



PENYELESAIAN TRAVELLING SALESMAN PROBLEM (TSP) DENGAN EVOLUTIONARY ALGORITHM & EXCEL SOLVER (STUDI KASUS: AK-47-TSP INSTANCE)

Ekra Sanggala*

Program Studi D4 Logistik Bisnis, Universitas Logistik & Bisnis Internasional
Jalan Sari Asih No 54, Bandung, 40151, Indonesia

*Corresponding author : ekrasanggala@mail.ru

ABSTRAK

Travelling Salesman Problem (TSP) merupakan permasalahan penentuan rute terpendek yang diawali dari titik *start* untuk mengunjungi sekumpulan titik tepat sekali dan diakhiri dengan kembali ke titik *start*. *Evolutionary Algorithm (EA)* merupakan sebuah *metaheuristic* yang dapat diaplikasikan pada berbagai permasalahan optimasi, termasuk *TSP*. *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 *TSP* dan *Solver* yang mampu menjalankan *EA*, maka dapat disimpulkan bahwa penyelesaian *TSP* dapat dilakukan dengan memanfaatkan *Solver*. Untuk membuktikan kemampuan tersebut maka diperlukan sebuah *TSP Instance* yang akan diselesaikan oleh *EA* dan *Solver*. *AK-47-TSP Instance* merupakan salah satu *TSP Instance* yang terdapat pada *Russian TSP Instances*. Dengan menggunakan *EA* & *Solver*, panjang rute terpendek dari *AK-47-TSP Instance* adalah 20.998 Km.

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

ABSTRACT

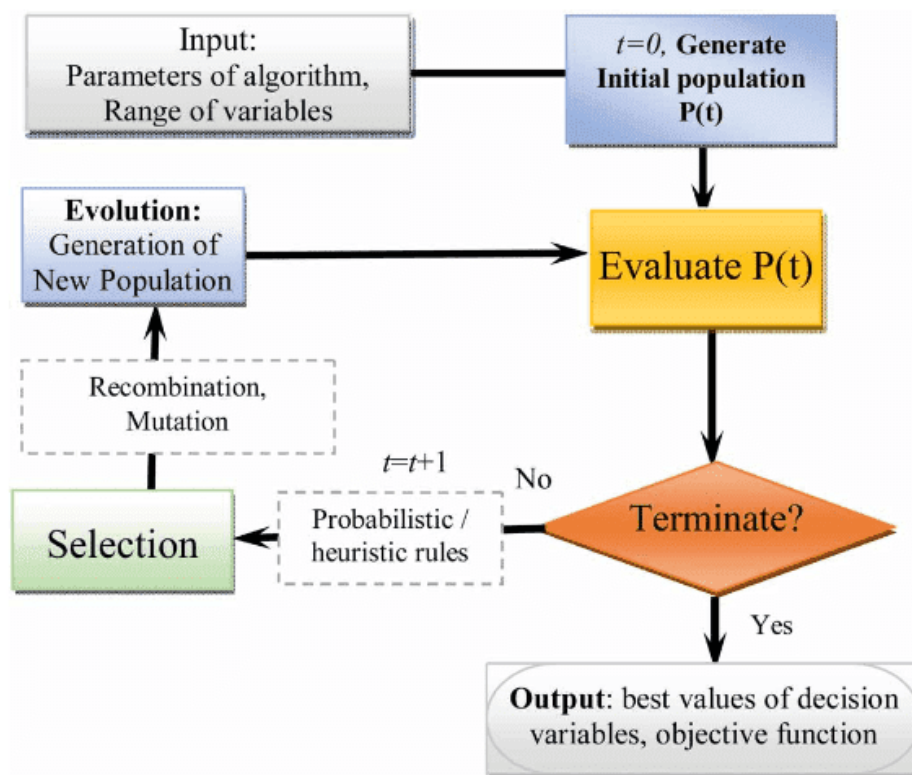
Travelling Salesman Problem (TSP) is the problem for finding the shortest route starting from start node then visiting number of nodes exactly once and finally go back to start node. *Evolutionary Algorithm (EA)* is a *metaheuristic* that can be applied to various optimization problems, include *TSP*. *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 *TSP* and *Solver* being able to run *EA*, it can be concluded that solving *TSP* can be done by utilizing *Solver*. To prove this capability, a *TSP Instance* is needed which will be solved by *EA* and *Solver*. *AK-47-TSP Instance* is one of the *TSP Instances* found in *Russian TSP Instances*. Using *EA* & *Solver*, the shortest route length of the *AK-47-TSP Instance* is 20,998 Km.

Keywords: *TSP, Evolutionary Algorithm, Excel Solver, TSP Instance*

PENDAHULUAN

Travelling Salesman Problem (TSP) merupakan permasalahan penentuan rute terpendek yang diawali dari titik *start* untuk mengunjungi sekumpulan titik tepat sekali dan diakhiri dengan kembali ke titik *start* (Bisma & Sanggala, 2023). Pada dunia nyata banyak permasalahan yang dapat didefinisikan sebagai *TSP*, diantaranya adalah rute perjalanan turis, rute bus sekolah, rute pengiriman barang dan lain-lain (Applegate et al., 2011). Jika sebuah *TSP* mempunyai banyak titik, maka akan menjadi sebuah *NP-Hard Problem* dimana untuk memperoleh solusi terbaiknya diperlukan waktu perhitungan yang tidak wajar untuk ditunggu (Wu, 2020). Algoritma yang bekerja berdasarkan *heuristic* dan *metaheuristic* dapat menjadi sebuah solusi untuk menyelesaikan *NP-Hard Problem* dengan waktu perhitungan yang wajar ditunggu, walaupun solusi yang dihasilkan belum tentu merupakan solusi yang terbaik (Elshaer & Awad, 2020).

Evolutionary Algorithm (EA) merupakan sebuah *metaheuristic* yang dapat diaplikasikan pada berbagai permasalahan optimasi, termasuk *TSP*. *Evolutionary Algorithm* merupakan sebuah *metaheuristic* yang terinspirasi dari teori evolusi, dimana individu yang terkuat yang akan bertahan hidup (Vikhar, 2017). *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.

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. *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*

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

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

AK-47-TSP Instance merupakan salah satu *TSP Instance* yang terdapat pada *Russian TSP Instances*. *Instance* ini dibuat untuk memperingati senjata *AK-47* yang ditemukan oleh *Mikhail Kalashnikov*. *Instance* ini terdiri dari 47 tempat di *Russia*, dimana titik-start-nya berada di *Kur'ya* yang merupakan tempat kelahiran dari *Mikhail Kalashnikov*. Sedangkan 46 tempat lainnya merupakan 46 kota di *Russia* yang mempunyai jumlah penduduk terbanyak. Koordinat yang digunakan pada *AK-47-TSP Instance* ini adalah *Geographic Coordinate System*, sehingga setiap tempat mempunyai nilai *Latitude* dan *Longitude* (Sanggala & Bisma, 2023b). *Instance* ini menjadi menarik untuk menguji sebuah algoritma dalam menyelesaikan *TSP* karena jumlah titiknya yang cukup banyak, sehingga dapat mengukur kualitas sebuah algoritma dalam menyelesaikan *TSP*.

AK-47-TSP Instance pernah diselesaikan dengan menggunakan *Random Nearest Neighbour* (Sanggala & Bisma, 2023c). Diharapkan *Evolutionary Algorithm* & *Solver* dapat menghasilkan solusi yang lebih baik dibandingkan dengan *Random Nearest Neighbour*. Dengan demikian pada *paper* ini akan dibahas mengenai penyelesaian *AK-47-TSP Instance* dengan *Evolutionary Algorithm* dan *Excel Solver*.

METODE PENELITIAN

Langkah-langkah dalam penelitian ini adalah sebagai berikut ini:

1. Menentukan sebuah *TSP Instance* yang akan digunakan untuk menguji *Evolutionary Algorithm* dan *Solver* dalam menyelesaikan *TSP*.
2. Menghitung jarak antara setiap tempat yang terdapat pada *TSP Instance*.

3. Membuat *Spreadsheet* untuk menghitung panjang rute dari sebuah solusi.
4. Pada *Solver* dimasukkan *objective function* dan *constraints* dari *TSP Instance* yang akan diselesaikan.
5. Menjalankan *Solver* sebanyak 10 kali.
6. Membandingkan solusi yang dihasilkan dengan solusi yang diperoleh dengan *Random Nearest Neighbour*.

A. *TSP Instance* Yang Akan Digunakan

AK-47-TSP Instance akan digunakan untuk menguji *Evolutionary Algorithm* dan *Solver* dalam menyelesaikan *TSP*.

B. Menghitung Jarak Antara Setiap Tempat Yang Terdapat Pada *TSP Instance*

Untuk menghitung jarak antara setiap tempat ini digunakan *Haversine* formula yang bentuk persamaannya seperti berikut ini (Bashkim 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)

Berikut ini merupakan contoh penggunaan *Haversine Formula* dalam menghitung jarak antara dua tempat di Bumi (Sanggala & Bisma, 2023a):

Kur'ya (*Latitude*: 51,60000; *Longitude*: 82,31667)

Moscow (*Latitude*: 55,75222; *Longitude*: 37,61556)

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

$$a = 0,051862$$

$$c = 2 \cdot \text{atan2}\left[\sqrt{0,051862}, \sqrt{0,948138}\right]$$

$$c = 0,459496$$

$$d = 6371 \cdot 0,459496$$

$$d = 2927,451 \approx 2928 \text{ km}$$

C. Membuat *Spreadsheet* Untuk Menghitung Panjang Rute Dari Sebuah Solusi

Spreadsheet pada *Excel* akan digunakan untuk menuliskan solusi dan menghitung panjang rute berdasarkan solusi yang dituliskan.

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

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

E. 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 rute terpendek.

F. Membandingkan Solusi

Solusi yang dihasilkan dari *Evolutionary Algorithm* dan *Solver* dibandingkan dengan solusi yang dihasilkan dari *Random Nearest Neighbour*. Membandingkan solusi ini diperlukan untuk mengetahui kehandalan *Evolutionary Algorithm* dan *Solver* dalam menyelesaikan *TSP* dibandingkan dengan algoritma lainnya.

HASIL DAN PEMBAHASAN

A. *AK-47-TSP Instance*

Berikut ini merupakan daftar 47 tempat yang terdapat pada *AK-47-TSP Instance*:

Tabel 1: *AK-47-TSP Instance*

Node No	Node Symbol	Name	Latitude	Longitude
1	0	Kur'ya	51,60000	82,31667
2	1	Moscow	55,75222	37,61556
3	2	Saint Petersburg	59,93863	30,31413
4	3	Novosibirsk	55,04150	82,93460
5	4	Yekaterinburg	56,85190	60,61220
6	5	Nizhniy Novgorod	56,32867	44,00205
7	6	Samara	53,20007	50,15000
8	7	Omsk	54,99244	73,36859
9	8	Kazan	55,78874	49,12214
10	9	Rostov-na-Donu	47,23135	39,72328
11	10	Chelyabinsk	55,15402	61,42915
12	11	Ufa	54,74306	55,96779
13	12	Volgograd	48,71939	44,50183
14	13	Perm	58,01046	56,25017
15	14	Krasnoyarsk	56,01839	92,86717
16	15	Saratov	51,54056	46,00861
17	16	Voronezh	51,67204	39,18430
18	17	Tol'yatti	53,53030	49,34610
19	18	Krasnodar	45,04484	38,97603
20	19	Ulyanovsk	54,32824	48,38657
21	20	Izhevsk	56,84976	53,20448
22	21	Yaroslavl	57,62987	39,87368
23	22	Barnaul	53,36056	83,76361
24	23	Vladivostok	43,10562	131,87353
25	24	Irkutsk	52,29778	104,29639
26	25	Khabarovsk	48,48271	135,08379
27	26	Khabarovsk Vtoroy	48,43776	135,13329
28	27	Orenburg	51,77270	55,09880
29	28	Novokuznetsk	53,75570	87,10990
30	29	Ryazan'	54,62690	39,69160
31	30	Tyumen	57,15222	65,52722
32	31	Lipetsk	52,60311	39,57076
33	32	Penza	53,20066	45,00464
34	33	Naberezhnyye Chelny	55,72545	52,41122
35	34	Kalininskiy	59,99675	30,38990
36	35	Astrakhan	46,34968	48,04076
37	36	Makhachkala	42,97638	47,50236
38	37	Tomsk	56,49771	84,97437
39	38	Kemerovo	55,33333	86,08333
40	39	Tula	54,19609	37,61822
41	40	Kirov	58,59665	49,66007
42	41	Cheboksary	56,13222	47,25194
43	42	Kaliningrad	54,70649	20,51095
44	43	Bryansk	53,25209	34,37167
45	44	Ivanovo	56,99719	40,97139
46	45	Magnitogorsk	53,41861	59,04722
47	46	Kursk	51,73733	36,18735

B. Jarak Antara Setiap Tempat Yang Terdapat Pada *AK-47-TSP Instance*

Berikut ini merupakan jarak antara setiap tempat yang terdapat pada *AK-47-TSP Instance*:

Tabel 2: Jarak Antara Setiap Tempat Pada *AK-47-TSP Instance*

Node_Symbol	Name	0	1	2	3	4	5	6	7	8	9
0	Kur'ya	0	2928	3288	385	1519	2527	2172	704	2212	3075
1	Moscow	2928	0	635	2812	1418	402	857	2236	719	959
2	Saint Petersburg	3288	635	0	3105	1783	896	1420	2584	1200	1540
3	Novosibirsk	385	2812	3105	0	1398	2412	2127	610	2115	3083
4	Yekaterinburg	1519	1418	1783	1398	0	1017	780	820	718	1773
5	Nizhniy Novgorod	2527	402	896	2412	1017	0	526	1834	324	1054
6	Samara	2172	857	1420	2127	780	526	0	1520	296	994
7	Omsk	704	2236	2584	610	820	1834	1520	0	1527	2474
8	Kazan	2212	719	1200	2115	718	324	296	1527	0	1151
9	Rostov-na-Donu	3075	959	1540	3083	1773	1054	994	2474	1151	0
10	Chelyabinsk	1435	1496	1912	1363	196	1096	765	760	778	1740
11	Ufa	1780	1166	1632	1715	374	773	418	1111	449	1406
12	Volgograd	2683	911	1544	2693	1406	847	636	2084	847	393
13	Perm	1802	1156	1492	1658	292	761	658	1100	499	1629
14	Krasnoyarsk	848	3353	3575	634	1967	2961	2725	1230	2681	3697
15	Saratov	2484	724	1350	2456	1116	549	337	1847	515	661
16	Voronezh	2933	466	1072	2880	1499	606	762	2277	797	496
17	Tol'yatti	2219	794	1355	2166	804	462	65	1561	252	977
18	Krasnodar	3236	1195	1754	3267	1990	1303	1216	2660	1392	250
19	Ulyanovsk	2271	704	1251	2200	817	356	171	1600	170	996
20	Izhevsk	1964	968	1370	1848	451	566	451	1269	278	1408
21	Yaroslavl	2765	251	608	2625	1246	289	814	2065	600	1157
22	Barnaul	219	2933	3253	195	1515	2531	2215	700	2226	3149
23	Vladivostok	3773	6416	6538	3715	5065	6037	5827	4323	5773	6796
24	Irkutsk	1503	4203	4416	1434	2811	3810	3558	2044	3527	4516
25	Khabarovsk	3702	6141	6195	3575	4851	5778	5628	4169	5538	6616
26	Khabarovsk Vtoroy	3708	6147	6201	3581	4857	5784	5634	4175	5544	6622
27	Orenburg	1866	1228	1779	1868	669	883	371	1260	595	1217
28	Novokuznetsk	403	3116	3410	306	1701	2716	2419	900	2417	3364
29	Ryazan'	2821	182	815	2726	1328	331	703	2138	612	823
30	Tyumen	1246	1711	2042	1102	300	1311	1068	543	1017	2058
31	Lipetsk	2879	373	994	2814	1425	504	713	2214	715	598
32	Penza	2509	555	1167	2450	1072	354	343	1848	392	763
33	Naberezhnyye Chelny	2007	925	1382	1914	522	527	317	1323	206	1287
34	Kalininskiy	3283	636	8	3100	1778	895	1418	2579	1197	1544
35	Astrakhan	2544	1272	1906	2608	1451	1145	777	2012	1053	641
36	Makhachkala	2765	1588	2216	2870	1800	1506	1154	2291	1430	772
37	Tomsk	572	2878	3132	207	1481	2483	2235	745	2198	3207
38	Kemerovo	485	2987	3258	203	1580	2590	2320	808	2298	3281
39	Tula	2962	174	776	2868	1470	469	832	2280	754	789
40	Kirov	2193	791	1106	2041	678	422	601	1490	314	1426
41	Cheboksary	2327	602	1080	2221	823	203	376	1638	123	1117
42	Kaliningrad	4001	1089	826	3858	2483	1483	1933	3303	1805	1574
43	Bryansk	3195	348	784	3105	1708	705	1049	2516	991	770
44	Ivanovo	2705	249	700	2576	1188	200	720	2008	519	1090
45	Magnitogorsk	1581	1400	1881	1556	395	1014	592	946	691	1529
46	Kursk	3128	457	982	3063	1674	722	959	2464	961	563

Tabel 2: Jarak Antara Setiap Tempat Pada AK-47-TSP Instance (Lanjutan)

Node_Symbol	Name	10	11	12	13	14	15	16	17	18	19
0	Kur'ya	1435	1780	2683	1802	848	2484	2933	2219	3236	2271
1	Moscow	1496	1166	911	1156	3353	724	466	794	1195	704
2	Saint Petersburg	1912	1632	1544	1492	3575	1350	1072	1355	1754	1251
3	Novosibirsk	1363	1715	2693	1658	634	2456	2880	2166	3267	2200
4	Yekaterinburg	196	374	1406	292	1967	1116	1499	804	1990	817
5	Nizhniy Novgorod	1096	773	847	761	2961	549	606	462	1303	356
6	Samara	765	418	636	658	2725	337	762	65	1216	171
7	Omsk	760	1111	2084	1100	1230	1847	2277	1561	2660	1600
8	Kazan	778	449	847	499	2681	515	797	252	1392	170
9	Rostov-na-Donu	1740	1406	393	1629	3697	661	496	977	250	996
10	Chelyabinsk	0	352	1359	449	1961	1097	1518	803	1943	841
11	Ufa	352	0	1034	364	2308	753	1166	452	1619	492
12	Volgograd	1359	1034	0	1291	3309	332	501	633	586	679
13	Perm	449	364	1291	0	2201	973	1295	659	1862	636
14	Krasnoyarsk	1961	2308	3309	2201	0	3058	3465	2759	3888	2782
15	Saratov	1097	753	332	973	3058	0	472	317	890	349
16	Voronezh	1518	1166	501	1295	3465	472	0	716	738	683
17	Tol'yatti	803	452	633	659	2759	317	716	0	1205	109
18	Krasnodar	1943	1619	586	1862	3888	890	738	1205	0	1233
19	Ulyanovsk	841	492	679	636	2782	349	683	109	1233	0
20	Izhevsk	545	291	1076	224	2406	753	1075	443	1642	413
21	Yaroslavl	1349	1044	1038	969	3147	784	664	749	1401	644
22	Barnaul	1458	1809	2757	1787	655	2537	2974	2258	3323	2299
23	Vladivostok	5064	5410	6407	5282	3103	6161	6563	5861	6981	5881
24	Irkutsk	2793	3143	4126	3050	850	3889	4305	3595	4698	3623
25	Khabarovsk	4877	5212	6235	5041	2942	5964	6335	5654	6818	5661
26	Khabarovsk Vtoroy	4883	5218	6241	5047	2948	5970	6341	5660	6824	5668
27	Orenburg	563	336	826	698	2487	628	1095	435	1402	531
28	Novokuznetsk	1658	2010	2972	1964	446	2745	3175	2460	3541	2498
29	Ryazan'	1386	1044	736	1085	3284	544	331	641	1067	563
30	Tyumen	338	652	1682	561	1668	1404	1797	1097	2268	1115
31	Lipetsk	1455	1104	555	1211	3392	456	107	661	842	614
32	Penza	1089	737	500	885	3036	197	430	291	1007	256
33	Naberezhnyye Chelny	572	251	946	345	2488	628	979	314	1517	300
34	Kalininskiy	1908	1629	1546	1488	3568	1350	1075	1354	1758	1250
35	Astrakhan	1355	1087	375	1410	3238	596	875	804	719	888
36	Makhachkala	1686	1446	680	1780	3504	959	1151	1182	720	1264
37	Tomsk	1471	1818	2820	1723	491	2568	2977	2270	3400	2295
38	Kemerovo	1555	1906	2892	1832	432	2650	3069	2357	3467	2388
39	Tula	1528	1184	773	1226	3426	636	300	772	1023	699
40	Kirov	810	576	1149	391	2566	819	1016	564	1672	482
41	Cheboksary	895	571	845	583	2779	518	724	319	1363	214
42	Kaliningrad	2578	2253	1771	2206	4362	1728	1285	1871	1696	1788
43	Bryansk	1762	1416	869	1464	3664	812	371	992	972	927
44	Ivanovo	1281	967	951	918	3109	689	604	656	1337	552
45	Magnitogorsk	248	250	1140	540	2169	906	1353	642	1721	706
46	Kursk	1705	1354	680	1454	3640	678	207	909	772	864

Tabel 2: Jarak Antara Setiap Tempat Pada AK-47-TSP Instance (Lanjutan)

Node_Symbol	Name	20	21	22	23	24	25	26	27	28	29
0	Kur'ya	1964	2765	219	3773	1503	3702	3708	1866	403	2821
1	Moscow	968	251	2933	6416	4203	6141	6147	1228	3116	182
2	Saint Petersburg	1370	608	3253	6538	4416	6195	6201	1779	3410	815
3	Novosibirsk	1848	2625	195	3715	1434	3575	3581	1868	306	2726
4	Yekaterinburg	451	1246	1515	5065	2811	4851	4857	669	1701	1328
5	Nizhniy Novgorod	566	289	2531	6037	3810	5778	5784	883	2716	331
6	Samara	451	814	2215	5827	3558	5628	5634	371	2419	703
7	Omsk	1269	2065	700	4323	2044	4169	4175	1260	900	2138
8	Kazan	278	600	2226	5773	3527	5538	5544	595	2417	612
9	Rostov-na-Donu	1408	1157	3149	6796	4516	6616	6622	1217	3364	823
10	Chelyabinsk	545	1349	1458	5064	2793	4877	4883	563	1658	1386
11	Ufa	291	1044	1809	5410	3143	5212	5218	336	2010	1044
12	Volgograd	1076	1038	2757	6407	4126	6235	6241	826	2972	736
13	Perm	224	969	1787	5282	3050	5041	5047	698	1964	1085
14	Krasnoyarsk	2406	3147	655	3103	850	2942	2948	2487	446	3284
15	Saratov	753	784	2537	6161	3889	5964	5970	628	2745	544
16	Voronezh	1075	664	2974	6563	4305	6335	6341	1095	3175	331
17	Tol'yatti	443	749	2258	5861	3595	5654	5660	435	2460	641
18	Krasnodar	1642	1401	3323	6981	4698	6818	6824	1402	3541	1067
19	Ulyanovsk	413	644	2299	5881	3623	5661	5668	531	2498	563
20	Izhevsk	0	806	1965	5495	3253	5261	5267	578	2151	880
21	Yaroslavl	806	0	2755	6194	3997	5907	5913	1172	2930	335
22	Barnaul	1965	2755	0	3663	1380	3559	3565	1933	226	2837
23	Vladivostok	5495	6194	3663	0	2284	648	645	5582	3442	6365
24	Irkutsk	3253	3997	1380	2284	0	2206	2212	3301	1158	4132
25	Khabarovsk	5261	5907	3559	648	2206	0	7	5420	3334	6109
26	Khabarovsk Vtoroy	5267	5913	3565	645	2212	7	0	5426	3339	6115
27	Orenburg	578	1172	1933	5582	3301	5420	5426	0	2147	1072
28	Novokuznetsk	2151	2930	226	3442	1158	3334	3339	2147	0	3028
29	Ryazan'	880	335	2837	6365	4132	6109	6115	1072	3028	0
30	Tyumen	746	1529	1226	4767	2512	4560	4566	900	1406	1626
31	Lipetsk	993	560	2913	6486	4235	6251	6257	1061	3111	226
32	Penza	661	590	2545	6136	3876	5916	5922	702	2746	383
33	Naberezhnyye Chelny	135	794	2022	5584	3332	5362	5368	474	2215	816
34	Kalininskiy	1367	606	3247	6531	4409	6188	6194	1777	3404	817
35	Astrakhan	1220	1372	2646	6309	4026	6180	6186	792	2868	1092
36	Makhachkala	1595	1715	2888	6537	4263	6443	6448	1132	3113	1415
37	Tomsk	1924	2680	358	3593	1330	3425	3431	1999	334	2804
38	Kemerovo	2027	2794	266	3516	1239	3372	3378	2067	188	2907
39	Tula	1022	407	2980	6505	4274	6245	6251	1198	3170	143
40	Kirov	287	585	2174	5626	3416	5359	5365	834	2347	752
41	Cheboksary	374	478	2336	5864	3627	5619	5625	706	2524	506
42	Kaliningrad	2039	1238	3992	7358	5209	7021	7027	2301	4164	1230
43	Bryansk	1260	598	3216	6742	4512	6477	6483	1408	3407	381
44	Ivanovo	742	97	2702	6169	3959	5892	5898	1081	2881	276
45	Magnitogorsk	533	1290	1631	5268	2990	5098	5104	324	1841	1268
46	Kursk	1238	697	3162	6733	4484	6491	6497	1299	3361	398

Tabel 2: Jarak Antara Setiap Tempat Pada AK-47-TSP Instance (Lanjutan)

Node_Symbol	Name	30	31	32	33	34	35	36	37	38	39
0	Kur'ya	1246	2879	2509	2007	3283	2544	2765	572	485	2962
1	Moscow	1711	373	555	925	636	1272	1588	2878	2987	174
2	Saint Petersburg	2042	994	1167	1382	8	1906	2216	3132	3258	776
3	Novosibirsk	1102	2814	2450	1914	3100	2608	2870	207	203	2868
4	Yekaterinburg	300	1425	1072	522	1778	1451	1800	1481	1580	1470
5	Nizhniy Novgorod	1311	504	354	527	895	1145	1506	2483	2590	469
6	Samara	1068	713	343	317	1418	777	1154	2235	2320	832
7	Omsk	543	2214	1848	1323	2579	2012	2291	745	808	2280
8	Kazan	1017	715	392	206	1197	1053	1430	2198	2298	754
9	Rostov-na-Donu	2058	598	763	1287	1544	641	772	3207	3281	789
10	Chelyabinsk	338	1455	1089	572	1908	1355	1686	1471	1555	1528
11	Ufa	652	1104	737	251	1629	1087	1446	1818	1906	1184
12	Volgograd	1682	555	500	946	1546	375	680	2820	2892	773
13	Perm	561	1211	885	345	1488	1410	1780	1723	1832	1226
14	Krasnoyarsk	1668	3392	3036	2488	3568	3238	3504	491	432	3426
15	Saratov	1404	456	197	628	1350	596	959	2568	2650	636
16	Voronezh	1797	107	430	979	1075	875	1151	2977	3069	300
17	Tol'yatti	1097	661	291	314	1354	804	1182	2270	2357	772
18	Krasnodar	2268	842	1007	1517	1758	719	720	3400	3467	1023
19	Ulyanovsk	1115	614	256	300	1250	888	1264	2295	2388	699
20	Izhevsk	746	993	661	135	1367	1220	1595	1924	2027	1022
21	Yaroslavl	1529	560	590	794	606	1372	1715	2680	2794	407
22	Barnaul	1226	2913	2545	2022	3247	2646	2888	358	266	2980
23	Vladivostok	4767	6486	6136	5584	6531	6309	6537	3593	3516	6505
24	Irkutsk	2512	4235	3876	3332	4409	4026	4263	1330	1239	4274
25	Khabarovsk	4560	6251	5916	5362	6188	6180	6443	3425	3372	6245
26	Khabarovsk Vtoroy	4566	6257	5922	5368	6194	6186	6448	3431	3378	6251
27	Orenburg	900	1061	702	474	1777	792	1132	1999	2067	1198
28	Novokuznetsk	1406	3111	2746	2215	3404	2868	3113	334	188	3170
29	Ryazan'	1626	226	383	816	817	1092	1415	2804	2907	143
30	Tyumen	0	1724	1370	821	2037	1692	2022	1182	1282	1768
31	Lipetsk	1724	0	371	904	997	926	1222	2906	3001	220
32	Penza	1370	371	0	555	1167	793	1152	2548	2640	499
33	Naberezhnyye Chelny	821	904	555	0	1379	1086	1461	2002	2099	958
34	Kalininskiy	2037	997	1167	1379	0	1908	2219	3126	3252	778
35	Astrakhan	1692	926	793	1086	1908	0	378	2756	2810	1143
36	Makhachkala	2022	1222	1152	1461	2219	378	0	3031	3072	1441
37	Tomsk	1182	2906	2548	2002	3126	2756	3031	0	147	2946
38	Kemerovo	1282	3001	2640	2099	3252	2810	3072	147	0	3049
39	Tula	1768	220	499	958	778	1143	1441	2946	3049	0
40	Kirov	950	918	667	360	1102	1367	1744	2096	2210	887
41	Cheboksary	1120	634	357	325	1078	1090	1463	2298	2401	649
42	Kaliningrad	2766	1274	1604	2009	834	2138	2345	3904	4023	1105
43	Bryansk	2006	356	708	1194	789	1243	1498	3184	3287	238
44	Ivanovo	1477	497	494	718	699	1279	1626	2637	2749	376
45	Magnitogorsk	584	1303	932	499	1878	1112	1439	1680	1753	1405
46	Kursk	1973	250	619	1153	987	1050	1292	3154	3250	290

Tabel 2: Jarak Antara Setiap Tempat Pada AK-47-TSP Instance (Lanjutan)

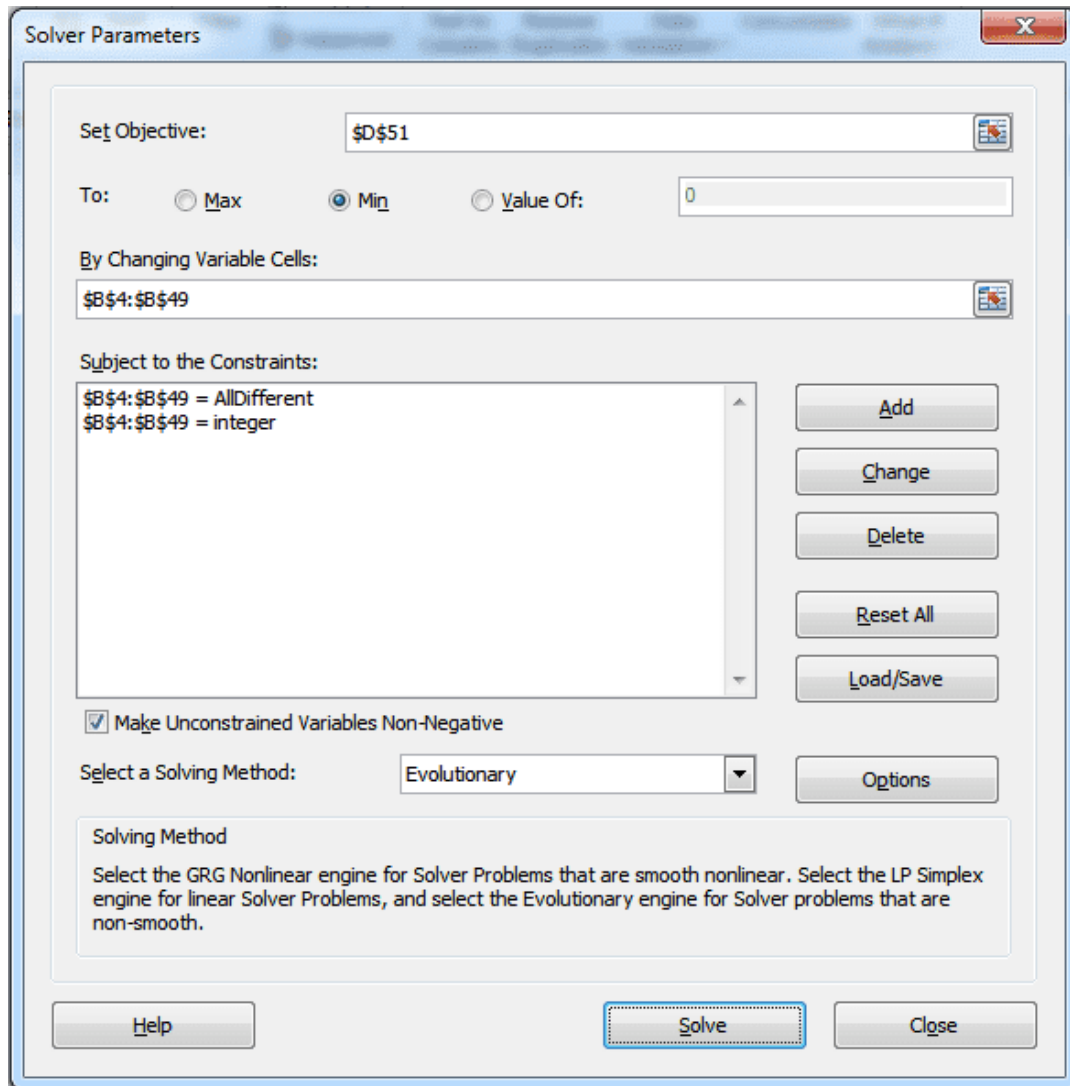
Node_Symbol	Name	40	41	42	43	44	45	46
0	Kur'ya	2193	2327	4001	3195	2705	1581	3128
1	Moscow	791	602	1089	348	249	1400	457
2	Saint Petersburg	1106	1080	826	784	700	1881	982
3	Novosibirsk	2041	2221	3858	3105	2576	1556	3063
4	Yekaterinburg	678	823	2483	1708	1188	395	1674
5	Nizhniy Novgorod	422	203	1483	705	200	1014	722
6	Samara	601	376	1933	1049	720	592	959
7	Omsk	1490	1638	3303	2516	2008	946	2464
8	Kazan	314	123	1805	991	519	691	961
9	Rostov-na-Donu	1426	1117	1574	770	1090	1529	563
10	Chelyabinsk	810	895	2578	1762	1281	248	1705
11	Ufa	576	571	2253	1416	967	250	1354
12	Volgograd	1149	845	1771	869	951	1140	680
13	Perm	391	583	2206	1464	918	540	1454
14	Krasnoyarsk	2566	2779	4362	3664	3109	2169	3640
15	Saratov	819	518	1728	812	689	906	678
16	Voronezh	1016	724	1285	371	604	1353	207
17	Tol'yatti	564	319	1871	992	656	642	909
18	Krasnodar	1672	1363	1696	972	1337	1721	772
19	Ulyanovsk	482	214	1788	927	552	706	864
20	Izhevsk	287	374	2039	1260	742	533	1238
21	Yaroslavl	585	478	1238	598	97	1290	697
22	Barnaul	2174	2336	3992	3216	2702	1631	3162
23	Vladivostok	5626	5864	7358	6742	6169	5268	6733
24	Irkutsk	3416	3627	5209	4512	3959	2990	4484
25	Khabarovsk	5359	5619	7021	6477	5892	5098	6491
26	Khabarovsk Vtoroy	5365	5625	7027	6483	5898	5104	6497
27	Orenburg	834	706	2301	1408	1081	324	1299
28	Novokuznetsk	2347	2524	4164	3407	2881	1841	3361
29	Ryazan'	752	506	1230	381	276	1268	398
30	Tyumen	950	1120	2766	2006	1477	584	1973
31	Lipetsk	918	634	1274	356	497	1303	250
32	Penza	667	357	1604	708	494	932	619
33	Naberezhnyye Chelny	360	325	2009	1194	718	499	1153
34	Kalininskiy	1102	1078	834	789	699	1878	987
35	Astrakhan	1367	1090	2138	1243	1279	1112	1050
36	Makhachkala	1744	1463	2345	1498	1626	1439	1292
37	Tomsk	2096	2298	3904	3184	2637	1680	3154
38	Kemerovo	2210	2401	4023	3287	2749	1753	3250
39	Tula	887	649	1105	238	376	1405	290
40	Kirov	0	310	1818	1119	545	819	1143
41	Cheboksary	310	0	1685	886	397	814	873
42	Kaliningrad	1818	1685	0	920	1298	2488	1092
43	Bryansk	1119	886	920	0	591	1631	209
44	Ivanovo	545	397	1298	591	0	1210	662
45	Magnitogorsk	819	814	2488	1631	1210	0	1550
46	Kursk	1143	873	1092	209	662	1550	0

- C. *Spreadsheet* Untuk Menghitung Panjang Rute Dari Sebuah Solusi
Berikut ini merupakan *Spreadsheet* untuk menghitung panjang rute dari sebuah solusi:

Tabel 3: *Spreadsheet* Untuk Menghitung Panjang Rute Dari Sebuah Solusi

Rute	Name	Jarak (Km)
0	Kur'ya	
45	Magnitogorsk	1581
9	Rostov-na-Donu	1529
34	Kalininskiy	1544
17	Tol'yatti	1354
29	Ryazan'	641
14	Krasnoyarsk	3284
26	Khabarovsk Vtoroy	2948
8	Kazan	5544
39	Tula	754
4	Yekaterinburg	1470
31	Lipetsk	1425
20	Izhevsk	993
3	Novosibirsk	1848
33	Naberezhnyye Chelny	1914
16	Voronezh	979
43	Bryansk	371
2	Saint Petersburg	784
30	Tyumen	2042
41	Cheboksary	1120
42	Kaliningrad	1685
7	Omsk	3303
23	Vladivostok	4323
15	Saratov	6161
12	Volgograd	332
13	Perm	1291
37	Tomsk	1723
11	Ufa	1818
35	Astrakhan	1087
1	Moscow	1272
10	Chelyabinsk	1496
6	Samara	765
32	Penza	343
5	Nizhniy Novgorod	354
36	Makhachkala	1506
18	Krasnodar	720
40	Kirov	1672
28	Novokuznetsk	2347
38	Kemerovo	188
19	Ulyanovsk	2388
25	Khabarovsk	5661
21	Yaroslavl	5907
24	Irkutsk	3997
27	Orenburg	3301
46	Kursk	1299
44	Ivanovo	662
22	Barnaul	2702
0	Kur'ya	219
Panjang Rute		90647

- D. Memasukkan *Objective Function* dan *Constraints* pada *Solver*
 Berikut ini merupakan *Solver* setelah dimasukkan *Objective Function* dan *Constraints*:



Gambar 2: *Solver* Setelah Dimasukkan *Objective Function* dan *Constraints*

- E. Melakukan Perhitungan Dengan *Solver* Sebanyak 10 Kali
 Berikut ini merupakan solusi yang dihasilkan dari 10 kali perhitungan:

Tabel 4: Solusi Yang Dihasilkan Dari 10 Kali Perhitungan

Perhitungan	Solusi (Km)
1	21320
2	21451
3	21709
4	20998
5	21791
6	21900
7	21557
8	21135
9	21595
10	21663

Dari tabel 4 dapat dilihat bahwa solusi terbaik yang diperoleh adalah solusi yang dihasilkan dari perhitungan 4, yaitu solusi dengan panjang rute 20.998 Km.

Tabel 5: Rute Dari Solusi Terbaik Yang Diperoleh

Rute	Name	Jarak (Km)
0	Kur'ya	
7	Omsk	704
10	Chelyabinsk	760
45	Magnitogorsk	248
11	Ufa	250
27	Orenburg	336
6	Samara	371
17	Tol'yatti	65
19	Ulyanovsk	109
32	Penza	256
15	Saratov	197
12	Volgograd	332
35	Astrakhan	375
36	Makhachkala	378
18	Krasnodar	720
9	Rostov-na-Donu	250
16	Voronezh	496
31	Lipetsk	107
29	Ryazan'	226
1	Moscow	182
39	Tula	174
46	Kursk	290
43	Bryansk	209
42	Kaliningrad	920
2	Saint Petersburg	826
34	Kalininskiy	8
21	Yaroslavl	606
44	Ivanovo	97
5	Nizhniy Novgorod	200
41	Cheboksary	203
8	Kazan	123
33	Naberezhnyye Chelny	206
20	Izhevsk	135
40	Kirov	287
13	Perm	391
4	Yekaterinburg	292
30	Tyumen	300
3	Novosibirsk	1102
37	Tomsk	207
38	Kemerovo	147
14	Krasnoyarsk	432
25	Khabarovsk	2942
26	Khabarovsk Vtoroy	7
23	Vladivostok	645
24	Irkutsk	2284
28	Novokuznetsk	1158
22	Barnaul	226
0	Kur'ya	219
Panjang Rute		20998

F. Perbandingan Solusi

Solusi berdasarkan *Random Nearest Neighbour* menghasilkan panjang rute sepanjang 27.803 Km.

Tabel 6: Rute Solusi Terbaik Yang Diperoleh Dari *Random Nearest Neighbour*

Rute	Name	Jarak (Km)
0	Kur'ya	
22	Barnaul	219
3	Novosibirsk	195
38	Kemerovo	203
37	Tomsk	147
28	Novokuznetsk	334
14	Krasnoyarsk	446
24	Irkutsk	850
7	Omsk	2044
30	Tyumen	543
4	Yekaterinburg	300
10	Chelyabinsk	196
45	Magnitogorsk	248
11	Ufa	250
33	Naberezhnyye Chelny	251
20	Izhevsk	135
13	Perm	224
40	Kirov	391
41	Cheboksary	310
8	Kazan	123
19	Ulyanovsk	170
17	Tol'yatti	109
6	Samara	65
15	Saratov	337
32	Penza	197
5	Nizhniy Novgorod	354
44	Ivanovo	200
21	Yaroslavl	97
1	Moscow	251
39	Tula	174
29	Ryazan'	143
31	Lipetsk	226
16	Voronezh	107
46	Kursk	207
43	Bryansk	209
9	Rostov-na-Donu	770
18	Krasnodar	250
12	Volgograd	586
35	Astrakhan	375
36	Makhachkala	378
27	Orenburg	1132
34	Kalininskiy	1777
2	Saint Petersburg	8
42	Kaliningrad	826
25	Khabarovsk	7021
26	Khabarovsk Vtoroy	7
23	Vladivostok	645
0	Kur'ya	3773
Panjang Rute		27803

Dikarenakan solusi berdasarkan *Evolutionary Algorithm & Solver* menghasilkan solusi dengan panjang rute 20.998 Km, maka solusi berdasarkan *Evolutionary Algorithm & Solver* lebih baik dibandingkan dengan solusi yang dihasilkan dari *Random Nearest Neighbour*.

KESIMPULAN

Kombinasi *Evolutionary Algorithm* dan *Solver* dapat digunakan untuk menyelesaikan *Travelling Salesman Problem*. 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 *AK-47-TSP Instance* berdasarkan hasil penyelesaian oleh *Evolutionary Algorithm* dan *Solver*, panjang jarak terbaiknya adalah 20.998 Km. Panjang jarak tersebut lebih pendek dibandingkan panjang jarak terbaik yang dihasilkan *Random Nearest Neighbour* yaitu 27.803 Km.

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

DAFTAR PUSTAKA

- Applegate, D. L., Bixby, R. E., Chvátal, V., & Cook, W. J. (2011). The traveling salesman problem: A computational study. In *The Traveling Salesman Problem: A Computational Study*. <https://doi.org/10.5860/choice.45-0928>
- Badar, A. Q. H. (2021). *Evolutionary Optimization Algorithms*. CRC Press.
- Bashkim IDRIZI, A. (2020). *NECESSITY FOR GEOMETRIC CORRECTIONS OF DISTANCES IN WEB AND MOBILE MAPS* Bashkim IDRIZI. 1, 462–470.
- Bisma, M. A., & Sanggala, E. (2023). Genetic Algorithm for Improving Route of Travelling Salesman Problem Generated by Savings Algorithm. *Sainteks: Jurnal Sains Dan Teknik*, 5(1), 102–111.
- Elshaer, R., & Awad, H. (2020). A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. *Computers and Industrial Engineering*, 140(November 2018), 106242. <https://doi.org/10.1016/j.cie.2019.106242>
- 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. *H2Open Journal*, 3(1), 135–188. <https://doi.org/10.2166/h2oj.2020.128>
- Sanggala, E., & Bisma, M. A. (2023a). Analysis of The Ant Number Effects on Ant Colony Optimization for Solving Russia-20-Nodes-SDVRP Instance. *Sainteks: Jurnal Sain Dan Teknik*, 5(2), 163–174.
- Sanggala, E., & Bisma, M. A. (2023b). Perbandingan Savings Algorithm dengan Nearest Neighbour dalam Menyelesaikan Russian TSP Instances. *Jurnal Media Teknik & Sistem Industri*, 7(1).

- Sanggala, E., & Bisma, M. A. (2023c). Random Nearest Neighbour Untuk Menyelesaikan Russian TSP Instances. *Media Jurnal Informatika*, 15(1), 63–69.
- 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.
- Vikhar, P. A. (2017). Evolutionary algorithms: A critical review and its future prospects. *Proceedings - International Conference on Global Trends in Signal Processing, Information Computing and Communication, ICGTSPICC 2016, figure 1*, 261–265.
<https://doi.org/10.1109/ICGTSPICC.2016.7955308>
- Wu, Z. (2020). A comparative study of solving traveling salesman problem with genetic algorithm, ant colony algorithm, and particle swarm optimization. *ACM International Conference Proceeding Series*, 95–99. <https://doi.org/10.1145/3450292.3450308>