

# Aplikasi Pemilihan Rute Pengiriman Barang pada Perusahaan Elektronik di Surabaya dengan Menggunakan Metode *K-Means Clustering* Dan *Google Maps API*

Liliana Ester<sup>1</sup>, Rolly Intan<sup>2</sup>, Andreas Handoyo<sup>3</sup>

Program Studi Sistem Informasi Bisnis, Fakultas Teknologi Industri, Universitas Kristen Petra

Jl. Siwalankerto 121 – 131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) – 8417658

E-mail: lili.esterw@gmail.com<sup>1</sup>, rintan@petra.ac.id<sup>2</sup>, handoyo@petra.ac.id<sup>3</sup>

## ABSTRAK

Perusahaan elektronik yang dituju merupakan perusahaan yang bergerak dalam bidang retail barang elektronika. Perusahaan berusaha untuk melakukan peningkatan kepuasan pelanggan dengan melakukan evaluasi dan perbaikan pada proses bisnis yang ada. Salah satunya adalah proses pengiriman barang. Permasalahan yang muncul pada proses ini yaitu susah menentukan urutan rute pengiriman barang, terutama jika dalam suatu pengiriman terdapat pelanggan yang memiliki preferensi jam pengiriman. Selain itu, perusahaan juga tidak bisa memberikan informasi kepada pelanggan mengenai waktu estimasi kedatangan barang. Oleh karena itu dibutuhkan suatu sistem informasi pengiriman barang yang mampu memberikan rekomendasi urutan rute pengiriman barang dan informasi mengenai estimasi waktu pengiriman barang.

Sistem informasi pengiriman barang dibuat dengan basis *website* dengan menggunakan *framework Codeigniter*. Pada sistem ini, proses pertama yang dilakukan yaitu mencocokkan alamat pengiriman dengan alamat hasil response dari *Google Maps API*. Selanjutnya program akan melakukan kalkulasi terhadap setiap alamat pengiriman. Dalam usaha untuk melakukan perhitungan, *Google Maps API* menyediakan layanan *service* bebas biaya yang mampu memberikan urutan rute terbaik dengan keterbatasan maksimum 8 titik alamat. Namun, *service* ini tidak dapat langsung digunakan karena jumlah alamat dalam satu pengiriman biasanya berjumlah lebih dari 8 alamat. Oleh karena itu dibutuhkan metode *K-Means Clustering* untuk mengelompokkan area pengiriman menjadi 8 titik area. Titik area tersebut nantinya digunakan untuk mengurutkan rute pengiriman.

Hasil akhir dari program ini yaitu suatu sistem informasi pengiriman barang untuk pengurus/administrator dan supir perusahaan. Sistem ini mampu memberikan rekomendasi untuk pemilihan rute pengiriman barang dan informasi mengenai estimasi waktu pengiriman barang. Dari hasil pengujian yang dilakukan, sistem dapat melakukan perhitungan rute dengan waktu rata-rata kalkulasi 133 detik untuk 30 rute tujuan.

**Kata Kunci:** *K-Means*, *Google Maps API*, Pengiriman Barang

## ABSTRACT

*Company X is an electronics retailer company operated in Surabaya city. The company is trying to improve their customer satisfaction by doing some evaluations and enhancement on their existing business processes. One of them is the process of goods*

*delivery services. Some shortcomings occur in the process. For example, it is difficult to determine the order of the delivery routes, especially if there are some customers who have preference delivery hours. Another thing, the company cannot provide any information about the time estimation of the goods arrival that makes their customer feel bad about the delivery services. Because of these problems, the company needs a system that is capable of providing the route sequence recommendation and also giving information about estimation of goods arrival.*

*The system is implemented on website by using codeigniter framework and SQL Server as the database. In this system, the first process is matching the addresses with the response from Google Maps API. This Google Maps API responses are being used in the next process which is calculating route. In order to perform the calculations, Google Maps API has already provided a free of charge service that is capable of giving the best waypoint through the maximum of 8 points addresses. However, the amount of the addresses in one shipment is usually more than 8 addresses. Therefore a method of clustering using K-Means is needed in order to make a group of delivery area into 8 clusters. By then, the route calculation can be done by using the clusters.*

*The end result of this program is a shipping information system for administrators and drivers of the company. This system is able to provide recommendations for the selection of goods delivery routes and information on the estimated time of goods delivery. Based on some testings, the system can perform route calculations with an average time calculation of 133 seconds for 30 destination routes.*

**Keywords:** *K-Means*, *Google Maps API*, *Delivery Service*

## 1. PENDAHULUAN

Saat ini, tingkat persaingan di dalam dunia bisnis semakin ketat. Setiap perusahaan dituntut untuk bisa memberikan nilai yang lebih baik untuk dapat mempertahankan bisnisnya. Salah satu cara yang dapat dilakukan yaitu dengan melakukan evaluasi dan/ atau perubahan terhadap proses bisnis yang berjalan pada perusahaan.

Di kota Surabaya, terdapat suatu perusahaan elektronik yang bergerak dalam bidang *retail* barang elektronika. Perusahaan berusaha untuk melakukan evaluasi terhadap proses bisnisnya. Evaluasi dilakukan dengan melihat permasalahan-permasalahan yang terjadi pada saat ini. Adapun salah satu permasalahan yang terjadi yaitu, permasalahan pada proses pengiriman barang. Perusahaan berusaha untuk dapat melakukan perbaikan dan pembaruan terhadap layanan pengiriman barang tersebut.

Suatu layanan pengiriman barang dapat dikatakan baik apabila barang tersebut bisa diantar tepat waktu dan sesuai dengan estimasi pengiriman yang telah disepakati. Selain itu, jika dilihat dari segi bisnis, pengiriman yang dilakukan haruslah dilakukan dengan biaya pengiriman seminimal mungkin sehingga menghasilkan keuntungan yang lebih besar. Dengan terjawabnya dua kriteria tersebut, maka layanan pengiriman barang dapat memberikan hasil yang memuaskan bagi pelanggan perusahaan.

Dalam kenyataannya, tidak bisa di pungkiri bahwa terjadi berbagai permasalahan terkait pengiriman barang pada perusahaan elektronik ini. Banyaknya tujuan pengiriman yang dituju, seringkali menyebabkan supir susah untuk menentukan rute tujuan yang mana yang akan di dahulukan untuk bisa mencapai rute yang optimal. Selain itu, kemacetan juga dapat menyebabkan barang terlambat untuk dikirim tepat waktu, dimana tentu saja akan berdampak pada penumpukan barang-barang kiriman lainnya yang ikut tertunda. Di sisi lain, perusahaan elektronik ini juga seringkali tidak dapat memberikan kepastian estimasi pengiriman barang kepada pelanggan mereka sehingga menyebabkan pelanggan kurang puas dengan adanya pelayanan pengiriman barang yang tidak informatif tersebut. Lebih dari pada itu, ketika pelanggan tidak mengetahui mengenai informasi estimasi barang sampai pada rumah pelanggan, maka seringkali terjadi barang dikirim pada rumah yang kosong atau penghuni rumah sedang tidak berada di tempat. Hal ini menyebabkan supir terpaksa kembali mengirim barang pada lain waktu atau hari. Dimana hal ini menyebabkan penambahan biaya atau *cost* untuk dapat melakukan pengiriman kembali.

Berdasarkan uraian di atas, pada penelitian ini akan dilakukan pembuatan sistem informasi yang mampu memberikan rekomendasi urutan rute pengiriman barang dan informasi mengenai estimasi pengiriman barang kepada pelanggan. Dengan adanya sistem yang didukung dengan informasi dari *Google Maps API* ini, diharapkan supir dapat mengambil rute yang lebih optimal sehingga mampu mengantisipasi jam-jam kemacetan di jalan yang dilalui. Sedangkan di sisi lain, pengurus pengiriman barang dapat mengetahui status pengiriman barang dan estimasi kedatangan barang dengan lebih akurat.

## 2. TINJAUAN PUSTAKA

### 2.1 Servis Pengiriman Perusahaan

Perusahaan elektronik yang dituju melakukan kegiatan utama perusahaan yaitu jual-beli barang-barang elektronik. Dalam usaha perusahaan untuk meningkatkan kepuasan pelanggan, perusahaan ini menyediakan layanan pengiriman barang. Tetapi dalam pelaksanaannya, masih banyak kekurangan atau missed yang terjadi. Tidak lain hal ini disebabkan karena tidak adanya informasi pendukung yang mampu memudahkan dalam pengambilan keputusan baik dalam *level* manajemen ataupun supir pengirim barang.

Menurut Orunga [7], servis pengiriman sudah menjadi suatu kebutuhan untuk masyarakat perkotaan. Peningkatan pelayanan servis, penting dilakukan untuk memuaskan baik pihak pelanggan ataupun perusahaan demi mengakomodasi trend yang akan datang. Pada sisi pelanggan, Orunga juga berpendapat bahwa pengiriman dikatakan baik apabila barang dapat dikirim sesuai jam permintaan yang diberikan dan ketika pelanggan ada dan bisa menerima barang kiriman. Dengan demikian pelanggan memiliki kepastian jam, sehingga tidak perlu menunggu lama sampai barang tiba.

### 2.2 K-Means Clustering

*K-Means clustering* merupakan suatu metode untuk mengelompokkan dua atau lebih data yang memiliki kesamaan karakteristik. Dengan adanya pengelompokkan ini, maka data yang berada pada satu cluster memiliki tingkat variasi karakteristik yang kecil [2].

Pada permasalahan pencarian rute ini, clustering didasarkan pada jarak antar masing-masing titik yang ada. Dengan menggunakan metode ini, maka  $n$  titik destinasi pengiriman dapat digolongkan dalam suatu kelompok sehingga dapat membantu mempercepat proses perhitungan kombinasi rute.

Berikut merupakan langkah-langkah melakukan *clustering* [9] :

1. Tentukan jumlah *cluster*  $k$ .
2. Pusat-pusat *cluster* diberi nilai dengan angka *random*.
3. Alokasikan semua data ke *cluster* terdekat. Kedekatan antar data dapat dinilai dari jarak antar kedua data tersebut. Sama halnya dengan kedekatan data dengan *cluster*. Setiap data yang ada akan dihitung jaraknya dengan *masing-masing cluster*. Jarak terdekat antar data dan *cluster* menentukan pada *cluster* mana data akan digolongkan. Untuk dapat menghitung jarak antar data, digunakan pemanggilan *service API* dari *Google Maps* untuk dapat mengetahui jarak antar kedua titik dengan mempertimbangkan rute yang ada.
4. Dilakukan perhitungan kembali untuk nilai tengah/*centroid* dari *cluster* dengan cara mengambil nilai rata-rata dari setiap data dalam *cluster*.
5. Kembali lakukan alokasi data kedalam *cluster* yaitu pada langkah nomor 3.

Faktor utama yang dapat mempengaruhi kinerja algoritma K-Means adalah sebagai berikut [1] :

1. Pemilihan titik *centroid* awal.
2. Perkiraan jumlah *cluster*.

### 2.3 Google Maps API

API merupakan rangkaian instruksi atau program yang digunakan untuk memberikan akses atau layanan yang berbasis *web*. Dengan adanya API, maka *software developer* lain dapat memanfaatkan layanan yang tersedia untuk mengembangkan aplikasi yang sedang dibuat. Untuk dapat memanggil API, dibutuhkan suatu metode pemanggilan dengan memberikan parameter tertentu.

*Google Maps* memiliki layanan *service API* yang cukup banyak dan lengkap. Terdapat API khusus yang disediakan untuk membangun program dengan basis *Android*, *iOS*, *Web API (javascript)*, dan *Web Service APIs*. Layanan yang diberikan pun cukup beragam diantaranya yaitu layanan untuk menggambarkan peta pada *website*, serta memberikan informasi kepadatan lalu lintas pada peta tersebut, memberikan gambaran rekomendasi rute yang harus diambil untuk suatu titik destinasi yang ingin dikunjungi, memberikan informasi jarak dan waktu tempuh antar kedua titik destinasi, menghitung urutan rute terbaik ketika diberikan  $n$  titik destinasi, dan sebagainya.

Untuk dapat menggunakan layanan yang diberikan, pengguna harus memiliki *API Key* terlebih dahulu. Kemudian mengaktifkan *service-service* yang diinginkan pada *google console*. Pemanggilan *Google Maps API* tidak berbayar atau gratis sampai dengan jumlah pemanggilan tertentu. Setiap *service* memiliki ketentuan pemanggilan yang berbeda antar satu *service* dengan *service* lainnya.

### 2.3.1 Google Maps Directions API

Service *Google Maps Direction API* ini seringkali digunakan untuk mendapatkan petunjuk arah perjalanan antar satu titik lokasi dengan titik lainnya. Parameter utama yang harus diberikan yaitu titik awal lokasi dan titik tujuan perjalanan. Parameter lokasi tersebut dapat berupa alamat, nilai titik *latitude* dan *longitude*, atau *place\_id* yang didapat dari hasil response *service Google Maps API Place Autocomplete* and *Directions*. Secara *default*, *service* menganggap bahwa perjalanan dilakukan dalam mode *driving*. Hal ini akan berpengaruh terhadap direksi perjalanan, yakni *google* hanya akan memberikan jalanan yang dapat dilalui oleh mobil atau disebut dengan *road network*. Walaupun demikian, mode tersebut dapat diganti dengan mengganti nilai parameter dengan mode *walking /bicycling*.

Selain parameter di atas, *service* ini juga dilengkapi dengan parameter *via* jika pengguna ingin melakukan perjalanan dengan melalui jalan tertentu, dan parameter *waypoints* jika destinasi perjalanan lebih dari satu titik lokasi. Pada *service* ini, *google* juga dapat mengurutkan perjalanan yang dilakukan sehingga urutan perjalanan menjadi terurut dan tidak berputar-putar.

### 2.3.2 Google Maps Distance Matrix API

Service *distance matrix* ini digunakan untuk mengetahui estimasi perjalanan yang dilakukan. Estimasi yang diberikan merupakan prediksi kemacetan yang berasal dari data *historical traffic conditions* dan data *live traffic*. Hasil estimasi yang diberikan digolongkan menjadi 3 mode yaitu, mode *best\_guess*, *pessimistic*, dan *optimistic*. Secara *default*, *google* memberikan hasil *response* dengan parameter *best\_guess*, yakni hasil estimasi yang sekiranya mampu memberikan nilai estimasi terbaik sesuai dengan waktu *travel* yang diberikan.

Pada *service* ini, parameter utama yang harus diberikan yakni, parameter *departure\_time*. Jika pada *request API* sudah terdapat parameter tersebut, maka hasil *response* akan memberikan informasi mengenai waktu tempuh sesuai *departure\_time* yang diberikan.

## 2.4 Travelling Salesman Problem (TSP)

*Travelling Salesman Problem* (TSP) merupakan suatu problema permasalahan dalam mencari rute terpendek terhadap  $n$  kota ketika diberikan *list* kota dan jarak antar kota tersebut [5]. Setiap kota dikunjungi tepat satu kali dan kembali ke kota keberangkatan yang sama. Permasalahan TSP akan mudah diselesaikan jika nilai  $n$  kota kecil, yakni dibawah angka 5. Tetapi ketika kota yang dikunjungi lebih dari 20 kota, maka kemungkinan kombinasi rute akan berlipat ganda dan sangat susah untuk diselesaikan secara manual.

TSP tergolong sebagai *NP-hard problem*, yaitu suatu kelompok permasalahan yang tidak memiliki algoritma tertentu untuk mendapatkan nilai optimal dalam kurun waktu *polynomial* [6]. Adapun jika terdapat suatu algoritma yang dapat mencari nilai optimal, maka dibutuhkan waktu yang sangat lama, terutama jika masalah yang ingin dipecahkan sangat besar. Dalam hal ini, algoritma paling sederhana yang digunakan yaitu *brute force*, yaitu suatu algoritma permutasi untuk mencari setiap kombinasi kemungkinan yang ada. Sehingga jika terdapat  $n$  titik, maka kemungkinan yang dihasilkan yakni  $(n - 1)!$  kemungkinan urutan rute. Oleh karena itu, untuk menyelesaikan *NP-hard problem* dalam kehidupan sehari-hari, maka seringkali pencarian yang dilakukan bukan untuk mencari nilai optimal, melainkan untuk

mencari nilai yang dapat dikatakan “cukup” baik yang dapat dicari dalam kurun waktu tertentu.

## 2.5 Travelling Salesman Problem with Time Windows (TSP-TW)

*Travelling Salesman with Time Windows* atau yang disingkat dengan TSP-TW merupakan suatu bentuk pengembangan dari TSP. TSP-TW melibatkan dua atau lebih *variable* untuk mencari rute optimal dengan mempertimbangkan total waktu perjalanan, waktu pengiriman, waktu pelayanan, dan waktu kedatangan [4]. Dalam hal ini, maka setiap pelanggan akan memiliki *time windows* yang berbeda antar satu pelanggan dengan yang lainnya. Pelanggan memiliki selang waktu batas awal dan akhir untuk menyatakan bahwa pada waktu tersebut, ia bisa dikunjungi. Dengan adanya *time windows* maka optimasi rute akan semakin susah dilakukan, karena optimasi harus mempertimbangkan *time windows* yang dimiliki oleh setiap pelanggan.

## 2.6 Review Jurnal Serupa

Menurut R. Nallusamy, et all [8], dalam *paper*-nya melakukan perhitungan optimasi rute terhadap *multiple Vehicle Routing Problems* (mVRP). Sebelum melakukan perhitungan kombinasinya, dilakukan pengelompokan dengan menggunakan metode *K-Means clustering* terhadap setiap kota yang dikunjungi. Kemudian dilakukan optimasi dengan menggunakan metode dari *Genetic Algorithm* (GA).

Hasil konklusi dari jurnal ini mengatakan bahwa metode *K-Means clustering* terbukti efektif untuk mengelompokan kota-kota yang dikunjungi menjadi beberapa *cluster* tertentu secara optimal dan dalam kurun waktu yang cukup singkat. Selain itu juga disebutkan bahwa metode GA memiliki waktu eksekusi yang lebih cepat jika dibandingkan metode lain seperti *Branch and Bound*, *Branch and Cut*, dan *Cut and Solve Techniques*. Serta GA dapat memberikan hasil yang mendekati nilai *global optimal*.

Menurut Chia, W. C., et all [3] dalam jurnalnya berjudul “*Trip Planning Route Optimization With Operating Hour And Duration Of Stay Constraints*” mengusulkan suatu metode untuk menjadwalkan rute perjalanan liburan yang optimal untuk memaksimalkan jumlah tempat yang ingin dikunjungi dengan durasi dan waktu kunjung tertentu secara *offline*. Adapun beberapa *variable* yang perlu diperhatikan seperti jam operasional tempat hiburan, *time travel*, durasi waktu maksimal untuk berada pada tempat hiburan, serta jarak antar tempat hiburan. Jurnal ini kemudian membandingkan hasil yang diperoleh ketika perhitungan didasarkan pada perhitungan *straight-line distance approximation* dengan menggunakan titik *latitudinal* dan *longitudinal* untuk menghitung jarak *spherical* di antara keduanya dengan perhitungan yang menggunakan data *travel distance* dan *travel time between places* hasil *response* dari *Google Maps Direction API*.

Hasil konklusi mengatakan bahwa kedua metode berhasil memenuhi dua kriteria utama, yakni kepastian kedatangan pada suatu tempat hiburan berada pada jam operasional yang diberikan serta dengan durasi waktu yang cukup. Sedangkan hasil pengujian metode, mengatakan bahwa metode *straight-line distance approximation* memiliki 15.1% *time penalty* jika dibandingkan dengan *waypoint optimization* dengan *Google Maps API*.

### 3. ANALISIS DAN DESAIN

#### 3.1 Analisis Permasalahan

Permasalahan yang terjadi di PT. X yakni:

1. PT. X tidak dapat memantau setiap pengiriman yang terjadi dalam waktu bersamaan. Hal ini dikarenakan PT. X belum memiliki sistem yang mampu memberikan informasi perkembangan pengiriman setiap armada pengiriman barang perusahaan. Oleh karena itu jika perusahaan ingin mengetahui status pengiriman suatu barang, maka pengurus pengiriman barang harus menanyakan secara langsung dengan supir pengiriman barang tersebut.
2. Banyaknya tujuan pengiriman barang menyebabkan supir harus melakukan spekulasi mengenai rute yang paling efektif berdasarkan dugaannya.
3. Supir pengirim barang tidak memiliki informasi mengenai arus lalu lintas yang akan dilewatinya sehingga seringkali supir terjebak dalam kemacetan.
4. Setiap pengiriman barang yang tertunda akan berdampak pada penjadwalan ulang, baik barang tersebut dikirim pada hari yang sama atau di hari lainnya. Penundaan pengiriman barang menyebabkan berkurangnya efisiensi pengiriman barang serta menimbulkan penambahan biaya dalam pengiriman barang.
5. Perusahaan tidak dapat memberikan informasi secara detail mengenai estimasi barang akan sampai pada tempat tujuan. Dalam hal ini, pelanggan hanya memiliki sedikit informasi mengenai barang akan tiba pada tempat yang telah disepakati. Selain itu, pelanggan juga akan memiliki rasa waswas jikalau barang tidak dapat dikirim tepat waktu karena suatu alasan tertentu.

#### 3.2 Analisis Kebutuhan

Dari permasalahan-permasalahan yang terjadi, maka dapat disimpulkan bahwa PT. X membutuhkan suatu sistem yang mampu memberikan informasi yang mendukung kegiatan pengiriman barang kepada pelanggan perusahaan. Sistem yang dibutuhkan mempunyai kriteria sebagai berikut:

1. Sistem dapat memberikan informasi mengenai proses perkembangan pengiriman yang dilakukan oleh supir pengirim barang. Dengan adanya sistem ini, perusahaan dapat memonitor kegiatan pengiriman barang kepada pelanggan.
2. Sistem ini juga harus dapat menghasilkan report pengiriman barang ketika supir telah selesai menyelesaikan semua pengirimannya. Report yang dihasilkan akan berguna untuk membantu perusahaan dalam mengambil langkah strategis.
3. Sistem mampu melakukan perhitungan mengenai rute tujuan pengiriman barang, sehingga dapat membantu supir dalam menentukan urutan rute pengiriman barang. Dengan adanya sistem ini, supir juga dapat mengatur peletakan barang yang akan dikirim, sehingga mempermudah supir saat ingin menurunkan barang pada tempat tujuan.
4. Sistem mampu memberikan informasi mengenai estimasi barang akan sampai di tempat pelanggan.

#### 3.3 DESAIN SISTEM

Desain sistem digambarkan dalam bentuk *diagram usecase* pada 1.1.1.1.a.i.1.a. Gambar 1. Pada diagram tersebut terdapat 2 *actor* yang berperan dalam aplikasi yang dibuat, yakni *admin* dan *driver*. *Admin* berperan sebagai *administrator* dalam aplikasi, yakni sebuah tim yang akan mengatur segala keperluan pada aplikasi. Sedangkan *driver*, akan berperan sebagai *user* atau

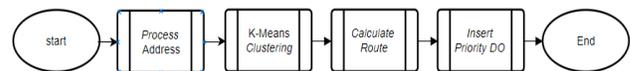
pengguna aplikasi. Secara garis besar, *user* akan menerima hasil perhitungan rute, kemudian melakukan *review* dari rute yang diberikan.



Gambar 1. Usecase Diagram Perancangan Sistem

#### 3.4 Desain Flowchart

Desain *flowchart* digambarkan untuk memberikan penggambaran mengenai sistem perhitungan rute yang dilakukan. Pada Gambar 2, terdapat 4 proses besar yang akan dilakukan guna mendapatkan urutan rute pengiriman barang.



Gambar 2. Flowchart Perhitungan Rute

##### 3.4.1 Process Address

Pada proses ini, setiap data alamat yang ada harus memiliki data alamat baru yang diberikan dari *response Google Maps API*. Jika data belum memiliki alamat tersebut, maka proses selanjutnya tidak dapat dilakukan. Adapun kegunaan dari alamat tersebut adalah untuk mendapatkan titik *latitude* dan *longitude* alamat. Titik-titik ini akan digunakan sebagai perhitungan pada proses berikutnya.

##### 3.4.2 K-Means Clustering

Pada proses ini, setiap alamat akan digolongkan kedalam suatu kelompok cluster tertentu, dengan jumlah *cluster* maksimal yaitu 8 titik. Hal ini dikarenakan parameter untuk *pemanggilan Google Maps API* untuk proses perhitungan rute pada proses berikutnya terbatas sampai dengan 8 titik. Setiap titik *centroid cluster* nantinya akan dipakai untuk melakukan perhitungan urutan rute pada proses *calculation route*.

Kesamaan data pada tiap *cluster* dihitung dari jarak setiap titik alamat terhadap nilai *centroid cluster*. Perhitungan jarak menggunakan nilai titik *latitude* dan *longitude* pada alamat tersebut.

### 3.4.3 Calculate Route

Pada bagian ini dilakukan perhitungan urutan rute pada setiap *cluster*. Kemudian jika *cluster* sudah terurut, berikutnya setiap data dalam *cluster* juga diurutkan dengan menggunakan hasil *response* dari *Google Maps API*.

### 3.4.4 Insert Priority Delivery Order

Priority delivery order merupakan pengiriman yang memiliki waktu ketersediaan khusus. Pada proses ini, dilakukan proses “menyelipkan” data kedalam urutan data alamat yang telah terurut pada proses sebelumnya. Proses dilakukan dengan membandingkan data yang ingin diselipkan dengan setiap kombinasi kemungkinan tempat data memungkinkan untuk diselipkan. Kemudian data akan diselipkan pada tempat dengan minimal cost yaitu dengan jarak terpendek yang didapat.

Proses ini akan mengurangi tingkat efisiensi dan efektifitas dari rute yang sudah terurut sebelumnya. Tetapi proses ini penting dilakukan untuk memenuhi kebutuhan dari pelanggan. Selain itu, proses ini juga mampu meminimalisir kemungkinan barang gagal dikirim karena ditemukan rumah pada alamat tersebut kosong.

## 4. HASIL PENGUJIAN

Pengujian dilakukan untuk memastikan bahwa setiap proses yang dilakukan, berjalan dengan benar. Pengujian dilakukan pada proses *clustering* dan proses memasukan setiap alamat yang memiliki *time windows*. Adapun proses pertama dan ketiga, yakni mengenali alamat pengiriman dan proses perhitungan rute dengan *Google Maps API* tidak dilakukan, karena setiap operasi perhitungan langsung dilakukan oleh *Google* sendiri.

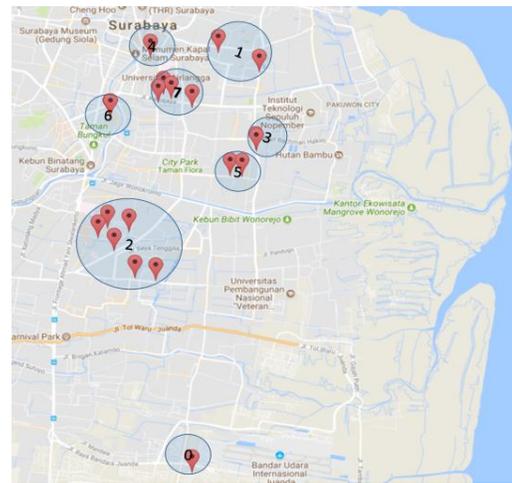
### 4.1 Pengujian Proses Clustering

Pada proses clustering ini, dilakukan proses mengelompokkan titik lokasi alamat pengiriman sesuai dengan titik lokasi yang berdekatan. Pengelompokkan ini dibatasi oleh jumlah titik *centroid* yakni berjumlah 8 titik. Hal ini dikarenakan, hasil dari setiap *cluster* yang terbentuk akan digunakan untuk melakukan perhitungan urutan rute dengan menggunakan *service API* dari *Google Maps*. Dimana pada *service* ini, jumlah titik maksimal yang diperbolehkan mencapai 8 titik lokasi. *Service* mampu memberikan hasil berupa urutan dalam bentuk *array* yang berisikan urutan terbaik untuk melakukan perjalanan dari titik awal, ke titik akhir melalui 8 titik yang diberikan.

Pada pengujian ini, diberikan 20 titik latitude dan longitude alamat pengiriman barang. Titik-titik tersebut di kelompokkan menjadi 8 *cluster*. Hasil dari pengelompokkan dapat dilihat pada Tabel 1. Setelah 8 *cluster* terbentuk, maka *cluster* digambarkan pada peta untuk memastikan bahwa *cluster* yang terbentuk sesuai dengan harapan. Gambar pemetaan tersebut terdapat pada Gambar 3.

Tabel 1. Tabel Pengujian Clustering

No Cluster	Centroid		Jumlah Isi Cluster (titik)	Isi Cluster
	Latitude	Longitude		
0	-7.3841	112.7648	1	[7.3841,112.7648]
1	-7.2682	112.7768	2	[7.2710,112.7820][7.2655,112.7715]
2	-7.3217	112.7472	6	[7.3151,112.7431][7.3292,112.7502][7.3300,112.7555][7.3178,112.7407][7.3217,112.7448][7.3161,112.7486]
3	-7.2931	112.7812	2	[7.2933,112.7813][7.2928,112.7811]
4	-7.2668	112.7541	1	[7.2668,112.7541]
5	-7.3003	112.7761	2	[7.3002,112.7745][7.3005,112.7776]
6	-7.2837	112.7438	2	[7.2836,112.7437][7.2837,112.7439]
7	-7.2791	112.7595	4	[7.2786,112.7594][7.2772,112.7575][7.2796,112.7561]



Gambar 3. Hasil Pemetaan Cluster

Hasil pemetaan berjalan sesuai dengan harapan, yakni tidak ada titik yang tidak mendapatkan *cluster* dan semua titik berada pada *cluster* yang sesuai. Oleh karena itu, pengelompokan dengan metode *clustering k-means* ini mampu memberikan hasil yang cocok dan dapat digunakan.

### 4.2 Pengujian Inserting Priority Delivery Order

Pengujian ini dilakukan untuk mengetahui apakah setiap alamat yang memiliki *time windows* sudah diurutkan sesuai dengan *time windows* yang diberikan. Selain itu, pengujian juga akan melihat sejauh mana *time windows* dapat dipenuhi.

Pada pengujian ini, diberikan data sederhana untuk diuji coba. Data tersebut berisikan 5 alamat pengiriman barang, dimana 2 diantaranya, memiliki *time windows*. Potongan data dapat dilihat pada Tabel 2.

Tabel 2. Tabel Data Alamat Pengiriman dengan Time Windows

ID	Address	Time Windows
2	Jalan Kendangsari Industri No.1	-
3	Jalan Semolowaru No. 45	15:00:00 - 00:00:00
4	Jalan IR.H. Soekarno 35 C	-
5	Jl. Margorejo Indah No.97	08:00:00 - 15:00:00
6	Gubeng Masjid gang 1 No. 22	-

Setiap data yang tidak memiliki *time windows* telah diurutkan oleh sistem pada proses kalkulasi rute. Urutan tersebut yakni, 2-4-6. Selanjutnya, sistem akan melakukan proses untuk setiap data yang memiliki *time windows* dimulai dengan data yang paling atas yakni no ID 3. Pada ID 3, terdapat *time windows* pada bagian *start available time*. Hal ini menandakan bahwa pelanggan baru dapat dikunjungi setelah jam 3 sore. Maka sistem akan mengusahakan agar barang dikirim pada jam tersebut. Karena jam pengiriman terakhir belum melebihi angka 15:00, maka pengiriman terhadap alamat ini di posisikan pada urutan paling terakhir.

Sistem akan berjalan untuk setiap alamat dengan *time windows*. Untuk setiap posisi yang memungkinkan, akan dihitung *total cost*, yakni total jarak antar titik jika alamat tersebut diselipkan pada

posisi tersebut, kemudian akan dicari *cost* dengan nilai paling kecil.

Setelah setiap destinasi dengan *time windows* terproses, maka setiap rute sudah memiliki nomor urutan pengiriman yang disarankan. Hasil urutan tersebut dapat dilihat pada Tabel 3.

**Tabel 3. Hasil Urutan Rute**

No	ID Destinasi 1	ID Destinasi 2	Lat 1	Lng 1	Lat 2	Lng 2	Time Travel (mins)	Time Departure	Distance (Metre)
1	0	2	0	0	-7.329	112.75	14	07:00:00	3650
2	2	4	-7.329	112.8	-7.293	112.78	20	07:44:00	7677
3	4	6	-7.293	112.8	-7.267	112.75	24	08:34:00	7252
4	6	3	-7.267	112.8	-7.3	112.77	23	09:28:00	6582
5	3	5	-7.3	112.8	-7.316	112.75	20	10:21:00	4862
6	5	-1	-7.316	112.7	-1	-1	7	11:11:00	1738

### 4.3 Pengujian Kecepatan Perhitungan Rute

Pengujian perhitungan rute ini dilakukan untuk menguji kecepatan metode perhitungan rute yang dipakai. Adapun manfaat dari pengujian ini yaitu menunjukkan seberapa banyak waktu yang dibutuhkan untuk melakukan perhitungan pada suatu *shipment* dari proses paling awal hingga akhir.

Pengujian dilakukan dengan membuat sejumlah data dengan jumlah destinasi yang berbeda-beda dan dengan jumlah destinasi dengan permintaan khusus yang beragam. Dari setiap data tersebut, dihitung waktu tempuh yang dibutuhkan untuk setiap proses yang ada.

Sebagai bahan pengujian, diperlukan pencatatan terhadap setiap waktu tempuh dalam menyelesaikan setiap proses yang dilakukan. Tabel 4 merupakan hasil nilai perhitungan rata-rata untuk setiap proses dari setiap percobaan yang dilakukan.

**Tabel 4. Tabel Pengujian Kecepatan Perhitungan Rute**

Jumlah Destinasi	Jumlah Destinasi dengan Permintaan Khusus	Pencarian Alamat dengan Google Maps API (detik)	Waktu Untuk Clustering (detik)	Waktu untuk Mengurutkan Rute (detik)	Waktu untuk Menghitung kembali Rute dengan Permintaan Khusus (detik)	Rata-rata response Google Maps API per satu data (detik)	Total Waktu yang dibutuhkan untuk menghitung rute (detik)
3	0	7.51	0.00026	0.87	1.48	2.50333	9.86026
3	1	7.42	0.00035	0.98	1.51	2.47333	9.91035
4	1	6.79	0.00075	0.476	1.57	1.6975	8.83675
5	2	7.79	0.0025	0.887	9.22	1.558	17.8995
5	3	7.99	0.0013	0.74	11.3	1.598	20.0313
5	3	7.6	0.0008	0.58	12.14	1.52	20.3208
10	8	8.14	0.0045	1.24	25.47	0.814	34.8545
10	6	8.4	0.0039	1.26	12	0.84	21.6639
10	4	8.3	0.0038	1.67	10.89	0.83	20.8638
20	10	14.49	0.026	2.15	57.45	0.7245	74.116
20	5	13.97	0.029	2.43	38.15	0.6985	54.579
30	11	20	0.034	2.62	110.58	0.6667	133.234

## 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Dari hasil perancangan dan pembuatan aplikasi, dapat diambil kesimpulan antara lain:

1. Aplikasi memberikan informasi mengenai estimasi pengiriman serta estimasi keterlambatan pengiriman barang kepada pelanggan beserta dengan *report* status pengiriman barang.

2. Aplikasi memberikan rekomendasi urutan rute pengiriman dengan memperhatikan setiap preferensi waktu pengiriman barang ke pelanggan.
3. Sistem dapat menghitung 30 titik destinasi hanya dengan waktu kurang lebih 133 detik.
4. Secara keseluruhan, sistem mampu menjawab kebutuhan perusahaan. Hal ini dibuktikan dengan hasil kuisioner yang mengatakan bahwa secara keseluruhan fungsional program 93,3% baik dan 6,7% cukup. Serta secara keseluruhan program 100% baik.

### 5.2 Saran

Saran yang dapat diberikan untuk menyempurnakan dan mengembangkan aplikasi ini lebih lanjut antara lain:

1. Penambahan fitur yang mampu mengatasi permasalahan traffic diluar prediksi, seperti kecelakaan, jalan yang di tutup, volume kendaraan yang tiba-tiba meningkat, dan lain sebagainya.
2. Penambahan fitur cronjob sehingga program dapat berjalan secara otomatis tanpa bantuan administrator.
3. Penambahan fitur *cleaning* pada alamat pengiriman
4. Penambahan fitur untuk melakukan tracking terhadap supir pengiriman barang dengan menggunakan location dari GPS.
5. Pembuatan aplikasi pada modul driver dengan berbasis android sehingga aplikasi dapat menggunakan resource pada smartphone dengan lebih baik.

## 6. DAFTAR PUSTAKA

- [1] Aggarwal, C. C., & Reddy, C. K. 2014. *Data clustering: algorithms and applications*. Boca Raton: CRC Press/Taylor & Francis Group.
- [2] Agusta, Y. 2007. *K-means - Penerapan, Permasalahan dan Metode Terkait*. *Jurnal Sistem dan Informatika* Vol. 3 (Februari 2007): 47-60
- [3] Chia, W. C., Yeong, L. S., Lee, F. J., & Chng, S. I. 2016. *Trip planning route optimization with operating hour and duration of stay constraints*. *2016 11th International Conference on Computer Science & Education (ICCSE)*. doi:10.1109/iccse.2016.7581
- [4] Gambardella, L. M., Taillard, E., & Agazzi, G., 1999. *A Multiple Ant Colony System For Vehicle Routing Problems With Time Windows*. *New Ideas in Optimization*, 3
- [5] J. R., & R. C. 2012. *Travelling Salesman Problem. Book On Demand*.
- [6] Khachay, M., & Neznakhina, K. (2016). *Polynomial Time Approximation Scheme for the Minimum-weight k-Size Cycle Cover Problem in Euclidean space of an arbitrary fixed dimension*. *IFAC-PapersOnLine*, 49(12), 6-10. doi:10.1016/j.ifacol.2016.07.541
- [7] Orunga, E. 2012. *Challenges facing the Courier industry and their solutions*
- [8] R. Nallusamy, K. Duraiswamy, R. Dhanalaksmi, and P. Parthiban. 2010. *Optimization of multiple vehicle routing problems using approximation algorithms*.
- [9] Santosa, B. 2007. *Data Mining: Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Yogyakarta: Graha Ilmu

