Penyelesaian CVRP dengan Menggunakan Savings Algorithm & VBA Excel

Ekra Sanggala

D4 Logistik Bisnis – Politeknik POS Indonesia
Jl. Sariasih no 54, Bandung
ekrasanggala@mail.ru

Dikirimkan: 03, 2020. Diterima: 03, 2020. Dipublikasikan: 03, 2020.

Abstract— *Capacitated Vehicle Routing Problem (CVRP) is a logistics problem that call scientists finding a tool to solve it. One of many CVRP tools is Savings Algorithm. To Execute this algorithm manually will consume long time, for that needed an application to help executing it. Visual Basic Application for Excel (VBA Excel) can be used to create this application. Created application can execute Saving Algorithm, although this time it fits only on practice and training this application still can be upgraded.*

Keywords – CVRP, Savings Algorithm, VBA Excel

Abstrak – *Capacitated Vehicle Routing Problem (CVRP) merupakan sebuah masalah logistik yang banyak diteliti oleh para ilmuwan. Salah satu algoritma yang populer untuk menyelesaikannya adalah Algoritma Savings. Jika perhitungan Algoritma Savings dilakukan secara manual tentu akan banyak menghabiskan waktu, oleh karena itu diperlukan sebuah aplikasi untuk membantu menyelesaikan perhitungannya. Visual Basic Application for Excel (VBA Excel) dapat digunakan untuk membuat aplikasi ini. Aplikasi yang dihasilkannya ini dapat menyelesaikan perhitungan Algoritma Savings, walaupun sampai saat ini hanya cocok untuk keperluan latihan dan sebagai media pembelajaran, aplikasi ini masih bisa dikembangkan lagi.*

Kata kunci – CVRP, Algoritma Savings, VBA Excel

I. PENDAHULUAN

Vehicle Routing Problem (VRP) merupakan salah satu masalah pada logistik yang sangat menarik perhatian para peneliti untuk mengembangkan metode-metode solusinya [1]. Jenis-jenis dari VRP ini antara lain *Capacitated VRP (CVRP)*, *VRP with Time Windows (VRPTW)*, *VRP with Backhauls (VRPB)* dan *VRP with Pickup and Delivery (VRPPD)* [2].

Dalam penelitian sebelumnya [3] dilakukan penentuan jarak pendistribusian dengan metode Algoritma Heuristik, yaitu dengan tipe permasalahan *Traveling Salesman Problem (TSP)*. Dan penelitian lainnya [4] mengenai optimalisasi penentuan rute distribusi untuk meminimalkan biaya transportasi dengan metode *saving matrix*. Penelitian yang dilakukan bertujuan untuk menentukan jarak tempuh minimum untuk setiap rute.

Pada *paper* ini dibahas mengenai pengembangan aplikasi menggunakan *Visual Basic Application (VBA) Excel* untuk menjalankan *Savings Algorithm* dalam menyelesaikan masalah CVRP.

CVRP merupakan masalah paling sederhana dari VRP. Dalam CVRP, jumlah pelanggan dan kebutuhannya telah diketahui dan pemenuhan

kebutuhan ini hanya bisa dipenuhi dalam sekali kunjungan. Seluruh kendaraan yang digunakan merupakan kendaraan yang setipe serta jumlah depotnya hanya satu. Satu-satunya pembatas adalah kapasitas kendaraan. Tujuannya adalah meminimalkan jarak atau biaya pengiriman untuk melayani seluruh pelanggan [2].

Clarke dan Wright pada tahun 1964 mempublikasikan sebuah algoritma untuk menyelesaikan CVRP. Konsep ini berdasarkan pada konsep *savings* [5].

Penggunaan *Microsoft Office Excel* dapat memberi kemudahan dan kenyamanan dalam melakukan berbagai perhitungan [6]. Untuk contoh penggunaannya dalam menyelesaikan kasus CVRP akan digunakan sebuah kasus sederhana.

II. METODOLOGI PENELITIAN

Seperti telah sedikit dibahas pada bagian I, metode yang digunakan adalah algoritma *Savings* dan *VBA Excel*.

Algoritma *Savings* merupakan sebuah algoritma heuristik, oleh karena itu algoritma ini

belum tentu memberikan hasil yang terbaik, tetapi setidaknya memberikan hasil yang baik. Konsep dasar dari *Savings* adalah penghematan biaya yang diperoleh berdasarkan penggabungan dua rute, gambaran mengenai penggabungan tersebut dapat dilihat pada gambar 1, dimana titik 0 merupakan depot [3].

Gambar 1: Contoh penggabungan 2 rute [3].

Pada gambar 1(a) terlihat bahwa *customer i* dan *customer j* dilayani oleh rute yang berbeda. Pada gambar 1(b) diperlihatkan bahwa *customer i* dan *customer j* dilayani oleh rute yang sama. Dapat diketahui ongkos transportasi D_a pada gambar 1(a) adalah sebagai berikut ini [3]:

$$D_a = c_{0i} + c_{i0} + c_{0j} + c_{j0}$$

Sedangkan untuk ongkos transportasi D_b pada gambar 1(b) dapat diketahui sebagai berikut ini [3]:

$$D_b = c_{0i} + c_{ij} + c_{j0}$$

Nilai *savings* diperoleh dengan mengurangi ongkos transportasi D_a dengan ongkos transportasi D_b , sehingga diperoleh persamaan nilai *savings* S_{ij} berikut ini [3]:

$$S_{ij} = D_a - D_b = c_{i0} - c_{0j} - c_{ij}$$

Algoritma *Savings* ini terdapat dua versi, yaitu versi *sequential* dan versi *parallel*. Pada versi *sequential* hanya satu rute yang dibuat pada proses pembentukan rute, sedangkan pada versi *parallel* lebih dari satu rute yang dibuat pada proses pembentukan rute. Langkah pertama yang harus dilakukan adalah menghitung *savings* untuk setiap pasangan *customer*. Kemudian nilai-nilai *savings* yang telah diperoleh, diurutkan mulai dari nilai *savings* terbesar sampai ke nilai *savings* terkecil. Pembentukan rute di versi *sequential* adalah dengan memasukkan ke dalam rute terlebih dahulu pasangan-pasangan *customer* yang memiliki nilai *savings* lebih tinggi pada satu waktu pembentukan rute. Sedangkan di versi *parallel* dibuat lebih dari satu rute pada satu waktu pembentukan rute, walaupun cara memasukkan pasangan-pasangan *customer* ke dalam rute tetap sama, yaitu berdasarkan nilai *savings* yang lebih tinggi [5].

VBA merupakan salah satu aplikasi *scripting language* terbaik dari *Microsoft*, dan VBA ini terdapat pada aplikasi *Microsoft Office*. Bahasa *BASIC* mendapatkan respek di tahun 1991 ketika *Microsoft* meluncurkan *Visual Basic (VB)* untuk sistem operasi *Windows*. VB membantu para *programmer* untuk mengembangkan aplikasi pada sistem operasi *Windows*. VB inilah yang menjadi dasar dari VBA [7].

Kombinasi antara VBA dengan *Excel* menjadi sebuah alat yang sangat baik. Ada sekitar 400 juta orang yang menggunakan *Microsoft Office* dan sebagian besar tidak memahami bagaimana memanfaatkan kehebatan VBA *Excel*. Dengan menggunakan VBA, tugas-tugas kita dalam *Excel* dapat diselesaikan lebih cepat. Jika kita menggunakan *Excel* untuk menyelesaikan tugas-tugas rutin, kita dapat menggunakan VBA untuk menyelesaikannya hanya dalam beberapa detik [8].

III. HASIL PENELITIAN

Pada bagian ini akan dibahas mengenai aplikasi yang telah dibuat menggunakan VBA *Excel* untuk menyelesaikan CVRP dengan *Savings Algorithm*. Karena aplikasi ini dibuat saat penulis sedang belajar bahasa Spanyol maka istilah-istilah dalam aplikasi ditulis dalam bahasa Spanyol, tapi para pembaca tidak perlu khawatir karena istilah-istilah tersebut akan disertakan bahasa Indonesianya. Adapun *form* dan format yang didapat dari aplikasi tersebut dapat dilihat pada gambar 2 – 9.

Clarke & Wright's Saving Algorithm

Número de nodos	
Tipo de distancias	
Número de vehículos	
Tipo de vehículos	
Aplicación Clarke y Wriqh	

Gambar 2: Form pengisian informasi.

Los Vehiculos	
Los Numeros	Las Capacidades
Todos Vehiculos	

Gambar 3: Form kendaraan.

Las Demandas	
Los Nodos	Las Demandas
1	
2	
3	
4	
5	

Gambar 4: Form demand.

Los Nodos / Las Distancias	0	1	2
0	0		
1		0	
2			0
3			
4			
5			

Gambar 5: Form titik/jarak.

Savings	1	2	3
1			
2			
3			
4			
5			

Gambar 6: Format nilai savings.

Savings			
Los Nodos	Savings	El Ranking	Las Demandas

Gambar 7: Format pengurutan nilai savings.

Las Rutas				
Las Rutas	Los Nodos	Los Vehiculos	Los Kilometrajes	Las Cargas

Gambar 8: Format rute yang dihasilkan.

Los Vehiculos			
Los Numeros	Las Rutas	Los Kilometrajes	Las Cargas

Gambar 9: Format informasi penggunaan kendaraan.

IV. PEMBAHASAN

Pada aplikasi ini terdapat dua *sheet* yaitu *sheet options* (*La Opciones*) dan *sheet* perhitungan (*El Calculo*). Pada *sheet options* hal pertama yang harus dilakukan adalah memasukkan informasi sebagai berikut ini:

1. Jumlah titik (*Numero de nodos*).

Berapa banyak titik/*customer* yang harus dikunjungi.

2. Jenis jarak (*Tipo de distancias*).
Jarak antar titiknya berupa koordinat atau jarak simetris atau jarak asimetris.
3. Jumlah kendaraan (*Numero de vehiculos*).
Jumlah kendaraan yang bisa digunakan.
4. Jenis kendaraan (*Tipo de vehiculos*).
Kendaraan-kendaraan yang akan digunakan kapasitasnya homogen atau heterogen.
5. Versi Algoritma *Savings* (*Aplicacion Clarke y Wright*).
Versi Algoritma *Savings* yang akan digunakan *sequential* (*secuencial*) atau *parallel* (*paralelo*).

Form pengisian informasi yang disebutkan sebelumnya dapat dilihat pada gambar 2.

Setelah informasi yang diperlukan sudah diisi secara lengkap dan benar, maka tombol "Langkah Selanjutnya" ("*Siguiente Paso*") perlu ditekan. Tombol ini berguna untuk membuat 3 buah *form*, yaitu *form* Kendaraan (*Los Vehiculos*), *form* Demand (*Las Demandas*) dan *form* Titik/Jarak (*Los Nodos/Las Distancias*). *Form* Kendaraan berguna untuk menampung informasi kapasitas kendaraan dan *form* Demand akan digunakan untuk menerima informasi *demand* setiap titik/*customer*, sedangkan *form* Titik/Jarak disiapkan untuk pengisian informasi koordinat atau jarak. Bentuk dari ketiga *form* tersebut dapat dilihat pada gambar 3, gambar 4 dan gambar 5.

Apabila ketiga *form* diatas sudah diisi dengan lengkap dan benar, perlu dilakukan penekanan pada tombol "Menghitung" (*Calcular*). Setelah tombol ini ditekan maka aplikasi akan melakukan perhitungan untuk menghasilkan sejumlah rute. Pada proses perhitungan ini akan ditampilkan nilai *Savings* yang dihasilkan, bentuk tampilannya kurang lebih seperti pada gambar 6. Nilai *Savings* yang dihasilkan akan diurutkan dari nilai yang terbesar ke nilai yang terkecil, untuk tampilan hasil pengurutan ini dapat dilihat pada gambar 7. Setelah nilai *Savings* diurutkan, maka akan ditampilkan sejumlah rute yang dihasilkan, bentuk informasi rute yang dihasilkan tampak seperti pada gambar 8. Aplikasi juga akan menghasilkan informasi mengenai penggunaan kendaraanya, untuk lebih jelasnya dapat dilihat pada gambar 9.

Kode program yang ditulis untuk menghasilkan aplikasi ini ada kurang lebih 1.250 baris yang terdiri dari sekitar 71.000 karakter. Jika kode tersebut ditampilkan pada *paper* ini tentu akan banyak menghabiskan tempat, oleh karena itu bagi pembaca yang memerlukan kodenya atau

bahkan *soft file* dari aplikasi ini, dapat langsung menghubungi penulis melalui surat elektronik.

Akan terasa kurang lengkap jika penggunaan aplikasi ini tidak disertai contoh penggunaannya. Sebagai contoh, sebuah kasus *CVRP* sederhana akan diselesaikan menggunakan aplikasi ini. Penyelesaian kasus tersebut dapat dilihat pada bagian lampiran.

V. KESIMPULAN

Aplikasi ini cocok digunakan untuk keperluan latihan dan sebagai media pembelajaran untuk memahami penyelesaian *CVRP* dengan Algoritma *Savings*.

Penelitian ini bisa menjadi dasar untuk pengembangan aplikasi yang cocok untuk penyelesaian kasus *CVRP* yang lebih kompleks, dan juga untuk keperluan industri. Satu hal yang perlu diperhatikan untuk penelitian selanjutnya adalah kemampuan komputasi yang lebih baik, yang tentunya memerlukan teknik pemrograman yang lebih tinggi. Hal lainnya yang perlu diperhatikan adalah *User Interface* yang lebih baik, sehingga lebih memudahkan para penggunanya.

REFERENSI

- [1] H. Paessens, *The Savings Algorithm for The Vehicle Routing Problem*, European Journal of Operational Research, Vol. 34, pp.336-344, 1988.
- [2] P. Toth and D. Vigo, *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications, 2002.
- [3] A. Sutoni and N. Asilah, "Penentuan Jarak Pendistribusian Gas LPG 3 Kg Dengan Metode Algoritma Heuristik," *J. Media Tek. dan Sist. Ind.*, vol. 2, no. 2, p. 37, 2018, doi: 10.35194/jmtsi.v2i2.416.
- [4] A. Sutoni and I. Apipudin, "Optimalisasi Penentuan Rute Distribusi Pupuk Untuk Meminimalkan Biaya Transportasi Dengan Metode Saving Matrix," *Spektrum Ind.*, vol. 17, no. 2, pp. 143-156, 2019, doi: <http://dx.doi.org/10.12928/si.v17i2.13139>.
- [5] J. Lysgaard, Clarke & Wright's Savings Algorithm, Department of Management Science and Logistics, The Aarhus School of Business, Sep. 2007.
- [6] M. Catana, C. Neagu and S. Tonoiu, Scheduling of Job-Shop Production Projects with Microsoft Office Excel, *Academic Journal of Manufacturing Engineering*, Vol. 7, pp.31-36, Apr. 2009.
- [7] J. Walkenbach, *Excel 2010 Power Programming with VBA*, Wiley Publishing, 2010.
- [8] B. Jelen and T. Syrtstad, *VBA and Macros for Microsoft Excel*, Sams Publishing, 2004.

LAMPIRAN

Diketahui perusahaan X adalah produsen vodka. Perusahaan tersebut mempunyai 10 gudang distribusi, setiap gudang mempunyai kebutuhan yang berbeda-beda. Kebutuhan-kebutuhan tersebut adalah sebagai berikut ini:

Gudang	Kebutuhan (Botol)
1	22
2	31
3	28
4	27
5	27
6	27
7	23
8	33
9	31
10	28

Jarak depot ke gudang-gudang dan jarak antar gudang adalah simetris, dengan jarak sebagai berikut ini:

Jarak (Km)	0	1	2	3	4	5	6	7	8	9	10
0	0	2	4	1	4	9	4	1	10	2	9
1		0	6	8	10	10	1	7	5	5	7
2			0	10	5	3	6	2	9	1	7
3				0	6	5	3	6	7	10	4
4					0	7	8	2	3	5	5
5						0	4	2	10	5	2
6							0	1	3	10	8
7								0	9	3	8
8									0	7	7
9										0	2
10											0

Truk angkut yang akan digunakan adalah homogen, dengan kapasitas angkut maksimal adalah 100 botol.

Untuk menyelesaikan contoh kasus diatas adalah diawali dengan memasukkan informasi pada *form* yang telah disediakan oleh aplikasi, sehingga tampilan *form*-nya seperti berikut ini:

Clarke & Wright's Saving Algorithm

Número de nodos	10
Tipo de distancias	2. Distancias Simetricas
Número de vehículos	1
Tipo de vehículos	1. Homogeneo
Aplicación Clarke y Wriqth	1. Secuencial

Kemudian tombol “Langkah Selanjutnya” ditekan sehingga aplikasi akan membuatkan *Form* Kendaraan, *Form Demand* dan *Form Titik/Jarak*. Selanjutnya ketiga *form* dilengkapi, maka hasilnya akan seperti berikut ini:

Los Vehiculos	
Los Numeros	Las Capacidades
Todos Vehiculos	100

Las Demandas	
Los Nodos	Las Demandas
1	22
2	31
3	28
4	27
5	27
6	27
7	23
8	33
9	31
10	28

Los Nodos / Las Distancias	0	1	2	3	4	5	6	7	8	9	10
0	0	2	4	1	4	9	4	1	10	2	9
1		0	6	8	10	10	1	7	5	5	6
2			0	10	5	3	6	2	9	1	7
3				0	6	5	3	6	7	10	4
4					0	7	8	2	3	5	5
5						0	4	2	10	5	2
6							0	1	3	10	8
7								0	9	3	8
8									0	7	7
9										0	2
10											0

Kemudian tombol “Menghitung” ditekan, maka aplikasi akan melakukan perhitungan untuk menentukan rute-rutenya. Hasilnya adalah seperti berikut ini:

Savings	1	2	3	4	5	6	7	8	9	10
1	0	0	-5	-4	1	5	-4	7	-1	5
2		0	-5	3	10	2	3	5	5	6
3			0	-1	5	2	-4	4	-7	6
4				0	6	0	3	11	1	8
5					0	9	8	9	6	16
6						0	4	11	-4	5
7							0	2	0	2
8								0	5	12
9									0	9
10										0

Savings

Los Nodos	Savings	El Ranking	Las Demandas
5-10	16	1	55
8-10	12	2	61
6-8	11	3	60
4-8	11	4	60
2-5	10	5	58
5-8	9	6	60
5-6	9	7	54
9-10	9	8	59
5-7	8	9	50
4-10	8	10	55
1-8	7	11	55
4-5	6	12	54
2-10	6	13	59
3-10	6	14	56
5-9	6	15	58
2-9	5	16	62
1-10	5	17	50
3-5	5	18	55
6-10	5	19	55
8-9	5	20	64
1-6	5	21	49
2-8	5	22	64
6-7	4	23	50
3-8	4	24	61
2-4	3	25	58
4-7	3	26	50
2-7	3	27	54
7-10	2	28	51
2-6	2	29	58
3-6	2	30	55
7-8	2	31	56
1-5	1	32	49
4-9	1	33	58
1-2	0	34	53
7-9	0	35	54
4-6	0	36	54
1-9	-1	37	53
3-4	-1	38	55
6-9	-4	39	58
1-7	-4	40	45
1-4	-4	41	49
3-7	-4	42	51
1-3	-5	43	50
2-3	-5	44	59
3-9	-7	45	59

Las Rutas

Las Rutas	Los Nodos	Los Vehiculos	Los Kilometrajes	Las Cargas
1	0 - 5 - 10 - 8 - 0	1	28	88
2	0 - 4 - 2 - 9 - 0	1	12	89
3	0 - 1 - 6 - 7 - 0	1	5	72
4	0 - 3 - 0	1	2	28
		Total	47	

Los Vehiculos

Los Numeros	Las Rutas	Los Kilometrajes	Las Cargas
1	4	47	277

Dari hasil perhitungan oleh aplikasi dapat disimpulkan beberapa hal, yaitu:

1. Dihasilkan 4 rute, yaitu 0-5-10-8-0 dengan jarak tempuh 28 km dan mengangkut 88 botol, 0-4-2-9-0 yang menempuh jarak 12 km dan membawa 89 botol, 0-1-6-7-0 yang jarak tempuhnya 5 km serta mengangkut 72 botol, dan rute terakhir adalah 0-3-0 yang jarak tempuhnya adalah 2 km serta mengangkut 28 botol.
2. Total jarak tempuh nya adalah $28+12+5+2=47$ km.
3. Jumlah botol yang diangkut adalah sebanyak 277 botol.