

## CHAPTER 5

### IMPLEMENTATION AND TESTING

#### 5.1 Implementation

This chapter is a chapter that contains the implementation and testing of the program used by researchers in finding the final results of problems published during the research. In this chapter there are two solutions that are applied to the problem taken for optimal results.

Below are some lines of code used in the Greedy algorithm in finding optimal results according to the algorithm:

```
1. berangkat=1
2. res=any(berangkat in sublist for sublist in kumpulanKota)
3. if res:
4.     for kota in kumpulanKota
5.         if berangkat==kota[0]:
6.             data=kumpulanKota.index(kota)
7.             break
8. jarakGrafik1=jarakgrafik(kumpulanKota)
9. kot=greedy(jarakGrafik1, data)
10.     jarakGrafik3=jarakGrafik(kumpulanKota)
11.     totalJarak=kot[1]+jarakGrafik3[kot[2]][data]
```

From the above code, the code is a few lines of code that runs the command to run the Greedy Algorithm calculation. In line 1, by calling a dataset that contains x and y coordinates as a city position that has been determined and declared as a variable "dataset". Line 2 to discard the first line in the dataset so that the array used starts from array 1. Then line 3 is used to provide the "area" variable as a place of city points so that it can be drawn into a graph on the code afterwards. Line 4 to start counting the time the program runs. Line 5 to call the variable from the code that comes from another file and then can be used to draw as a graph. Line 6 to stop the calculation of the time when the program has finished working. Line 7 and 8 to print the output of the calculation that has been declared in each variable.

Then, below are a few lines of code used in the Genetic Algorithm to calculate according to the algorithm.

```
1. pop=Populasi(kelilingkota, 550, True)
2. print      ("Jarak      Awal:      "      +
      str(pop.ambilHasilTerbaik().ambilJarak()))
```

```

3. ga=AlgoritmaGenetika(kelilingkota)
4. pop=ga.genetika(pop)
5. for i in range(0, 550):
6.     pop=ga.genetika(pop)
7. print      ("Jarak      Tercepat:      "      +
      str(pop.ambilHasilTerbaik().ambilJarak()))

```

From some of the code shown, this code is the code used by the Genetic Algorithm in carrying out its calculations on the dataset so that it produces optimal results according to the algorithm. Line 1 to determine the number of Populations in the calculation. The size of the population affects the processing time and the shortness of the final route produced. Line 2 is the code to call the initial distance before counting. Lines 3 and 4 are used to provide variables for the dataset to be calculated in Genetic Algorithms using mutations, crossovers, tournament selection. Line 5 to determine the number of generations carried out in one Genetic process. The size of the number Generation affects the length of time and the shortness of the final route. Line 6 to run a parent and child route search which makes the possible route possible in the search for the fastest route as many as the Number of Generations given. Line 7 to provide the fastest output distance obtained after Genetic calculation is done.

## 5.2 Testing

In this chapter, researchers conduct a program testing with several datasets with some problems that can be drawn from the results of the analysis in the previous chapter. In this test, the Greedy Algorithm is tested whether the end result remains constant / the same continuously or not. For Genetic Algorithms, it is tested whether the effect on the performance of the algorithm if the dataset is used in large numbers and in small amounts.

For testing the Greedy Algorithm, researchers used a dataset of 70 coordinate points that had been drawn into a graph and 10 coordinate points as an experimental dataset.

Table 5.1: 70 Datasets

No	X, Y	Region	No	X, Y	Region
1	31, 30	City Hall	36	33, 6	Diponegoro Statue

2	30, 28	Jl. Mgr Sugiyopranoto	37	32, 3	Tembalang
3	27, 29	Kalibanteng	38	38, 1	Gombel Lama
4	12, 28	Jl. Pamularsih Raya	39	32, 8	Karangrejo Raya
5	18, 25	Sam Poo Kong	40	30, 10	Unika
6	22, 20	Jl. Kaligarang	41	27, 11	Akpelni
7	25, 20	Kariadi Hospital	42	24, 14	Jl. Menoreh Raya
8	28, 21	Jl. Veteran	43	22, 14	Taman Sampangan
9	31, 19	Jl. Pahlawan	44	22, 17	Jl. Kelud Raya
10	37, 21	Simpang Lima	45	24, 18	Taman Madukoro
11	38, 24	Jl. Ahmad Yani	46	26, 20	Jl. Indraprasta
12	42, 22	Jl. Majapahit	47	26, 27	Jl. Imam Bonjol
13	54, 17	Jl. Gajah	48	30, 30	Jl. Hasanudin
14	56, 25	Great Mosque of Central Java	49	30, 32	Jl. Peres
15	54, 28	Jl. Jolotundo	50	32, 31	Jl. Petek
16	54, 27	MT. Haryono	51	30, 33	New Metro Hotel
17	45, 27	Bubakan	52	36, 31	Jl. K.H Agus Salim
18	44, 38	Kota Lama	53	38, 31	Jl. Patimura
19	42, 39	Tawang Station	54	40, 31	Jl. Citarum
20	43, 42	Pasar Johar	55	42, 30	RS Panti Wilasa Citarum
21	40, 37	Sri Ratu Pemuda	56	44, 29	Patung Soekarno - Hatta

22	37, 36	BCA Pemuda	57	47, 25	Universitas Semarang
23	36, 34	Jl. Thamrin	58	51, 24	Jl. Tlogosari
24	35, 29	Kampung Kali	59	47, 21	Jl. Medoho Raya
25	37, 27	Jl. Gajahmada	60	53, 12	Jl. Fatmawati
26	42, 22	Poncol Station	61	50, 11	Universitas Muhammadiyah Semarang
27	33, 36	Paragon Mall	62	48, 12	Jl. Kedungmundu
28	34, 31	Kampung Pelangi Kalisari	63	42, 15	Jl. Tentara Pelajar
29	28, 24	William Booth Public Hospital	64	40, 14	Vihara Tanah Putih
30	27, 22	Jl. S.Parman	65	39, 12	Jl. Dokter Wahidin
31	29, 19	Taman Diponegoro Semarang	66	37, 13	Jl. Sisingamangaraja
32	31, 18	Akpol	67	31, 14	Jl. Papandayan
33	32, 15	Karangrejo	68	28, 12	Unisbank Kendeng
34	33, 12	Jatingaleh	69	24, 9	Jl. Dewi Sartika Timur
35	32, 10	Gombel Baru	70	17, 1	Unnes

Illustration 5.1: Graph of 70 Points with a Total Distance of 359

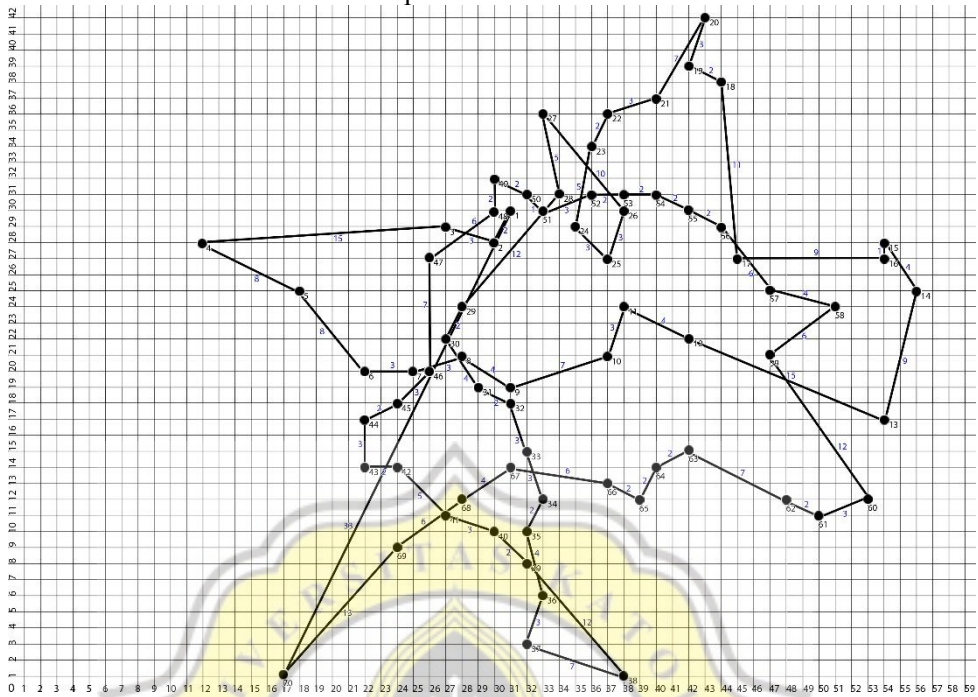
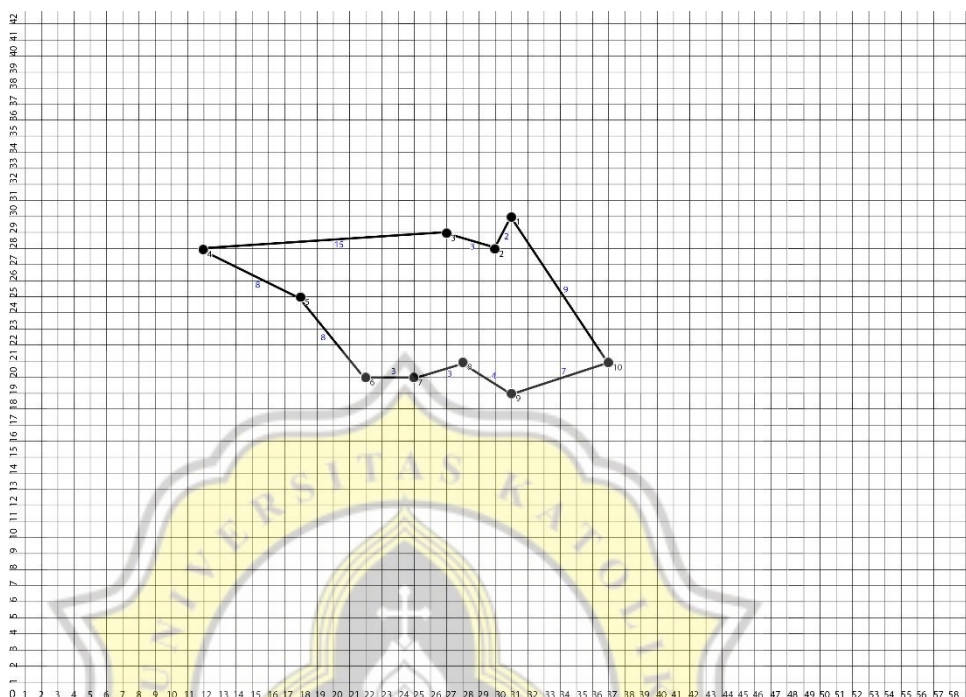


Table 5.2: 10 Datasets

No	X, Y	Region	No	X, Y	Region
1	31, 30	City Hall	6	22, 20	Jl. Kaligarang
2	30, 28	Jl. Mgr Sugiyopranoto	7	25, 20	Kariadi Hospital
3	27, 29	Kalibanteng	8	28, 21	Jl. Veteran
4	12, 28	Jl. Pamularsih Raya	9	31, 19	Jl. Pahlawan
5	18, 25	Sam Poo Kong	10	37, 21	Simpang Lima



Illustration 5.2: Graph of 10 Points with a Total Distance of 62



Then from the dataset above, it is used into the calculation of the Greedy Algorithm which is tested for the final result whether it remains constant or not. A total of 5 times for 70 datasets and 5 times for 10 datasets were done to get the intended results. The data that is tried is shaped into a table so that it is easier to observe. Then by running the program the results obtained are as follows.

Illustration 5.3: Output of Greedy Algorithm with 70 Datasets

```
Rute Tercepat : [0, 47, 1, 2, 46, 28, 29, 7, 30, 8, 31, 32, 66, 33, 34, 38, 35, 36, 37, 64, 63, 62, 65, 39, 67, 40, 68,
41, 42, 43, 44, 6, 45, 5, 4, 3, 48, 50, 49, 27, 51, 52, 25, 53, 54, 55, 16, 56, 58, 57, 15, 14, 13, 12, 59, 60, 61, 11,
10, 9, 24, 23, 22, 21, 20, 18, 17, 19, 26, 69]
Total Jarak : 321
```

Illustration 5.4: Output of Greedy Algorithm with 10 Datasets

```
Rute Tercepat : [0, 1, 2, 7, 6, 5, 4, 3, 8, 9]
Total Jarak : 71
```

The researcher then repeated the calculation of the Greedy Algorithm 10 times and the result obtained was that the Greedy Algorithm gave a constant result for each experiment. For data generated during the experiment written in table 5.4 below

Table 5.3: Constant results 10 times experiment

No	Datasets	Route Order	Distance	Constant?
1	70	1 – 48 – 2 – 3 – 47 – 29 – 30 – 8 – 31 – 9 – 32 – 33 – 67 – 34 – 35 – 39 – 36 – 37 – 38 – 65 – 64 – 63 – 66 – 40 – 68 – 41 – 69 – 42 – 43 – 44 – 45 – 7 – 46 – 6 – 5 – 4 – 49 – 51 – 50 – 28 – 52 – 53 – 26 – 54 – 55 – 56 – 17 – 57 – 59 – 58 – 16 – 15 – 14 – 13 – 60 – 61 – 62 – 12 – 11 – 10 – 25 – 24 – 23 – 22 – 21 – 19 – 18 – 20 – 27 – 70	314	Yes
2	70	1 – 48 – 2 – 3 – 47 – 29 – 30 – 8 – 31 – 9 – 32 – 33 – 67 – 34 – 35 – 39 – 36 – 37 – 38 – 65 – 64 – 63 – 66 – 40 – 68 – 41 – 69 – 42 – 43 – 44 – 45 – 7 – 46 – 6 – 5 – 4 – 49 – 51 – 50 – 28 – 52 – 53 – 26 – 54 – 55 – 56 – 17 – 57 – 59 – 58 – 16 – 15 – 14 – 13 – 60 – 61 – 62 – 12 – 11 – 10 – 25 – 24 – 23 – 22 – 21 – 19 – 18 – 20 – 27 – 70	314	Yes
3	70	1 – 48 – 2 – 3 – 47 – 29 – 30 – 8 – 31 – 9 – 32 – 33 – 67 – 34 – 35 – 39 – 36 – 37 – 38 – 65 – 64 – 63 – 66 – 40 – 68 – 41 – 69 – 42 – 43 – 44 – 45 – 7 – 46 – 6 – 5 – 4 – 49 – 51 – 50 – 28 – 52 – 53 – 26 – 54	314	Yes

		- 55 - 56 - 17 - 57 - 59 - 58 - 16 - 15 - 14 - 13 - 60 - 61 - 62 - 12 - 11 - 10 - 25 - 24 - 23 - 22 - 21 - 19 - 18 - 20 - 27 - 70		
4	70	1 - 48 - 2 - 3 - 47 - 29 - 30 - 8 - 31 - 9 - 32 - 33 - 67 - 34 - 35 - 39 - 36 - 37 - 38 - 65 - 64 - 63 - 66 - 40 - 68 - 41 - 69 - 42 - 43 - 44 - 45 - 7 - 46 - 6 - 5 - 4 - 49 - 51 - 50 - 28 - 52 - 53 - 26 - 54 - 55 - 56 - 17 - 57 - 59 - 58 - 16 - 15 - 14 - 13 - 60 - 61 - 62 - 12 - 11 - 10 - 25 - 24 - 23 - 22 - 21 - 19 - 18 - 20 - 27 - 70	314	Yes
5	70	1 - 48 - 2 - 3 - 47 - 29 - 30 - 8 - 31 - 9 - 32 - 33 - 67 - 34 - 35 - 39 - 36 - 37 - 38 - 65 - 64 - 63 - 66 - 40 - 68 - 41 - 69 - 42 - 43 - 44 - 45 - 7 - 46 - 6 - 5 - 4 - 49 - 51 - 50 - 28 - 52 - 53 - 26 - 54 - 55 - 56 - 17 - 57 - 59 - 58 - 16 - 15 - 14 - 13 - 60 - 61 - 62 - 12 - 11 - 10 - 25 - 24 - 23 - 22 - 21 - 19 - 18 - 20 - 27 - 70	314	Yes
6	10	1 - 2 - 3 - 8 - 7 - 6 - 5 - 4 - 9 - 10	68	Yes
7	10	1 - 2 - 3 - 8 - 7 - 6 - 5 - 4 - 9 - 10	68	Yes
8	10	1 - 2 - 3 - 8 - 7 - 6 - 5 - 4 - 9 - 10	68	Yes



9	10	1 - 2 - 3 - 8 - 7 - 6 - 5 - 4 - 9 - 10	68	Yes
10	10	1 - 2 - 3 - 8 - 7 - 6 - 5 - 4 - 9 - 10	68	Yes

For 70 datasets in the Greedy Algorithm, the Algorithm was successful in providing a shorter route than the original route, but in 10 datasets, the resulting route was higher in distance than the original route.

Table 5.4: Comparison of Initial and Greedy on 10 datasets

No	Route	Route Order	Distance
1	Initial	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 1	62
2	Greedy	1 - 2 - 3 - 8 - 7 - 6 - 5 - 4 - 9 - 10 - 1	68

Table 5.5: Comparison of Initial and Greedy on 70 datasets

No	Route	Route Order	Distance
1	Initial	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 - 32 - 33 - 34 - 35 - 36 - 37 - 38 - 39 - 40 - 41 - 42 - 43 - 44 - 45 - 46 - 47 - 48 - 49 - 50 - 51 - 52 - 53 - 54 - 55 - 56 - 57 - 58 - 59 - 60 - 61 - 62 - 63 - 64 - 65 - 66 - 67 - 68 - 69 - 70 - 1	359
2	Greedy	1 - 48 - 2 - 3 - 47 - 29 - 30 - 8 - 31 - 9 - 32 - 33 - 67 - 34 - 35 - 39 - 36 - 37 - 38 - 65 - 64 - 63 - 66 - 40 - 68 - 41 - 69 - 42 - 43 - 44 - 45 - 7 - 46 - 6 - 5 - 4 - 49 - 51 - 50 - 28 - 52 - 53 - 26 - 54 - 55 - 56 - 17 - 57 - 59 - 58 - 16 - 15 - 14 - 13 - 60 - 61 - 62 - 12 - 11 - 10 - 25 - 24 - 23 - 22 - 21 - 19 - 18 - 20 - 27 - 70	314

Then for the Genetic Algorithm, in this test, the researcher tries whether the results of the fastest route output produced by the Genetic Algorithm dataset if the dataset is reduced will affect the Genetic Algorithm performance in terms of time and the fastest route results. For the first test, the dataset will be reduced to only 10 datasets, and for the second test, the probability of mutation will be increased by 5 attempts but the number of generations and population sizes are made the same during the testing.

Table 5.6: Genetic Algorithm's Dataset

No	Dataset	Mutation Probability	Number of Generations	Population Size
1	10	0.05	100	100
2	10	0.1	100	100
3	10	0.15	100	100
4	70	0.05	100	100
5	70	0.1	100	100
6	70	0.15	100	100

The dataset used is taken as many as 10 points and 70 points. After that, the data is placed in a table so that it is easy to compare with one another.

In the experiment, the Genetic Algorithm uses 10 datasets if the mutation probability is changed in value, the route results obtained from each mutation probability do not differ greatly from one another.

Table 5.7: 10 Datasets

No	Dataset	Mutation Probability	Route Order	Distance
1	10	0.05	4 – 5 – 6 – 7 – 8 – 9 – 10 – 1 – 2 – 3	60
2	10	0.1	8 – 7 – 6 – 5 – 4 – 3 – 2 – 1 – 10 – 9	60
3	10	0.15	9 – 8 – 7 – 6 – 5 – 4 – 3 – 2 – 1 – 10	60

--	--	--	--	--

However, in the 70 datasets used in the Genetic Algorithm, the results of the route distances show more different distances that can be compared to one another.

Table 5.8: 70 Datasets

No	Dataset	Mutation Probability	Route Order	Distance
1	70	0.05	4 – 21 – 51 – 49 – 5 – 6 – 44 – 43 – 60 – 13 – 20 – 53 – 25 – 28 – 24 – 50 – 2 – 22 – 69 – 70 – 40 – 59 – 10 – 11 – 19 – 1 – 47 – 29 – 68 – 67 – 7 – 12 – 18 – 55 – 16 – 14 – 23 – 52 – 58 – 57 – 17 – 61 – 3 – 48 – 9 – 32 – 34 – 36 – 63 – 66 – 42 – 41 – 38 – 37 – 33 – 31 – 46 – 27 – 26 – 54 – 15 – 56 – 8 – 45 – 35 – 65 – 64 – 39 – 61 – 30	719
2	70	0.1	10 – 27 – 52 – 1 – 40 – 41 – 30 – 15 – 60 – 46 – 42 – 43 – 66 – 31 – 39 – 70 – 45 – 65 – 61 – 58 – 54 – 22 – 34 – 29 – 48 – 3 – 53 – 17 – 4 – 2 – 7 – 64 – 11 – 33 – 69 – 36 – 9 – 51 – 32 – 47 – 28 – 55 – 56 – 57 – 61 – 19 – 20 – 18 – 12 – 50 – 8 – 6 – 68 – 67 – 35 – 38 – 37 – 59 – 44 – 5 – 26 – 24 – 21 – 25 – 49 – 23 – 14 – 13 – 16 – 63	844
3	70	0.15	11 – 42 – 26 – 44 – 25 – 2 – 69 – 32 – 70 – 31 – 33 – 40 – 66 – 16 – 15 – 61 – 57 – 55 – 3 – 10 – 21 – 38 – 41 – 8 – 29 – 47 – 5 – 68 – 35 – 23 – 49 – 27 – 51 – 45 – 14 – 58 – 12 – 39 – 13 – 65 – 53 – 54 – 34 – 67 – 36 – 64 – 7 – 24 – 48 – 9 – 17 – 61 – 43 – 37 – 60 – 59 – 56 – 4 – 46 – 6 – 30 – 19 – 18 – 20 – 28 – 50 – 1 – 52 – 22 – 63	887

From this experiment, it can be taken that for the Greedy Algorithm on 10 datasets, the Greedy Algorithm does not provide results that can beat the initial route. The route initially has a lower distance result than after the use of the Greedy Algorithm. For 70 datasets in the Greedy Algorithm, the final results issued provide a significant difference between the initial distance and the distance after calculated with the Greedy Algorithm. For Genetic Algorithms, for the 10 Datasets, all three of the differences in the probability of mutations given produce the same final result of the three differences. This is likely due to the least chance of a route that produces the fastest route on 10 datasets. But for 70 datasets, each mutation probability gives a difference in the end result that can be seen which probability of mutation is good in supporting the Genetic Algorithm to carry out its calculations.

