

## Performansi Algoritma CODEQ dalam Penyelesaian Vehicle Routing Problem

Annisa Kesy Garside<sup>1\*</sup>, Satya Sudaningtyas<sup>1</sup>

**Abstract:** Genetic Algorithm, Tabu Search, Simulated Annealing, and Ant Colony Optimization showed a good performance in solving vehicle routing problem. However, the generated solution of those algorithms was changeable regarding on the input parameter of each algorithm. CODEQ is a new, parameter free meta-heuristic algorithm that had been successfully used to solve constrained optimization problems, integer programming, and feed-forward neural network. The purpose of this research are improving CODEQ algorithm to solve vehicle routing problem and testing the performance of the improved algorithm. CODEQ algorithm is started with population initiation as initial solution, generated of mutant vector for each parent in every iteration, replacement of parent by mutant when fitness function value of mutant is better than parent's, generated of new vector for each iteration based on opposition value or chaos principle, replacement of worst solution by new vector when fitness function value of new vector is better, iteration ceasing when stopping criterion is achieved, and sub-tour determination based on vehicle capacity constraint. The result showed that the average deviation of the best-known and the best-test value is 6.35%. Therefore, CODEQ algorithm is good in solving vehicle routing problem.

**Keywords:** Algorithm, meta-heuristic, vehicle routing problem, CODEQ.

### Pendahuluan

*Combinatorial problem* merupakan suatu permasalahan matematis untuk menyusun, mengelompokkan, mengurutkan, atau memilih sejumlah objek diskrit tertentu (Lawler [1]). Sampai saat ini, *combinatorial problem* masih menjadi salah satu bidang yang sering diteliti karena dengan semakin berkembangnya ilmu pengetahuan diharapkan akan ditemukan metode-metode maupun pendekatan-pendekatan baru yang bisa menjadi sebuah cara penyelesaian terbaik untuk permasalahan ini. Contoh *combinatorial problem* adalah *travelling salesman problem*, *knapsack problem*, dan *vehicle routing problem*.

*Vehicle Routing Problem* (VRP) merupakan permasalahan *m-travelling salesman* (Kallehauge [2]). Dalam permasalahan ini, terdapat *m-salesman* yang berangkat dari suatu kota kemudian melakukan kunjungan ke sejumlah kota. Setiap kota hanya dapat dikunjungi oleh satu *salesman* saja. Tujuan utamanya adalah meminimalkan total jarak atau total biaya perjalanan. VRP merupakan permasalahan klasik yang sangat dekat dengan kehidupan manusia. Contoh permasalahan yang dapat diselesaikan dengan pendekatan VRP adalah penentuan rute pengiriman surat, *laundry*, dan koran. Sampai saat ini, sudah banyak algoritma heuristik yang dihasilkan dari penelitian-penelitian untuk

menyelesaikan permasalahan VRP, diantaranya adalah *genetic algorithm*, *tabu search*, *simulated annealing*, dan *ant colony*. Dalam prosesnya, meskipun memberikan hasil yang memuaskan, algoritma-algoritma penyelesaian tersebut memiliki beberapa parameter tertentu. Jika nilai parameter-parameter tersebut diubah, maka hasil penyelesaiannya akan berbeda. Sampai saat ini penelitian untuk permasalahan VRP masih terus dilakukan karena masih belum ada algoritma heuristik yang konsisten untuk memberikan hasil terbaik.

Perkembangan ilmu pengetahuan memberikan alternatif baru dalam penyelesaian permasalahan optimasi. Omran dan Salman [3] mengembangkan algoritma CODEQ yang merupakan sintesis dari empat algoritma yang telah ada, yaitu *chaotic search*, *opposition-based learning*, *differential evolution* (DE), dan *quantum mechanism*. CODEQ merupakan algoritma *meta-heuristic* yang tidak membutuhkan inputan parameter (kecuali untuk ukuran populasi) dan telah berhasil diterapkan untuk menyelesaikan permasalahan-permasalahan *continuous* berkonstrain (Omran dan Salman [3]); *integer programming* (Omran dan al-Sharhan [4] dan Chiou dan Chang [5]); serta *feed-forward artificial neural network* (Omran dan al-Adwani, [6]). Performansi algoritma CODEQ terbukti menghasilkan solusi yang lebih baik dibandingkan algoritma optimasi lainnya pada permasalahan optimasi kontinu (Omran [7]). Sementara itu, Chiou dan Chang [5] menggunakan algoritma CODEQ untuk menyelesaikan permasalahan diskrit yaitu penempatan kapasitor dalam sistem distribusi dengan *integer*

<sup>1</sup> Jurusan Teknik Industri, Universitas Muhammadiyah Malang, Jl. Raya Tlogomas No. 246, Malang 65144, Indonesia.  
Email: annisa\_garside@yahoo.com

\* Penulis korespondensi

*programming*. Hasil penelitian Chiou dan Chang [5] menunjukkan performansi algoritma CODEQ lebih baik dibanding algoritma DE dan *Simulated Annealing* (SA). Berdasarkan keunggulan algoritma CODEQ dalam menyelesaikan permasalahan kontinu dan diskrit, penelitian ini bertujuan mengembangkan algoritma CODEQ untuk menyelesaikan salah satu permasalahan optimasi diskrit yaitu VRP. Penelitian terdahulu mengenai algoritma CODEQ untuk menyelesaikan VRP telah dilakukan oleh Narendra [8]. Narendra [8] menggunakan pendekatan *cluster first-route second* yang berarti *node-node* dikelompokkan terlebih dahulu menjadi satu klaster, kemudian *node-node* tersebut disusun menjadi rute berurutan berdasarkan jarak terdekatnya. Menurut Fuellerer [9], pendekatan *route-first-cluster second* memiliki tingkat kesulitan yang lebih mudah dibanding *cluster first-route second* sehingga waktu komputasi cenderung lebih singkat. Berdasarkan pertimbangan tersebut, penelitian ini akan menggunakan pendekatan *route first-cluster second* dalam menyelesaikan permasalahan penentuan rute.

## Metode Penelitian

### Tahap Pengembangan Algoritma

Terdapat dua macam pendekatan heuristik yang digunakan para peneliti untuk menyelesaikan permasalahan VRP, yaitu *route first-cluster second* dan *cluster first-route second*. Penelitian ini menggunakan pendekatan *route first-cluster second*, sehingga penyelesaian dimulai dengan membentuk *giant tour* (*TSP tour*) dengan merelaksasi batasan kapasitas kendaraan dan kemudian memecah *giant tour* tersebut menjadi beberapa *sub tour* dengan mempertimbangkan kapasitas kendaraan. Berdasarkan pendekatan tersebut, pengembangan algoritma dilakukan dengan memodifikasi algoritma CODEQ yang diusulkan Omran dan Salman [3] pada langkah pembangkitan vektor mutan dan vektor baru sehingga diperoleh *giant tour* (urutan node yang dikunjungi) dalam bilangan integer. Selain itu menambahkan langkah untuk membentuk sub tour berdasarkan *giant tour* yang telah dibuat.

Langkah-langkah penyelesaian VRP dengan algoritma CODEQ yang diusulkan adalah:

1. Inisialisasi populasi sebagai solusi awal  
Sebagaimana algoritma *evolutionary optimization*, CODEQ merupakan algoritma yang membutuhkan solusi awal. Jumlah solusi awal yang dibangkitkan (ukuran populasi) dinotasikan sebagai *NP*. Dalam hal ini, solusinya merupakan vektor  $r_i(t)$  yang menggambarkan urutan *node* yang dikunjungi. Inisialisasi populasi dibangkitkan dengan dua cara, yaitu acak dan menggunakan metode *nearest neighbour*.

2. Pembangkitan vektor mutan  $v_i(t)$  dari vektor *parent*  $r_i(t)$  untuk setiap iterasi  $t$  dengan rumus:

$$v_i(t) = x_i(t) + (x_{i_1}(t) - x_{i_2}(t)) \ln\left(\frac{1}{u}\right) \quad (1)$$

Bila vektor  $v_i(t)$  merupakan bilangan real, maka harus diubah menjadi bilangan integer dengan metode *integer order criterion* (Mingyong dan Erbao [10]). Gen dengan bilangan real terbesar akan menjadi konsumen dengan *ordinal number* terbesar ( $n$ ), selanjutnya gen dengan bilangan real terbesar kedua akan menjadi konsumen  $n - 1$ , dan seterusnya.

3. Penggantian *parent*  $r_i(t)$  oleh mutan  $v_i(t)$  bila nilai *fitness function* mutan lebih baik dari *parent*, dan sebaliknya. Pada tahap ini, dilakukan perhitungan total jarak yang ditempuh oleh kendaraan.
4. Pembangkitan vektor baru  $w(t)$  pada setiap iterasi  $t$  dengan rumus:

$$w(t) = LB + UB - r \times x_b(t) \text{ jika } \text{rand} \leq 0,5 \quad (2)$$

$$w(t) = x_g(t) + |x_{i_1}(t) - x_{i_2}(t)| \times (2c(t) - 1) \text{ jika } \text{rand} > 0,5 \quad (3)$$

dimana  $r$  didapatkan secara acak  $U(0,1)$ ,  $LB$  dan  $UB$  merupakan batas bawah dan atas pada permasalahan,  $x_b(t)$  merupakan *worst* (*least fit*) *vector* pada iterasi  $t$ ,  $x_g(t)$  merupakan *best* (*fittest*) *vector* pada iterasi  $t$ ,  $x_{i_1}(t)$  dan  $x_{i_2}(t)$  merupakan vektor yang dipilih secara random dengan  $i_1 \neq i_2 \neq i$  dan  $c(t)$  merupakan *chaotic variable* yang didapatkan dari rumus:

$$c(t) = \frac{c(t-1)^p}{p} \text{ jika } c(t-1) \in (0, p) \quad (4)$$

$$c(t) = \frac{(1-c(t-1))^{1-p}}{(1-p)} \text{ jika } c(t-1) \in (p, 1) \quad (5)$$

dengan  $c(0)$  dan  $p$  didapatkan secara random pada interval  $(0,1)$ .

Pembangkitan vektor baru  $w(t)$  dihitung menggunakan persamaan 2 atau 3 tergantung pada bilangan random ( $r$ ) yang diperoleh apakah bernilai lebih dari 0,5 atau sebaliknya. Dalam penelitian ini, penetapan bilangan sebesar 0,5 mengacu pada algoritma yang telah dikembangkan oleh Omran dan Salman [3].

Sama dengan prosedur yang diterapkan pada langkah 2, bila vektor  $w(t)$  merupakan bilangan real, maka harus diubah menjadi bilangan integer dengan metode *integer order criterion* (Mingyong dan Erbao [10]). Gen dengan bilangan real terbesar akan menjadi konsumen dengan *ordinal number* terbesar ( $n$ ), selanjutnya gen dengan bilangan real terbesar kedua akan menjadi konsumen ( $n - 1$ ), dan seterusnya.

5. Penggantian solusi rute terburuk  $r_b(t)$  oleh vektor  $w(t)$  bila nilai *fitness function* vektor baru tersebut lebih baik dari rute terburuk. Pada

tahap ini, dilakukan perhitungan total jarak yang ditempuh oleh kendaraan.

6. Penghentian iterasi jika *stopping criteria* telah tercapai, jika belum, maka kembali ke langkah 2, jika sudah maka ke langkah 7. Dalam penelitian ini, iterasi akan dihentikan bila dalam 700 iterasi berikutnya tidak terdapat perubahan solusi yang dihasilkan.
7. Pembentukan *sub-tour* dengan memperhatikan batasan kapasitas kendaraan.

### Tahap Uji Coba Algoritma

Algoritma CODEQ yang diusulkan selanjutnya dikoding dengan menggunakan bahasa pemrograman Matlab sehingga diperoleh program komputer yang akan digunakan dalam percobaan numerik. Validasi program komputer dilakukan untuk mengetahui apakah program komputer telah memberikan solusi yang benar sesuai dengan langkah-langkah algoritma yang diusulkan. Validasi dilakukan dengan membandingkan solusi antara penyelesaian secara manual dan dengan program komputer. Jika tidak perbedaan solusi maka dapat disimpulkan program yang dibuat telah valid. Penyelesaian VRP secara manual menggunakan sebuah set data berukuran kecil untuk meminimalkan jumlah iterasi yang harus dilakukan sesuai langkah-langkah algoritma yang diusulkan. Set data tersebut diambil dari penelitian yang dilakukan oleh Rera [11]. Rera [11] menggunakan set data dengan jumlah node sebanyak 4 untuk memvalidasi program komputer yang dibuat berdasarkan algoritma *cross entropy* yang dikembangkan untuk menyelesaikan VRP.

Percobaan numerik dilakukan dengan menggunakan 4 set data dari Christofides dan Eilon [12] dan 1 set data dari penelitian yang dilakukan oleh Indriana [13]. Set data dari Christofides dan Eilon yang diujikan adalah set data dengan tipe E-n13-k4 (jumlah *node*= 13), E-n22-k4 (jumlah *node*= 22), E-n23-k3 (jumlah *node*= 23), dan E-n30-k3 (jumlah *node*= 30). Pada masing-masing set data, terdapat koordinat masing-masing *node* (yang nantinya digunakan untuk menghitung jarak antar *node* dengan metode *euclidean distance*), permintaan masing-masing *node*, serta jumlah dan kapasitas kendaraan yang digunakan. Berbeda dengan set data dari Christofides dan Eilon [12], set data dari Indriana adalah set data berdasarkan kasus nyata pada CV. Cempaka Tulungagung yang merupakan perusahaan rokok dengan wilayah distribusinya adalah Kediri, Pare, Jombang, Madiun, Nganjuk,

Trenggalek, Pacitan, Ponorogo, Magetan, Ngawi, Blitar, Pasuruan, Mojokerto, Sidoarjo, dan Surabaya (jumlah *node*=16). Tiap set data akan dijalankan dengan menggunakan program komputer yang telah dibuat untuk mendapatkan solusi berupa rute dan jarak tempuh.

### Tahap Analisis Performansi

Pada tahap ini, analisis performansi algoritma dilakukan dengan membandingkan solusi berupa total jarak tempuh yang dihasilkan dari percobaan numerik dengan solusi terbaik (*best known solution*) yang dihasilkan oleh algoritma yang lainnya pada tiap set data. Solusi terbaik untuk set data dari Christofides dan Eilon [12] dapat diunduh pada <http://www.branchandcut.org>, dimana sebagian besar dari solusi terbaik tersebut merupakan solusi global optimal. Solusi untuk set data dari Indriana [13] akan dicari terlebih dahulu dengan menggunakan model matematik sehingga diperoleh solusi global optimal, agar perbandingan solusi untuk keseluruhan set data menjadi hampir sama. Dalam penelitiannya, Indriana menggunakan salah satu metode heuristik yaitu metode saving matriks untuk menyelesaikan permasalahan rute yang dihadapi CV. Cempaka dan solusi yang diperoleh sebesar 1177 (sedangkan solusi optimal sebesar 1151). Dalam penelitian ini, perhitungan nilai *Relative Percentage Deviation* (RPD) digunakan untuk mengetahui persentase perbedaan solusi antara algoritma CODEQ yang diusulkan dengan solusi terbaik/solusi optimal. RPD dirumuskan sebagai:

$$RPD = \frac{Best\ test\ CODEQ - Best\ known}{Best\ known} \times 100\% \quad (6)$$

Algoritma CODEQ yang diusulkan memiliki performansi yang baik jika rata-rata RPD dari seluruh set data kurang dari 10%.

### Hasil dan Pembahasan

Program komputer dengan menggunakan algoritma CODEQ yang dihasilkan dalam penelitian ini dinamakan dengan CODEQ-VRP. Program CODEQ-VRP tersebut kemudian divalidasi dengan mengujicobakan pada set data dari Rera [11]. Berdasarkan pengujian diperoleh program CODEQ-VRP berhasil menyelesaikan permasalahan VRP dengan benar. Selanjutnya, program CODEQ-VRP diujicobakan menggunakan set data dari Christofides dan Eilon [12] dan Indriana [13].

**Tabel 1.** Perbandingan solusi pada tiap set data

Tipe set data	<i>Best known</i>	<i>Best test</i> CODEQ	Waktu komputasi (menit)	RPD (%)
E-n13-k4	290,00*	290,00	0,68	0,00
E-n22-k4	375,28*	392,73	106,52	4,73
E-n23-k3	568,56*	645,54	323,52	13,45
E-n30-k3	534,00	605,47	513,28	13,38
Indriana	1151,00*	1153,00	33,46	0,17
RPD rata-rata				6,35

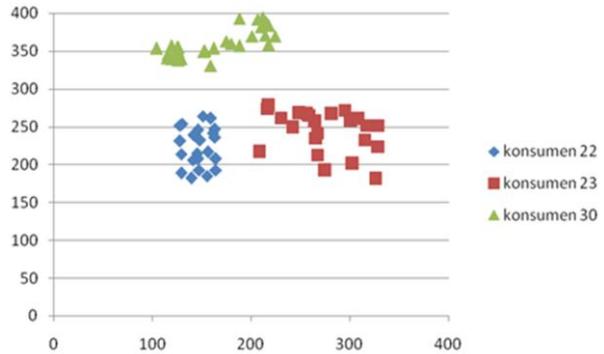
\*Solusi global optimal

Pengujian dijalankan menggunakan komputer dengan spesifikasi 2,20 Ghz Pentium i3 dan RAM 2 GB. Dalam setiap pengujian, dilakukan *running* program sebanyak 5 replikasi untuk mengetahui konsistensi hasil pengujian dan akan dipilih hasil terbaik (*best test*). Nilai *best test* tersebut kemudian akan dibandingkan dengan hasil *best known* dengan menghitung nilai RPD-nya. Nilai RPD menunjukkan tingkat deviasi antara *best test* dan *best known*. Semakin kecil RPD menunjukkan semakin kecil deviasi, yang berarti bahwa program komputer yang digunakan berhasil mencapai hasil optimal global. Selain RPD, pada pengujian ini juga dicatat waktu komputasi untuk menyelesaikan masing-masing set data. Tabel 1 menunjukkan rekapan hasil pengujian yang telah dilakukan.

Dari Tabel 1 dapat diketahui bahwa dari pengujian yang dilakukan, tidak semuanya mencapai kondisi global optimal. Hal ini tampak dari nilai beberapa RPD yang masih belum bernilai nol. Nilai RPD ini juga sangat bervariasi. RPD paling minimum didapatkan saat pengujian dilakukan pada tipe data E-n13-k4. Hal ini disebabkan karena problem tipe set data tersebut yang termasuk problem kecil sehingga relatif mudah diselesaikan. Lebih jauh lagi, RPD minimum (kurang dari 10%) didapatkan untuk set data Indriana [13] dan set data E-n22-k4. Hal ini disebabkan oleh karakteristik dari tipe data tersebut. Apabila dilihat di peta, *node-node* pada set data dari Indriana letaknya tidak terlalu menyebar dan cenderung mengelompok (membentuk klaster). Demikian juga dengan *node-node* pada tipe data E-n22-k4. Gambar 1 menunjukkan pemetaan letak antar *node* pada set data E-n22-k4, E-n23-k3, dan E-n30-k3. Adapun pemetaan untuk E-n13-k4 tidak dapat dilakukan karena data yang tersedia tidak mencantumkan koordinat masing-masing *node*, akan tetapi jarak antar masing-masing *node*. Karena pada tahap inialisasi dilakukan

**Tabel 2.** Perbandingan solusi pada set data E-n30-k3

Tipe set data	<i>Best known</i>	<i>Best test</i> CODEQ (Garside dan Sudaningtyas)	Waktu komputasi (menit)	<i>Best test</i> CODEQ (Narendra [8])	Waktu komputasi
E-n30-k3	534	605,47	513,28	606,14	570,8



**Gambar 1.** Pemetaan *node* pada set data Christofides dan Eilon [12]

pembangkitan solusi awal dengan menggunakan metode *nearest neighbor*, maka kondisi *node* yang cenderung terklaster ini membuat program CODEQ-VRP dapat dengan cepat menghasilkan solusi yang mendekati solusi *best-known*.

Nilai RPD dapat diperbaiki apabila dilakukan penambahan jumlah populasi yang dibangkitkan di iterasi awal. VRP merupakan permasalahan kombinatorial yang berarti bahwa dengan semakin banyaknya jumlah *node*, maka ruang solusi permasalahan ini akan semakin besar sehingga dengan semakin bertambahnya jumlah populasi, maka kemungkinan untuk mencapai solusi optimal akan semakin bertambah besar pula. Akan tetapi, tidak dapat dipungkiri bahwa dengan semakin banyaknya jumlah populasi yang dibangkitkan, maka jumlah vektor  $v_i(t)$  yang dibangkitkan juga akan semakin banyak, yang akan berpengaruh pada waktu komputasi.

Atas dasar pertimbangan tersebut, maka pencarian solusi dilakukan sedemikian sehingga hasilnya masih dapat diterima (ditunjukkan dengan nilai RPD yang kurang dari 10%) sekaligus juga dengan waktu komputasi yang masih dapat diperimbangkan.

Performansi algoritma CODEQ dalam menyelesaikan VRP akan dilihat juga berdasarkan pendekatan penyelesaian yang digunakan.

Tabel 2 menunjukkan perbandingan solusi pada set data E-n30-k3 dengan menggunakan 2 pendekatan yang ada yaitu *route first-cluster second* dan *cluster first-route second*. Solusi dengan menggunakan *cluster first-route second* sebesar 606,14 dengan waktu komputasi sebesar 570,8 menit (Narendra,

[8]). Sedangkan solusi dengan menggunakan pendekatan *route first-cluster second* yang diusulkan peneliti menghasilkan solusi yang lebih kecil yaitu sebesar 605,47 dengan waktu komputasi yang lebih singkat yaitu 513,28 menit.

### Simpulan

Algoritma CODEQ berhasil dikembangkan untuk menyelesaikan *vehicle routing problem* dengan menerapkan prinsip *integer ordinal number* untuk mengubah bilangan kontinu menjadi bilangan diskrit pada saat mutasi dan *crossover*. Berdasarkan hasil deviasi rata-rata pengujian yaitu sebesar 6,35%, maka dapat disimpulkan bahwa CODEQ dapat menyelesaikan permasalahan VRP dengan baik. Lebih jauh lagi, dapat pula disimpulkan bahwa algoritma CODEQ juga dapat digunakan untuk menyelesaikan permasalahan diskrit.

### Ucapan Terima Kasih

Peneliti mengucapkan terima kasih kepada Reviewer yang telah banyak memberikan masukan untuk perbaikan paper ini.

### Daftar Pustaka

1. Lawler, E. L., *Combinatorial Optimization: Networks and Matroids*, Dover Publications Inc., New York, 2012.
2. Kallehauge, B., J. L., and Marsen, O. B. G., *Lagrangian Duality Applied on Vehicle Routing with Time Windows*. Technical Report. IMM, Technical University of Denmark, 2001.
3. Omran, M. G. H., and Salman, A., Constrained Optimization using CODEQ, *Journal of Chaos, Solutions and Fractal*, 42, 2009, pp. 662-668.
4. Omran, M. G. H., and al-Sharhan, S., *Optimization of Discrete Values Using Recent Variants of Differential Evolution*, Proceedings of IAESTED International Conference on Computational Intelligence, 2009.
5. Chiou, J., and Chang, C., A Novel Evolutionary Algorithm for Capacitor Placement in Distribution Systems, *Journal of Engineering Technology*, 2(3), 2013, DOI: 10.7603/s40707-013-0003-x
6. Omran, M. G. H., and al-Adwani, F., Using CODEQ to Train Feed-Forward Neural Networks. e-print, Cornell University Library, 2009 <URL: ,<http://arxiv4.library.cornell.edu/ftp/arxiv/papers/1002/1002.0745.pdf>>, diakses pada 3 Maret 2012.
7. Omran, M., CODEQ: An Efficient Meta-Heuristic for Continuous Global Optimization. *International Journal of Metaheuristic*, 1(2), 2010, pp. 108-131.
8. Narendra, Y. N., *Penggunaan Metode CODEQ untuk Menyelesaikan Permasalahan Capacitated Vehicle Routing Problem*, Tugas Akhir Jurusan Teknik Industri, Institut Teknologi Sepuluh Nopember, Surabaya, 2010.
9. Fuellerer, G., *Transportation Logistics*, URL <http://neumann.hec.ca/chairedistributique/common/transportationlogistics.pdf>>, diakses 16 April 2014.
10. Mingyong, L., and C. Erbao, An Improved Differential Algorithm for Vehicle Routing Problem with Simultaneous Pickups and Deliveries and Time Windows, *Journal of Engineering Applications of Artificial Intelligence*, 23, 2010, pp. 188-195.
11. Rera, G. F., *Penerapan Metode Cross Entropy dalam Penyelesaian Capacitated Vehicle Routing Problem, Studi Kasus: Distribusi Koran Jawa Pos se-Surabaya*, Tugas Akhir Jurusan Teknik Industri. Institut Teknologi Sepuluh Nopember, Surabaya, 2010.
12. Christofides, N., and Eilon, S., Expected Distances in Distribution Problems, *Operational Research Quarterly*, 20, 1969, pp. 437-443.
13. Indriana, Y., *Perencanaan Rute Pendistribusian Produk dengan Menggunakan Metode Saving Matriks untuk Menghemat Biaya Transportasi Studi Kasus di CV. Cempaka Tulungagung*, Skripsi Jurusan Teknik Industri. Universitas Muhammadiyah Malang, 2011.