Genetic Algorithm for Optimizing Traveling Salesman Problems with Time Windows (TSP-TW)

Juwairiah[1], Dicky pratama[2] , Heru Cahya Rustamaji[3], Herry Sofyan[4], Dessyanto Boedi Prasetyo[5]
Program Studi Teknik Informatika Fakultas Teknik Industri UPN "Veteran" Yogyakarta
Jln. Babarsari 2, Tambak Bayan, Yogyakarta
e-mail : [1]juwai_riah@yahoo.com, [2]dckypratama@gmail.com, [3]herucr@gmail.com
*corresponding author

ABSTRACT

The concept of Traveling Salesman Problem (TSP) used in the discussion of this paper is the Traveling Salesman Problem with Time Windows (TSP-TW), where the time variable considered is the time of availability of attractions for tourists to visit. The algorithm used for optimizing the solution of Traveling Salesman Problem with Time Windows (TSP-TW) is a genetic algorithm. The search for a solution for determining the best route begins with the formation of an initial population that contains a collection of individuals. Each individual has a combination of different tourist sequence. Then it is processed by genetic operators, namely crossover with Partially Mapped Crossover (PMX) method, mutation using reciprocal exchange method, and selection using ranked-based fitness method. The research method used is GRAPPLE. Based on tests conducted, the optimal generation size results obtained in solving the TSP-TW problem on the tourist route in the Province of DIY using genetic algorithms is 700, population size is 40, and the combination of crossover rate and mutation rate is 0.70 and 0.30 There is a tolerance time of 5 seconds between the process of requesting distance and travel time and the process of forming a tourist route for the genetic algorithm process.

Keywords: TSP-TW, Time Window, Genetic Algorithm, Route Optimization, Adult Tourism Object

I. INTRODUCTION

Yogyakarta Special Region is one of the Provinces in Java which has an area of 3,185.80 km2. Administratively *D.I. Yogyakarta, Indonesia* has 1 city and 4 districts, namely; *City of Yogyakarta, Kab. Bantul, Kab. Kulon Progo, Kab. Gunungkidul*, and *Kab. Sleman*. Besides being known as a student city, Yogyakarta City is also rich in culture and has many tourist attractions that can be visited by domestic and foreign tourists. Tourism objects in Yogyakarta Province are diverse, ranging from cultural tourism, historic buildings, to natural attractions.

In 2017, the Provincial Tourism Office D.I. Yogyakarta stated the number of foreign and domestic tourists who came to Yogyakarta was 4,549,574 people [1]. This is used by hotel stays or travel agents to provide tour packages that can be enjoyed by tourists who come to D.I Province. Yogyakarta. However, not a few tourists who do not like the tour packages offered and prefer to travel independently. The problem that is often faced by tourists who are new to D.I. Yogyakarta and tourists who want to travel independently in the form of ignorance of the locations to be visited, the operating hours of a tourist attraction, tourist itineraries, and short time also require tourists to arrange schedules as effectively as possible. To overcome these problems, we need a system that is able to solve problems related to the optimization of tourist attractions in Yogyakarta by considering the travel time and schedule of open and close attractions that will be visited by using the Android platform to run it.

The optimization of tourist attractions in the above problem not only solves problems related to travel route optimization, but also considers the travel time and open and close schedules of a tourist attraction to be visited so that the development of a Traveling Salesman Problem (TSP) involving two or more variables is involved. known as Traveling Salesman Problem with Time Windows (TSP-TW). TSP-TW is the optimal route search that considers the total time, travel time, travel time, and time of availability of a tourist attraction to be visited.

The algorithm used to solve the Traveling Salesman Problem with Time Windows (TSP-TW) optimization in this study is a genetic algorithm. Genetic algorithm is an algorithm that was developed from the process of finding solutions using random search, then searching is done based on genetic theories that pay attention to how to get better individuals so that in the process of evolution can be expected to get the best individual. Each individual represents a possible solution to the existing problem.

The coding technique of genetic algorithms is done by encoding genes from a chromosome, where genes are part of a chromosome. A gene will usually represent a variable, genes can be represented in the form of bits, real numbers, list of rules, permutation elements, program elements, and other representations that can be implemented for genetic algorithms. Genetic algorithms are widely used in solving optimization problems because genetic algorithms are able to provide the best solution by basing on the evolutionary process so that it is effective in solving problems related to optimizing route determination based on tourist attractions in the D.I Yogyakarta Province.

 DOI: 10.25139/ijair.v1i1.2024

II. METHOD

The research methodology used is a quantitative method. The quantitative method itself is more systematic, planned, structured and clear from the beginning to the end of the study. Quantitative methods can be interpreted as research methods based on the philosophy of positivity, used to examine specific populations or samples [2]. The sample used in this study uses the Indonesian Disaster Information Data (DIBI). The method applied for system development in this study is the Guidelines for Rapid Application Engineering (GRAPPLE) method. The GRAPPLE system development method has the following stages: requirements gathering, analysis, design, development and deployment.[3]

Application of optimization of completion of the TSP-TW on the route of attractions in the Province of D.I. Yogyakarta by using genetic algorithm is a client server based application. Admin does the process of processing tourist data, while users can optimize travel plans using Android. The flowchart in Fig.1 is the main flow of the TSP-TW optimization calculation process using genetic algorithms. In the application of genetic algorithms will involve several operators, namely evolutionary operations involving the selection process therein and genetic operations involving operators of crossovers and mutations [4].

Genetic algorithms can be implemented in combination optimization problems. To check the results of optimization, a fitness value function is needed. If the genetic algorithm is well designed the population will experience convergence and an optimum solution will be obtained. The final result of genetic algorithms is to display chromosomes that have the highest fitness values of all generations [5].

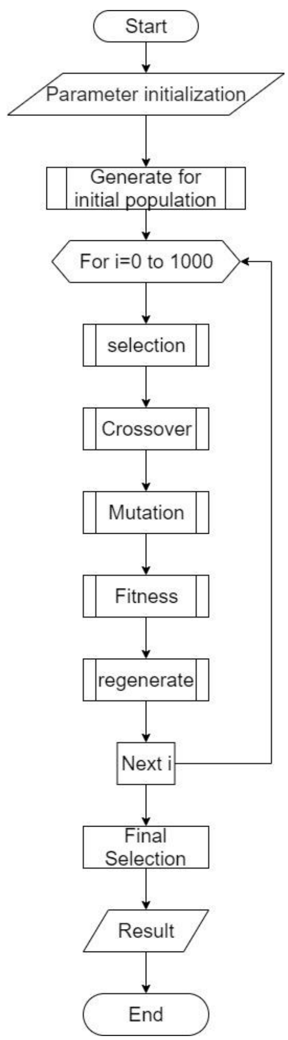

Fig. 1. Genetic Algorithm Flowchart

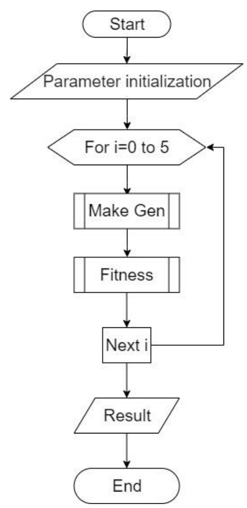

Fig. 1(a). Initial Population Generation Flowchart

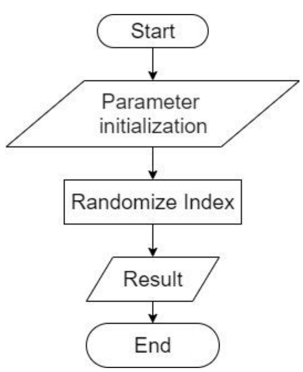

Fig. 1(b). Generator Flowchart

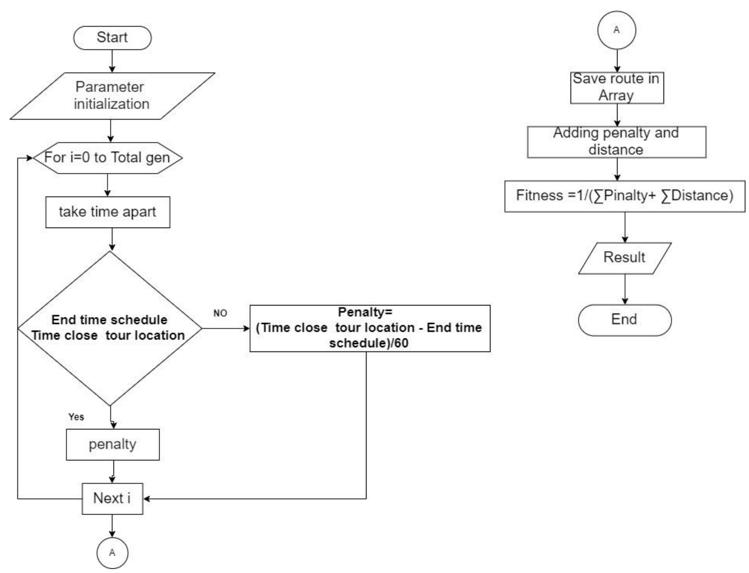

Fig. 1(c). Calculate Fitness Flowchart

 DOI: 10.25139/ijair.v1i1.2024

 E- ISSN: 2686-6269

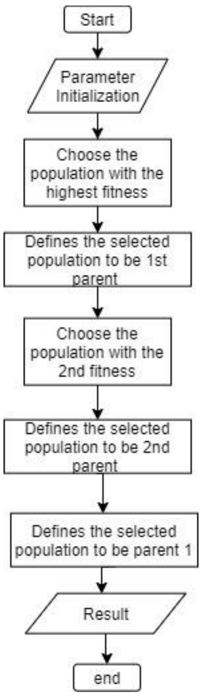

Fig. 1(d). Flowchart Seleksi

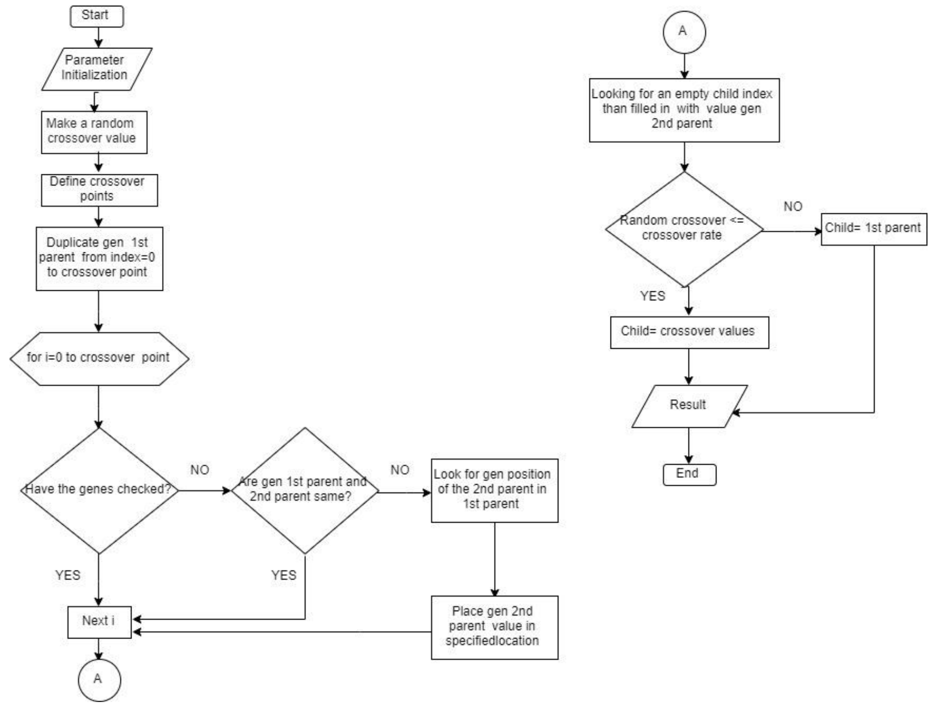

Fig. 1(e). Crossover Flowchart

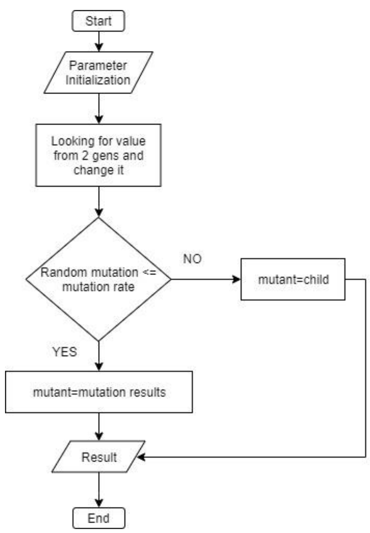

Fig. 1(f). Mutation Flowchart

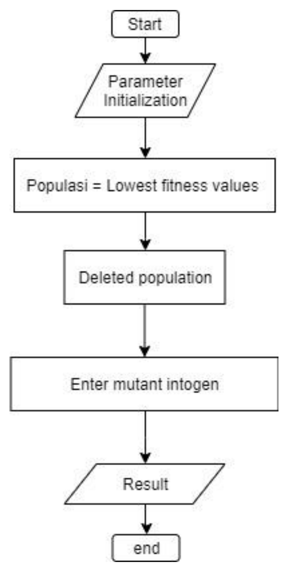

Fig. 1(g). Regression Flowchart

In the initial population generation sub-process there are two sub-processes, namely sub-processes for genes and fitness count sub-processes. The input parameters used in this sub-process are the id of each tourist attraction, the start time of the tour, the time of completion of the tour, and the length of the tour. In this sub-process a population of five individuals is produced along with its fitness value. The initial population generation flowchart can be seen in Fig. 1 (a). Sub-process of making genes is done by forming arrays which are then randomized by tourist routes by randomizing the index of each tourist object without randomizing the first index as the user's initial location. As in Fig. 1 (b). Fitness fitness sub process is done by calculating the value of fitness with Flow on Fig. 1 (c) for each individual. Fitness values are produced with the formula (1) as follows:

$$Nilai\ fitness = \frac{1}{\sum P + \sum J + \sum W} \tag{1}$$

where the $\sum P$ variable represents the total penalty, the $\sum J$ variable represents the total mileage, and the $\sum W$ variable represents the total waiting time.

 DOI: 10.25139/ijair.v1i1.2024

The selection method in Fig. 1 (d) used in this study is a rank-based fitness selection method. Rank-based fitness selection method is a selection method by sorting individuals in the population based on their fitness value[6]. Individuals with the highest fitness value will rank first, individuals with the second highest fitness value will rank second, and so on.

In the crossover flow sub process Fig. 1 (e) used is PMX (Partially Mapped Crossover)[7]. In solving problems related to route optimization, the crossover using the PMX method offers better performance than most other crossover techniques. Crossover with the PMX method is able to produce children with different gene structures (there are no identical genes in 1 individual). In the mutase sub-process in flow Fig. 1 (f) used is reciprocal exchange. The reciprocal exchange method works by exchanging two genes randomly. Regeneration sub-process Fig.1 (g) is re-ranking individuals in the population, then looking for the lowest ranked individuals. Ranking is done based on the fitness value of each individual.

III. RESULT AND DISCUSSION

The results and discussion will explain the application of the use of genetic algorithms and application test results of the optimization application of the completion of the Traveling Salesman Problem with Time Windows (TSP-TW) on tourist route routes in the Province of D.I. Yogyakarta. A case of a user with a starting location at UPNVYK Condongcatur wants to travel to Prambanan Temple, Mangunan Fruit Garden, Yogyakarta Palace and Parangtritis Beach, with travel time starting at 8:00 to 18:00, and the length of tourist visit is 60 minutes. The following is detailed information that will be used in the example calculation in Table I.

TABLE I
INFORMATION ON OPENING HOURS AND CLOSING HOURS OF ATTRACTION

Id wisata	Nama wisata	Jam buka	Jam tutup
1	Lokasi awak (UPNVYK Condongcatur)	-	-
2	Prambanan temple	06:00	17:00
3	Mangunan Fruit Garden	05:00	18:00
4	Yogyakarta palace	08:00	14:00
5	Parangtritis beach	00:00	24:00

TABLE II
TOURIST DISTANCE AND TIME INFORMATION

Travel ID 1	Travel ID 2	Mileage (KM)	Travel time (Minutes)
1	2	13	22
1	3	25,5	49
1	4	10	28
1	5	38	66
2	3	28	53
2	4	18,2	35
2	5	45	78
3	4	21,2	43
3	5	23	42
4	5	29	56

Calculation simulation based on genetic algorithm flowchart which is the main flow in the process of optimizing tourist routes using time windows. In the flowchart there are inputs with several parameters, namely: the user's initial location, the time of the start of the trip, the time of the end of the trip, the length of the tour visit, and the list of tours to be visited in Table II. Then there are six sub-processes and outputs in the form of TSP-TW optimization results in the travel design table and tourist route map. Sub-processes contained in the genetic algorithm flowchart are initial population generation, selection, crossover, mutation, fitness calculation, and regeneration. These sub-processes are interrelated in producing the best individual in a population which is then used as a TSP-TW optimization solution. In the initial population generation sub-process there are two sub-processes, namely sub-processes for genes and sub-processes for fitness calculation. The input parameters used in this sub-process are the id of each tourist attraction, the start time of the tour, the end of the tour, and the length of the tour. In this sub-process a population of five individuals is produced along with its fitness value in Table II. In the process of creating genes, an array is formed which is then randomized by the tourist object route by randomizing the index of each tourist object without randomizing the first index as the user's initial location. Randomization is performed with the shuffle function in php to randomize the order of several elements in an array. Following are the results of the randomized index of attractions that users want to visit in Table III.

 DOI: 10.25139/ijair.v1i1.2024

 E- ISSN: 2686-6269

TABLE III
EARLY POPULATION GENE

Person	Tourist Route
1	[1-2-5-4-3]
2	[1-5-3-2-4]
3	[1-3-2-5-4]
4	[1-4-3-2-5]
5	[1-2-4-3-4]

After the five persons are formed then a fitness calculation is performed for each individual in the fitness calculation sub-process. Table IV is a calculation of the fitness value of each individual.

TABLE IV
NILAI FITNESS INDIVIDU

Person	Node	Start Time	Travelling Time (Minute)	Time arrived	Start Time	Duration of visit	End wisata	Penalty (Minute)	Range (KM)
Individu 1 [1-2-5-4-3]	1-2	08:00	22	08:22	08:22	60	09:22	0	13
	2-5	09:22	78	10:40	10:40	60	11:40	0	45
	5-4	11:40	56	12:36	12:36	60	13:36	0	29
	4-3	13:36	43	14:19	14:19	60	15:19	0	21,2
	3-1	15:19	49	15:58	-	-	-	0	25,5
							Total:	0	133,7
							Fitness Value =	$\frac{1}{0 + 133,7+0}$	= 0,007479
Individu 2 [1-5-3-2-4]	1-5	08:00	66	09:06	09:06	60	10:06	0	38
	5-3	10:06	42	10:48	10:48	60	11:48	0	23
	3-2	11:48	53	12:43	12:43	60	13:43	0	28
	2-4	13:43	38	14:21	14:21	60	15:21	81	18,2
	4-1	15:21	28	15:49	-	-	-	0	10
							Total:	81	117,2
							Fitness Value=	$\frac{1}{81 + 117,2+0}$	= 0,005045
Individu 3 [1-3-2-5-4]	1-3	08:00	49	08:49	08:49	60	09:49	0	25,5
	3-2	09:49	53	10:42	10:42	60	11:42	0	28
	2-5	11:42	78	13:00	13:00	60	14:00	0	45
	5-4	14:00	56	14:56	14:56	60	15:56	116	29
	4-1	15:56	28	16:24	-	-	-	0	10
							Total:	116	137,5
							Fitness Value =	$\frac{1}{116 + 137,5+0}$	= 0,003944
Individu 4 [1-4-3-2-5]	1-4	08:00	38	08:28	08:28	60	09:28	0	10
	4-3	09:28	43	10:11	10:11	60	11:11	0	21,2
	3-2	11:11	53	12:04	12:04	60	13:04	0	28
	2-5	13:04	78	14:22	14:22	60	15:22	0	45
	5-1	15:22	66	16:28	-	-	-	0	38
							Total:	0	142,2
							Fitness Value =	$\frac{1}{0 + 142,2+0}$	= 0,007032
Individu 5 [1-2-4-3-5]	1-2	08:00	22	08:22	08:22	60	09:22	0	13
	2-4	09:22	38	10:00	10:00	60	11:00	0	18,2
	4-3	11:00	43	11:43	11:43	60	12:43	0	21,2
	3-5	12:43	42	13:25	13:25	60	14:25	0	23
	5-1	14:25	66	15:31	-	-	-	0	38
							Total:	0	113,4
							Fitness Value =	$\frac{1}{0 + 113,4+0}$	= 0,008818

The results of the initial population generation sub-process will then be selected in the selection sub-process. The selection method used in this study is rank-based fitness selection method. Rank-based fitness selection method is a selection method by sorting persons in the population based on their fitness value. The person with the highest fitness value will be first. Persons with the second highest fitness value will rank second, and so on. Based on the results of the initial initial population generation, the following are the results of ranking of persons in the initial population based on their fitness value in Table V.

 DOI: 10.25139/ijair.v1i1.2024

 E- ISSN: 2686-6269

TABLE V
RANKING INDIVIDU

Person	Tourist Route	Fitness Value	Ranking
5	[1-2-4-3-5]	0,008818	1 (Parent 1)
1	[1-2-5-4-3]	0,007479	2 (Parent 2)
4	[1-4-3-2-5]	0,007032	3
2	[1-5-3-2-4]	0,005045	4
3	[1-3-2-5-4]	0,003944	5

Furthermore, parent 1 and parent 2 will be mated using the crossover sub-process to produce a child (child). In the crossover sub-process, the crossover method used is PMX. In solving problems related to route optimization, the crossover using the PMX method offers better performance than most other crossover techniques. Crossover with the PMX method is able to produce children with different gene structures (there are no identical genes in 1 individual). Following are the crossover steps using the PMX method:

a) Determine the crossover point, crossover point is determined by halving the number of genes in a parent that is,
   Gen parent 1 = [1-2-4-3-5]
   Gen parent 2 = [1-2-5-4-3]
   Crossover rate = 0,35; Parent gene length 1 or parent 2 = 5;
   So the crossover rate = length of parent gene / 2 = 5/2 = 2.5. Rounded to 3.
b) PMX crossover 1 process, PMX crossover 1 process is done by dividing the length of parent 1 and parent 2 genes according to the crossover point, the first three genes from parent 1 will be passed down to the child according to its index. In the 2 PMX crossover process, the parent 2 gene is checked with the parent 1 gene according to its index. First starts from the 0th index, where the value of the parent gene of the 2nd index and the value of the parent gene of the 1st index is equal to 1, so the process will continue to the next index. The value of the parent index gene 2 index 1 and the value of the gene gene parent 1 index 1 also have the same value of 2, so that it is continued to the gene index parent 2. The PMX 3 crossover process continues checking beforehand. Checking is continued on the 2nd index parent 2 gene with the 2nd index parent 1 gene. Both genes have different values. Next is how to place the value of 5 in the parent 2 gene to child. The way to do this is to look at the value of the parent gene 1 in the same index that is 4. The value 4 in the parent gene 2 is at index 3, so the value of 5 in the parent gene 2 will go down to the child at index 3. Finally the 4 process crossover PMX is checking the child index that has not been filled. The child at index 4 detected has not been filled, so the value of 3 in the parent 2 gene will go down to the child according to its index.

Index	k	0	1	2	3	4
Parent 1	=	1	2	4	3	5
Parent 2	=	1	2	5	4	3
Child	=	1	2	4		

Index	k	0	1	2	3	4
Parent 1	=	1	2	4	3	5
Parent 2	=	1	2	5	4	3
Child	=	1	2	4		

Index	k	0	1	2	3	4
Parent 1	=	1	2	4	3	5
Parent 2	=	1	2	5	4	3
Child	=	1	2	4	5	

Index	k	0	1	2	3	4
Parent 1	=	1	2	4	3	5
Parent 2	=	1	2	5	4	3
Child	=	1	2	4	5	3

After obtaining a child from the PMX crossover process, the crossover rate is checked. In this study a crossover rate of 0.35 was used based on research with the title "Optimization of Vehicle Routing Problem with Time Windows in Catering Distribution Using Genetic Algorithms"[8]. So if the random crossover value rises <= 0.35, the child generated is the result of the PMX crossover. If the value of the random crossover that rises> = 0.35, the resulting child is parent 1. For example the value of the random crossover that rises is 0.30 then the result of the crossover is the result of the PMX crossover ie the child with genes [1-2-4-5 -3]. Furthermore, the child will be mutated using the mutation subprocess to produce mutants.

In the reciprocal exchange process, random numbers that arise are used as a reference index whose gene values will be exchanged. After mutants are obtained from the reciprocal exchange mutation process, a mutation rate check is performed. In this study a mutation rate of 0.05 was used based on research with the title "Optimization of Vehicle Routing Problems with Time Windows in Catering Distribution Using Genetic Algorithms"[8]. So if the random mutation value arises <= 0.05, the mutants produced will be the result of a reciprocal exchane mutation. If the value of random mutation arises> = 0.05 then the mutant produced is child. Suppose the value of the random mutation that arises is 0.03 then the result of the mutation is the result of a reciprocal exchange mutation ie mutants with genes [1-4-2-5-3].

The next experiment is to determine the optimal population size by calculating the value of fitness with tourist sites that will be addressed as many as 6 tourist sites. The number of generations to be used is 700. The number of individuals used is a multiple of 10 with a population of at least 10 and a maximum of 50, a combination of crossover rate and mutation rate that is used respectively 0.35 and 0.15. Each generation is tested 10 times using a ranked based fitness selection Table VI and a generation size trial table with a generation size trial graph in Fig.2.

 DOI: 10.25139/ijair.v1i1.2024

TABLE VI
UKURAN POPULASI

No.	Population Size				
	10	20	30	40	50
1	0,003788	0,003788	0,003788	0,003788	0,003788
2	0,004167	0,003571	0,003788	0,004167	0,004167
3	0,003597	0,003788	0,003876	0,004167	0,004167
4	0,003185	0,003876	0,003788	0,003788	0,00361
5	0,004016	0,003788	0,003788	0,004167	0,003788
6	0,004167	0,00369	0,004167	0,004167	0,004167
7	0,003788	0,003788	0,003788	0,00369	0,003521
8	0,003788	0,003876	0,004016	0,003876	0,004167
9	0,004167	0,003185	0,003597	0,003788	0,003788
10	0,003788	0,004167	0,003788	0,003788	0,004167
Average	0,003845	0,003752	0,003838	0,003938	0,003933

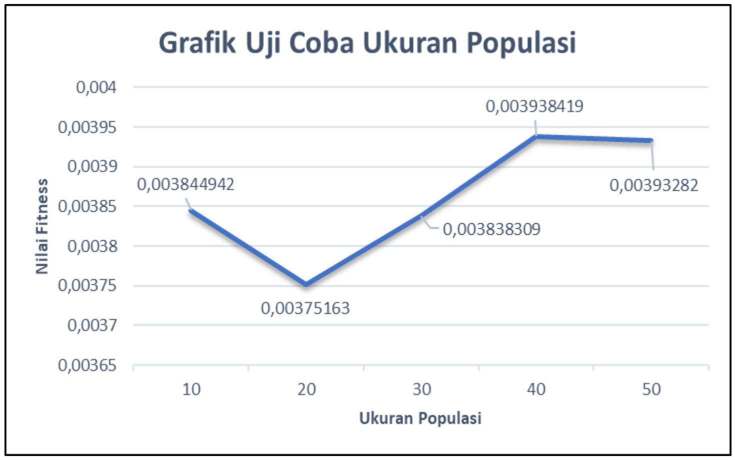

Fig. 2. Graph of Population Size Trial

Based on the graph in Figure 4, the highest average fitness value is obtained at a population size of 40. The more population size, the more individual combinations will be produced in the population. So it will affect the value of fitness produced later. The greater the size of the population also affects the computational time of genetic algorithms.

The next experiment is to determine the optimal combination of Crossover Rate (CR) and Mutation Rate (MR) by calculating the value of fitness with 6 tourist sites to be addressed. The number of generations to be used is 700. The number of individuals used is 40, a combination of crossover rate and mutation rate. The right combination of crossover rate and mutation rate will help genetic algorithms in the process of crossover and mutation in producing the best individual. The combination of crossover rate and mutation rate used ranges from 0.1-1 with a combination of crossover rate> mutation rate. Each generation is tested 10 times using a ranking based fitness selection Table VII method and a generation size trial table with a generation size trial graph in Fig. 3. Based on the graph in Figure 4.3, the highest average fitness value is obtained at the combination of crossover rate and mutation rate respectively 0.7 and 0.3 with an average fitness value of 0.003977.

TABLE VII
SIZE OF A COMBINATION OF CROSSOVER RATE AND MUTATION RATE

No.	Combination CR : MR				
	0,5 : 0,5	0,6 : 0,4	0,7 : 0,3	0,8 : 0,2	0,9 : 0,1
1	0,003788	0,003788	0,004167	0,003788	0,003788
2	0,004167	0,003788	0,003788	0,003788	0,003788
3	0,003788	0,004167	0,003788	0,004167	0,004167
4	0,003788	0,003788	0,004167	0,004167	0,003788
5	0,004167	0,003788	0,003788	0,004167	0,003788
6	0,003788	0,004167	0,003788	0,003788	0,004167
7	0,003788	0,003788	0,004167	0,003788	0,003788
8	0,003788	0,004167	0,004167	0,003788	0,003788
9	0,004167	0,003788	0,004167	0,004167	0,003788
10	0,003788	0,003788	0,003788	0,003788	0,003788
Rata-rata	0,003902	0,003902	0,003977	0,003939	0,003864

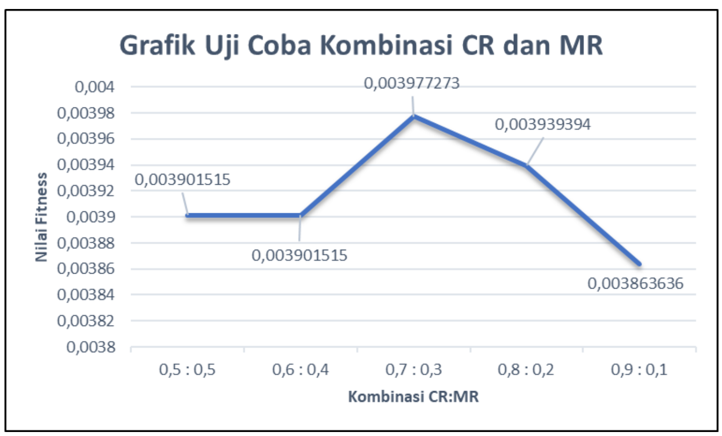

Fig. 3. Test Trial Combination of Crossover Rate and Mutation Rate

Analysis of computational time testing of genetic algorithms is carried out to find out how long it takes to do route optimization using genetic algorithms. The execution time of the genetic algorithm depends on the number of tours entered by the user and the hardware used for the computational process in Table VIII and the graph of the results of the genetic algorithm computation in Fig. 4, the more number of tours entered by the user, the longer the time needed for genetic algorithm execution.

 DOI: 10.25139/ijair.v1i1.2024

TABLE VIII
COMPUTATIONAL GENETIC ALGORITHMS

Number of Tourist Attractions	Execution Time (Seconds)
2	2,32
3	2,50
4	2,47
5	2,58
6	3,02
7	3,04
8	3,85
9	4,13
10	4,38

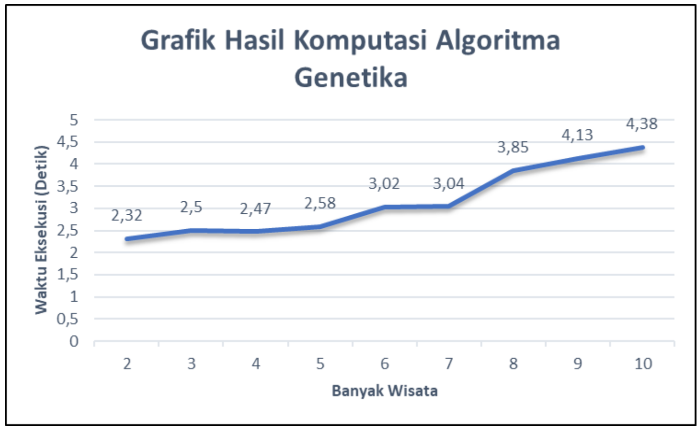

Fig. 4 Computation Results of Genetic Algorithms

IV. CONCLUSION

Based on the results of the analysis, design, and discussion that have been carried out previously, then an optimization application can be generated for the completion of the Traveling Salesman Problem with Time Windows (TSP-TW) on the tourist route in the province of D.I. Yogyakarta by using an Android-based genetic algorithm. Application testing is done by testing on the generation size, population size, and a combination of crossover rate and mutation rate to produce optimal results in solving the TSP-TW problem on tourist route routes in the D.I province. Yogyakarta by using genetic algorithms. Based on the tests conducted, the result of generation size is 700, population size is 40, and the combination of crossover rate and mutation rate is 0.70 and 0.30.

The process of forming a tourist route using Google's fire is different from the process of requesting the distance and travel time of each tour. The process of requesting mileage and travel time is done first by using google fire, namely the distance matrix of fire, then proceed with the genetic algorithm process, then the process of forming a tourist route by using google fire is fire direction. In this study there is a tolerance time of 5 seconds between the process of requesting distance and travel time and the process of forming a tourist route for the genetic algorithm process.

Making the TSP-TW completion optimization application on the tourist attraction route in the D.I province. Yogyakarta by using an Android-based genetic algorithm there are a few suggestions so that further application development can be carried out, namely to save the route for each destination that is traversed based on the distance and travel time data received from the Google Distance Matrix Fire. Update each parameter when the user has arrived at each tourist location for optimizing the completion of a Traveling Salesman Problem with Time Windows (TSP-TW) so that the optimization results obtained are close to real time and develop android applications in the form of native Android.

REFERENCES

[1] Dinas Pariwisata Daerah Istimewa Yogyakarta, “Statistik Kepariwisataan.” 2017.
[2] I. Etikan, “Comparison of Convenience Sampling and Purposive Sampling,” *AJTAS*, vol. 5, no. 1, p. 1, 2016.
[3] N. A. PELKA, “SISTEM INFORMASI GEOGRAFIS LOKASI POOL BUS DI KOTA MEDAN MENGGUNAKAN METODE GRAPPLE BERBASIS ANDROID,” 2017.
[4] T. Baeck, D. B. Fogel, and Z. Michalewicz, *Evolutionary Computation 1: Basic Algorithms and Operators*. CRC Press, 2018.
[5] N. D. Priandani and W. F. Mahmudy, “Optimasi travelling salesman problem with time windows (TSP-TW) pada penjadwalan paket rute wisata di pulau Bali menggunakan algoritma genetika,” *SESINDO 2015*, vol. 2015, 2015.
[6] N. M. Razali and J. Geraghty, “Genetic algorithm performance with different selection strategies in solving TSP,” presented at the Proceedings of the world congress on engineering, 2011, vol. 2, pp. 1–6.
[7] C.-K. Ting, C.-H. Su, and C.-N. Lee, “Multi-parent extension of partially mapped crossover for combinatorial optimization problems,” *Expert Systems with Applications*, vol. 37, no. 3, pp. 1879–1886, 2010.
[8] D. C. A. Nugraha and W. F. Mahmudy, “Optimasi vehicle routing problem with time windows pada distribusi katering menggunakan algoritma genetika,” *SESINDO 2015*, vol. 2015, 2015.

DOI: 10.25139/ijair.v1i1.2024