

Penentuan Lokasi Fasilitas *Supply Chain* Dengan Metode *Gravity Location Models*

Luh Gede Surya Kartika¹⁾, I Made Ari Andreawan²⁾, Yohana Nugraheni³⁾

Sistem Komputer STMIK STIKOM Bali

Jalan Raya Puputan Renon no. 86 Denpasar, Bali, Indonesia tlp. (0361) 244445

fax: (0361) 264773

suryakartika1109@gmail.com¹⁾, ariandreawan@gmail.com²⁾, yohana_biz@yahoo.com³⁾

ABSTRAK

Perancangan jaringan Supply Chain merupakan kegiatan strategis yang harus dilakukan pada Supply Chain Management dan mencakup keputusan tentang lokasi, jumlah dan kapasitas fasilitas produksi dan distribusi dalam Supply Chain. Tujuan dari jaringan Supply Chain adalah untuk memenuhi kebutuhan pelanggan yang bisa berubah secara dinamis dari waktu ke waktu. Penempatan lokasi fasilitas merupakan upaya dalam mendistribusikan barang lebih dekat ke pelanggan. Dengan adanya fasilitas yang dimaksud perusahaan dapat mendistribusikan barang untuk membantu dalam menentukan kapan, dimana, dan berapa jumlah barang yang disiapkan dan selanjutnya dikirim untuk memenuhi permintaan. Metode Gravity Location Models merupakan salah satu metode dalam pemilihan lokasi fasilitas, dimana jarak atau biaya menuju fasilitas-fasilitas yang akan dibangun ataupun yang telah ada akan dapat diminimalkan. Oleh karena itu, pada penelitian ini telah dibangun sistem pendukung keputusan berbasis web dengan menggunakan metode Gravity Location Models. Pembuatan aplikasi ini menggunakan Bahasa pemrograman PHP dengan berbasis web dan MySQL sebagai DBMS.

Kata kunci: Gravity Location Models, Lokasi Fasilitas, Sistem Pendukung Keputusan, Supply Chain.

1. Pendahuluan

Penempatan lokasi fasilitas merupakan salah satu upaya dalam mendistribusikan barang lebih dekat ke pelanggan. Fasilitas adalah lokasi fisik dalam jaringan rantai pasok yang menjadi tempat untuk perakitan, penyimpanan, ataupun produksi. Fasilitas dikelompokkan menjadi fasilitas produksi dan fasilitas penyimpanan dapat berupa gudang, pabrik ataupun pasar. Beberapa komponen fasilitas yang harus dipertimbangkan antara lain peranan lokasi dan kapasitas [1]. Lokasi disetiap fasilitas memiliki dampak terhadap aktivitas dan biaya dalam rangka memproduksi produk yang diinginkan pelanggan. Dengan adanya fasilitas yang dimaksud perusahaan dapat mendistribusikan barang untuk membantu dalam menentukan kapan, dimana, dan berapa jumlah barang yang disiapkan dan selanjutnya dikirim dalam rangka untuk memenuhi permintaan. Tujuan lain adalah untuk penghematan dalam menjalankan aktivitas pendistribusian produknya. Faktor jarak dari lokasi fasilitas menuju ke tempat-tempat tujuan merupakan upaya untuk menurunkan biaya transportasi yang sekecil mungkin. Biaya perlu diperhatikan karena merupakan faktor penting dan terbatas. Untuk itu perlu perencanaan yang strategis dalam menentukan lokasi fasilitas yang dimaksud [2]. Dari sisi pelanggan, jaringan yang baik tentunya harus bisa memberikan kecepatan respons yang tinggi (*lead time* yang pendek bagi pelanggan untuk memperoleh barang) dan *service level* yang tinggi, yakni kemampuan jaringan untuk memasok dengan ketersediaan barang cukup tinggi [3]. Dari sisi *Supply Chain*, biaya untuk menyediakan layanan dengan *lead time* yang pendek dan atau tingkat layanan yang tinggi harus dilaksanakan secara efisien.

Karena rancangan jaringan adalah sesuatu yang strategis maka perubahan terhadap konfigurasi jaringan hanya terjadi dalam interval waktu yang relatif panjang, namun proses operasional dalam jaringan tersebut akan berlangsung secara terus menerus. Implementasi strategi *Supply Chain* hanya bisa berlangsung secara efektif apabila *Supply Chain* memiliki jaringan dengan konfigurasi yang sesuai. Artinya, struktur atau konfigurasi jaringan bisa menentukan apakah suatu *Supply Chain* akan bisa menjadi responsif atau efisien. Contohnya jika *Supply Chain* menginginkan kondisi responsif maka konfigurasi jaringannya harus ditunjang oleh fasilitas produksi dan gudang yang lebih banyak dan tersebar di berbagai lokasi pemasaran. Sebaliknya, suatu *Supply Chain* akan efisien apabila jaringan yang ada relatif tersentralisasi dengan fasilitas yang lebih sedikit. Untuk keputusan lokasi fasilitas yang dimaksud, strategi yang digunakan biasanya adalah strategi untuk meminimalkan biaya. Walaupun demikian, strategi pemilihan lokasi fasilitas ditentukan oleh kombinasi antara biaya dan kecepatan pengiriman. Secara umum, tujuan strategi lokasi adalah untuk memaksimalkan keuntungan lokasi bagi perusahaan. Terdapat beberapa pilihan yang ada dalam lokasi yaitu yang pertama tidak pindah, melainkan meluaskan fasilitas

yang ada. Yang kedua mempertahankan lokasi sekarang, selagi menambah fasilitas lain di tempat lain, dan yang terakhir menutup fasilitas yang ada dan pindah ke lokasi yang lain [4]. Dapat disimpulkan pemilihan lokasi fasilitas dapat menentukan keberhasilan perusahaan hubungannya dengan biaya operasi, harga jual, serta kemampuan perusahaan untuk bersaing dipasar. Metode *Gravity Location Models* merupakan salah satu metode dalam pemilihan lokasi fasilitas. Dengan penggunaan metode ini, jarak atau biaya menuju fasilitas-fasilitas yang akan dibangun ataupun yang telah ada akan dapat diminimalkan. Pada prinsipnya metode ini mencari nilai optimal yang dapat diperoleh dengan mempertimbangkan pemenuhan *demand* dan *supply* pada biaya transportasi yang terendah.

Dengan demikian, dibutuhkan suatu sistem untuk mempermudah pengambilan keputusan dalam menentukan lokasi, biaya, dan jarak tempuh. Sistem yang digunakan adalah sistem berbasis web dengan menggunakan *Gravity Location Models* yang menjadi penghubung antara sumber-sumber pasokan dan beberapa lokasi terkait. Sehingga biaya dari segi transportasi akan dapat diminimasi.

2. Metode Perencanaan

Bagian ini akan menjelaskan mengenai tahapan dalam penelitian. Alur penelitian yang dilakukan adalah sebagai berikut:

1. Studi Literatur, merupakan penelusuran literatur yang bersumber dari buku, media pakar, ataupun hasil penelitian orang lain yang bertujuan untuk menyusun dasar teori yang digunakan dalam melakukan penelitian.
2. Analisa Data, digunakan untuk menganalisa data serta informasi yang didapatkan dari proses pengumpulan data.
3. Perancangan Sistem, Dalam perancangan sistem ini menggunakan DFD (*Data Flow Diagram*) yang memberikan gambaran bagaimana data masuk dan keluar dari dalam, dan ke suatu *entity* atau representasi dari sumber dan tujuan aliran data tersebut, aturan dari proses data, penyimpanan data dan entitas eksternal. Dengan database yang mencakup konseptual basis data dan struktur tabel.
4. Pembuatan dan Pengujian Sistem, dalam tahap pembuatan sistem ini menggunakan bahasa pemrograman HTML, PHP dan MySQL untuk pembuatan *database* serta menggunakan *Black Box testing* dalam pengujian sistem ini.

2.1 Alur Sistem

1. *Gravity Location Models*

Gravity Location Models didasarkan pada pemilihan koordinat titik suatu pusat distribusi yang memberikan jarak total terpendek terhadap keseluruhan pusat zona produksi yang harus dipasok [5]. Model ini menggunakan beberapa asumsi, yaitu:

- 1) Ongkos-ongkos transportasi diasumsikan naik sebanding dengan volume yang dipindahkan.
- 2) Baik sumber-sumber pasokan maupun lokasi produksi bisa ditentukan lokasinya pada suatu peta dengan koordinat X dan Y yang jelas. Perhitungan koordinat lokasi menggunakan rumus sebagai berikut:

$$x_{0n} = \frac{\sum_i \frac{C_i V_i x_i}{j_i}}{\sum_i \frac{C_i V_i}{j_i}} \quad \begin{array}{l} C_i = \text{ongkos transportasi per unit beban per kilometer antara kandidat lokasi} \\ \text{fasilitas dengan lokasi pasar atau lokasi sumber pasokan} \\ V_i = \text{beban yang akan dipindahkan antar fasilitas dengan sumber pasokan} \\ \text{atau sumber lokasi .} \end{array}$$

$$y_{0n} = \frac{\sum_i \frac{C_i V_i y_i}{j_i}}{\sum_i \frac{C_i V_i}{j_i}} \quad \begin{array}{l} j_i = \text{jarak antara lokasi fasilitas dengan sumber pasokan atau pasar } i. \\ (x_i, y_i) = \text{koordinat } x \text{ dan } y \text{ untuk lokasi pasar atau sumber pasokan } i \\ (x_{0n}, y_{0n}) = \text{koordinat } x \text{ dan } y \text{ yang dihasilkan pada iterasi ini} \end{array}$$

Proses perhitungan jarak antara dua lokasi pada model ini yang dihitung sebagai jarak geometris antara dua lokasi menggunakan formula berikut:

$$j_i = \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2}$$

dengan :

(x_0, y_0) = kandidat koordinat fasilitas yang dipertimbangkan.

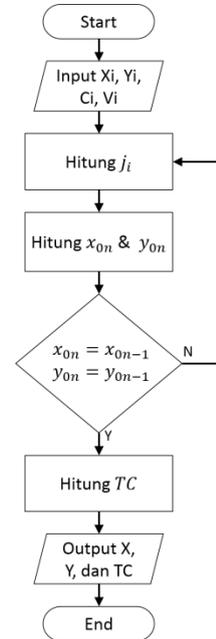
Tujuan dari model ini adalah mendapatkan lokasi fasilitas yang meminimumkan total ongkos-ongkos pengiriman .

$$TC = \sum_i C_i V_i j_i$$

dengan :

TC = total biaya transportasi.

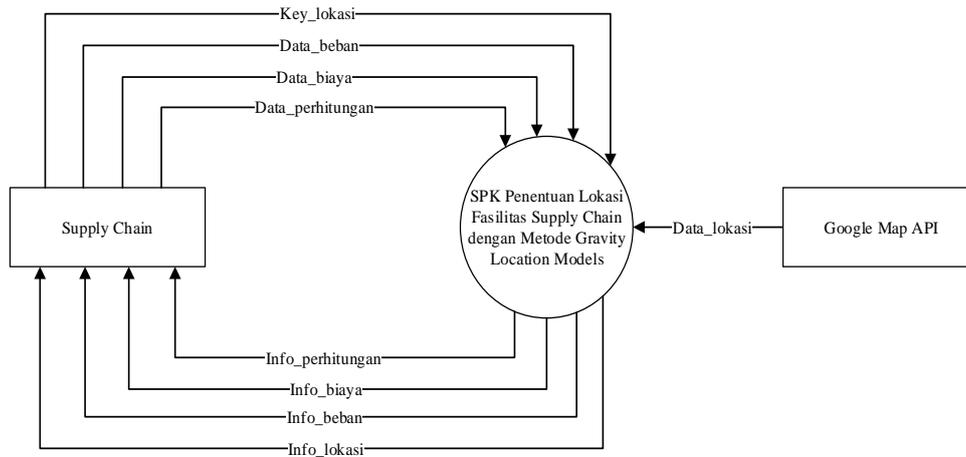
Pada Gambar 1 merupakan gambaran *flowchart* dari perhitungan *Gravity Location Models*. Proses dimulai saat *user* memasukkan data X_i (*Longitude*), Y_i (*Latitude*), C_i (*Biaya transportasi*), dan V_i (*beban perpindahan*). Proses selanjutnya adalah melakukan perhitungan j_i (*jarak antar 2 lokasi*), setelah nilai j_i diketahui, maka akan melakukan proses perhitungan X_{0n} dan Y_{0n} . Terdapat sebuah kondisi dimana saat X_{0n} sama dengan X_{0n-1} dan Y_{0n} sama dengan Y_{0n-1} , maka akan melakukan proses selanjutnya yaitu menghitung TC (*Total Cost*), jika nilainya tidak sama dengan kondisi yang diinginkan, maka akan melakukan proses perulangan untuk menghitung nilai j_i . Proses akan berhenti ketika menampilkan nilai optimal X (*Latitude*), Y (*Longitude*), dan TC (*Total Cost*).



Gambar 1. Flowchart Gravity Location Models

2. Diagram Konteks

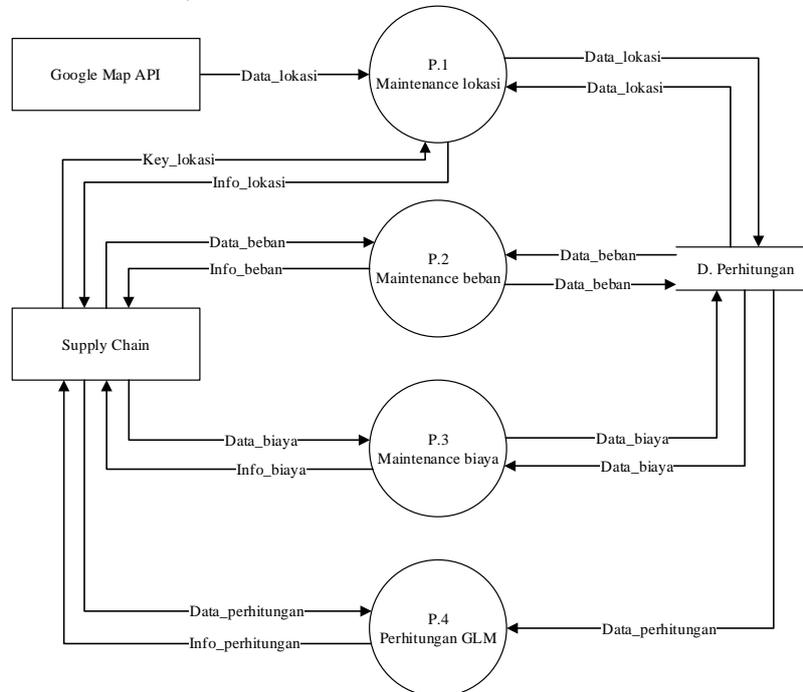
Diagram konteks merupakan penjabaran proses yang terjadi pada sistem yang dijelaskan secara garis besar dimana terdapat *entity-entity* yang akan memasukan data maupun menerima data dari sistem. Adapun *entity-entity* tersebut adalah *Supply Chain* dan *Google Map API*. Pada gambar 2 Diagram Konteks, *Supply Chain* dapat melakukan *request key* lokasi fasilitas (berupa *Latitude* dan *Longitude* lokasi fasilitas), input data beban, input data biaya, dan melihat data perhitungan. *Supply Chain* memperoleh output berupa informasi lokasi (*Latitude* dan *Longitude*), informasi beban, informasi biaya, dan informasi perhitungan. Sedangkan *Google Map API* memberikan data lokasi berupa *Latitude* dan *Longitude* ke dalam sistem.



Gambar 2. Diagram Konteks

2. *Data Flow Diagram*

Data Flow Diagram dibuat untuk menggambarkan arus *input* dari *eksternal entity* ke proses dan arus *output* dari proses ke *eksternal entity*. Pada gambar 3 *Data Flow Diagram Level 0* menjabarkan proses pada proses level sebelumnya, proses tersebut dijabarkan menjadi 4 proses, yaitu: Proses *Maintenance* Lokasi, Proses *Maintenance* Beban, Proses *Maintenance* Biaya, dan Proses Perhitungan GLM (*Gravity Location Models*).



Gambar 3. Data Flow Diagram

3. **Hasil dan Pembahasan**

Pengujian yang akan digunakan untuk menguji sistem adalah metode pengujian *Black Box*. Pengujian *Black Box* berfokus pada persyaratan fungsional perangkat lunak. Pengujian ini dilakukan saat *user* berhasil memasukkan data ke dalam sistem dalam kelas uji kelola data untuk mengetahui proses *view* hasil dan jarak lokasi. Dari hasil pengujian yang didapatkan, diketahui bahwa selama proses pengujian Sistem Pendukung Keputusan Penentuan Lokasi Fasilitas Supply Chain, semua kemungkinan kondisi yang dipersyaratkan telah sesuai dengan kasus dan hasil uji yang dirancang. Hasil pengujian ditunjukkan oleh Tabel 1.

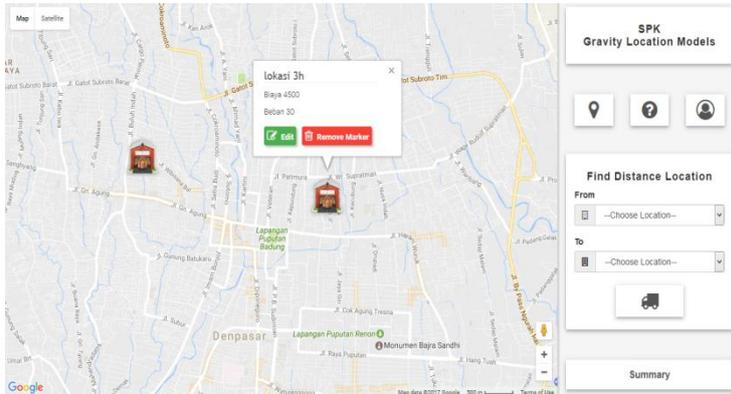
Tabel 1 Rancangan Pengujian Sistem Pendukung Penentuan Lokasi Fasilitas Supply Chain

Kelas Uji	Butir Uji	Jenis Uji
Kelola data	1. Tambah data (nama lokasi, biaya, beban, <i>Latitude</i> dan <i>Longitude</i>)	<i>Black Box</i>
	2. Tampil data (biaya dan beban)	
	3. Ubah data biaya dan beban.	
	4. Hapus <i>marker</i>	
View hasil	1. Tampil hasil <i>Gravity Location Models</i> 2. Tampil log perhitungan 3. Tampil <i>summary</i>	<i>Black Box</i>
View bantuan	Menampilkan informasi bantuan	<i>Black Box</i>
View profile	Menampilkan informasi <i>profile</i>	<i>Black Box</i>
View jarak lokasi	1. Tampil data lokasi	<i>Black Box</i>
	2. Tampil jarak lokasi	

3.1 **Tampilan Sistem**

Berikut merupakan tampilan Sistem Pendukung Keputusan Penentuan Lokasi Fasilitas *Supply Chain* dengan Metode *Gravity Location Models* saat dijalankan.

1. Halaman Tampil Data Lokasi

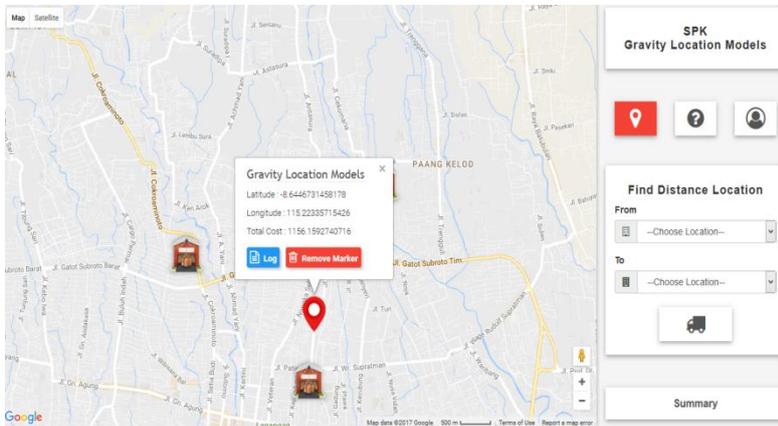


Gambar 4. Halaman Tampil Data Lokasi

Gambar 4 merupakan tampilan saat *user* telah menambahkan data lokasi fasilitas, dimana titik-titik lokasi (digambarkan dalam bentuk gudang) ditampilkan pada bagian *Google Maps* serta menampilkan informasi nama lokasi, biaya, dan beban.

Kode yang digunakan untuk menangkap Longitude dan latitude yang sudah dikirim adalah: `google.maps.LatLng(parseFloat(data.Y),parseFloat(data.X))`;

2. Halaman Hasil



Gambar 5. Halaman Hasil

Gambar 5 merupakan tampilan saat *user* melakukan perhitungan *Gravity Location Models*, dimana sebuah marker baru muncul di antara atau berada pada salah satu lokasi fasilitas. Marker baru ditandai dengan warna merah dan menampilkan data optimal *Latitude*, *Longitude*, dan *Total Cost*. Hasil pengujian terhadap waktu komputasi adalah menunjukkan kurang dari 5 detik dengan jumlah iterasi maksimum yang pernah terjadi adalah sebanyak 65 iterasi.

3. Halaman Log



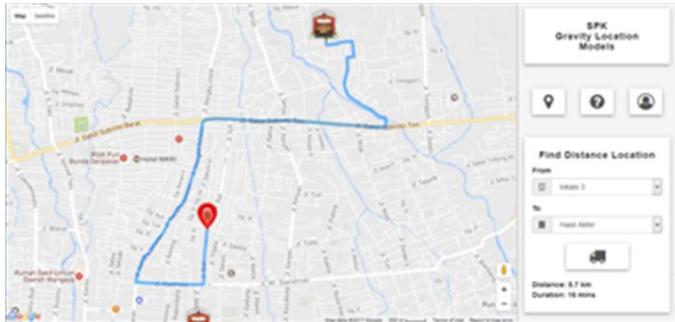
Gambar 6. Log hasil Pengujian

diambil dari fungsi perhitungan. Proses *foreach* digunakan untuk menampilkan data *array* ke dalam tabel berupa data *latitude*, *longitude*, biaya, beban, jarak, dan TC. Data-data tersebut diambil dari variabel $\$y_i$, $\$x_i$, $\$C_i$, $\$V_i$, $\$J_i$, $\$TC_i$. Selanjutnya untuk data hasil optimal didapatkan dari $\$data['Y']$ untuk *latitude*, $\$data['X']$ untuk *longitude*, $\$data['TC']$ untuk total *Total Transportation Cost*. Proses ini terus berulang hingga variabel $\$x$ tidak sama dengan variabel $\$X$ dan variabel $\$y$ tidak sama dengan variabel $\$Y$.

Gambar 6 merupakan tampilan pencatatan proses perhitungan *Gravity Location Models*, yang menampilkan data yang telah dimasukkan oleh *user* serta lokasi optimal secara spesifik, data berupa *Latitude*, *Longitude*, biaya, beban, *Transportation Cost*, serta iterasi tiap proses perhitungan.

Data ditampilkan ke dalam tabel dengan mengambil data *latitude*, *longitude*, beban, dan biaya di dalam *database*, data perhitungan

4. Halaman Jarak Lokasi Fasilitas



Gambar 7. Halaman Jarak Lokasi Fasilitas

Gambar 7 merupakan halaman pencarian jarak dan waktu yang dituju menuju antar lokasi fasilitas atau fasilitas dengan hasil perhitungan metode *Gravity Location Models*. *Class get-route* digunakan untuk menampilkan rute, jarak, dan waktu tempuh. Variabel *directionDisplay* digunakan untuk menampilkan rute dengan atribut *draggable false*, serta *id txtSource* dan *id txtDestination* untuk memilih lokasi dari tempat tujuan ke tempat

yang dituju. Dalam fungsi *class get-route*, terdapat variabel *request* yang digunakan untuk menyimpan data aslinya: *source* yang digunakan untuk mengambil data asal (*source*), *source* didapatkan dari variabel *srcLat* dan *srcLng*, *destination* digunakan untuk mengambil data tujuan (*destination*), *destination* didapatkan dari variabel *dstLat* dan *dstLng*. Variabel *distance* dan variabel *duration* digunakan untuk menampilkan jarak (*distance*) dan waktu (*duration*) ke halaman sistem, untuk menampilkannya terdapat sebuah kondisi dimana saat status sama dengan *DistanceMatrixStatus ok*, maka *distance* dan *duration* akan ditampilkan ke halaman sistem, jika tidak maka akan muncul peringatan.

4. Kesimpulan dan Saran

4.1 Kesimpulan

Berdasarkan pembahasan, Sistem Pendukung Keputusan Penentuan Lokasi Fasilitas *Supply Chain* dengan Metode *Gravity Location Models* dapat disimpulkan sebagai berikut.

1. Telah dihasilkan Sistem Pendukung Keputusan Penentuan Lokasi Fasilitas *Supply Chain* menggunakan metode *Gravity Location Models* untuk mempermudah *Supply Chain* dalam menentukan lokasi fasilitas.
2. Sistem berhasil menampilkan lokasi fasilitas optimal dari titik-titik lokasi fasilitas yang telah ditentukan oleh user.
3. Sistem ini telah berjalan dengan baik dan telah di uji dengan metode pengujian *Blackbox* dan mendapatkan hasil yang sesuai dengan yang di harapkan

Sistem ini masih memiliki kekurangan karena di dalamnya masih terdapat kekurangan yang dapat diperbaiki dan dikembangkan untuk menghasilkan sistem yang lebih baik lagi. Berikut beberapa hal yang dapat dilakukan untuk pengembangan sistem.

1. Sistem ini dapat dikembangkan dengan menambahkan informasi atau kriteria dalam penentuan lokasi fasilitas seperti relief permukaan bumi yang menunjukkan seberapa tinggi suatu lokasi (contohnya gunung dan sungai) dan pilihan rute perjalanan menuju lokasi.
2. Sistem ini dapat dibandingkan dengan metode *Point Rating* yang diterapkan menggunakan metode *AHP (Analytical Hierarchy Process)* dengan Pendekatan Skala *Rating Liberatore* untuk mengetahui metode manakah yang efektif dalam penentuan lokasi fasilitas.
3. Dimungkinkan untuk dikembangkan dalam platform lain seperti android dan iOS agar mudah digunakan kapanpun dan dimanapun.

Daftar Pustaka

- [1]. S. Chopra, P. Meindl. *Supply Chain Management Strategy, Planning, and Operation*, Third Edit. New Jersey: Pearson Education, Inc. 2007.
- [2]. Klibi W., Martel A., dan Guitouni A. The design of robust value-creating Supply Chain networks: A critical review. *European Journal of Operational Research*. 2010; 203(2): 283-293.
- [3]. Sourirajan K., Ozsen L., dan Uzsoy R. A genetic algorithm for a single product network design model with lead time and safety stock considerations. *European Journal of Operational Research*. 2009; 197(2): 599-608.
- [4]. Hindrayani, Aniek. *Manajemen Operasi*. Yogyakarta: Pohon Cahaya. 2010.
- [5]. J E Anderson. A Theoretical Foundation for the Gravity Equation. *Am Econ Assoc*. 2012; 69(1): 106-116.