



Excel Solver dengan Evolutionary Algorithm sebagai alternatif penyelesaian Capacitated Vehicle Routing Problem (CVRP)

Ekra Sanggala^{1✉}, Febriani Sulistyaningsih¹, Saepudin Nirwan²

⁽¹⁾Prodi D4 Logistik Bisnis, Sekolah Vokasi, Universitas Logistik dan Bisnis Internasional, Bandung

⁽²⁾Prodi D3 Teknik Informatika, Sekolah Vokasi, Universitas Logistik dan Bisnis Internasional, Bandung

DOI: [10.31004/jutin.v9i1.53909](https://doi.org/10.31004/jutin.v9i1.53909)

✉ Corresponding author:
ekrasanggala@mail.ru

Article Info	Abstrak
Kata kunci: <i>Excel Solver;</i> <i>Metaheuristic;</i> <i>Evolutionary Algorithm;</i> <i>CVRP;</i> <i>instance A-n32-k5</i>	<i>Capacitated Vehicle Routing Problem (CVRP)</i> merupakan salah satu varian <i>VRP</i> yang paling populer. Pada <i>CVRP</i> ini, benda-benda yang akan diangkut hanya dipertimbangkan massanya saja, biasanya benda-benda tersebut dianggap sebagai benda-benda yang sejenis. <i>Evolutionary Algorithm (EA)</i> merupakan sebuah <i>metaheuristic</i> yang dapat diaplikasikan pada berbagai permasalahan optimasi, termasuk <i>CVRP</i>. <i>Solver</i> merupakan <i>Excel Add-In</i> yang mampu menjalankan <i>EA</i>. Dengan adanya kemampuan <i>EA</i> untuk menyelesaikan <i>CVRP</i> dan <i>Solver</i> yang mampu menjalankan <i>EA</i>, maka dapat disimpulkan bahwa penyelesaian <i>CVRP</i> dapat dilakukan dengan memanfaatkan <i>Solver</i>. Untuk <i>instance A-n32-k5</i>, solusi yang dihasilkan dari <i>Solver</i> dengan <i>EA</i> adalah 824 atau 5,1% lebih panjang dari <i>Best Known Solution</i>.
Keywords: <i>Excel Solver;</i> <i>Metaheuristic;</i> <i>Evolutionary Algorithm;</i> <i>CVRP;</i> <i>instance A-n32-k5</i>	Abstract <i>The Capacitated Vehicle Routing Problem (CVRP) is one of the most popular variants of VRP. In CVRP, only the mass of objects to be transported is considered, typically assuming they are of the same type. Evolutionary Algorithm (EA) is a metaheuristic that can be applied to various optimization problems, including CVRP. Solver is an Excel Add-In that is capable of running EA. With the ability of Evolutionary Algorithm to tackle CVRP and the Solver which is capable of running EA, it can be concluded that solving CVRP can be done using the Solver. For instance A-n32-k5, the solution generated by Solver with EA is 824 or 5.1% longer than the Best Known Solution.</i>

1. PENDAHULUAN

Pada beberapa dekade terakhir ini pada manajemen pendistribusian produk telah terjadi peningkatan penggunaan metode-metode optimasi yang berdasarkan teknik-teknik *Operations Research* dan *Mathematical Programming*. Pada aplikasi di dunia nyata, di Amerika Utara dan Eropa penggunaan metode-metode optimasi

ini telah berhasil menghemat biaya transportasi sebanyak 2% sampai dengan 20%. Salah satu faktor keberhasilan ini adalah karena ditemukannya algoritma-algoritma yang mampu memberikan solusi yang baik dengan lama waktu perhitungan yang wajar (Toth & Vigo, 2002).

Vehicle Routing Problem (VRP) adalah suatu permasalahan optimasi yang fokus pada pendistribusian barang antara *depot* dan pelanggan. Beberapa contoh VRP pada kehidupan nyata adalah rute pengambilan sampah dari rumah-rumah, rute membersihkan jalan, rute bis sekolah, rute transportasi untuk orang-orang berkebutuhan khusus, rute *sales* dan rute pemeliharaan tower (Abidin et al., 2025). VRP mempunyai beberapa varian, antara lain *Capacitated Vehicle Routing Problem (CVRP)*, *Multiple Depot Vehicle Routing Problem (MDVRP)*, *Periodic Vehicle Routing Problem (PVRP)*, *Split Delivery Vehicle Routing Problem (SDVRP)*, *Vehicle Routing Problem with Satellite Facilities (VRP-SF)*, *Heterogeneous Fleet Vehicle Routing Problem (HF-VRP)*, *Vehicle Routing Problem with Time Windows (VRPTW)*, *Green Vehicle Routing Problem (G-VRP)*, *Generalized Vehicle Routing Problem (GVRP)*, *Stochastic Vehicle Routing Problem (SVRP)*, *Rich Vehicle Routing Problem (RVRP)*, *Open Vehicle Routing Problem (OVRP)*, *Vehicle Routing Problem with Pick-Up and Delivery (VRPPD)* dan *Asymmetric Vehicle Routing Problem (AVRP)* (Sharma et al., 2018).

Capacitated Vehicle Routing Problem (CVRP) merupakan salah satu varian VRP yang paling populer. Pada CVRP ini, benda-benda yang akan diangkut hanya dipertimbangkan massanya saja, biasanya benda-benda tersebut dianggap sebagai benda-benda yang sejenis. Pada CVRP, kebutuhan setiap pelanggan diekspresikan dengan sebuah bilangan bulat positif yang mewakili jumlah massa atau volume dari benda yang dibutuhkan. Pada kasus ini, pengujian kelayakan sebuah solusi hanya berdasarkan total kebutuhan dari sejumlah pelanggan yang dialokasikan pada kendaraan pengangkut, tidak melebihi kapasitas maksimal daya angkutnya (Hameed et al., 2025).

CVRP termasuk kedalam *NP-Hard Problem* (Kalatzantonakis et al., 2020). *NP-Hard Problem* merupakan permasalahan yang jika diselesaikan dengan *Exact Method* memerlukan waktu yang sangat lama, sehingga tidak wajar untuk ditunggu (Garey, 1979). Algoritma yang bekerja berdasarkan *metaheuristic* dapat menjadi sebuah solusi untuk menyelesaikan *NP-Hard Problem* dengan waktu perhitungan yang wajar untuk ditunggu, walaupun solusi yang dihasilkan belum tentu merupakan solusi yang terbaik (Agrawal et al., 2021).

Beberapa *metaheuristic* yang populer dalam menyelesaikan *Capacitated Vehicle Routing Problem (CVRP)* antara lain *Evolutionary Algorithm (EA)*, *Genetic Algorithm (GA)*, *Ant Colony Optimization (ACO)*, *Particle Swarm Optimization (PSO)*, *Tabu Search (TS)* dan *Simulated Annealing (SA)* (Dorransoro et al., 2022). Keunggulan dari *metaheuristic* adalah dapat dirancang sedemikian rupa sehingga sesuai dengan permasalahan yang akan diselesaikan (Chopard & Tomassini, 2018). Hasil perancangan sebuah *metaheuristic* sangat tergantung dari logika dan kreatifitas perancangannya, sehingga hasil rancangannya dapat berbeda-beda walaupun tujuannya sama (Yu & Gen, 2010).

Evolutionary Algorithm merupakan sebuah *metaheuristic* yang terinspirasi dari teori evolusi, dimana individu yang terkuat yang akan bertahan hidup (Badar, 2021). *Evolutionary Algorithm (EA)* dapat dirancang sehingga mampu menyelesaikan *Capacitated Vehicle Routing Problem* (Talbi, 2009).

Agar *Evolutionary Algorithm* yang telah dirancang dapat dijalankan, maka *Evolutionary Algorithm* tersebut perlu diterjemahkan ke dalam bahasa pemrograman (Sanggala & Bisma, 2025). Dengan demikian dalam menyelesaikan permasalahan *Capacitated Vehicle Routing Problem* dengan *Evolutionary Algorithm* diperlukan kemampuan *programming* dan ketersediaan waktu untuk melakukan *programming*. Dua hal tersebut sering menjadi hambatan dalam penggunaan *metaheuristic*, dengan demikian perlu adanya alternatif penyelesaian *Capacitated Vehicle Routing Problem* tanpa melakukan *programming*.

Microsoft Excel merupakan aplikasi *spreadsheet* yang dapat digunakan untuk menyelesaikan berbagai permasalahan optimasi, termasuk *Capacitated Vehicle Routing Problem* (Erdoğan, 2017). *Tools* khusus pada *Microsoft Excel* untuk menyelesaikan permasalahan optimasi adalah *Solver* (Singh et al., 2022). *Solver* merupakan *Excel Add-In* untuk menyelesaikan permasalahan optimasi, yang bertugas untuk menemukan nilai-nilai untuk sekumpulan variabel keputusan dengan tujuan memaksimalkan atau meminimumkan tujuan (Selvi et al., 2022). *Solver* dapat menggunakan tiga algoritma untuk menemukan nilai optimal yang akan digunakan pada variabel-variabel keputusan. Tiga algoritma tersebut adalah *Linear Programming Simplex*, *GRG Nonlinear* dan *Evolutionary Algorithm* (Gilbert et al., 2023).

Dengan adanya kemampuan *Evolutionary Algorithm* untuk menyelesaikan *Capacitated Vehicle Routing Problem* dan *Solver* yang mampu menjalankan *Evolutionary Algorithm*, maka dapat disimpulkan bahwa penyelesaian *Capacitated Vehicle Routing Problem* dapat dilakukan dengan memanfaatkan *Solver* tanpa perlu melakukan *programming*. Proses *programming* akan digantikan oleh proses pembuatan *template Spreadsheet*

untuk menerjemahkan *Evolutionary Algorithm* yang dirancang menjadi solusi dari *Capacitated Vehicle Routing Problem*. Proses pembuatan *template Spreadsheet* tersebut akan memanfaatkan *formula-formula* yang terdapat pada *Microsoft Excel*, dengan demikian proses ini lebih mudah dan lebih cepat dibandingkan dengan proses *programming*. Berdasarkan uraian diatas maka penggunaan *Excel Solver* dengan *Evolutionary Algorithm* dapat menjadi alternatif dalam penyelesaian *Capacitated Vehicle Routing Problem*.

2. METODE

Sebelum mulai mengembangkan *Evolutionary Algorithm* perlu diketahui kasus *Capacitated Vehicle Routing Problem* yang akan diselesaikan. Sebagai contoh, pada *paper* ini akan digunakan *instance A-n32-k5*. *Instance* tersebut terdiri dari 32 *node*, yaitu 1 *node* depot dan 31 *node customer*, kapasitas truk yang digunakan adalah 100 serta jarak antar *node*-nya berdasarkan *Euclidean Distance* (Augerat et al., 1995). *Best Known Solution* (BKS) untuk *instance* tersebut adalah 784 (Awad et al., 2018). Pada tabel 1 dapat dilihat koordinat dan *demand* untuk setiap *node*-nya.

Tabel 1. Koordinat dan Demand Instance A-n32-k5 (Martinelli, 2014)

Node	Koordinat		Demand
	X	Y	
0	82	76	0
1	96	44	19
2	50	5	21
3	49	8	6
4	13	7	19
5	29	89	7
6	58	30	12
7	84	39	16
8	14	24	6
9	2	39	16
10	3	82	8
11	5	10	14
12	98	52	21
13	84	25	16
14	61	59	3
15	1	65	22
16	88	51	18
17	91	2	19
18	19	32	1
19	93	3	24
20	50	93	8
21	98	14	12
22	5	42	4
23	42	9	8
24	61	62	24
25	9	97	24
26	80	55	2
27	57	69	20
28	23	15	15
29	20	70	2
30	85	60	14
31	98	5	9

Berikut ini merupakan contoh menghitung jarak antar *node* berdasarkan *Euclidean Distance* (Sanggala et al., 2024):

Node 0 (X: 82; Y: 76)

Node 1 (X: 96; Y: 44)

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

$$d_{01} = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}$$

$$d_{01} = \sqrt{(82 - 96)^2 + (76 - 44)^2}$$

$$d_{01} = \sqrt{(82 - 96)^2 + (76 - 44)^2}$$

$$d_{01} = 34,9285 \approx 35$$

Tahapan pertama dalam mengembangkan *Evolutionary Algorithm* adalah merancang kromosom yang dapat diterjemahkan menjadi sebuah solusi (Alromema & Makarem, 2024). Kromosom inilah yang akan terus mengalami evolusi sampai evolusi tersebut berhenti (Pétrowski & Ben-Hamida, 2017). Pada *paper* ini, kromosom akan dibagi menjadi dua bagian yaitu kromosom "Node" dan kromosom "Kunjungi Node Berikutnya". Panjang setiap bagian kromosom adalah 31 gen, sesuai dengan jumlah *customer* yang harus dikunjungi. Kromosom "Node" akan berisi urutan *node* yang akan dikunjungi. Sedangkan Kromosom "Kunjungi Node Berikutnya" akan berisi keputusan yang harus dilakukan oleh truk untuk "kembali ke *depot*" atau "mengunjungi *node* berikutnya". Jika keputusannya adalah "mengunjungi *node* berikutnya" maka akan dilakukan pengecekan terhadap sisa kapasitas truk dan *demand* dari urutan *node-node* yang belum terkunjungi, jika berdasarkan urutan *node-node* yang belum terkunjungi terdapat *demand* yang masih muat untuk diangkut maka truk akan mengunjungi *node* tersebut, tetapi jika setelah dilakukan pengecekan tidak terdapat satu pun *node* yang *demand*-nya masih muat untuk diangkut, maka truk akan kembali ke *depot*. Nilai gen pada Kromosom "Kunjungi Node Berikutnya" adalah 0 atau 1. 0 berarti "kembali ke *depot*" dan 1 berarti "mengunjungi *node* berikutnya". Sebelum *Evolutionary Algorithm* dijalankan oleh *solver*, nilai gen pada Kromosom "Node" diisi nilai antara 1 sampai dengan 31 secara *random* dan tidak boleh ada yang sama. Sedangkan Kromosom "Kunjungi Node Berikutnya" diisi nilai 0 atau 1 secara *random*. Pada gambar 1 dapat dilihat bentuk kromosom yang dirancang beserta nilai-nilai pada gen-nya.

	Gen																															
Kromosom	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
Node	30	19	11	15	31	29	16	22	23	25	27	1	26	12	28	10	3	24	9	5	2	21	20	14	6	18	13	17	4	8	7	
Kunjungi Node Berikutnya	1	1	1	0	1	1	1	1	1	0	1	1	0	0	1	1	0	1	1	1	0	1	1	0	1	1	0	1	1	1	1	1

Gambar 1. Bentuk Kromosom yang Dirancang

Berdasarkan nilai-nilai pada gen di gambar 1 maka Kromosom tersebut dapat diterjemahkan menjadi solusi dari *instance A-n32-k5*. Pada tabel 2 dapat dilihat solusinya.

Tabel 2. Solusi Berdasarkan Nilai-Nilai Gen Pada Gambar 1

Rute	Urutan Kunjungan	Panjang Rute
1	0 – 30 – 19 – 11 – 15 – 0	299
2	0 – 31 – 29 – 16 – 22 – 23 – 25 – 0	549
3	0 – 27 – 1 – 26 – 0	112
4	0 – 12 – 0	58
5	0 – 28 – 10 – 3 – 0	318
6	0 – 24 – 9 – 5 – 2 – 0	310
7	0 – 21 – 20 – 14 – 0	219
8	0 – 6 – 18 – 13 – 0	207
9	0 – 17 – 4 – 8 – 7 – 0	279
Total Panjang Rute		2351

Template Spreadsheet untuk menerjemahkan kromosom menjadi solusi dari *Capacitated Vehicle Routing Problem* dibuat sedemikian rupa sehingga mudah untuk dipahami. *Template Spreadsheet* ini terdiri dari dua sheet, yaitu sheet "Jarak" dan sheet "EA & CVRP". Sheet "Jarak" berisi matrik jarak antar *node*, sedangkan sheet "EA & CVRP" terdiri dari dua bagian, yaitu bagian "Kromosom" dan bagian "Solusi CVRP". Bagian "Kromosom" menampilkan hasil perancangan kromosom, *demand*, rute untuk setiap *node* dan kapasitas truk, sedangkan bagian "Solusi CVRP" menampilkan rute, urutan *node* pada setiap rute, nama *node*, *demand*, *demand* kumulatif, sisa kapasitas, jarak tempuh dari *node* ke *node* berikutnya, keputusan "Kunjungi Node Berikutnya", Keterangan 1, Keterangan 2, panjang setiap rute dan Total Panjang Rute. Pada gambar 2 dapat dilihat tampilan *Template Spreadsheet* untuk Sheet "EA & CVRP" bagian "Kromosom".

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG
2			Gen																														
3		Kromosom	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
4		Node																															
5		Kunjungi Node Berikutnya																															
6		Demand																															
7		Rute																															
8																																	
9		Kapasitas																															

Gambar 2. Template Spreadsheet untuk Sheet "EA & CVRP" bagian "Kromosom"

Pada gambar 3 dapat dilihat tampilan *Template Spreadsheet* untuk Sheet "EA & CVRP" bagian "Solusi CVRP".

10												
11	Rute	Node	Nama Node	Demand	Demand Kumulatif	Sisa Kapasitas	Jarak	Kunjungi Node Berikutnya	Keterangan 1	Keterangan 2	Panjang Rute	
12												
13												
14												
15												
101												
102												
103												
104												
105										Total Panjang Rute		

Gambar 3. Template Spreadsheet untuk Sheet "EA & CVRP" bagian "Solusi CVRP"

Agar *template Spreadsheet* tersebut dapat digunakan maka harus diisi oleh *formula-formula* yang mampu menerjemahkan kromosom menjadi solusi *Capacitated Vehicle Routing Problem*. *Formula-formula* tersebut adalah sebagai berikut ini:

- Bagian "Kromosom" – Demand
cell C6 : =INDEX(Jarak!\$F\$4:\$F\$35;MATCH('EA & CVRP'!C4;Jarak!\$B\$4:\$B\$35;0);1)
Copy-kan sampai cell AG6 !
- Bagian "Kromosom" – Rute
cell C7 : =IFERROR(INDEX(\$B\$12:\$B\$104;MATCH(C\$4;\$C\$12:\$C\$104;0);1);0)
Copy-kan sampai cell AG7 !
- Bagian "Kromosom" – Kapasitas
cell C9 : 100 (sesuai dengan kapasitas truk pada *instance A-n32-k5*)
- Bagian "Solusi CVRP" – Rute
cell B12: 1 (rute pertama adalah rute 1)
- Bagian "Solusi CVRP" – Rute
cell B13: =IF(K12="Semua Node Terkunjungi";"-";IF(J12="Kembali Ke Depot";B12+1;B12))
Copy-kan sampai cell B104 !
- Bagian "Solusi CVRP" – Node
cell C12: 0 (Node pertama dari rute 1 adalah 0 atau depot)
- Bagian "Solusi CVRP" – Node
cell C13: {=IF(B13<>"-";IF(J12="Berangkat Dari Depot";IFERROR(INDEX(\$C\$4:\$AG\$4;1;MATCH(1;IF(TRANPOSE(IF(IF(ISERROR(MATCH(\$C\$4:\$AG\$4;C\$12:C12;0))=TRUE;\$C\$6:\$AG\$6;"SAMPEU"))<=G12;\$C\$4:\$AG\$4;"SAMPEU"))<>"SAMPEU";1;"SAMPEU");0);0);IF(J12="Kembali Ke Depot";0;IF(I12=0;0;IFERROR(INDEX(\$C\$4:\$AG\$4;1;MATCH(1;IF(TRANPOSE(IF(IF(ISERROR(MATCH(\$C\$4:\$AG\$4;C\$12:C12;0))=TRUE;\$C\$6:\$AG\$6;"SAMPEU"))<=G12;\$C\$4:\$AG\$4;"SAMPEU"))<>"SAMPEU";1;"SAMPEU");0);0))));"-")}
Copy-kan sampai cell C104 !

- Bagian "Solusi CVRP" – Nama *Node*
 cell D12: =IF(B12<>"-";INDEX(Jarak!\$C\$4:\$C\$35;MATCH('EA & CVRP'!\$C12;Jarak!\$B\$4:\$B\$35;0);1);"-")
 Copy-kan sampai cell D104 !

- Bagian "Solusi CVRP" – *Demand*
 cell E12 : =IF(B12<>"-";INDEX(Jarak!\$F\$4:\$F\$35;MATCH('EA & CVRP'!\$C12;Jarak!\$B\$4:\$B\$35;0);1);"-")
 Copy-kan sampai cell E104 !

- Bagian "Solusi CVRP" – *Demand* Kumulatif
 cell F12 : =IF(B12<>"-";SUMIF(\$B\$12:B12;B12;\$E\$12:E12);"-")
 Copy-kan sampai cell F104 !

- Bagian "Solusi CVRP" – Sisa Kapasitas
 cell G12: =C9-F12

- Bagian "Solusi CVRP" – Sisa Kapasitas
 cell G13: =IF(B13<>"-";IF(J12="Kembali Ke Depot";\$C\$9;G12-E13);"-")
 Copy-kan sampai cell G104 !

- Bagian "Solusi CVRP" – Jarak
 cell H12: 0 (Jarak tempuh awal rute 1 adalah 0)

- Bagian "Solusi CVRP" – Jarak
 cell H13: =IF(B13<>"-";INDEX(Jarak!\$G\$4:\$AL\$35;MATCH('EA & CVRP'!\$C12;Jarak!\$B\$4:\$B\$35;0);MATCH('EA & CVRP'!\$C13;Jarak!\$G\$2:\$AL\$2;0));"-")
 Copy-kan sampai cell H104 !

- Bagian "Solusi CVRP" – Kunjungi *Node* Berikutnya
 cell I12 : =IF(B12<>"-";IF(C12=0;"-";INDEX(\$C\$5:\$AG\$5;1;MATCH(C12;\$C\$4:\$AG\$4;0)));"-")
 Copy-kan sampai cell I104 !

- Bagian "Solusi CVRP" – Keterangan 1
 cell J12 : Berangkat Dari Depot (Keterangan 1 dari rute 1 adalah Berangkat Dari Depot)

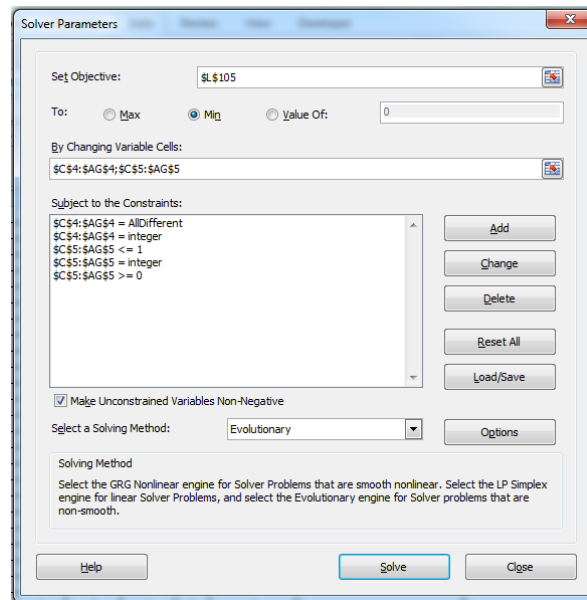
- Bagian "Solusi CVRP" – Keterangan 1
 cell J13 : =IF(B13<>"-";IF(AND(C13=0;C12=0)=TRUE;"Berangkat Dari Depot";IF(AND(C13=0;C12>0)=TRUE;"Kembali Ke Depot";"Perjalanan Dari "&D12&" Ke "&D13));"-")
 Copy-kan sampai cell J104 !

- Bagian "Solusi CVRP" – Keterangan 2
 cell K12: =IF(B12<>"-";IF(C12=0;IF(COUNTIF(\$C\$12:C12;">"&0)<\$AG\$3;"-";"Semua Node Terkunjungi");"-");"-")
 Copy-kan sampai cell K104 !

- Bagian "Solusi CVRP" – Panjang Rute
 cell L12 : =IF(B12<>"-";IF(J12="Kembali Ke Depot";SUMIF(\$B\$12:B12;B12;\$H\$12:H12);"-");"-")
 Copy-kan sampai cell L104 !

- Bagian "Solusi CVRP" – Total Panjang Rute
 cell L105 : =SUM(L12:L104)

Setelah *formula-formula* yang dibutuhkan selesai diisikan pada *template Spreadsheet* maka selanjutnya adalah mengisi *Solver Parameters*. Pada gambar 4 dapat dilihat nilai-nilai pada *Solver Parameters* yang harus diisikan.

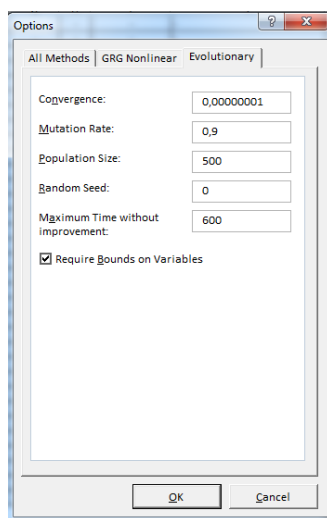


Gambar 4. Nilai-Nilai Pada *Solver Parameters* yang Harus Diisikan

Setelah mengisi nilai-nilai pada *Solver Parameters* maka selanjutnya adalah mengisi nilai-nilai pada *Options – Evolutionary*. Nilai-nilai pada *Options – Evolutionary* adalah sebagai berikut ini (Frontline, 2025):

- *Convergence*
Perbedaan antara nilai solusi terbaik yang telah didapatkan dengan nilai-nilai solusi terbaik pada generasi-generasi berikutnya, semakin kecil nilai *convergence* maka perhitungan akan semakin lama tetapi lebih besar kemungkinan untuk mendapatkan solusi yang lebih baik.
- *Mutation Rate*
Nilai yang menyatakan berapa besar peluang suatu kromosom akan mengalami mutasi. Nilai *Mutation Rate* ini berada diantara 0 dan 1, semakin besar nilainya maka semakin besar peluang suatu kromosom mengalami mutasi sehingga peluang untuk mendapatkan solusi yang lebih baik makin besar tetapi komputasi akan semakin lama.
- *Population Size*
Nilai yang menyatakan berapa banyak kromosom pada setiap generasinya, semakin besar nilainya maka semakin banyak kromosom yang terdapat pada setiap generasinya sehingga peluang untuk mendapatkan solusi yang lebih baik makin besar tetapi komputasi akan semakin lama.
- *Random Seed*
Nilai yang menyatakan *Seed* pada proses *random*, jika nilai ini diisi dengan suatu nilai maka proses *random* akan menghasilkan bilangan *random* yang sama setiap kali *Solver* dijalankan, tetapi jika nilai ini dikosongkan maka proses *random* akan menghasilkan bilangan *random* yang berbeda setiap kali *Solver* dijalankan.
- *Maximum Time Without Improvement*
Nilai yang menyatakan lama waktu maksimum dari suatu solusi untuk tidak mengalami perbaikan, jika waktu maksimum tersebut tercapai maka *Solver* akan menghentikan komputasinya. Nilai ini dinyatakan dalam satuan detik.
- *Require Bounds on Variables*
Ceklis *Options* ini jika pada *constraint* terdapat batas bawah dan batas atas dari nilai variabel keputusan.

Pada gambar 5 dapat dilihat nilai-nilai *Options – Evolutionary* yang digunakan pada *paper* ini.



Gambar 5. Nilai-Nilai Options – Evolutionary yang Digunakan

3. HASIL DAN PEMBAHASAN

Dikarenakan *Evolutionary Algorithm* bekerja menggunakan bilangan *random*, tentunya hasil dari setiap perhitungan dapat berbeda-beda, (Corne & Lones, 2025). Dengan demikian *Solver* akan dijalankan sebanyak sepuluh kali untuk kemudian dari sepuluh solusi yang dihasilkan diambil satu solusi terbaiknya. Pada tabel 3 dapat dilihat total panjang rute dari setiap perhitungan yang dilakukan.

Tabel 3. Total Panjang Rute Dari Setiap Perhitungan

Perhitungan	Total Panjang Rute
1	861
2	845
3	872
4	830
5	824
6	864
7	862
8	854
9	871
10	846

Pada tabel 3 dapat dilihat bahwa dari 10 kali perhitungan yang dilakukan, diperoleh total panjang rute terpendek adalah 824 yang berasal dari perhitungan ke-5. Dengan demikian solusi yang dihasilkan dari perhitungan ke-5 terpilih menjadi solusi dari *instance A-n32-k5*. Pada tabel 4 dapat dilihat solusi dari *instance A-n32-k5*.

Tabel 4. Solusi dari Instance A-n32-k5

Rute	Urutan Kunjungan	Panjang Rute
1	0 – 30 – 12 – 0	60
2	0 – 29 – 18 – 11 – 4 – 28 – 23 – 3 – 2 – 6 – 26 – 0	258
3	0 – 8 – 9 – 22 – 15 – 10 – 25 – 5 – 20 – 0	244
4	0 – 13 – 17 – 19 – 31 – 21 – 1 – 0	156
5	0 – 27 – 24 – 14 – 7 – 16 – 0	106
Total Panjang Rute		824

Best Known Solution dari *instance A-n32-k5* adalah 784, dengan demikian solusi yang dihasilkan dari Excel Solver dengan *Evolutionary Algorithm* lebih panjang 40 atau sekitar 5,1%.

4. KESIMPULAN

Keunggulan penggunaan *Excel Solver* dengan *Evolutionary Algorithm* dalam menyelesaikan *Capacitated Vehicle Routing Problem* adalah tidak diperlukannya kemampuan *programming* dan tidak perlu kehabisan waktu untuk melakukan *programming*. Kegiatan *programming* dapat digantikan dengan pembuatan *template Spreadsheet* menggunakan *formula-formula* yang ada pada *Microsoft Excel*. Tentunya pembuatan *template Spreadsheet* ini lebih mudah dan lebih cepat dibandingkan dengan melakukan *programming*.

Berdasarkan solusi yang diperoleh dapat diketahui solusi yang dihasilkan oleh *Excel Solver* dengan *Evolutionary Algorithm* lebih panjang sekitar 5,1%, tetapi "kerugian" ini masih layak untuk dimaklumi dibandingkan dengan pengeluaran *effort* dan kehilangan waktu dalam melakukan *programming*. Dengan demikian penggunaan *Excel Solver* dengan *Evolutionary Algorithm* dapat menjadi alternatif yang baik untuk menyelesaikan *Capacitated Vehicle Routing Problem*, terutama jika solusi perlu untuk segera diperoleh.

Agar kehandalan *Excel Solver* dengan *Evolutionary Algorithm* dalam menyelesaikan *Capacitated Vehicle Routing Problem* dapat lebih terukur, maka perlu diujicobakan pada berbagai *instance*.

5. UCAPAN TERIMA KASIH

Paper ini didukung oleh Universitas Logistik Dan Bisnis Internasional. Melalui programnya yang bernama "Hibah Penelitian Internal" telah mendanai *paper* ini. Ucapan terima kasih diberikan kepada Universitas Logistik Dan Bisnis Internasional atas segala dukungannya. Bagi pembaca yang membutuhkan *file* dari *template Spreadsheet* ini dapat langsung menghubungi penulis melalui surat elektronik.

6. REFERENSI

- Abidin, S. Z., Jaini, N. I., & Daud, H. (2025). Decision-Making Support in Vehicle Routing Problems: A Review of Recent Literature. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, 44(2), 124–134. <https://doi.org/10.37934/araset.44.2.124134>
- Agrawal, P., Abutarboush, H. F., Ganesh, T., & Mohamed, A. W. (2021). Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019). *IEEE Access*, 9, 26766–26791. <https://doi.org/10.1109/ACCESS.2021.3056407>
- Alromema, M., & Makarem, M. A. (2024). Enhancing Evolutionary Solver Efficiency for NP Hard Single Machine Scheduling Problems. *ArXiv Preprint ArXiv:2403.19109*.
- Augerat, P., Belenguer, J. M., Benavent, E., Corberán, Á., Naddef, D., & Rinaldi, G. (1995). *Computational results with a branch and cut code for the capacitated vehicle routing problem*.
- Awad, H., Elshaer, R., AbdElmo'ez, A., & Nawara, G. (2018). An effective genetic algorithm for capacitated vehicle routing problem. *Proceedings of the International Conference on Industrial Engineering and Operations Management*, 374–384.
- Badar, A. Q. H. (2021). *Evolutionary Optimization Algorithms*. CRC Press.
- Chopard, B., & Tomassini, M. (2018). *An introduction to metaheuristics for optimization* (Vol. 226). Springer.
- Corne, D., & Lones, M. A. (2025). Evolutionary algorithms. In *Handbook of heuristics* (pp. 125–146). Springer.
- Dorronsoro, B., Yalaoui, F., Talbi, E.-G., & Danoy, G. (2022). *Metaheuristics and Nature Inspired Computing*. Springer.
- Erdoğan, G. (2017). An open source spreadsheet solver for vehicle routing problems. *Computers & Operations Research*, 84, 62–72.
- Frontline, S. (2025). *EXCEL SOLVER - CHANGE OPTIONS FOR EVOLUTIONARY SOLVING METHOD*. Frontline Systems. <https://www.solver.com/excel-solver-change-options-evolutionary-solving-method>
- Garey, M. R. (1979). D, S. Johnson, Computers and Intractability. In *A Guide to the Theory of NP-Completeness*.
- Gilbert, J., Koska, O., & Oladi, R. (2023). Excel Sheet for Demonstrating the Specific Factors Model (Solver). *Excel Models for Trade Theory*.
- Hameed, A. S., Alrikabi, H. M. B., Abdul-razaq, A. A., Nasser, H. K., Mutar, M. L., & Katea, H. H. (2025). A Detailed Review of the Capacitated Vehicle Routing Problem: Model, Computational Complexity, Solutions, and Practical Applications. *Journal of Internet Services and Information Security*, 15(1), 218–235. <https://doi.org/10.58346/JISIS.2025.11.014>
- Kalatzantonakis, P., Sifaleras, A., & Samaras, N. (2020). Cooperative versus non-cooperative parallel variable

- neighborhood search strategies: a case study on the capacitated vehicle routing problem. *Journal of Global Optimization*, 78(2), 327–348. <https://doi.org/10.1007/s10898-019-00866-y>
- Martinelli, R. (2014). *CVRPLib Capacitated Vehicle Routing Problem Library*. Galgos. <https://galgos.inf.puc-rio.br/cvrplib/en/instances>
- Pétrowski, A., & Ben-Hamida, S. (2017). *Evolutionary algorithms* (Vol. 9). John Wiley & Sons.
- Sanggala, E., & Bisma, M. (2025). Penyelesaian Capacitated Vehicle Routing Problem (CVRP) dengan Nearest Neighbour (Studi Kasus: Russian CVRP Instances). *Jurnal Teknik Industri Terintegrasi*, 8, 2586–2599. <https://doi.org/10.31004/jutin.v8i3.46463>
- Sanggala, E., Bisma, M. A., & Wirakanda, G. G. (2024). Penyelesaian Unequal Area Facility Layout Problem (UA-FLP) dengan Evolutionary Algorithm & Excel Solver (Studi Kasus: CartonPacks Instance). *REMIK: Riset Dan E-Jurnal Manajemen Informatika Komputer*, 8(1), 208–221.
- Selvi, A. A., Selvabharathi, S. M., & Lavanya, S. (2022). Real Life Optimization Problem using Excel and Solver. *International Journal of Research in Engineering, Science and Management*, 5(5), 155–157.
- Sharma, S. K., Routroy, S., & Yadav, U. (2018). Vehicle routing problem: recent literature review of its variants. *International Journal of Operational Research*, 33(1), 1–31.
- Singh, J., Singh, R., & Sharma, R. (2022). *QUADRATIC OPTIMIZATION WITH EXCEL SOLVER CODE*. 10(12), 530–535.
- Talbi, E.-G. (2009). *Metaheuristics: from design to implementation*. John Wiley & Sons.
- Toth, P., & Vigo, D. (2002). *The vehicle routing problem*. SIAM.
- Yu, X., & Gen, M. (2010). *Introduction to evolutionary algorithms*. Springer.