



PENYELESAIAN *CAPACITATED VEHICLE ROUTING PROBLEM (CVRP)* DENGAN *EVOLUTIONARY ALGORITHM & EXCEL SOLVER* (STUDI KASUS: *RUSSIA-20-NODES-CVRP INSTANCE*)

Ekra Sanggala

Program Studi D4 Logistik Bisnis, Universitas Logistik & Bisnis Internasional

E-mail: ekrasanggala@mail.ru

ABSTRAK

CVRP merupakan masalah paling sederhana dari *VRP*. *Evolutionary Algorithm (EA)* merupakan sebuah *metaheuristic* yang dapat diaplikasikan pada berbagai permasalahan optimasi, termasuk *CVRP*. *Solver* merupakan *Excel Add-In* untuk menyelesaikan permasalahan optimasi. *Solver* menggunakan tiga algoritma, yaitu *LP Simplex*, *GRG Nonlinear* dan *EA*. Dengan adanya kemampuan *EA* untuk menyelesaikan *CVRP* dan *Solver* yang mampu menjalankan *EA*, maka dapat disimpulkan bahwa penyelesaian *CVRP* dapat dilakukan dengan memanfaatkan *Solver*. *Russia-20-Nodes-CVRP Instance* merupakan salah satu *CVRP Instance* yang terdapat pada *Russian CVRP Instances*. Dengan menggunakan *EA & Solver*, panjang rute terpendek dari *Russia-20-Nodes-CVRP Instance* adalah 15.884 Km.

Kata kunci: *CVRP*, *Evolutionary Algorithm*, *Excel Solver*, *CVRP Instance*

ABSTRACT

CVRP is the simplest problem of *VRP*. *Evolutionary Algorithm (EA)* is a *metaheuristic* that can be applied to various optimization problems, include *CVRP*. *Solver* is *Excel Add-In* for solving optimization problems. *Solver* uses three algorithms. These algorithms are *LP Simplex*, *GRG Nonlinear* and *EA*. With the *EA*'s ability to solve *CVRP* and *Solver* being able to run *EA*, it can be concluded that solving *CVRP* can be done by utilizing *Solver*. *Russia-20-Nodes-CVRP Instance* is one of the *CVRP Instances* found in *Russian CVRP Instances*. Using *EA & Solver*, the shortest length of the *Russia-20-Nodes-CVRP Instance* is 15.884 Km

Keyword: *CVRP*, *Evolutionary Algorithm*, *Excel Solver*, *CVRP Instance*

1. PENDAHULUAN

Vehicle Routing Problem (VRP) merupakan salah satu masalah pada logistik yang sangat menarik perhatian para peneliti untuk mengembangkan metode-metode solusinya (Tan & Yeh, 2021). 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)* (Golden et al., 2023).

CVRP merupakan masalah paling sederhana dari *VRP*. Dalam *CVRP*, jumlah pelanggan dan kebutuhannya

telah diketahui dan pemenuhan 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 (Elshaer & Awad, 2020).

Evolutionary Algorithm (EA) merupakan sebuah *metaheuristic* yang dapat diaplikasikan pada berbagai permasalahan optimasi, termasuk *CVRP*. *Evolutionary Algorithm* merupakan sebuah *metaheuristic* yang

terinspirasi dari teori evolusi, dimana individu yang terkuat yang akan bertahan hidup (Badar, 2021).

Solver merupakan *Excel Add-In* untuk menyelesaikan permasalahan optimasi, yang bertugas untuk menemukan nilai-nilai untuk sekumpulan variabel keputusan dengan tujuan memaksimalkan atau meminimumkan fungsi tujuan (Selvi et al., 2022).

Dengan adanya kemampuan *Evolutionary Algorithm* untuk menyelesaikan *CVRP* dan *Solver* yang mampu menjalankan *Evolutionary Algorithm*, maka dapat disimpulkan bahwa penyelesaian *CVRP* dapat dilakukan dengan memanfaatkan *Solver*.

Kemampuan *Evolutionary Algorithm* dan *Solver* dalam menyelesaikan *CVRP* perlu diuji, agar dapat dibuktikan kemampuannya dalam menyelesaikan *CVRP*. Untuk membuktikan kemampuan tersebut maka diperlukan sebuah *CVRP Instance* yang akan diselesaikan oleh *Evolutionary Algorithm* dan *Solver*.

Russia-20-Nodes-CVRP Instance merupakan salah satu *CVRP Instance* yang terdapat pada *Russian CVRP Instances*. *CVRP Instance* ini dibuat berdasarkan 20 kota berpenduduk terbanyak di *Russia* (Sanggala & Bisma, 2023). *CVRP Instance* ini menjadi menarik untuk diselesaikan karena jumlah titiknya tidak terlalu banyak, hanya 20 titik saja. Dengan jumlah titik sebanyak 20 titik, menjadikan *CVRP Instance* tersebut sebagai permasalahan yang menarik karena waktu perhitungannya yang relatif cepat dan mempunyai kemungkinan solusi yang sangat banyak. Dengan demikian pada *paper* ini akan dibahas mengenai *Evolutionary Algorithm* dan *Solver* untuk menyelesaikan *Russia-20-Nodes-CVRP Instance*.

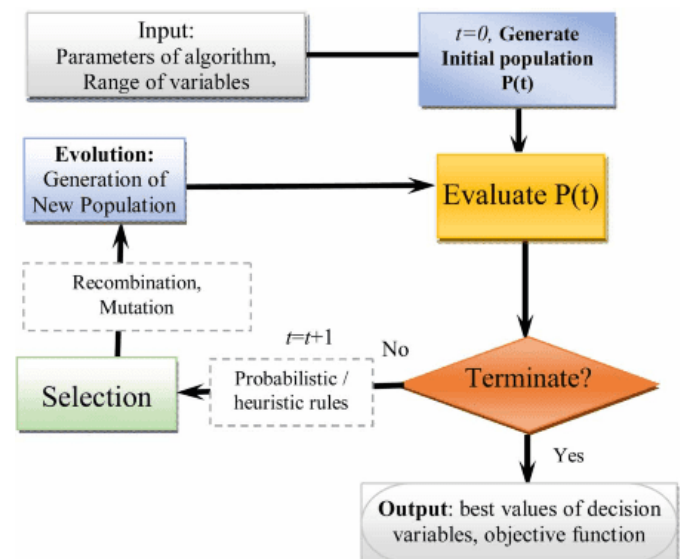
2. TINJAUAN PUSTAKA

Vehicle Routing Problem (VRP) merupakan salah satu masalah pada logistik yang sangat menarik perhatian para peneliti untuk mengembangkan metode-metode solusinya (Tan & Yeh, 2021). 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)* (Golden et al., 2023).

2.1. Evolutionary Algorithm

Evolutionary Algorithm merupakan sebuah *metaheuristic* yang dapat diaplikasikan pada berbagai permasalahan optimasi, termasuk *CVRP*. *Evolutionary Algorithm* merupakan sebuah *metaheuristic* yang

terinspirasi dari teori evolusi, dimana individu yang terkuat yang akan bertahan hidup. *Evolutionary Algorithm* terdiri dari sekumpulan individu yang disebut populasi, setiap individu mewakili sebuah *feasible solution*. Individu-individu pada generasi awal dibentuk dengan cara *random* dan akan mengalami proses *selection*, *recombination* dan *mutation* sehingga menghasilkan individu-individu baru yang akan hidup pada generasi selanjutnya, proses ini akan terus berlangsung hingga sejumlah generasi. Proses-proses tersebut dilakukan dengan cara mengevaluasi nilai *fitness* (kekuatan) setiap individu. Individu-individu dengan nilai *fitness* lebih baik akan terpilih untuk dikombinasikan hingga diharapkan dapat menghasilkan individu-individu baru yang mempunyai nilai *fitness* yang lebih baik. Setelah melalui sejumlah generasi, nilai *fitness* akan konvergen pada sebuah nilai, dan individu dengan nilai *fitness* terbaik akan menjadi solusi (Badar, 2021).



Gambar 1: Struktur Dasar Sebuah Evolutionary Algorithm (Janga Reddy & Nagesh Kumar, 2020)

Berikut ini merupakan langkah-langkah utama pada *Evolutionary Algorithm* (Janga Reddy & Nagesh Kumar, 2020):

1. Secara *random* bangkitkan individu-individu sebagai populasi awal.
2. Evaluasi nilai *fitness* setiap individu.
3. Ulangi langkah-langkah evolusi hingga kriteria selesai terpenuhi:



- (a) Pilih individu-individu untuk melakukan reproduksi.
 - (b) Lakukan *genetic operations* untuk menghasilkan keturunan.
 - (c) Evaluasi nilai *fitness* setiap keturunan.
 - (d) Individu-individu dengan nilai *fitness* terlemah diganti oleh individu-individu baru yang mempunyai nilai *fitness* terkuat.
4. Diperoleh solusi berdasarkan individu dengan nilai *fitness* terkuat.

2.2. Solver

Solver merupakan *Excel Add-In* untuk menyelesaikan permasalahan optimasi, yang bertugas untuk menemukan nilai-nilai untuk sekumpulan variabel keputusan dengan tujuan memaksimalkan dan atau meminimumkan fungsi tujuan. *Solver* menggunakan beberapa algoritma untuk menemukan nilai optimal yang akan digunakan pada variabel-variabel keputusan. Tiga algoritma tersebut adalah sebagai berikut ini (Selvi et al., 2022):

- (a) *Linear Programming Simplex*
- (b) *GRG Nonlinear*
- (c) *Evolutionary Algorithm*

2.3. Russia-20-Nodes-CVRP Instance

Berikut ini merupakan 20 titik yang terdapat pada *Russia-20-Nodes-CVRP Instance*:

Tabel 1: Titik-Titik Yang Terdapat Pada Russia-20-Nodes-CVRP Instance

Node	Name	Latitude	Longitude	Demand (Unit)
0	Moscow	55,75222	37,61556	0
1	Saint Petersburg	59,93863	30,31413	26
2	Novosibirsk	55,04150	82,93460	32
3	Yekaterinburg	56,85190	60,61220	14
4	Nizhniy Novgorod	56,32867	44,00205	35
5	Samara	53,20007	50,15000	10
6	Omsk	54,99244	73,36859	20
7	Kazan	55,78874	49,12214	14
8	Rostov-na-Donu	47,23135	39,72328	15
9	Chelyabinsk	55,15402	61,42915	33
10	Ufa	54,74306	55,96779	19
11	Volgograd	48,71939	44,50183	23
12	Perm	58,01046	56,25017	18
13	Krasnoyarsk	56,01839	92,86717	26
14	Saratov	51,54056	46,00861	32
15	Voronezh	51,67204	39,18430	17
16	Tol'yatti	53,53030	49,34610	14
17	Krasnodar	45,04484	38,97603	11
18	Ulyanovsk	54,32824	48,38657	22
19	Izhevsk	56,84976	53,20448	19

2.4. Haversine Formula

Untuk menghitung jarak antara setiap titik digunakan *Haversine Formula* yang bentuk persamaannya seperti berikut ini (Idrizi, 2020):

$$a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos\phi_1 \cdot \cos\phi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)$$

$$c = 2 \cdot \text{atan2} \left[\sqrt{a}, \sqrt{1-a} \right]$$

$$d = R \cdot c$$

- Φ_1 : *Latitude* tempat pertama
- Φ_2 : *Latitude* tempat kedua
- λ_1 : *Longitude* tempat pertama
- λ_2 : *Longitude* tempat kedua
- $\Delta\Phi$: $\Phi_1 - \Phi_2$
- $\Delta\lambda$: $\lambda_1 - \lambda_2$
- R : Radius Bumi (6371 Km)



3. METODE PENELITIAN

Langkah-langkah dalam penelitian ini sebagai berikut ini:

1. Menentukan sebuah *CVRP Instance* yang akan digunakan untuk menguji *Evolutionary Algorithm* dan *Solver* dalam menyelesaikan *CVRP*.
2. Menghitung jarak antara setiap titik yang terdapat pada *CVRP Instance*.
3. Membuat *Spreadsheet* untuk menentukan banyaknya rute dan titik-titik yang termasuk kedalamnya serta untuk menghitung panjang seluruh rute.
4. Pada *Solver* dimasukkan *objective function* dan *constaints* dari *CVRP Instance* yang akan diselesaikan.
5. Menjalankan *Solver* sebanyak 10 kali.

3.1. *CVRP Instance* Yang Akan Digunakan

Russia-20-Nodes-CVRP Instance akan digunakan untuk menguji *Evolutionary Algorithm* dan *Solver* dalam menyelesaikan *CVRP*.

3.2. Menghitung Jarak Antara Setiap Titik

Berikut ini merupakan contoh penggunaan *Haversine Formula* dalam menghitung jarak antara dua titik di Bumi:

Moscow (Lat: 55,75222; Long: 37,61556)

St.Petersburg (Lat: 59,93863; Long: 30,31413)

$$a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos\phi_1 \cdot \cos\phi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)$$

$$a = 0,002477$$

$$c = 2 \cdot \text{atan2}\left[\sqrt{0,002477}, \sqrt{0,998761}\right]$$

$$c = 0,099581$$

$$d = 6371 \cdot 0,099581$$

$$d = 634,4309 \approx 635 \text{ km}$$

3.3. Membuat *Spreadsheet*

Spreadsheet pada *Excel* akan digunakan untuk menuliskan solusi dan menentukan banyaknya rute dan titik-titik yang termasuk kedalamnya serta untuk menghitung panjang seluruh rute.

3.4. Memasukkan *Objective Function* dan *Constraints* pada *Solver*

Agar *Solver* dapat menyelesaikan sebuah *CVRP* perlu dimasukkan *Objective Function* dan *Constraints*-nya.

3.5. Menjalankan *Solver* Sebanyak 10 Kali

Karena *Evolutionary Algorithm* menggunakan nilai *random* dalam melakukan evolusi, maka solusi yang dihasilkan dapat berbeda untuk setiap kali perhitungan, dengan demikian akan dilakukan 10 kali perhitungan untuk kemudian diambil satu solusi yang menghasilkan panjang seluruh rute terpendek.

4. HASIL DAN PEMBAHASAN

4.1. Jarak Antara Setiap Titik Yang Terdapat Pada *Russia-20-Nodes-CVRP Instance*

Berikut ini merupakan jarak antara setiap titik yang terdapat pada *Russia-20-Nodes-CVRP Instance*:

Tabel 2: Jarak Antara Setiap Titik Pada *Russia-20-Nodes-CVRP Instance* (Km)

Node	Name	0	1	2	3	4
0	Moscow	0	635	2812	1418	402
1	Saint Petersburg	635	0	3105	1783	896
2	Novosibirsk	2812	3105	0	1398	2412
3	Yekaterinburg	1418	1783	1398	0	1017
4	Nizhniy Novgorod	402	896	2412	1017	0
5	Samara	857	1420	2127	780	526
6	Omsk	2236	2584	610	820	1834
7	Kazan	719	1200	2115	718	324
8	Rostov-na-Donu	959	1540	3083	1773	1054
9	Chelyabinsk	1496	1912	1363	196	1096
10	Ufa	1166	1632	1715	374	773
11	Volgograd	911	1544	2693	1406	847
12	Perm	1156	1492	1658	292	761
13	Krasnoyarsk	3353	3575	634	1967	2961
14	Saratov	724	1350	2456	1116	549
15	Voronezh	466	1072	2880	1499	606
16	Tol'yatti	794	1355	2166	804	462
17	Krasnodar	1195	1754	3267	1990	1303
18	Ulyanovsk	704	1251	2200	817	356
19	Izhevsk	968	1370	1848	451	566

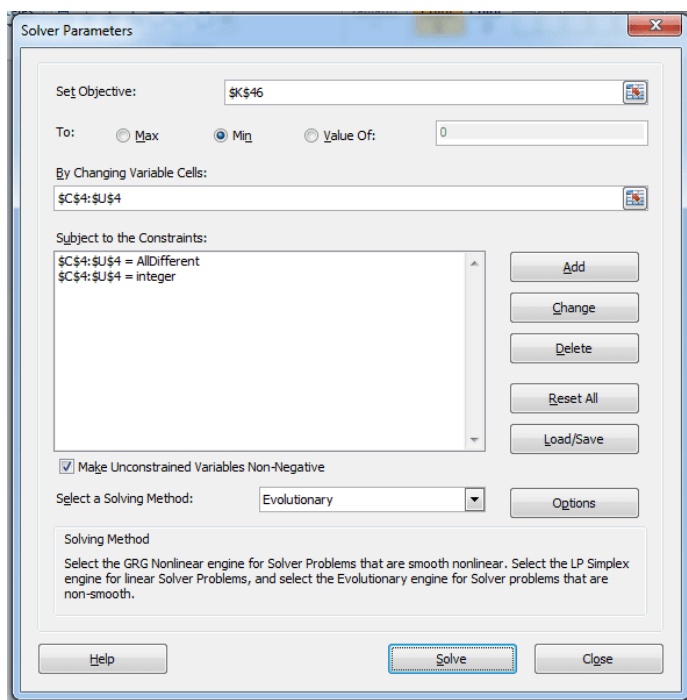


Tabel 2: Jarak Antara Setiap Titik Pada Russia-20-Nodes-CVRP Instance (Km) (Lanjutan)

Node	Name	5	6	7	8	9
0	Moscow	857	2236	719	959	1496
1	Saint Petersburg	1420	2584	1200	1540	1912
2	Novosibirsk	2127	610	2115	3083	1363
3	Yekaterinburg	780	820	718	1773	196
4	Nizhniy Novgorod	526	1834	324	1054	1096
5	Samara	0	1520	296	994	765
6	Omsk	1520	0	1527	2474	760
7	Kazan	296	1527	0	1151	778
8	Rostov-na-Donu	994	2474	1151	0	1740
9	Chelyabinsk	765	760	778	1740	0
10	Ufa	418	1111	449	1406	352
11	Volgograd	636	2084	847	393	1359
12	Perm	658	1100	499	1629	449
13	Krasnoyarsk	2725	1230	2681	3697	1961
14	Saratov	337	1847	515	661	1097
15	Voronezh	762	2277	797	496	1518
16	Tol'yatti	65	1561	252	977	803
17	Krasnodar	1216	2660	1392	250	1943
18	Ulyanovsk	171	1600	170	996	841
19	Izhevsk	451	1269	278	1408	545

Tabel 2: Jarak Antara Setiap Titik Pada Russia-20-Nodes-CVRP Instance (Km) (Lanjutan)

Node	Name	10	11	12	13	14
0	Moscow	1166	911	1156	3353	724
1	Saint Petersburg	1632	1544	1492	3575	1350
2	Novosibirsk	1715	2693	1658	634	2456
3	Yekaterinburg	374	1406	292	1967	1116
4	Nizhniy Novgorod	773	847	761	2961	549
5	Samara	418	636	658	2725	337
6	Omsk	1111	2084	1100	1230	1847
7	Kazan	449	847	499	2681	515
8	Rostov-na-Donu	1406	393	1629	3697	661
9	Chelyabinsk	352	1359	449	1961	1097
10	Ufa	0	1034	364	2308	753
11	Volgograd	1034	0	1291	3309	332
12	Perm	364	1291	0	2201	973
13	Krasnoyarsk	2308	3309	2201	0	3058
14	Saratov	753	332	973	3058	0
15	Voronezh	1166	501	1295	3465	472
16	Tol'yatti	452	633	659	2759	317
17	Krasnodar	1619	586	1862	3888	890
18	Ulyanovsk	492	679	636	2782	349
19	Izhevsk	291	1076	224	2406	753



Gambar 2: Solver Setelah Dimasukkan Objective Function Dan Constraints

4.4. Melakukan Perhitungan Dengan Solver Sebanyak 10 Kali

Berikut ini merupakan solusi yang dihasilkan dari 10 kali perhitungan:

Tabel 4: Panjang Rute Yang Dihasilkan Dari 10 Kali Perhitungan

Perhitungan	Panjang Rute (Km)
1	15.994
2	16.010
3	16.487
4	15.884
5	15.900
6	16.044
7	16.440
8	15.900
9	15.902
10	15.994

Dari tabel 4 dapat dilihat bahwa panjang rute terpendek yang diperoleh adalah panjang rute yang dihasilkan dari perhitungan ke-4, yaitu sepanjang 15.884 Km.

Tabel 5: Rute Dari Panjang Rute Terpendek

Rute	Node	Name	Demand (Unit)	Demand Kumulatif (Unit)	Sisa Kapasitas (Unit)	Jarak (Km)	Panjang Rute (Km)
1	0	Moscow	0	0	100	0	-
1	12	Perm	18	18	82	1.156	-
1	13	Krasnoyarsk	26	44	56	2.201	-
1	2	Novosibirsk	32	76	24	634	-
1	6	Omsk	20	96	4	610	-
1	0	Moscow	0	96	4	2.236	6.837
2	0	Moscow	0	0	100	0	-
2	14	Saratov	32	32	68	724	-
2	11	Volgograd	23	55	45	332	-
2	17	Krasnodar	11	66	34	586	-
2	8	Rostov-na-Donu	15	81	19	250	-
2	15	Voronezh	17	98	2	496	-
2	0	Moscow	0	98	2	466	2.854
3	0	Moscow	0	0	100	0	-
3	19	Izhevsk	19	19	81	968	-
3	3	Yekaterinburg	14	33	67	451	-
3	9	Chelyabinsk	33	66	34	196	-
3	10	Ufa	19	85	15	352	-
3	7	Kazan	14	99	1	449	-
3	0	Moscow	0	99	1	719	3.135
4	0	Moscow	0	0	100	0	-
4	4	Nizhny Novgorod	35	35	65	402	-
4	18	Ulyanovsk	22	57	43	356	-
4	5	Samara	10	67	33	171	-
4	16	Tol'yatti	14	81	19	65	-
4	0	Moscow	0	81	19	794	1.788
5	0	Moscow	0	0	100	0	-
5	1	Saint Petersburg	26	26	74	635	-
5	0	Moscow	0	26	74	635	1.270
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
Total							15.884

5. KESIMPULAN DAN SARAN

Kombinasi *Evolutionary Algorithm* dan *Solver* dapat digunakan untuk menyelesaikan *CVRP*. Dikarenakan *Evolutionary Algorithm* bekerja dengan menggunakan nilai *random*, solusi yang dihasilkan setiap perhitungan dapat berbeda-beda, oleh karena itu perlu dilakukan beberapa kali perhitungan agar dapat diperoleh solusi terbaiknya. Semakin banyak perhitungan yang dilakukan, akan memperbesar peluang didapatkannya solusi terbaik, walaupun mengakibatkan waktu penentuan solusi terbaiknya menjadi lebih lama.

Untuk *Russia-20-Nodes-CVRP Instance* berdasarkan hasil penyelesaian oleh *Evolutionary Algorithm* dan *Solver*, panjang rute terbaiknya adalah 15.884 Km dengan jumlah rute sebanyak 5 rute. Kelima rute tersebut adalah sebagai berikut ini:



Rute 1: *Moscow – Perm – Krasnoyarsk – Novosibirsk – Omsk – Moscow* (6.837 Km)

Rute 2: *Moscow – Saratov – Volgograd – Krasnodar – Rostov-Na-Donu – Voronezh – Moscow* (2.854 Km)

Rute 3: *Moscow – Izhevsk – Yekaterinburg – Chelyabinsk – Ufa – Kazan – Moscow* (3.135 Km)

Rute 4: *Moscow – Nizhniy Novgorod – Ulyanovsk – Samara – Tol'yatti – Moscow* (1.788 Km)

Rute 5: *Moscow – Saint Petersburg – Moscow* (1.270 Km)

Pengujian pada satu *CVRP Instance* belum mampu memberi gambaran mengenai kehandalan *Evolutionary Algorithm* dan *Solver* dalam menyelesaikan *CVRP*. Dengan demikian perlu dilakukan pengujian terhadap *Evolutionary Algorithm* dan *Solver* dalam menyelesaikan berbagai *CVRP Instance* lainnya, sehingga kehandalannya dapat lebih diketahui.

Ant Number Effects on Ant Colony Optimization for Solving Russia-20-Nodes-SDVRP Instance. *Sainteks: Jurnal Sain Dan Teknik*, 5(2), 163–174.

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.

Tan, S.-Y., & Yeh, W.-C. (2021). The vehicle routing problem: State-of-the-art classification and review. *Applied Sciences*, 11(21), 10295.

DAFTAR PUSTAKA

- Badar, A. Q. H. (2021). *Evolutionary Optimization Algorithms*. CRC Press.
- Elshaer, R., & Awad, H. (2020). A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. *Computers & Industrial Engineering*, 140, 106242.
- Golden, B., Wang, X., & Wasil, E. (2023). The Evolution of the Vehicle Routing Problem—A Survey of VRP Research and Practice from 2005 to 2022. In *The Evolution of the Vehicle Routing Problem: A Survey of VRP Research and Practice from 2005 to 2022* (pp. 1–64). Springer.
- Idrizi, B. (2020). Necessity for geometric corrections of distances in web and mobile maps. *International Conference on Cartography and GIS, Bulgaria*.
- Janga Reddy, M., & Nagesh Kumar, D. (2020). Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering: a state-of-the-art review. *H2oj*, 3(1), 135–188.
- Sanggala, E., & Bisma, M. A. (2023). Analysis of The