

PENCARIAN JALUR TERPENDEK UNTUK PENJEMPUTAN BARANG KIRIMAN PELANGGAN MITRA (STUDI KASUS PADA KANTOR POS MALANG)

Angga Setiyawan¹, Deddy Kusbianto Purwoko Aji², Ariadi Retno Tri Hayati Ririd³,
Erfan Rohadi⁴, Ahmadi Yuli Ananta⁵

Teknik Informatika, Teknologi Informasi, Politeknik Negeri Malang

¹angga.setiyawan@outlook.co.id, ²deddy_kusbianto@polinema.ac.id, ³ariadi.retno@polinema.ac.id ⁴erfanr@polinema.ac.id, ⁵ahmadi@polinema.ac.id

Abstrak

PT. Pos Indonesia (Persero) merupakan Badan Usaha Milik Negara yang bertujuan untuk turut serta melaksanakan dan menunjang kebijaksanaan & program pemerintah di bidang pelayanan jasa pos dan giro kepada masyarakat Indonesia. Kantor Pos Malang menawarkan layanan jemput kiriman atau *pickup service* kepada pelanggannya yang telah menandatangani surat perjanjian kontrak untuk menjadi Mitra Pos. Permasalahan yang timbul dari fasilitas ini adalah belum terdapatnya sebuah sistem untuk memajemen penjemputan dan tidak sedikit kurir penjemput kiriman yang tidak mengetahui rute terpendek menuju lokasi Mitra Pos sehingga proses penjemputan terkesan belum efektif. Skripsi ini mengembangkan sebuah aplikasi yang dapat *manage* data mitra pos, data penjemputan barang dan data kurir penjemput kiriman, serta memberikan informasi mengenai rute terpendek menuju Mitra Pos dengan menggunakan algoritma A* yang menerapkan pencarian heuristik, pencarian tersebut memiliki informasi mengenai jarak tempuh untuk mencapai tujuan dari Kantor Pos Malang menuju Mitra Pos. Berdasarkan hasil pengujian kuesioner kepada 10 responden, 100% responden menyatakan bahwa aplikasi ini user friendly dan bermanfaat serta 80% menyatakan bahwa aplikasi ini mudah digunakan untuk membantu melakukan permintaan penjemputan barang kiriman.

Kata kunci : Pencarian Jalur Tependek, Jemput Kiriman, *Algoritma A**, *PHP*, *Google Maps API*.

1. Pendahuluan

PT. Pos Indonesia (Persero) merupakan Badan Usaha Milik Negara (BUMN) yang bertujuan untuk turut serta melaksanakan dan menunjang kebijaksanaan & program Pemerintah dibidang ekonomi dan pembangunan Nasional pada umumnya dan pada khususnya di bidang pelayanan jasa pos dan giro kepada masyarakat Indonesia dengan menerapkan prinsip-prinsip perseroan terbatas, sesuai dengan Undang – Undang Republik Indonesia No. 38 Tahun 2009 tentang tujuan di dirikannya pos guna memudahkan masyarakat untuk menjalin komunikasi jarak jauh untuk penggunaan pribadi maupun bisnis.

Layanan jemput kiriman atau *pickup service* adalah bagian dari fasilitas baru yang ditawarkan Kantor Pos Malang untuk pelanggan yang telah memiliki kerjasama dengan Kantor Pos Malang dengan terlebih dahulu menandatangani surat perjanjian kontrak antara kedua belah pihak, pelanggan ini di sebut dengan Pelanggan Mitra Pos. dengan adanya layanan ini diharapkan dapat mempermudah pengirim untuk melakukan pengiriman barang tanpa harus datang langsung ke Kantor Pos.

Belum terdapatnya sebuah system untuk memajemen transaksi penjemputan dan tidak sedikit kurir penjemput kiriman yang mengetahui rute terpendek yang harus dilalui untuk melakukan penjemputan terhadap kiriman pelanggan, dengan adanya permasalahan yang telah diuraikan diatas, maka akan dikembangkan sebuah aplikasi Pencarian Jalur Terpendek Untuk Penjemputan Barang Kiriman Pelanggan Mitra guna memudahkan kurir dalam mendapatkan informasi rute terpendek dalam penjemputan barang kiriman pelanggan.

Metode yang digunakan adalah A-Star, Algoritma A-Star merupakan suatu algoritma yang dapat memecahkan masalah pencarian jalur atau rute terpendek. Algoritma A-Star menggunakan estimasi jarak terdekat untuk mencapai tujuan yang ingin dicapai, dan memiliki nilai heuristik yang digunakan sebagai dasar pertimbangan. Heuristik merupakan kriteria, metoda, atau prinsip-prinsip untuk menentukan pilihan sejumlah alternatif untuk mencapai sasaran dengan efektif.

2. Landasan Teori

2.1 PT. POS Indonesia

Sejarah mencatat keberadaan Pos Indonesia begitu panjang, Kantorpos pertama didirikan di Batavia (sekarang Jakarta) oleh Gubernur Jenderal G.W Baron van Imhoff pada tanggal 26 Agustus 1746 dengan tujuan untuk lebih menjamin keamanan surat-surat penduduk, terutama bagi mereka yang berdagang dari kantor-kantor di luar Jawa dan bagi mereka yang datang dari dan pergi ke Negeri Belanda.

Sejak itulah pelayanan pos telah lahir mengemban peran dan fungsi pelayanan kepada publik. Setelah Kantorpos Batavia didirikan, maka empat tahun kemudian didirikan Kantorpos Semarang untuk mengadakan perhubungan pos yang teratur antara kedua tempat itu dan untuk mempercepat pengirimannya. Rute perjalanan pos kala itu ialah melalui Karawang, Cirebon dan Pekalongan.

Tabel 1. Sampel data Mitra Pos

No.	Nama Mitra	Alamat Mitra
1	BPJS Kesehatan	Jl. Tumenggung Suryo No.48, Kota Malang
2	MANDIRI TUNAS FINANCE	Jl. Tumenggung Suryo No. 98, Kota Malang
3	INDOMOBIL	JL. S.P. Sudarmo No. 12, Kota Malang
4	ASABRI	Jl. A. Yani No.123-125, Kota Malang
5	BEA CUKAI	Jl. Raden Intan No. 3, Kota Malang

(sumber : SLPK Kantor Pos Malang)

Pos Indonesia telah beberapa kali mengalami perubahan status mulai dari Jawatan PTT (Post, Telegraph dan Telephone). Badan usaha yang dipimpin oleh seorang Kepala Jawatan ini operasinya tidak bersifat komersial dan fungsinya lebih diarahkan untuk mengadakan pelayanan publik. Perkembangan terus terjadi hingga statusnya menjadi Perusahaan Negara Pos dan Telekomunikasi (PN Postel). Mengamati perkembangan zaman dimana sektor pos dan telekomunikasi berkembang sangat pesat, maka pada tahun 1965 berganti menjadi Perusahaan Negara Pos dan Giro (PN Pos dan Giro), dan pada tahun 1978 berubah menjadi Perum Pos dan Giro

yang sejak ini ditegaskan sebagai badan usaha tunggal dalam menyelenggarakan dinas pos dan giropos baik untuk hubungan dalam maupun luar negeri. Selama 17 tahun berstatus Perum, maka pada Juni 1995 berubah menjadi Perseroan Terbatas dengan nama PT Pos Indonesia (Persero).

2.2 Mitra Pos

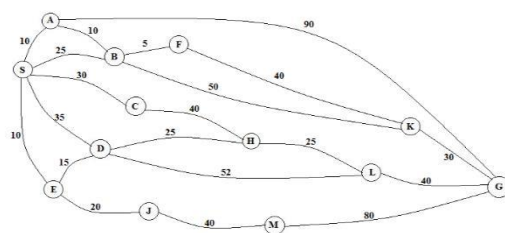
Mitra Pos adalah Perorangan atau Korporat yang telah memiliki kerjasama dengan terlebih dahulu melakukan penandatanganan surat perjanjian kontrak antara Pihak Pengirim dalam hal ini (Mitra Pos) dengan Kantor Pos Malang sebagai Pihak Jasa Pengiriman.

2.3 Algoritma A*

Algoritma ini merupakan algoritma *Best First Search* yang menggabungkan *Uniform Cost Search* dan *Greedy Best-First Search*. Jarak yang diperhitungkan didapat dari jarak sebenarnya (*actual cost*) ditambah dengan jarak perkiraan (*estimated cost*) (Suyanto, 2014). Dalam notasi matematika dituliskan sebagai:

$$f(n) = g(n) + h(n) \tag{1}$$

Dengan perhitungan jarak seperti ini, algoritma A* adalah *complete* dan optimal. Misalkan terdapat kota yang dinyatakan oleh simpul-simpul dalam suatu *graph* dua arah. Setiap angka pada busur menyatakan jarak sebenarnya (*actual cost*) antara satu kota dengan yang lainnya. Nilai $h(n)$ adalah fungsi heuristik, yaitu jarak garis lurus dari simpul n menuju simpul G dalam satuan kilometer. Contoh kasus pencarian rute terpendek ditampilkan pada Gambar 1.



Gambar 1. Contoh kasus algoritma A*

Tabel 2 adalah tabel nilai $h(n)$ yang menyatakan jarak perkiraan dari setiap simpul-simpul menuju simpul G :

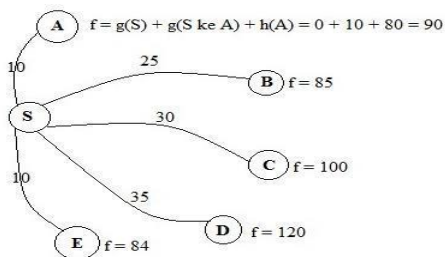
Tabel 2. Nilai $h(n)$ contoh kasus algoritma A*

n	S	A	B	C	D	E	F	G
$h(n)$	80	80	60	70	85	74	70	0
H	J	K	L	M				
40	100	30	20	70				

Langkah-langkah pencarian rute berdasarkan algoritma tersebut adalah sebagai berikut:

1. Langkah pertama

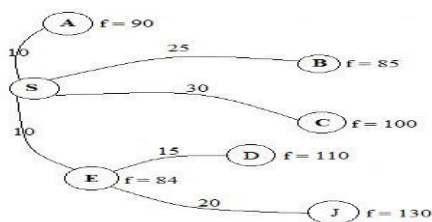
Karena di *OPEN* hanya terdapat satu simpul (S), maka S terpilih sebagai *BestNode* dan dipindahkan ke *CLOSED*. Kemudian bangkitkan semua suksesor S, yaitu: A, B, C, D, E. Karena kelima suksesor tidak ada di *OPEN* maupun *CLOSED*, maka kelima dimasukkan ke *OPEN*. Menghasilkan *OPEN* = [A, B, C, E] dan *CLOSED* = [S]. Gambar 2.2 menjelaskan langkah pertama dari pencarian rute terpendek



Gambar 2. Langkah pertama

2. Langkah kedua

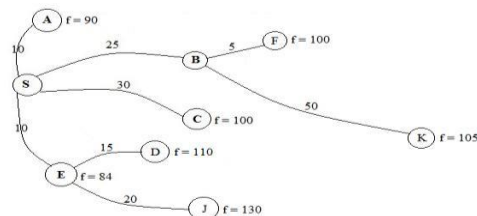
E dengan biaya terkecil (yaitu 84) terpilih sebagai *BestNode* dan dipindahkan ke *CLOSED*. Selanjutnya, semua suksesor E dibangkitkan, yaitu: D dan J. Karena belum pernah ada di *OPEN* maupun *CLOSED*, maka J dimasukkan ke *OPEN*. Sedangkan simpul D sudah ada di *OPEN*, maka harus dicek apakah *parent* dari D perlu diganti atau tidak. Ternyata, jarak dari S ke D melalui E (yaitu 10 + 15 = 25) lebih kecil daripada jarak dari S ke D (yaitu 35). Oleh karena itu, *parent* dari D harus diubah, yang semula S menjadi E. Dengan perubahan *parent* ini, maka nilai *g* dan *f* pada D juga diperbarui (nilai *g* yang semula 35 menjadi 25, dan nilai *f* dari 120 menjadi 110). Langkah kedua ini menghasilkan *OPEN* = [A, B, C, D, J] dan *CLOSED* = [S, E]. Penjelasan langkah kedua dari pencarian rute terpendek ditampilkan pada Gambar 3.



Gambar 3. Langkah kedua

3. Langkah ketiga

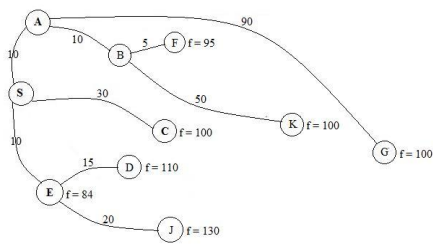
B dengan jarak terkecil (yaitu 85) terpilih sebagai *BestNode* dan dipindahkan ke *CLOSED*. Selanjutnya, semua suksesor B dibangkitkan, yaitu: A, F dan K. Karena belum pernah ada di *OPEN* maupun *CLOSED*, maka F dan K dimasukkan ke *OPEN*. Sedangkan simpul A sudah ada di *OPEN*, maka harus dicek apakah *parent* dari A perlu diganti atau tidak. Ternyata, jarak dari S ke A melalui B (yaitu 25 + 10 = 35) lebih besar daripada jarak dari S ke A (yaitu 10). Oleh karena itu, *parent* dari A tidak perlu diubah (tetap S), Akhir dari langkah ketiga ini menghasilkan *OPEN* = [A, C, D, F, J, K] dan *CLOSED* = [S, E, B]. Gambar 2.4 menjelaskan langkah ketiga dari pencarian rute terpendek.



Gambar 4. Langkah ketiga

4. Langkah keempat

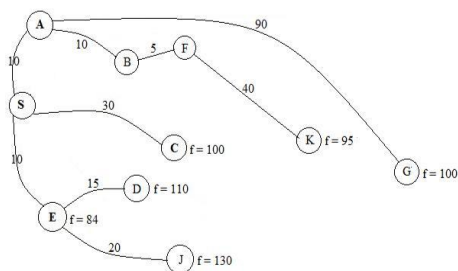
A dengan jarak terkecil (yaitu 90) terpilih sebagai *BestNode* dan dipindahkan ke *CLOSED*. Selanjutnya, semua suksesor A dibangkitkan, yaitu: B dan G. Karena belum pernah ada di *OPEN* maupun *CLOSED*, maka G dimasukkan ke *OPEN*. Sedangkan simpul B sudah ada di *CLOSED*, maka harus dicek apakah *parent* dari B perlu diganti atau tidak. Ternyata, jarak dari S ke B melalui A (yaitu 10 + 10 = 20) lebih kecil daripada jarak dari S ke B (yaitu 25). Oleh karena itu, *parent* dari B harus diubah, yang semula S menjadi A, nilai *g* dan *f* pada B juga harus diperbarui (nilai *g* yang semula 25 menjadi 20, dan nilai *f* dari 85 menjadi 80). Dalam kasus ini, B hanya mempunyai dua jalur, yaitu F dan K. Nilai *g*(F) yang semula 30 diubah menjadi 25, dan nilai *f*(F) dari 100 menjadi 95. Nilai *g*(K) yang semula 75 diubah menjadi 70, dan nilai *f*(K) dari 105 menjadi 100. Akhirnya, *OPEN* = [C, D, F, G, J, K] dan *CLOSED* = [S, E, B, A]. Gambar 5 menjelaskan langkah keempat dari pencarian rute terpendek.



Gambar 5. Langkah keempat

5. Langkah kelima

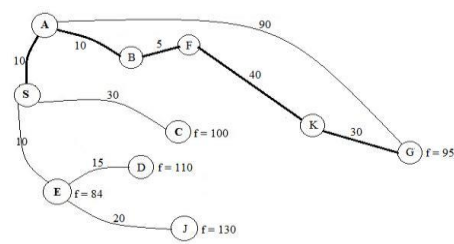
F dengan jarak terkecil (yaitu 95) terpilih sebagai *BestNode* dan dipindahkan ke *CLOSED*. Selanjutnya, semua suksesor F dibangkitkan, yaitu: K. Karena K sudah ada di *OPEN*, maka harus dicek apakah *parent* dari K perlu diganti atau tidak. Jarak dari S ke K melalui F ternyata lebih kecil daripada jarak dari S ke K melalui *parent* lama (B). Oleh karena itu, *parent* dari K harus diubah, yang semula B menjadi F. Selanjutnya, nilai $g(K)$ yang semula 70 diubah menjadi 65, dan nilai $f(K)$ dari 100 menjadi 95. Akhirnya, $OPEN = [C, D, F, G, J, K]$ dan $CLOSED = [S, E, B, A, F]$. Penjelasan langkah kelima dari pencarian rute terpendek ditunjukkan pada Gambar 2.6.



Gambar 6. Langkah kelima

6. Langkah keenam

K dengan jarak terkecil (yaitu 95) terpilih sebagai *BestNode* dan dipindahkan ke *CLOSED*. Selanjutnya, semua suksesor K dibangkitkan, yaitu: G. Karena G sudah ada di *OPEN*, maka harus dicek apakah *parent* dari G perlu diganti atau tidak. Jarak dari S ke G melalui K ternyata lebih kecil daripada jarak S ke G melalui *parent* lama (A). Oleh karena itu, *parent* dari G harus diubah, yang semula A menjadi K. Selanjutnya, nilai $g(G)$ yang semula 100 diubah menjadi 95, dan nilai $f(G)$ dari 100 menjadi 95. Pada akhir langkah keenam ini, $OPEN = [C, D, G, J]$ dan $CLOSED = [S, E, B, A, F, K]$. Gambar 2.7 menjelaskan langkah keenam dari pencarian rute terpendek.



Gambar 7. Langkah keenam Selanjutnya, G dengan jarak terkecil (yaitu 95)

terpilih sebagai *BestNode*. Karena *BestNode* sama dengan *goal*, berarti solusi sudah ditemukan. Rute dan total jarak bisa ditelusuri balik dari G menuju S karena setiap simpul hanya memiliki satu *parent* dan setiap simpul memiliki informasi jarak sebenarnya (g). Penelusuran menghasilkan rute S-A-B-F-K-G dengan total jarak sama dengan 95. (Suyanto:2014)

2.4 PHP

PHP Hypertext Preprocessor ini merupakan bahasa yang hanya dapat berjalan pada server yang hasilnya dapat ditampilkan pada klien. Interpreter PHP dalam mengeksekusi kode PHP pada sisi server (disebut server-side) berbeda dengan mesin maya Java yang mengeksekusi program pada sisi klien (client-side). PHP merupakan bahasa standar yang digunakan dalam dunia web site. PHP adalah bahasa program yang berbentuk script yang diletakkan di dalam server web. Jika kita lihat dari sejarah, mulanya PHP diciptakan dari ide Rasmus Lerdof yang membuat sebuah script perl. Script tersebut sebenarnya dimaksudkan untuk digunakan sebagai program untuk dirinya sendiri. Akan tetapi, kemudian dikembangkan lagi sehingga menjadi sebuah bahasa yang disebut "Personal Home Page". (Bunafit Nugroho, 2004:139)

2.5 Google Maps API

Google Maps API adalah kumpulan API yang memungkinkan untuk menghamparkan data di Google Map yang disesuaikan. Dengan menggunakan Google Maps dapat membuat aplikasi web dan seluler yang menarik dengan platform pemetaan canggih dari Google, termasuk data citra satelit, Street View, profil ketinggian, petunjuk arah mengemudi, peta bergaya, analisis, dan basis data tempat yang luas.

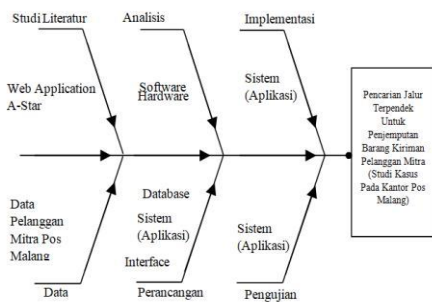
Manfaat dari Google Maps API yaitu mudah dan familiar karena Google Maps merupakan layanan pemetaan online yang paling banyak digunakan di seluruh penjuru dunia, dengan lebih dari 2 juta situs dan aplikasi. Selain itu, manfaat dari Google Maps API adalah katalog global yang berisi citra udara beresolusi tinggi yang ditawarkan oleh Google Earth juga tersedia di Google Maps API. Google Maps API

menyertakan akses ke citra 45 derajat di lebih dari 120 kota di seluruh dunia, sehingga pengguna dapat menjelajahi data dengan perspektif yang unik dalam area tertentu serta Google Maps API memudahkan pembuatan aplikasi atau perangkat lunak berbasis lokasi yang diterapkan dalam bentuk browser web, perangkat selular, dan aplikasi yang dibuat secara khusus

3. Metodologi Penelitian

3.1 Fishbone Diagram

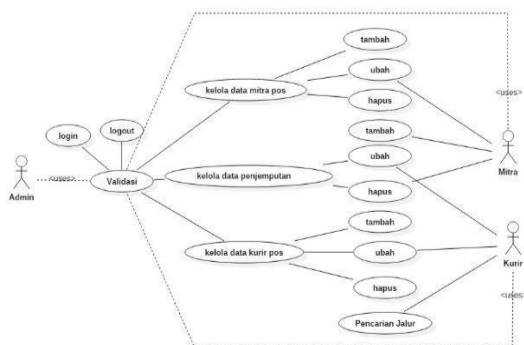
Dalam pembuatan Aplikasi Pencarian Jalur Terpendek Untuk Penjemputan Barang Kiriman Pelanggan Mitra (Studi Kasus Pada Kantor Pos Malang) metodologi pengerjaannya dapat dilihat pada gambar 8 dibawah ini:



Gambar 8. Fishbone

3.1.1 Use Case Diagram

Use case diagram adalah diagram yang menyajikan interaksi antara usecase dan aktor. Dimana aktor dapat berupa orang, peralatan atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. Use case menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai

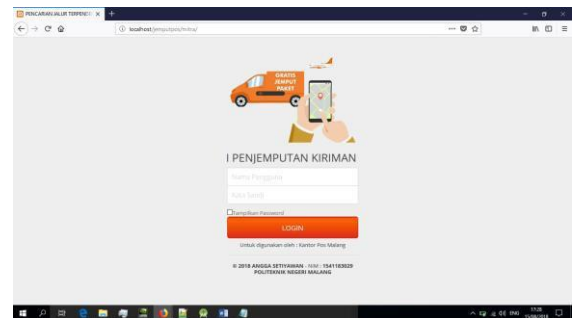


Gambar 9. Use Case Diagram

4. Implementasi dan Pengujian

4.1 Implementasi Antar Muka Pengguna

Halaman Antar Muka Mitra adalah rancangan yang digunakan untuk diterapkan pada tampilan antar muka aplikasi yang digunakan oleh Mitra Kantor Pos Malang, *User Interface* ini dirancang untuk diterapkan pada layar monitor *Personal Computer (PC)* akan tetapi dapat menyesuaikan sesuai dengan ukuran dan jenis layar yang akan menampilkannya.



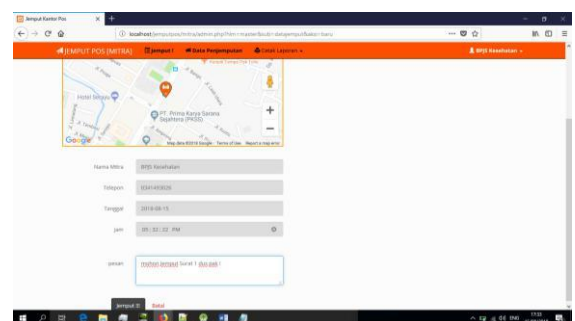
Gambar 10. Login

Login digunakan untuk memvalidasi pengguna sebelum masuk kedalam system untuk memastikan keabsahan pengguna aplikasi.



Gambar 11. Dashboard

Dashboard adalah halaman administrator pengguna yangmana dapat memilih menu fasilitas layanan seperti pengajuan penjemputan ataupun cetak riwayat penjemputan.



Gambar 12. Form Jemput Paket

Form jemput paket merupakan layanan yang digunakan mitra untuk melakukan permohonan penjemputan paket.

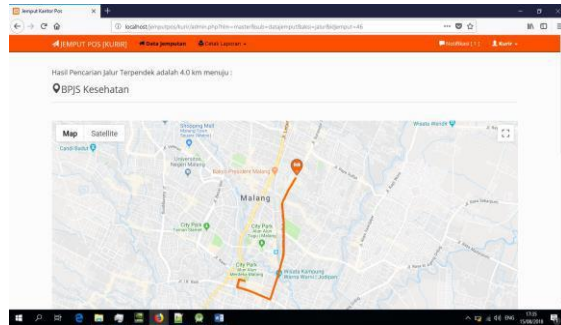
4.2 Implementasi Algoritma A*

Prinsip algoritma ini adalah mencari jalur terpendek dari sebuah simpul awal (starting point) menuju simpul tujuan dengan memperhatikan jarak (F) terkecil. Algoritma A* (A Star) memperhitungkan *cost* dari *current state* ke tujuan dengan fungsi heuristik, Algoritma ini juga mempertimbangkan *cost* yang telah ditempuh selama ini dari *initial state* ke *current state*. Jadi jika ada jalan yang telah ditempuh sudah terlalu panjang dan ada jalan lain yang *cost*-nya lebih kecil tetapi memberikan posisi yang sama dilihat dari *goal*, jalan yang lebih pendek yang akan dipilih.

Berikut adalah langkah dalam melakukan perhitungan Algoritma A* (A Star):

1. Masukkan node awal ke openlist
2. Loop langkah - langkah di bawah ini :
 - a. Cari *node* (*n*) dengan nilai *f(n)* yang paling rendah dalam *open list*. *Node* ini sekarang menjadi *current node*.
 - b. Keluarkan *current node* dari *open list* dan masukan ke *close list*.
 - c. Untuk setiap tetangga dari *current node* lakukan berikut :
 - Jika tidak dapat dilalui atau sudah ada dalam *close list*, abaikan.
 - Jika belum ada di *open list*. Buat *current node parent* dari *node* tetangga ini. Simpan nilai *f, g* dan *h* dari *node* ini.
 - Jika sudah ada di *open list*, cek bila *node* tetangga ini lebih baik, menggunakan nilai *g* sebagai ukuran. Jika lebih baik ganti *parent* dari *node* ini di *open list* menjadi *current node*, lalu kalkulasi ulang nilai *g* dan *f* dari *node* ini.
 - d. Hentikan *loop* jika :
 - *Node* tujuan telah ditambahkan ke *open list*, yang berarti rute telah ditemukan.
 - Belum menemukan *node goal* sementara *open list* kosong atau berarti tidak ada rute.
3. Simpan rute. Secara 'backward' urut mulai dari *node goal* ke *parent*-nya terus sampai mencapai *node* awal sambil menyimpan *node* ke dalam sebuah *array*.

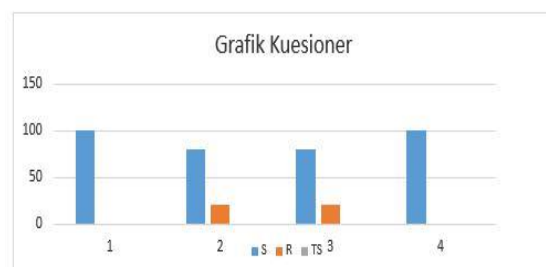
Hasil rute dan jarak pada aplikasi pada kurir ditunjukkan pada Gambar 10.



Gambar 10. Hasil Rute Terpendek

4.3 Hasil Uji Coba Dengan Responden

Untuk menguji sistem dari segi tampilan, *user friendly*, kehandalan atau keakuratan serta manfaat dari aplikasi yang dikembangkan, telah dilakukan penyebaran kuesioner kepada 10 responden. Responden yang dimaksud adalah Bagian IT, Bagian SPLK dan Kurir Penjemput Kiriman Kantor Pos Malang serta Mitra Pos. Pendapat dari responden tersebut ditampilkan pada Gambar 11.



Gambar 11. Grafik Presentase Kuesioner

5. Kesimpulan Dan Saran

5.1 Kesimpulan

Berdasarkan hasil dari perancangan Pencarian Jalur Terpendek Untuk Penjemputan Barang Kiriman Pelanggan Mitra (Studi Kasus Pada Kantor Pos Malang) dapat di ambil beberapa kesimpulan, yaitu: dengan menggunakan data yang dihasilkan dari uji coba dan dari hasil kuesioner yang disebarkan kepada 10 responden, aplikasi mampu berjalan dengan baik secara fungsional untuk menampilkan rute menuju lokasi mitra pemohonan penjemputan kiriman menggunakan algoritma A*(A star), dapat membantu baik Kurir, Administrator (Bagian SPLK), IT Support Kantor Pos Malang maupun terhadap Pelanggan Mitra Kantor Pos Malang berdasarkan kriteria yang diinginkan. 100 % responden menyatakan bahwa aplikasi ini memiliki tampilan menarik dan mudah dipahami serta memiliki proses yang cepat dalam pencarian rute dan membantu kurir mendapatkan rute terpendek menuju pelanggan mitra ,dan 80% menyatakan aplikasi mudah digunakan dan membantu melakukan permintaan penjemputan.

5.2 Saran

Saran terhadap pengembangan aplikasi ini di masa mendatang adalah, agar menerapkan multipoint terhadap data penjemputan kiriman, dan juga diterapkan GPS agar mengetahui lokasi kurir secara *realtime*, serta di tambahkan zonasi untuk kurir penjemput ketika kurir dan mitra mengalami pertambahan pesat.

Daftar Pustaka:

Binus.ac.id. (2018, Februari 05) "Fishbone Diagram". [online]. Available : <https://sis.binus.ac.id/2017/05/15/fishbone-diagram>.

- Bunafit Nugroho. 2004. PHP dan MySQL dengan editor Dreamweaver MX. ANDI Yogyakarta, Yogyakarta.
- Mufti, Yusuf. 2015. Panduan Mudah Pengembangan Google Map Android. ANDI OFFSET, Yogyakarta.
- Posindonesia.co.id. (2018, Februari 05) Sejarah Pos Indonesia [online]. Available : <http://www.posindonesia.co.id/index.php/sejarah-h-pos/>.
- Republik Indonesia. 2009. UU No. 38 Tahun 2009 Tentang Pos. Undang - Undang Republik Indonesia.
- Wahyu Rian S. 2017. Penerapan Algoritma A*(Star) Untuk Pencarian Lokasi Gedung Di Politeknik Negeri Malang Berbasis Android, Skripsi, Politeknik Negeri Malang.