan baku kacang tanah yaitu : kecamatan Tayu, kecamatan Gunung Wungkal, kecamatan Tlogowungu, kecamatan Trangkil, kecamatan Gembong, kecamatan Margorejo, kecamatan Pati, kecamatan Puncakwangi dan kecamatan Tambakromo. Titik tersebut dapat dilihat pada Tabel 1.

Tabel 1. Titik mewakili lokasi/tempat penting

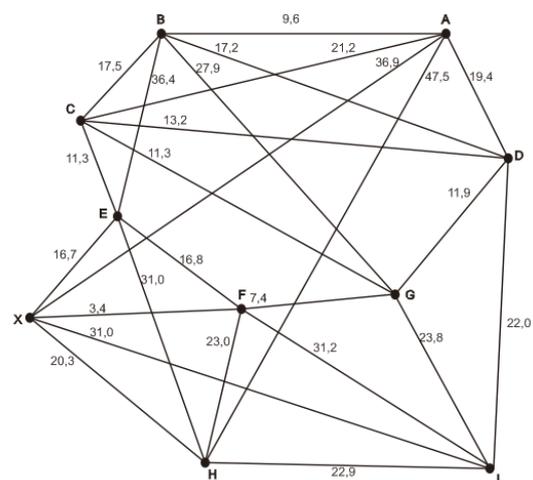
No	Nama Kecamatan	Simbol
1	Gunung Wungkal	A
2	Tayu	B
3	Tlogowungu	C
4	Trangkil	D
5	Gembong	E
6	Margorejo	F
7	Pati	G
8	Tambakromo	H
9	Puncakwangi	I
10	PT. Dua Kelinci	X

Jarak dalam penelitian ini merupakan nilai jarak antara satu kecamatan dengan kecamatan yang lain. Akan ditampilkan pula rute antar kecamatan dalam bentuk graf dan dapat dilihat dalam Tabel 2

Tabel 2. Jarak antar titik

No	Jarak		No	Jarak			
1	A	B	9,6	13	D	G	11,9
2	A	C	21,2	14	D	I	22
3	A	D	19,4	15	E	F	16,8
4	A	X	36,9	16	E	X	16,7
5	A	H	47,5	17	E	H	31
6	B	C	17,5	18	F	X	3,4
7	B	D	17,2	19	F	G	7,4
8	B	E	36,4	20	F	H	23
9	B	G	27,9	21	F	I	31,2
10	C	D	13,2	22	G	I	23,8
11	C	E	11,3	23	H	X	20,3
12	C	G	11,3	24	H	I	22,9
				25	I	X	31

sehingga diperoleh Graf G seperti Gambar 1.



Gambar 1. Rute pendistribusian kacang tanah

Jarak *Euclidian* merupakan jarak garis lurus dari masing-masing titik menuju ke titik tujuan. Maka diperoleh  $h(n)$  masing-masing titik. Data diperoleh dengan bantuan aplikasi *Google Maps*. Dengan menggunakan rumus

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

Ket:  $d$  : jarak *Euclidian*

$x$  : *Latitud* (letak titik pada koordinat x)

$y$  : *Longitude* (letak titik pada koordinat y)

Hasil penghitungan 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). Jarak *Euclidian* dalam penelitian ini dapat dilihat pada Tabel 3.

Tabel 3. Jarak *Euclidian* dari titik  $n$  ke titik tujuan

No	titik asal	jarak euclidian (km)
1	Gunungwungkal	2,291,468,735
2	Tayu	2,524,649,075
3	Tlogowungu	1,225,987,189
4	Trangkil	1,704,672,719
5	Gembong	9,825,774,414
6	Pati	6,726,439,945
7	Margorejo	2,963,975,758
8	Puncakwangi	2,001,487,825
9	Tambakromo	1,458,023,664
10	PT. Dua Kelinci	0

Berikut adalah langkah-langkah untuk menyelesaikan masalah rute pendistribusian bahan baku kacang tanah di kabupaten Pati. Karena dalam algoritma A\* tidak diperbolehkan ada pengulangan titik maka Penghitungan dimulai dari titik terdekat dengan titik tujuan titik X yaitu titik F dengan jarak 3,4 Km, dengan asumsi rute yang akan dihasilkan nanti merupakan rute terpendek.

Dari titik F menuju titik X melalui semua titik yang ada dengan menggunakan algoritma A\*. Jarak dari titik awal ke node n adalah  $g(n)$  dan  $h(n)$  adalah perkiraan jarak terpendek dari titik n ke titik tujuan X, sedangkan  $f(n)$  adalah perkiraan solusi dengan jarak terpendek melalui n.

**1. Fungsi Evaluasi (melalui F):**

$$f(E) = g(F) + g(F \text{ ke } E) + h(E) = 0 + 16,8 + 9,8 = 26,6$$

$$f(G) = g(F) + g(F \text{ ke } G) + h(G) = 0 + 7,4 + 6,7 = 14,1^*$$

$$f(H) = g(F) + g(F \text{ ke } H) + h(H) = 0 + 23,0 + 14,5 = 37,5$$

$$f(I) = g(F) + g(F \text{ ke } I) + h(I) = 0 + 31,25 + 20,0 = 51,25$$

Pada *open* hanya terdapat satu titik, yaitu titik A. Titik A terpilih sebagai *best node* dan dipindahkan ke *close*. Bangkitkan semua suksesor A kecuali X karena merupakan titik tujuan, yaitu E, G, H, dan I. Keempat titik tersebut tidak ada di *open* maupun *close*, maka keempatnya dimasukkan ke *open*. Langkah ini

menghasilkan *open* = [E, G, H, I] dan *close* = [A]. Titik G dengan bobot terkecil yaitu 14,1 terpilih sebagai *best node* dan dipindah ke *close*.

**2. Fungsi Evaluasi (melalui F dan G):**

$$f(E) = 26,6^*$$

$$f(H) = 37,5$$

$$f(I) = 51,25$$

$$f(B) = g(G) + g(G \text{ ke } B) + h(B) = 7,4 + 27,9 + 25,2 = 60,5$$

$$f(C) = g(G) + g(G \text{ ke } C) + h(C) = 7,4 + 11,3 + 12,2 = 30,9$$

$$f(D) = g(G) + g(G \text{ ke } D) + h(D) = 7,4 + 11,9 + 17,0 = 36,3$$

$$f(I) = g(G) + g(G \text{ ke } I) + h(I) = 7,4 + 23,81 + 20,0 = 51,21$$

Suksesor G dibangkitkan, yaitu B, C, D dan I. Titik B, C dan D belum pernah berada di *open* maupun *close*, maka ketiga titik tersebut dimasukkan ke *open*. Titik I sudah berada di *open*, sehingga harus dilakukan pengecekan *parent* titik I perlu diganti atau tidak. Berdasarkan penghitungan diatas *parent* titik I melalui F dan G lebih kecil daripada melalui titik F. Dengan demikian *parent* titik I perlu diganti. Langkah ini menghasilkan *open* = [E, H, I, B, C, D] dan *close* = [F, G]. Titik E dengan bobot terkecil yaitu 26,6 dipilih sebagai *best node* dan dimasukkan ke *close*.

**3. Fungsi Evaluasi (melalui F dan E):**

$$f(H) = 37,5$$

$$f(I) = 51,21$$

$$f(B) = 60,5$$

$$f(C) = 30,9^*$$

$$f(D) = 36,3$$

$$f(B) = g(E) + g(E \text{ ke } B) + h(B) = 16,8 + 36,4 + 25,2 = 78,4$$

$$f(H) = g(E) + g(E \text{ ke } H) + h(H) = 16,8 + 31,0 + 14,5 = 62,3$$

$$f(C) = g(E) + g(E \text{ ke } C) + h(C) = 16,8 + 11,3 + 12,2 = 40,3$$

merupakan titik tujuan, yaitu B, H, dan C. Titik B, C dan D sudah pernah berada di *open*, sehingga harus dilakukan pengecekan *parent*. Berdasarkan penghitungan diatas *parent* titik B, H dan C tidak perlu diganti karena menghasilkan bobot yang lebih besar. Langkah ini menghasilkan *open* = [H, I, B, C, D] dan *close* = [F, G, E]. Titik C dengan bobot terkecil yaitu 30,9 dipilih sebagai *best node* dan dimasukkan ke *close*.

**4. Fungsi Evaluasi (melalui F, G dan C):**

$$f(H) = 37,5$$

$$f(I) = 51,21$$

$$f(B) = 60,5$$

$$f(D) = 36,3^*$$

$$f(A) = g(C) + g(C \text{ ke } A) + h(A) = 18,7 + 21,2 + 22,9 = 62,8$$

$$f(B) = g(C) + g(C \text{ ke } B) + h(B) = 18,7 + 17,5 + 25,5 = 61,7$$

$$f(D) = g(C) + g(C \text{ ke } D) + h(D) = 18,7 + 13,2 + 17,0 = 48,9$$

$$f(E) = g(C) + g(C \text{ ke } E) + h(E) = 18,7 + 11,3 + 9,8 = 39,8$$

Suksesor C dibangkitkan, yaitu A, B, D dan E. Titik A dan E tidak berada di *open*, maka kedua titik tersebut dimasukkan ke *open*. Titik B dan D berada di *open*, sehingga harus dilakukan pengecekan *parent* titik B dan D perlu diganti atau tidak. Berdasarkan penghitungan diatas *parent* titik B dan D tidak perlu diganti karena menghasilkan bobot yang lebih besar. Langkah ini menghasilkan *open* = [H, I, B, D, A, E] dan *close* = [F, G, E, C]. Titik D dengan bobot terkecil yaitu 36,3 dipilih sebagai *best node* dan dimasukkan ke *close*.

**5. Fungsi Evaluasi (melalui F, G dan D):**

$$f(H) = 37,5^*$$

$$\begin{aligned}
 f(I) &= 51,21 \\
 f(B) &= 60,5 \\
 f(A) &= 62,8 \\
 f(E') &= 39,8 \\
 f(A) &= g(D) + g(D \text{ ke } A) + h(A) = 19,3 + 19,4 + 22,9 = 61,6 \\
 f(B) &= g(D) + g(D \text{ ke } B) + h(B) = 19,3 + 17,2 + 25,2 = 61,7 \\
 f(C) &= g(D) + g(D \text{ ke } C) + h(C) = 19,3 + 13,2 + 12,2 = 44,7 \\
 f(I) &= g(D) + g(D \text{ ke } I) + h(I) = 19,3 + 22,0 + 20,0 = 61,3
 \end{aligned}$$

Suksesor D dibangkitkan, yaitu A, B, C dan I. Titik C tidak berada di *open*, maka kedua titik tersebut dimasukkan ke *open*. Titik A, B dan I berada di *open*, sehingga harus dilakukan pengecekan *parent* titik A, B dan I perlu diganti atau tidak. Berdasarkan penghitungan diatas, *parent* titik A perlu diganti karena menghasilkan bobot yang lebih kecil, sedangkan *parent* titik B dan I tidak perlu diganti karena menghasilkan bobot yang lebih besar. Langkah ini menghasilkan *open* = [H, I, B, D, A, E', C'] dan *close* = [F, G, E, C, D]. Titik H dengan bobok terkecil yaitu 37,5 dipilih sebagai *best node* dan dimasukkan ke *close*.

**6. Fungsi Evaluasi (melalui F dan H):**

$$\begin{aligned}
 f(I) &= 51,21 \\
 f(B) &= 60,5 \\
 f(A) &= 61,6 \\
 f(E') &= 39,8 * \\
 f(C) &= 44,7 \\
 f(I) &= g(H) + g(H \text{ ke } I) + h(I) = 23,0 + 22,9 + 20,0 = 65,9 \\
 f(E) &= g(H) + g(H \text{ ke } E) + h(E) = 23,0 + 31,0 + 9,8 = 63,8 \\
 f(A) &= g(H) + g(H \text{ ke } A) + h(A) = 23,0 + 47,5 + 22,9 = 93,4
 \end{aligned}$$

Suksesor H dibangkitkan, yaitu I, E dan A. Titik I, E dan A berada di *open*, sehingga harus dilakukan pengecekan *parent* titik I, E dan A perlu diganti atau tidak. Berdasarkan penghitungan diatas, *parent* titik I, E dan A tidak perlu diganti karena menghasilkan bobot yang lebih besar. Langkah ini menghasilkan *open* = [I, B, A, E', C'] dan *close* = [F, G, E, C, D, H]. Titik E' dengan bobok terkecil yaitu 39,8 dipilih sebagai *best node* dan dimasukkan ke *close*.

**7. Fungsi Evaluasi (melalui F, G, C dan E):**

$$\begin{aligned}
 f(I) &= 51,21 * \\
 f(B) &= 60,5 \\
 f(A) &= 61,6 \\
 f(H) &= g(E) + g(E \text{ ke } H) + h(H) = 30,0 + 31,0 + 14,5 = 75,5 \\
 f(B) &= g(E) + g(E \text{ ke } B) + h(B) = 30,0 + 36,4 + 25,2 = 91,6
 \end{aligned}$$

Suksesor E dibangkitkan kecuali X karena merupakan titik tujuan, yaitu H dan B. Titik H tidak berada di *open*, maka titik tersebut dimasukkan ke *open*. Titik B berada di *open*, sehingga harus dilakukan pengecekan *parent* titik B perlu diganti atau tidak. Berdasarkan penghitungan diatas, *parent* titik B tidak perlu diganti karena menghasilkan bobot yang lebih besar. Langkah ini menghasilkan *open* = [I, B, A, H] dan *close* = [F, G, E, C, D, H, E']. Titik I dengan bobok terkecil yaitu 51,21 dipilih sebagai *best node* dan dimasukkan ke *close*.

**8. Fungsi Evaluasi (melalui F, G dan I):**

$$\begin{aligned}
 f(B) &= 60,5 * \\
 f(A) &= 61,6 \\
 f(H') &= 75,5 \\
 f(H) &= g(I) + g(I \text{ ke } H) + h(H) = 31,2 + 22,9 + 14,5 = 68,6 \\
 f(D) &= g(I) + g(I \text{ ke } D) + h(D) = 31,2 + 22,0 + 17,0 = 70,2
 \end{aligned}$$

Suksesor I dibangkitkan kecuali X karena merupakan titik tujuan, yaitu H dan D. Titik D tidak berada di *open*, maka titik tersebut dimasukkan ke *open*. Titik H' berada di *open*, sehingga harus dilakukan pengecekan *parent* titik H' perlu diganti atau tidak. Berdasarkan penghitungan diatas, *parent* titik H' perlu diganti karena menghasilkan bobot yang lebih kecil. Langkah ini menghasilkan *open* = [B, A, H', D] dan *close* = [F, G, E, C, D, H, E', I]. Titik B dengan bobok terkecil yaitu 60,5 dipilih sebagai *best node* dan dimasukkan ke *close*.

**9. Fungsi Evaluasi (melalui F, G dan B):**

$$\begin{aligned}
 f(A) &= 61,6 * \\
 f(H') &= 68,6 \\
 f(D) &= 70,2 \\
 f(A) &= g(B) + g(B \text{ ke } A) + h(A) = 35,3 + 9,6 + 22,9 = 67,8 \\
 f(C) &= g(B) + g(B \text{ ke } C) + h(C) = 35,3 + 17,5 + 12,2 = 65,0 \\
 f(E) &= g(B) + g(B \text{ ke } E) + h(E) = 35,3 + 36,4 + 9,8 = 81,5 \\
 f(D) &= g(B) + g(B \text{ ke } D) + h(D) = 35,3 + 17,2 + 17,0 = 69,5
 \end{aligned}$$

Suksesor B dibangkitkan, yaitu A, C, E dan D. Titik C dan E tidak berada di *open*, maka titik kedua tersebut dimasukkan ke *open*. Titik A dan D berada di *open*, sehingga harus dilakukan pengecekan *parent* titik A dan D perlu diganti atau tidak. Berdasarkan penghitungan diatas, *parent* titik D perlu diganti karena menghasilkan bobot yang lebih kecil. Sedangkan *parent* titik A tidak perlu diganti karena menghasilkan bobot yang lebih besar. Langkah ini menghasilkan *open* = [A, H, D, C, E] dan *close* = [F, G, E, C, D, H, E', I, B]. Titik A dengan bobok terkecil yaitu 61,6 dipilih sebagai *best node* dan dimasukkan ke *close*.

**10. Fungsi Evaluasi (melalui F, G, D dan A):**

$$\begin{aligned}
 f(H') &= 68,6 \\
 f(C') &= 65,0 * \\
 f(E) &= 81,5 \\
 f(B) &= g(A) + g(A \text{ ke } B) + h(B) = 38,7 + 9,6 + 25,2 = 73,5 \\
 f(C) &= g(A) + g(A \text{ ke } C) + h(C) = 38,7 + 21,2 + 12,2 = 72,1 \\
 f(H) &= g(A) + g(A \text{ ke } H) + h(H) = 38,7 + 47,5 + 14,5 = 100,7
 \end{aligned}$$

Suksesor A dibangkitkan kecuali titik X karena merupakan titik tujuan, yaitu B, C dan H. Titik B tidak berada di *open*, maka titik tersebut dimasukkan ke *open*. Titik C dan H berada di *open*, sehingga harus dilakukan pengecekan *parent* titik C dan H perlu diganti atau tidak. Berdasarkan penghitungan diatas, *parent* titik C dan H tidak perlu diganti karena menghasilkan bobot yang lebih besar. Langkah ini menghasilkan *open* = [H, C, E, B] dan *close* = [F, G, E, C, D, H, E', I, B, A]. Titik C' dengan bobok terkecil yaitu 65,0 dipilih sebagai *best node* dan dimasukkan ke *close*.

**11. Fungsi Evaluasi (melalui F, G, B dan C):**

$$\begin{aligned}
 f(H') &= 68,6 * \\
 f(E) &= 81,5 \\
 f(A) &= g(C) + g(C \text{ ke } A) + h(A) = 52,8 + 21,2 + 22,9 = 96,9 \\
 f(D) &= g(C) + g(C \text{ ke } D) + h(D) = 52,8 + 13,2 + 17,0 = 83,0 \\
 f(E) &= g(C) + g(C \text{ ke } E) + h(E) = 52,8 + 11,3 + 9,8 = 73,9
 \end{aligned}$$

Suksesor C' dibangkitkan, yaitu A, D dan E. Titik A dan D tidak berada di *open*, maka titik tersebut dimasukkan ke *open*. Titik E berada di *open*, sehingga harus dilakukan pengecekan *parent* titik E perlu

diganti atau tidak. Berdasarkan penghitungan diatas, *parent* titik E perlu diganti karena menghasilkan bobot yang lebih kecil. Langkah ini menghasilkan *open* = [H, E, A, D] dan *close* = [F, G, E, C, D, H, E', I, B, A, C']. Titik H' dengan bobok terkecil yaitu 68,6 dipilih sebagai *best node* dan dimasukkan ke *close*.

**12. Fungsi Evaluasi (melalui F, G, I dan H):**

$$f(E'') = 73,9^*$$

$$f(A) = 96,9$$

$$f(D) = 83,0$$

$$f(E) = g(H) + g(H \text{ ke } E) + h(E) = 54,1 + 31,0 + 9,8 = 94,9$$

$$f(A) = g(H) + g(H \text{ ke } A) + h(A) = 54,1 + 47,5 + 22,9 = 124,5$$

Suksesor H' dibangkitkan, yaitu E dan A. Titik E dan A berada di *open*, sehingga harus dilakukan pengecekan *parent* titik E dan A perlu diganti atau tidak. Berdasarkan penghitungan diatas, *parent* titik E dan A tidak perlu diganti karena menghasilkan bobot yang lebih besar. Langkah ini menghasilkan *open* = [E, A, D] dan *close* = [F, G, E, C, D, H, E', I, B, A, C', H'']. Titik E'' dengan bobok terkecil yaitu 73,9 dipilih sebagai *best node* dan dimasukkan ke *close*.

**13. Fungsi Evaluasi (melalui F, G, B, C dan E):**

$$f(A) = 96,9$$

$$f(D') = 83,0^*$$

$$f(H) = g(E) + g(E \text{ ke } H) + h(H) = 64,1 + 31,0 + 14,5 = 109,5$$

Suksesor E'' dibangkitkan, yaitu titik H. Titik H tidak berada di *open*, maka titik tersebut dimasukkan ke *open*. Langkah ini menghasilkan *open* = [A, D, H] dan *close* = [F, G, E, C, D, H, E', I, B, A, C', H', E'']. Titik D' dengan bobok terkecil yaitu 83,0 dipilih sebagai *best node* dan dimasukkan ke *close*.

**14. Fungsi Evaluasi (melalui F, G, B, C dan D):**

$$f(A') = 96,9^*$$

$$f(H) = 109,5$$

$$f(A) = g(D) + g(D \text{ ke } A) + h(A) = 66,0 + 19,4 + 22,9 = 108,3$$

$$f(I) = g(D) + g(D \text{ ke } I) + h(I) = 66,0 + 22,0 + 20,0 = 108,2$$

Suksesor D' dibangkitkan, yaitu titik A dan I. Titik I tidak berada di *open*, maka titik tersebut dimasukkan ke *open*. Titik A berada di *open*, sehingga harus dilakukan pengecekan *parent* titik A perlu diganti atau tidak. Berdasarkan penghitungan diatas, *parent* titik A tidak perlu diganti karena menghasilkan bobot yang lebih besar. Langkah ini menghasilkan *open* = [A, H, I] dan *close* = [F, G, E, C, D, H, E', I, B, A, C', H', E'', D'']. Titik A' dengan bobok terkecil yaitu 96,9 dipilih sebagai *best node* dan dimasukkan ke *close*.

**15. Fungsi Evaluasi (melalui F, G, B, C, dan A):**

$$f(H) = 109,5$$

$$f(I') = 108,2^*$$

$$f(D) = g(A) + g(A \text{ ke } D) + h(D) = 74,0 + 19,4 + 17,0 = 110,4$$

$$f(H) = g(A) + g(A \text{ ke } H) + h(H) = 74,0 + 47,5 + 14,5 = 136,0$$

Suksesor A' dibangkitkan, yaitu titik D dan H. Titik D tidak berada di *open*, maka titik tersebut dimasukkan ke *open*. Titik H berada di *open*, sehingga harus dilakukan pengecekan *parent* titik H perlu diganti atau tidak. Berdasarkan penghitungan diatas, *parent* titik H tidak perlu diganti karena menghasilkan bobot yang lebih besar. Langkah ini menghasilkan *open* = [H, I, D] dan *close* = [F, G, E, C, D, H, E', I, B, A, C', H', E'', D', A']. Titik I' dengan bobok terkecil

yaitu 108,2 dipilih sebagai *best node* dan dimasukkan ke *close*.

**16. Fungsi Evaluasi (melalui F, G, B, C, D dan I):**

$$f(H'') = 109,5$$

$$f(H) = g(I) + g(I \text{ ke } H) + h(H) = 88,0 + 22,9 + 14,5 = 125,4$$

Suksesor I' dibangkitkan, yaitu titik H. Titik H berada di *open*, sehingga harus dilakukan pengecekan *parent* titik H perlu diganti atau tidak. Berdasarkan penghitungan diatas, *parent* titik H tidak perlu diganti karena menghasilkan bobot yang lebih besar. Langkah ini menghasilkan *open* = [H] dan *close* = [F, G, E, C, D, H, E', I, B, A, C', H', E'', D', A', I'']. Titik H'' dengan bobok terkecil yaitu 109,5 dipilih sebagai *best node* dan dimasukkan ke *close*.

**17. Fungsi Evaluasi (melalui F, G, B, C, E dan H):**

$$f(I'') = g(H) + g(H \text{ ke } I) + h(I) = 95,1 + 22,9 + 20,0 = 138,0^*$$

$$f(A) = g(H) + g(H \text{ ke } A) + h(A) = 95,1 + 47,5 + 22,9 = 165,5$$

Suksesor H'' dibangkitkan, yaitu titik I dan A. Titik I dan A tidak berada di *open*, maka titik tersebut dimasukkan ke *open*. Langkah ini menghasilkan *open* = [I, A] dan *close* = [F, G, E, C, D, H, E', I, B, A, C', H', E'', D', A', I', H'']. Titik I'' dengan bobok terkecil yaitu 138,0 dipilih sebagai *best node* dan dimasukkan ke *close*.

**18. Fungsi Evaluasi (melalui F, G, B, C, E, H dan I):**

$$f(A'') = 165,5$$

$$f(D'') = g(I) + g(I \text{ ke } D) + h(D) = 118,0 + 22,0 + 17,0 = 157,0^*$$

Suksesor I'' dibangkitkan, yaitu titik D. Titik D tidak berada di *open*, maka titik tersebut dimasukkan ke *open*. Langkah ini menghasilkan *open* = [A, D] dan *close* = [F, G, E, C, D, H, E', I, B, A, C', H', E'', D', A', I', H'', I'']. Titik D'' dengan bobok terkecil yaitu 157,0 dipilih sebagai *best node* dan dimasukkan ke *close*.

**19. Fungsi Evaluasi (melalui F, G, B, C, E, H, I dan D):**

$$f(A'') = 165,5^*$$

$$f(A) = g(D) + g(D \text{ ke } A) + h(A) = 140,0 + 19,4 + 22,9 = 182,3$$

Suksesor D'' dibangkitkan, yaitu titik A. Titik A berada di *open*, sehingga harus dilakukan pengecekan *parent* titik A perlu diganti atau tidak. Berdasarkan penghitungan diatas, *parent* titik A tidak perlu diganti karena menghasilkan bobot yang lebih besar. Langkah ini menghasilkan *open* = [A] dan *close* = [F, G, E, C, D, H, E', I, B, A, C', H', E'', D', A', I', H'', I'', D'']. Titik A'' dengan bobok terkecil yaitu 165,5 dipilih sebagai *best node* dan dimasukkan ke *close*.

**20. Fungsi Evaluasi (melalui F, G, B, C, E, H, dan A):**

$$f(D''') = g(A) + g(A \text{ ke } D) + h(D) = 142,6 + 19,4 + 17,0 = 179,0$$

Suksesor A'' dibangkitkan, yaitu titik D. Titik D tidak berada di *open*, maka titik tersebut dimasukkan ke *open*. Langkah ini menghasilkan *open* = [D] dan *close* = [F, G, E, C, D, H, E', I, B, A, C', H', E'', D', A', I', H'', I'', D'', A'']. Titik D''' dengan

bobok terkecil yaitu 179,0 dipilih sebagai *best node* dan dimasukkan ke *close*.

**21. Fungsi Evaluasi (melalui F, G, B, C, E, H, A dan D):**

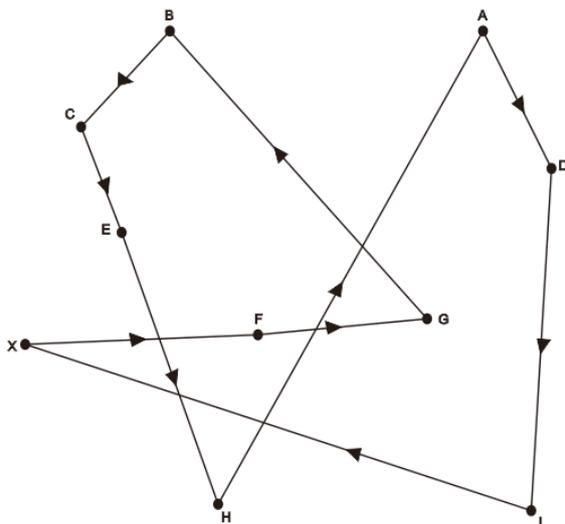
$$f(I''') = g(D) + g(D \text{ ke } I) + h(I) = 162,0 + 22,0 + 20,0 = 204,0$$

Suksesor D''' dibangkitkan, yaitu titik I. Titik I tidak berada di *open*, maka titik tersebut dimasukkan ke *open*. Langkah ini menghasilkan *open* = [I] dan *close* = [F, G, E, C, D, H, E', I, B, A, C', H', E'', D', A', I', H'', I'', D'', A'', D''']. Titik I''' dengan bobok terkecil yaitu 204,0 dipilih sebagai *best node* dan dimasukkan ke *close*.

**22. Fungsi Evaluasi (melalui F, G, B, C, E, H, A, D dan I):**

$$f(X) = g(I) + g(I \text{ ke } X) + h(X) = 184,0 + 31,0 + 0 = 215,0$$

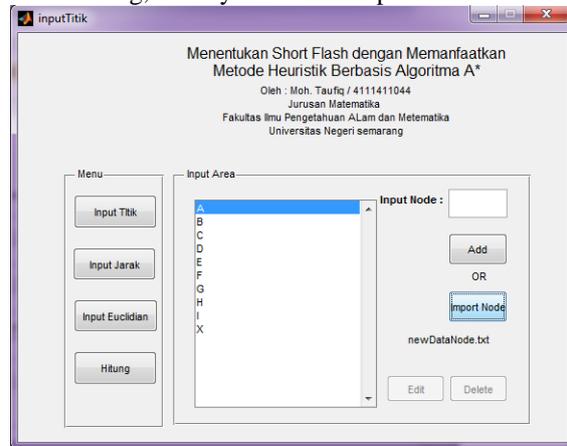
Suksesor I''' dibangkitkan, yaitu titik X. Titik X tidak berada di *open*, maka titik tersebut dimasukkan ke *open*. Langkah ini menghasilkan *open* = [X] dan *close* = [F, G, E, C, D, H, E', I, B, A, C', H', E'', D', A', I', H'', I'', D'', A'', D''', I''']. Titik X dengan bobok terkecil yaitu 215,0 dipilih sebagai *best node* dan dimasukkan ke *close*. Pencarian berhenti ketika semua titik yang ada telah dilalui dengan titik awal titik F dan titik akhir titik X. Karena *best node* sama dengan *goal* dan telah memenuhi semua titik yang tersedia berarti solusi ditemukan. Penghitungan menghasilkan rute terbaik yaitu X-F-G-B-C-E-H-A-D-I-X dengan panjang rute terpendeknya adalah 218,4 Km. Rutenya adalah PT. Dua Kelinci - Kecamatan Margorejo - Kecamatan Pati - Kecamatan Tayu - Kecamatan Tlogowungu - Kecamatan Gembong - Kecamatan Tambakkromo - Kecamatan Gunung Wungkal - Kecamatan Trangkil - Kecamatan Puncakwangi - PT. Dua Kelinci. Rute yang di peroleh bisa dilihat pada Gambar 2.



Gambar 2. Rute pendistribusian terpendek

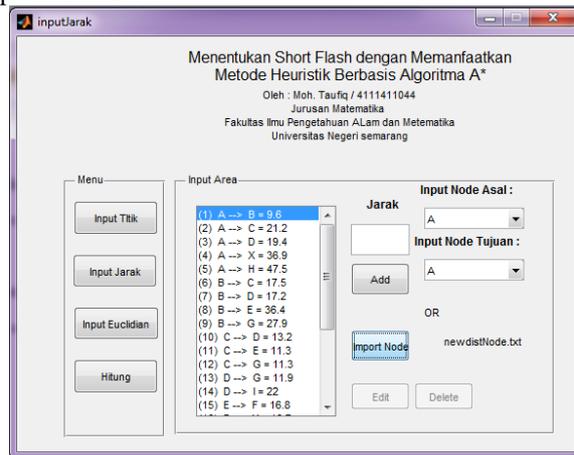
**Menentukan Rute Terpendek dengan Menggunakan Aplikasi Matlab.**

*Input Node* yaitu memasukkan titik yang akan dihitung, hasilnya bisa dilihat pada Gambar 3.



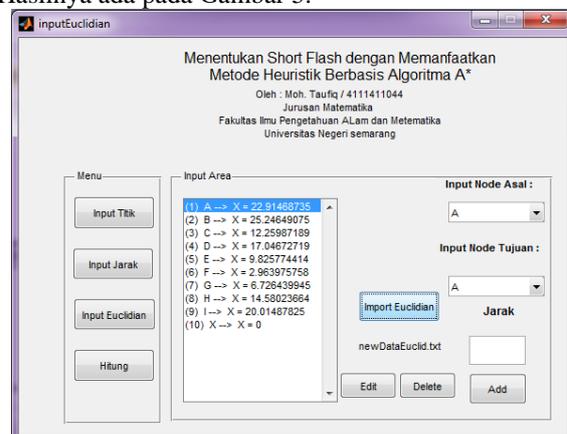
Gambar 3. Input titik

*Input Distance* yaitu memasukkan jarak antar titik kedalam program yang sudah dibuat, hasilnya ada pada Gambar 4.



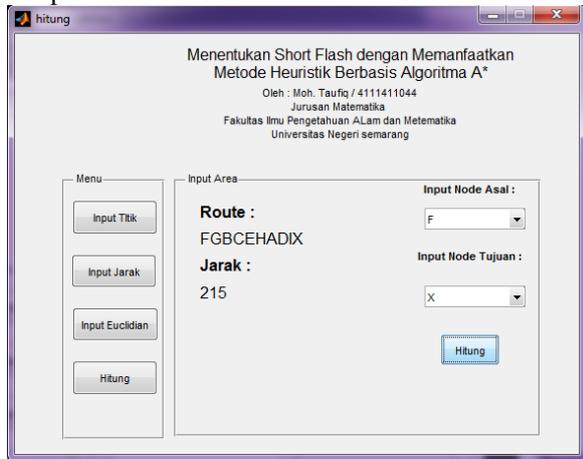
Gambar 4. Input jarak

*Input Euclidian* adalah memasukkan jarak *Euclidian* yang sudah di peroleh ke dalam program. Hasilnya ada pada Gambar 5.



Gambar 5. Input Jarak Euclidian

**Hitung** adalah menampilkan hasil penghitungan yang dilakukan oleh program, hasilnya ada pada Gambar 6.



Gambar 6. Hitung

Dari Gambar 6 kita bisa melihat *input node* asal adalah F, *input node* tujuan adalah X, *route* adalah F – G – B – C – E – H – A – D – I – X, dengan jarak adalah 215. Program ini menghitung mulai dari titik F kalau kita tambah rute yang sudah ditemukan menggunakan program dengan jarak dari titik X menuju titik F maka hasilnya menjadi X – F – G – B – C – E – H – A – D – I – X, dengan total jarak yang ditempuh adalah sejauh 218,4 Km. Rutenya yaitu PT. Dua Kelinci - Kecamatan Margorejo - Kecamatan Pati - Kecamatan Tayu - Kecamatan Tlogowungu - Kecamatan Gembong - Kecamatan Tambakkromo - Kecamatan Gunung Wungkal - Kecamatan Trangkil - Kecamatan Puncakwangi - PT. Dua Kelinci. Hal ini sama dengan perhitungan manual menggunakan algoritma A\*.

## SIMPULAN

Algoritma A\* merupakan algoritma yang mengevaluasi titik dan menghubungkan  $g(n)$  yaitu jarak untuk mencapai titik dan  $h(n)$  yaitu jarak yang diperlukan dari titik untuk mencapai tujuan, sehingga:

Jarak dari titik awal ke titik  $n$  adalah  $g(n)$  dan  $n$  adalah perkiraan jarak terendah dari titik  $n$  ke titik tujuan. Sedangkan  $f(n)$  adalah perkiraan solusi dengan jarak terpendek melalui  $n$ . dengan demikian untuk menemukan solusi terbaik, hal pertama yang di coba adalah titik dengan nilai  $g(n) + h(n)$  terendah. Strategi ini jelas lebih baik dengan disediakannya nilai heuristik  $h(n)$  yang dapat memenuhi kondisi tertentu sehingga A\* menjadi optimal.

Hasil penghitungan manual dengan menggunakan algoritma A\* menghasilkan rute PT. Dua Kelinci - Kecamatan Margorejo - Kecamatan Pati - Kecamatan Tayu - Kecamatan Tlogowungu - Kecamatan Gembong - Kecamatan Tambakkromo - Kecamatan Gunung Wungkal - Kecamatan Trangkil - Kecamatan

Puncakwangi - PT. Dua Kelinci dengan panjang rute terpendek adalah 218,4 Km.

Berdasarkan perhitungan dengan *software* Matlab diperoleh hasil yang sama dengan perhitungan manual.

## DAFTAR PUSTAKA

Budayasa, I. K. 2007. *Teory Graf dan Aplikasinya*. Unesa University Press.

Bell, E. T. 1952. *Mathematics: Queen and Servant of Science*. London: G. Bell & Sons, Ltd.

Firmansyah, A. 2007. *Dasar-dasar Pemograman MATLAB*. IlmuKomputer.com.

Harianja, F. 2013. *Penerapan Algoritma A\* dalam permasalahan optimalisasi pencarian solusi dynamic water jug*. Jurnal Pelita Informatika Budi Darma, Volume : IV, Nomor: 3. ISSN : 2301-9425.

Hart, P. E., Nilsson, N. J. and Raphael, B. 1968. *A Formal Basis for the Heuristic Determination of Minimum Cost Paths*. *IEEE Transactions on Systems Science and Cybernetics* SSC4 4 (2): 100-107.

Muttaqien, M. A., Mulyono, dan Suyitno, A. 2013. *Pelabelan total sisi ajaib pada graf double star dan graf Sun*. *Journal of Mathematics*, 2 (2). 85 – 89.

Pugas, D. O., Somantri, M., dan Satoto, K. I. 2011. *Pencarian Rute Terpendek Menggunakan Algoritma Dijkstra dan Astar (A\*) pada SIG Berbasis Web untuk Pemetaan Pariwisata Kota Sawahlunto*. *Jurnal TRANSMISI*, 13 (1), 2011, 27-32.