

BAB 3

ANALISIS DAN PERANCANGAN SISTEM

3.1. ANALISIS SISTEM

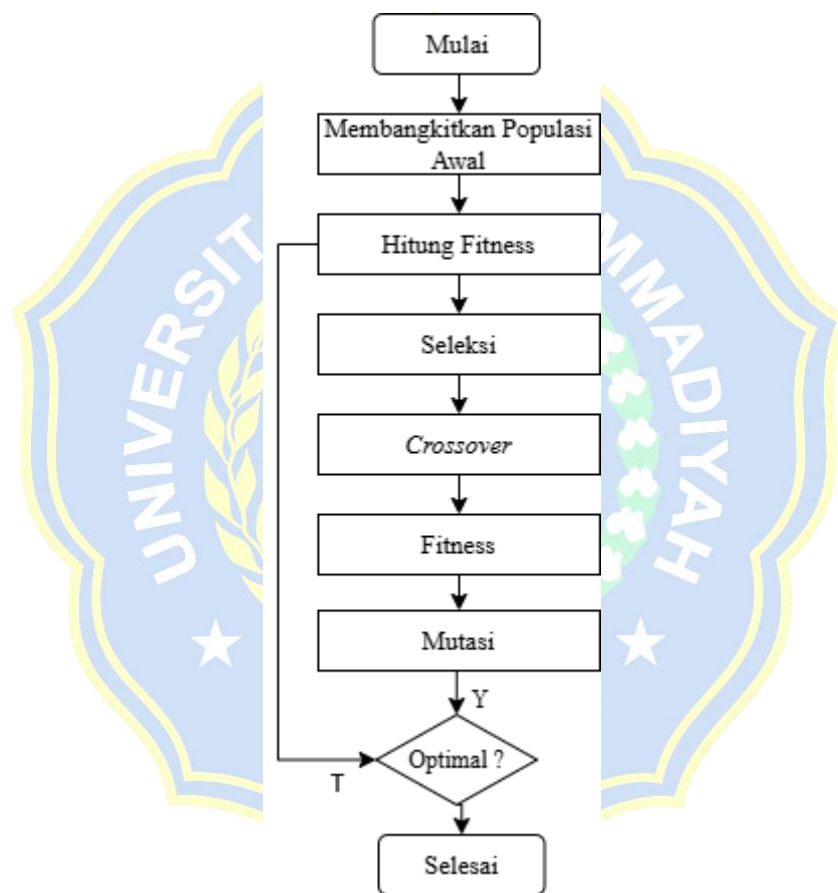
Pariwisata di Jawa Timur memiliki potensi yang besar dengan berbagai jenis destinasi seperti Wisata Edukasi, pantai, kuliner, dan wisata religi. Gresik, sebuah kota di Jawa Timur, terkenal dengan tradisi agama dan budaya yang kuat. Beberapa tempat wisata menarik di Gresik antara lain Edu Wisata Lontar Sewu, Bukit Jamur, Bandar Grisse, Wisata Alam Gosari, dan Taman Bundar. Pemilihan rute wisata di Kota Gresik seperti mempertimbangkan minat dan preferensi wisatawan, destinasi wisata apa saja yang sering dikunjungi, pemilihan rute yang memberikan jarak tempuh yang efisien antar destinasi wisata, memilih rute yang mudah diakses oleh transportasi umum atau kendaraan pribadi dan lain sebagainya. Namun dengan semakin banyaknya destinasi wisata, efisiensi dan efektivitas rute yang dipilih perlu diperhatikan agar pengalaman wisatawan tetap optimal.

Analisis Sistem Penentuan Rute Jarak Wisata di Kota Gresik menggunakan metode Algoritma genetika didapatkan masalah utama yaitu belum adanya sistem untuk menentukan rute jarak wisata di Kota Gresik yang dapat membantu wisatawan dalam berwisata ke tempat wisata, agar mendapatkan efisiensi jarak yang ditempuh dan efektifitas jalur wisata yang ditempuh karena pemilihan rute yang tidak tepat dapat mempengaruhi jarak yang ditempuh. Dari permasalahan tersebut peneliti membuat Sistem Penentuan Rute Jarak Wisata di Kota Gresik. menggunakan metode Algoritma Genetika.

3.2. HASIL ANALISIS SISTEM

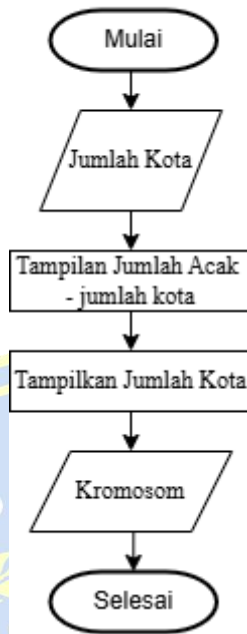
Penelitian ini menggunakan Algoritma Genetika untuk menyelesaikan *Traveling Salesman Problem* (TSP) dalam menentukan jalur wisata di Kota Gresik. Algoritma ini dipilih karena efektivitasnya dalam menangani TSP dan kemampuannya menghasilkan solusi yang mendekati optimal. Penerapan Algoritma Genetika dalam menentukan jalur wisata di Kota Gresik diharapkan

memberikan kontribusi signifikan terhadap pengembangan pariwisata di wilayah tersebut. Keputusan menggunakan Algoritma Genetika didasarkan pada kemudahan penerapannya dan hasil optimal yang dapat dicapai. Dalam penelitian ini, dibuat Sistem Penentuan Rute Jarak Wisata di Kota Gresik dengan menggunakan metode Algoritma Genetika. Penelitian ini diharapkan memperkuat dasar pengembangan jalur wisata optimal di Kota Gresik dan memberikan kontribusi yang signifikan terhadap pengembangan pariwisata di wilayah tersebut.



Gambar 3.1 Diagram Alir Algoritma Genetika Secara Umum

a. **Membangkitkan Populasi Awal**



Gambar 3.2 Diagram Alir Inisialisasi Populasi Awal

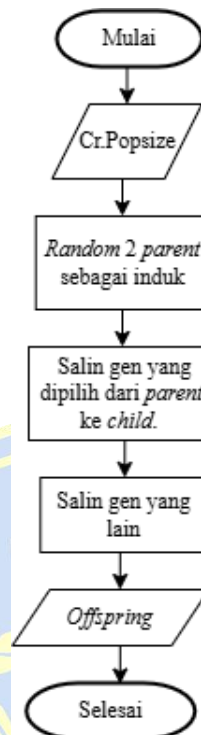
Dalam membangkitkan populasi awal digunakan teknik berupa *random number generator*. Inti dari cara ini adalah melibatkan pembangkitan bilangan *random* untuk nilai setiap gen sesuai dengan representasi kromosom yang digunakan. Sebagai contoh, sebuah kromosom untuk 7 kota bisa direpresentasikan :

[0.23 0.87 0.13 0.57 0.24 0.89 0.32]

Posisi i dalam list menunjukkan kota i . Nilai acak dalam posisi i menentukan urutan didatanginya kota i dalam lintasan TSP. Dengan kunci-kunci nilai *random* di atas, kita dapat menentukan bahwa nilai 0.13 merupakan nilai yang paling kecil, sehingga kota- i menempati urutan kota kedua dan seterusnya. Sehingga dengan demikian, dari kunci-kunci nilai *random* di atas kita dapat menentukan lintasan :

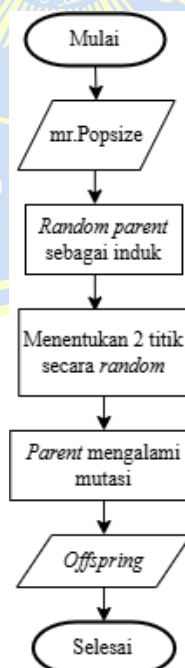
3 – 1 – 5 – 7 – 4 – 2 – 6

b. Crossover



Gambar 3.3 Diagram Alir *Crossover*

c. Mutasi



Gambar 3.4 Diagram Alir Mutasi

Dalam Algoritma Genetika untuk menyelesaikan TSP, nilai *fitness* terpendek diukur sebagai *invers* dari total jarak rute yang ditempuh. Semakin pendek jaraknya, semakin tinggi nilai *fitnessnya*. Langkah-langkahnya melibatkan perhitungan jarak total untuk setiap individu, diikuti dengan penghitungan nilai *fitness* dengan mengambil *invers* dari total jarak. Setelah itu, dilakukan seleksi berdasarkan nilai *fitness* untuk menentukan individu yang akan menjadi orangtua dalam reproduksi. Proses ini berulang secara iteratif hingga solusi yang mendekati optimal ditemukan.

Data dipetakan di lokasi setiap wisata yang dikunjungi berdasarkan titik koordinat Google Maps. Penulis mendapatkan 20 destinasi lokasi wisata yang ada di Kota Gresik, seperti Telaga Ngipik, Kampung Kemas, Bandar Grisee, Dynasty Water Park dan lain sebagainya.

Tabel 3.1 Data Lokasi Wisata di Kota Gresik

No.	Lokasi	Latitude	Longitude
1	Telaga Ngipik	-7,15181	112,63296
2	Kampung Kemas	-7,14774	112,65544
3	Nyai Ageng Pinatih	-7,15192	112,65827
4	Bandar Grisee	-7,15267	112,65609
5	Alun-Alun Gresik	-7,15492	112,65683
6	Museum Sunan Giri	-7,15841	112,65464
7	Wisata Kampung Kreasi	-7,16307	112,65774
8	Dynasty Water Park	-7,13872	112,61525
9	Bukit Kapur Suci	-7,14254	112,60669
10	Bukit Awan	-7,16202	112,62243
11	Wisata Religi Sunan Giri	-7,16844	112,63102
12	Mangrove Kalimereng	-7,10794	112,60678
13	Ekowisata Mangrove Kali Lamong	-7,19466	112,63482
14	Pantai Karang Kiring	-7,18868	112,66579
15	Edu Wisata Lontar Sewu	-7,23828	112,58365
16	Wisata Jati Sewu	-7,27289	112,62431
17	Bukit Jamur	-7,04212	112,56638
18	Wisata Telaga Palemwatu	-7,26462	112,58192
19	Wisata Kemudi Warrior	-7,12111	112,54738
20	Telaga Perengan	-7,15358	112,62628

Jarak antar dua titik dalam koordinat desimal dihitung menggunakan rumus *Haversine* berikut dari beberapa lokasi yang akan dikunjungi :

```
from math import radians, cos, sin, asin, sqrt
def haversine(lon1, lat1, lon2, lat2):
# convert decimal degrees to radians
    lon1, lat1, lon2, lat2= map(radians, [lon1,lat1,
lon2,lat2])
    # haversine formula
    dlon = lon2 - lon1
    dlat = lat2 - lat1
    a = sin(dlat/2)**2 + cos(lat1) * cos(lat2) *
sin(dlon/2)**2
    c = 2 * asin(sqrt(a))
    r = 6371 # Radius of earth in kilometers. Use 3956 for
miles. Determines return value units.
    return c * r
```

Perhitungan jarak antar titik koordinat lokasi akan dihasilkan, yang akan disajikan dalam matriks jarak dalam **Tabel 3.2**.

Tabel 3.2 Matriks Jarak

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	2,5212	2,7925	2,5537	2,65617	2,50198	3,007	2,4365	3,0762	1,6244	1,8615	5,6692	4,7691	5,4705	11,047	13,497	14,239	13,75	10,04	0,7628
2	2,5212	0	0,5599	0,5529	0,81297	1,18973	1,7234	4,5463	5,4096	3,9731	3,5436	6,9578	5,6917	4,6933	12,809	14,333	15,314	15,32	12,285	3,2821
3	2,7925	0,5599	0	0,2546	0,36949	0,82534	1,2412	4,9682	5,7857	4,1106	3,5232	7,4961	5,411	4,1709	12,648	13,963	15,87	15,099	12,705	3,5343
4	2,5537	0,5529	0,2546	0	0,26317	0,658	1,1707	4,7654	5,5655	3,8564	3,2749	7,3715	5,2256	4,1447	12,429	13,82	15,783	14,897	12,497	3,2905
5	2,6562	0,813	0,3695	0,2632	0	0,45714	0,9118	4,9285	5,7007	3,8765	1,5034	7,6016	5,0421	3,8819	12,292	13,599	16,029	14,734	12,648	3,3738
6	2,502	1,1897	0,8253	0,658	0,45714	0	0,6209	4,8663	5,5769	3,5763	2,8346	7,7058	4,5857	3,5836	11,841	13,162	16,188	14,277	12,54	3,1747
7	3,007	1,7234	1,2412	1,1707	0,91178	0,62086	0	5,4137	6,0774	3,8974	3,0078	8,3182	4,3281	2,983	11,694	12,756	16,808	14,052	13,04	3,6278
8	2,4365	4,5463	4,9682	4,7654	4,92855	4,86627	5,4137	0	1,0356	2,7092	3,7347	3,5479	6,5843	7,871	11,606	14,952	12,019	14,474	7,7402	2,0521
9	3,0762	5,4096	5,7857	5,5655	5,70067	5,57688	6,0774	1,0356	0	2,7763	3,9369	3,8474	6,5741	8,2968	10,945	14,624	12,019	13,847	6,9643	2,4857
10	1,6244	3,9731	4,1106	3,8564	3,87653	3,57625	3,8974	2,7092	2,7763	0	1,1865	6,2564	3,8783	5,6277	9,4978	12,33	14,697	12,253	9,4477	1,0301
11	1,8615	3,5436	3,5232	3,2749	3,22003	2,83455	3,0078	3,7347	3,9369	1,1865	0	7,2394	2,9455	4,4474	9,3604	11,638	15,753	11,988	10,623	1,7331
12	5,6692	6,9578	7,4961	7,3715	7,60162	7,70579	8,3182	3,5479	3,8474	6,2564	7,2394	0	10,127	11,09	14,716	18,443	8,5697	17,637	6,7157	5,5122
13	4,7691	5,6917	5,411	5,2256	5,04211	4,58571	4,3281	6,5843	6,5741	3,8783	2,9455	10,127	0	3,4807	7,4424	8,7757	18,567	9,7246	12,647	4,664
14	5,4705	4,6933	4,1709	4,1447	3,88191	3,58361	2,983	7,871	8,2968	5,6277	4,4474	11,09	3,4807	0	10,608	10,422	19,644	12,526	15,071	5,8509
15	11,047	12,809	12,648	12,429	12,292	11,8409	11,694	11,606	10,945	9,4978	9,3604	14,716	7,4424	10,608	0	5,9098	21,895	2,9351	13,629	10,527
16	13,497	14,333	13,963	13,82	13,5994	13,1619	12,756	14,952	14,624	12,33	11,638	18,443	8,7757	10,422	5,9098	0	26,444	4,7652	18,891	13,268
17	14,239	15,314	15,87	15,783	16,0291	16,188	16,808	12,019	12,019	14,697	15,753	8,5697	18,567	19,644	21,895	26,444	0	24,8	9,03	14,046
18	13,75	15,32	15,099	14,897	14,7337	14,2769	14,052	14,474	13,847	12,253	11,988	17,637	9,7246	12,526	2,9351	4,7652	24,8	0	16,406	13,281
19	10,04	12,285	12,705	12,497	12,6476	12,54	13,04	7,7402	6,9643	9,4477	10,623	6,7157	12,647	15,071	13,629	18,891	9,03	16,406	0	9,4243
20	0,7628	3,2821	3,5343	3,2905	3,37385	3,17468	3,6278	2,0521	2,4857	1,0301	1,7331	5,5122	4,664	5,8509	10,527	13,268	14,046	13,281	9,4243	0

3.3. REPRESENTASI MODEL

3.3.1. Perhitungan Algoritma Genetika

Algoritma genetika terdiri dari empat proses utama. Pertama, inialisasi awal digunakan menemukan solusi awal untuk masalah. Kemudian, proses reproduksi yang mencakup *mutation* dan *crossover*,

menghasilkan keturunan baru. Selanjutnya, evaluasi dilakukan untuk mengukur nilai kecocokan setiap individu dalam populasi. Terakhir, seleksi digunakan untuk memilih individu baru berdasarkan nilai kecocokan, dengan memprioritaskan mereka yang memiliki nilai kecocokan yang paling tinggi.

a. Inisialisasi Parameter Algoritma Genetika

Langkah awal algoritme genetika, kita menentukan parameter-parameter seperti jumlah generasi, ukuran populasi, tingkat persilangan, dan tingkat mutasi. Parameter yang digunakan dalam penelitian ini untuk menyelesaikan masalah dengan algoritma genetika adalah :

- Jumlah Generasi : 20
- Popsize : 10
- *Crossover rate* : 0,5
- *Mutation rate* : 0,1

b. Mendefinisikan Gen

Gen dalam hal ini merupakan representasi dari tempat yang dikunjungi oleh seorang wisatawan, dengan kata lain gen adalah titik suatu graf. Representasi gen dari **Tabel 3.2** adalah sebagai berikut :

Gen 1 = Telaga Ngipik	= Lokasi 1
Gen 2 = kampung kemas	= Lokasi 2
Gen 3 = Nyai Ageng Pinatih	= Lokasi 3
Gen 4 = Bandar Grisee	= Lokasi 4
Gen 5 = Alun-Alun Gresik	= Lokasi 5
Gen 6 = Museum Sunan Giri	= Lokasi 6
Gen 7 = Wisata Kampung Kreasi	= Lokasi 7
Gen 8 = Dynasty Water Park	= Lokasi 8
Gen 9 = Bukit Kapur Suci	= Lokasi 9
Gen 10 = Bukit Awan	= Lokasi 10
Gen 11 = Wisata Religi Sunan Giri	= Lokasi 11
Gen 12 = Mangrove Kalimereng	= Lokasi 12

Gen 13 = Ekowisata Mangrove Kali Lamong	= Lokasi 13
Gen 14 = Pantai Karang Kiring	= Lokasi 14
Gen 15 = Edu Wisata Lontar Sewu	= Lokasi 15
Gen 16 = Wisata Jati Suwe	= Lokasi 16
Gen 17 = Bukit Jamur	= Lokasi 17
Gen 18 = Wisata Telaga Palemwatu	= Lokasi 18
Gen 19 = Wisata Kemudi Warior	= Lokasi 19
Gen 20 = Telaga Perengan	= Lokasi 20

Dalam satu kromosom terdapat 20 gen. Dalam kromosom berisi gen dari 1 sampai 20 yang membentuk suatu rute perjalanan yang berawal dan berakhir pada titik yang sama.

c. Membangkitkan Populasi Awal

Selanjutnya untuk pencarian rute jarak wisata di Kota Gresik adalah membangkitkan populasi awal untuk diagram alir seperti pada **Gambar 3.2**. Menggunakan *random number generator* pada excel dengan perintah =rand() didapatkan beberapa rute secara acak. Hasil pengambilan rute secara acak membentuk populasi pada generasi awal sebagai berikut :

Tabel 3.3 Populasi Awal

Kromosom	Susunan Gen
1	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20
2	3-5-16-17-9-2-8-1-13-12-7-11-10-6-14-15-4-18-20-19
3	4-11-6-8-12-16-19-10-3-18-15-2-13-14-5-7-17-1-9-20
4	9-12-17-1-4-13-15-6-10-3-7-11-20-2-18-19-5-14-16-8
5	8-17-16-10-2-3-12-19-20-11-13-15-18-14-4-7-5-6-9-1
6	7-13-10-17-15-4-1-16-6-3-12-9-5-8-19-20-14-11-18-2
7	12-1-16-11-13-7-2-20-15-4-3-6-8-9-17-18-19-14-5-10
8	15-8-16-17-20-11-7-2-5-3-10-9-14-12-19-1-4-13-18-6

9	13-11-20-1-3-16-10-15-17-9-6- 14-4-5-7-8-12-2-18-19
10	19-9-16-5-3-12-8-20-6-7-1-10-13- 2-15-14-18-4-11-17

Pembangkitan populasi awal dalam skripsi ini menghasilkan ukuran populasi sebanyak 10 individu. Individu-individu tersebut selanjutnya akan dihitung masing-masing nilai *fitnessnya*.

d. Menentukan Nilai *Fitness*

Dalam algoritma genetika, fungsi *fitness* sangat penting karena mengukur seberapa baik kromosom dapat menyelesaikan masalah yang diberikan. Untuk permasalahan TSP, yang bertujuan meminimalkan jarak total, nilai *fitness* dihitung dengan membalikkan total jarak yang ditempuh oleh penjualan. Dengan kata lain, semakin pendek total jaraknya, semakin tinggi nilai *fitnessnya*, menunjukkan kualitas solusi yang lebih baik. Nilai *fitness* dapat dihitung menggunakan **persamaan 2.1**.

$$\begin{aligned} \text{Kromosom 1} &= (1 \rightarrow 2) + (2 \rightarrow 3) + (3 \rightarrow 4) + (4 \rightarrow 5) + (5 \rightarrow 6) + \\ &(6 \rightarrow 7) + (7 \rightarrow 8) + (8 \rightarrow 9) + \dots + (20 \rightarrow 1) \\ &= 130,2913 \end{aligned}$$

$$\begin{aligned} \text{Fitness Kromosom 1} &= \frac{1}{130,2913} \\ &= 0,00767 \end{aligned}$$

$$\begin{aligned} \text{Kromosom 2} &= (3 \rightarrow 5) + (5 \rightarrow 16) + (16 \rightarrow 17) + (17 \rightarrow 9) + (9 \rightarrow 2) \\ &+ (2 \rightarrow 8) + (8 \rightarrow 1) + (1 \rightarrow 13) + \dots + (19 \rightarrow 3) \\ &= 172,7382 \end{aligned}$$

$$\begin{aligned} \text{Fitness Kromosom 2} &= \frac{1}{172,7382} \\ &= 0,00578 \end{aligned}$$

Untuk mendapatkan nilai *fitness* setiap kromosom lainnya, lakukan perhitungan yang sama, sehingga *fitness* untuk setiap kromosom :

Tabel 3.4 Nilai *Fitness* Kromosom

Kromosom	Total jarak	<i>Fitness</i>
1	130,2913	0,00767511
2	172,7382	0,00578911
3	150,1243	0,00666115
4	145,3345	0,00688068
5	124,3184	0,00804386
6	174,5012	0,00573062
7	155,3326	0,0064378
8	161,3478	0,00619779
9	152,7117	0,00654829
10	145,5479	0,00687059
Total <i>Fitness</i>		0,06683499

Kromosom yang mempunyai nilai *fitness* tertinggi adalah 0,008043 memiliki susunan kromosom = (8-17-16-10-2-3-12-19-20-11-13-15-18-14-4-7-5-6-9-1) dan jarak rute yang didapatkan adalah 124,3184. Kemudian kromosom ini disimpan dan disalin langsung ke populasi berikutnya untuk proses *elitism*.

e. Proses Seleksi

Total *fitness* semua kromosom pada populasi adalah 0,06683. Untuk menghitung nilai *fitness* relatif setiap kromosom adalah :

- Mencari probabilitas *fitness* tiap kromosom (p_i) dengan menggunakan rumus dari **persamaan 2.2**.

$$p_i = \frac{f_i}{\sum_{i=1}^{10} f_i}$$

$$p_1 = \frac{0,00767}{0,06683} = 0,1148$$

$$p_2 = \frac{0,00578}{0,06683} = 0,0866$$

$$p_3 = \frac{0,00666}{0,06683} = 0,0996$$

$$p_4 = \frac{0,00688}{0,06683} = 0,1029$$

$$p_5 = \frac{0,00834}{0,06683} = 0,1203$$

$$p_6 = \frac{0,00573}{0,06683} = 0,0857$$

$$p_7 = \frac{0,00643}{0,06683} = 0,0963$$

$$p_8 = \frac{0,00619}{0,06683} = 0,0927$$

$$p_9 = \frac{0,00654}{0,06683} = 0,0979$$

$$p_{10} = \frac{0,00687}{0,06683} = 0,1027$$

Kemudian menghitung *fitness* kumulatif setiap *fitness* relatif yang telah diperoleh :

$$c_1 = 0,1148$$

$$c_2 = 0,1148 + 0,0866 = 0,2014$$

$$c_3 = 0,2014 + 0,0966 = 0,3011$$

$$c_4 = 0,3011 + 0,1029 = 0,4040$$

$$c_5 = 0,4040 + 0,1203 = 0,5244$$

$$c_6 = 0,5244 + 0,0857 = 0,6101$$

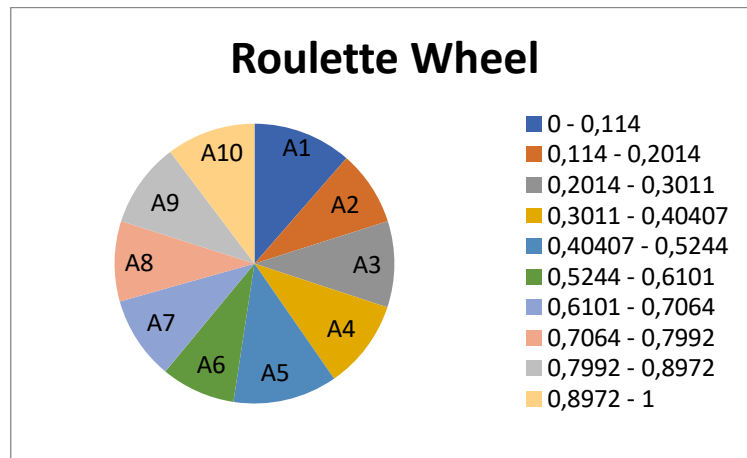
$$c_7 = 0,6101 + 0,0963 = 0,7064$$

$$c_8 = 0,7064 + 0,0927 = 0,7992$$

$$c_9 = 0,7922 + 0,0979 = 0,8972$$

$$c_{10} = 0,8972 + 0,1027 = 1$$

Dari hasil probabilitas kumulatif tersebut jika digambarkan dengan *Roulette Wheel* adalah sebagai berikut :



Gambar 3.5 Probabilitas Kumulatif dalam *Roulette Wheel*

Keterangan :

A_i dengan $i = 1,2,3,\dots,10$ merupakan interval dari c_i ke c_{i+1}

Langkah-langkah seleksi dengan metode *Roulette Wheel* sebagai berikut :

1. Bangkitkan bilangan acak R antara 0 samapi 1 sebanyak kromosom dalam satu populasi.
2. Jika $R_k < c_i$ untuk setiap k dan $i = 1,2,3,\dots,10$, maka kromosom ke- k dipilih sebagai *parent*. Selain itu pilih kromosom ke- k sebagai *parent* dengan syarat $c_{i-1} < R_k < c_i$. Dengan kata lain, jika R_k berada di dalaim A_i maka kromosom ke- k digantikan kromosom ke- i .

Bilangan acak R dibangkitkan sebanyak jumlah kromosom dalam satu populasi yaitu 10 bilangan. Perintah untuk membangkitkan 10 bilangan acak pada excel dengan perintah =rand(). Hasilnya sebagai berikut :

R1 = 0,3592
 R2 = 0,4896
 R3 = 0,4451
 R4 = 0,6083
 R5 = 0,7996
 R6 = 0,0925
 R7 = 0,1873
 R8 = 0,8843
 R9 = 0,9925
 R10 = 0,3783

Kromosom induk yang terpilih untuk proses reproduksi adalah :

Tabel 3.5 Kromosom Terpilih sebagai Induk

Induk	Ri	K.Terpilih	Susunan Gen
1	0,0433	K.1	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20
2	0,1526	K.2	3-5-16-17-9-2-8-1-13-12-7-11-10-6-14-15-4-18-20-19
3	0,2356	K.3	4-11-6-8-12-16-19-10-3-18-15-2-13-14-5-7-17-1-9-20
4	0,3876	K.4	9-12-17-1-4-13-15-6-10-3-7-11-20-2-18-19-5-14-16-8
5	0,5176	K.5	8-17-16-10-2-3-12-19-20-11-13-15-18-14-4-7-5-6-9-1
6	0,6032	K.6	7-13-10-17-15-4-1-16-6-3-12-9-5-8-19-20-14-11-18-2
7	0,6137	K.7	12-1-16-11-13-7-2-20-15-4-3-6-8-9-17-18-19-14-5-10
8	0,7886	K.8	15-8-16-17-20-11-7-2-5-3-10-9-14-12-19-1-4-13-18-6
9	0,8542	K.9	13-11-20-1-3-16-10-15-17-9-6-14-4-5-7-8-12-2-18-19
10	0,98316	K.10	19-9-16-5-3-12-8-20-6-7-1-10-13-2-15-14-18-4-11-17

f. Crossover

Proses *crossover* dengan metode *order crossover* untuk diagram alir seperti pada **Gambar 3.3**. Pasangkan secara acak kromosom induk sebanyak setengah ukuran populasi dan tentukan nilai probabilitas *crossover* (P_c). Nilai P_c yang dinyatakan ialah 0,5. Misal susunan pasangan orangtua adalah :

Pasangan 1	induk 4 induk 7
Pasangan 2	induk 6 induk 1
Pasangan 3	induk 2 induk 5
Pasangan 4	induk 3 induk 9

Pasangan 5 induk 10
 induk 8

Bangkitkan bilangan acak menggunakan excel dengan perintah =rand(). Pasangan ke-i akan mengalami *crossover* apabila bilangan acak R_i kurang atau sama dengan P_c dan sebaliknya, pasangan tersebut akan masuk ke tahap berikutnya. Berikut ini adalah contoh bilangan acak yang dihasilkan:

R1 = 0,25584
R2 = 0,74758
R3 = 0,96104
R4 = 0,88288
R5 = 0,63178

Di antara bilangan acak yang dihasilkan terdapat nilai yang lebih kecil dari nilai P_c , yaitu R1. Artinya pasangan pertama yang berpotongan adalah *parent* 4 dan *parent* 7 dan sisanya masuk ke tahap berikutnya tanpa pindah silang. Berikut proses *order crossover* untuk pasangan induk terpilih :

- a. Salin gen yang dipilih dari *parent* ke *child*. Susunan gen yang telah dipilih disalin langsung ke *offspring*. Misal pada *parent* 1 akan disalin ke *child* 1 dan *parent* 2 akan disalin ke *child* 2.

Parent 1

9	12	17	1	4	13	15	6	10	3
7	11	20	2	18	19	5	14	16	8

Parent 2

12	1	16	11	13	7	2	20	15	4
3	6	8	9	17	18	19	14	5	10

Child 1

9	12	17	1						
					19	5	14	16	8

Child 2

12	1	16	11						
					18	19	14	5	10

- b. Salin gen yang lain. Pada *parent* pasangannya, hilangkan gen-gen yang sudah disalin dan salin sisa gen ke *child* sesuai dengan urutan.

Parent 2

			11	13	7	2	20	15	4
3	6				10				18

Child 1

9	12	17	1						
					19	5	14	16	8

Salin sisa gen dari *parent 2* ke *child 1*

9	12	17	1	11	13	7	2	20	15
4	3	6	10	18	19	5	14	16	8

Parent 1

9		17		4	13	15	6		3
7		20	2						8

Child 2

12	1	16	11						
					18	19	14	5	10

Salin sisa gen dari *parent 1* ke *child 2*

12	1	16	11	9	17	4	13	15	6
3	7	20	2	8	18	19	14	5	10

Setelah pindah silang dilakukan, maka akan mendapatkan kromosom baru, yaitu *child 1* dan *child 2*.

Tabel 3.6 Hasil pindah silang dengan metode *Order Crossover*

Induk 1	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20
Induk 2	3-5-16-17-9-2-8-1-13-12-7-11-10-6-14-15-4-18-20-19
Induk 3	4-11-6-8-12-16-19-10-3-18-15-2-13-14-5-7-17-1-9-20
Offspring 4	9-12-17-1-11-13-7-2-20-15-4-3-6-10-18-19-5-14-16-8
Induk 5	8-17-16-10-2-3-12-19-20-11-13-15-18-14-4-7-5-6-9-1
Induk 6	7-13-10-17-15-4-1-16-6-3-12-9-5-8-19-20-14-11-18-2
Offspring 7	12-1-16-11-9-17-4-13-15-6-3-7-20-2-8-18-19-14-5-10
Induk 8	15-8-16-17-20-11-7-2-5-3-10-9-14-12-19-1-4-13-18-6
Induk 9	13-11-20-1-3-16-10-15-17-9-6-14-4-5-7-8-12-2-18-19
Induk 10	19-9-16-5-3-12-8-20-6-7-1-10-13-2-15-14-18-4-11-17

f. Mutasi

Proses mutasi untuk diagram alir seperti pada **Gambar 3.4**. Nilai probabilitas mutasi (P_m) dinyatakan sebesar 0,1. Membangkitkan bilangan acak antara 0 sampai 1 sebanyak jumlah populasi menggunakan excel dengan perintah =rand(). Misal bilangan acak yang dihasilkan adalah:

R1 = 0,1229
R2 = 0,7708
R3 = 0,3557
R4 = 0,0436
R5 = 0,4828
R6 = 0,7565
R7 = 0,6784
R8 = 0,6258
R9 = 0,6154
R10 = 0,2776

Berdasarkan bilangan acak yang dihasilkan, diketahui R4 saja yang mempunyai nilai lebih kecil dari Pm, sehingga kromosom ke-4 saja yang dipilih untuk mutasi. Pada penelitian ini akan menggunakan metode *Mutation Inversion* yaitu Metode mutasi pembalikan dilakukan dengan mengambil suatu *substring* yang terletak diantara dua titik pada kromosom. Lalu pemilihan akan dilakukan secara acak. Selanjutnya akan memilih sejumlah gen secara berurutan (*invers*) gen pada *substring* yang akan dibalik posisi gennya. Contohnya sebagai berikut :

9	12	17	1	11	13	7	2	20	15
4	3	6	10	18	19	5	14	16	8

Akan dipilih gen yang akan dimutasi yaitu gen ke-4 dan gen ke-5. Sebagai contoh gen akan melakukan pembalikan dalam kromosom tersebut akan menjadi seperti berikut.

9	12	17	11	1	13	7	2	20	15
4	3	6	10	18	19	5	14	16	8

Terlihat pada perubahan bahwa gen ke-4 saat ini menempati urutan gen ke-5 dan gen ke-5 saat ini menempati urutan gen ke-4. Setelah dilakukan mutasi, kromosom tersebut dikelompokkan menjadi kromosom-kromosom yang tidak mengalami mutasi sehingga terbentuk populasi baru. Populasi baru hasil dari mutasi adalah:

Tabel 3.7 Hasil mutasi menggunakan metode *Mutation Inversion*

Kromosom 1	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20
Kromosom 2	3-5-16-17-9-2-8-1-13-12-7-11-10-6-14-15-4-18-20-19
Kromosom 3	4-11-6-8-12-16-19-10-3-18-15-2-13-14-5-7-17-1-9-20
Kromosom 4	9-12-17-11-1-13-7-2-20-15-4-3-6-10-18-19-5-14-16-8
Kromosom 5	8-17-16-10-2-3-12-19-20-11-13-15-18-14-4-7-5-6-9-1

Kromosom 6	7-13-10-17-15-4-1-16-6-3-12-9-5-8-19-20-14-11-18-2
Kromosom 7	12-1-16-11-9-17-4-13-15-6-3-7-20-2-8-18-19-14-5-10
Kromosom 8	15-8-16-17-20-11-7-2-5-3-10-9-14-12-19-1-4-13-18-6
Kromosom 9	13-11-20-1-3-16-10-15-17-9-6-14-4-5-7-8-12-2-18-19
Kromosom 10	19-9-16-5-3-12-8-20-6-7-1-10-13-2-15-14-18-4-11-17

g. *Elitism*

Proses *elitism* ditambahkan karena kromosom terbaik dari generasi sebelumnya mungkin tidak termasuk dalam populasi baru atau nilai *fitness* kromosom terbaik telah menurun karena proses persilangan dan mutasi. Dalam *elitism*, kromosom terbaik dari evaluasi sebelumnya disalin untuk menggantikan kromosom dengan nilai *fitness* terendah dalam populasi baru. Berikut adalah nilai *fitness* populasi baru :

Tabel 3.8 Nilai *Fitness* Kromosom Populasi Baru

Kromosom	Total Jarak	<i>fitness</i>
1	130,2913	0,00767511
2	172,7382	0,00578911
3	150,1243	0,0066115
4	143,3452	0,00697617
5	124,3184	0,0080439
6	168,8343	0,00592297
7	160,5408	0,00622894
8	161,3478	0,00619779
9	152,7117	0,00654829
10	145,5479	0,00687059

Berdasarkan nilai *fitness* kromosom **Tabel 3.8**, terlihat bahwa kromosom ke-2 mempunyai nilai *fitness* terendah dan digantikan oleh kromosom terbaik yang disimpan dari evaluasi sebelumnya. Jadi, populasi baru yang dihasilkan adalah :

Tabel 3.9 Hasil Populasi Baru

Kromosom 1	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20
Kromosom 2	8-17-16-10-2-3-12-19-20-11-13-15-18-14-4-7-5-6-9-1
Kromosom 3	4-11-6-8-12-16-19-10-3-18-15-2-13-14-5-7-17-1-9-20
Kromosom 4	9-12-17-11-1-13-7-2-20-15-4-3-6-10-18-19-5-14-16-8
Kromosom 5	8-17-16-10-2-3-12-19-20-11-13-15-18-14-4-7-5-6-9-1
Kromosom 6	7-13-10-17-15-4-1-16-6-3-12-9-5-8-19-20-14-11-18-2
Kromosom 7	12-1-16-11-9-17-4-13-15-6-3-7-20-2-8-18-19-14-5-10
Kromosom 8	15-8-16-17-20-11-7-2-5-3-10-9-14-12-19-1-4-13-18-6
Kromosom 9	13-11-20-1-3-16-10-15-17-9-6-14-4-5-7-8-12-2-18-19
Kromosom 10	19-9-16-5-3-12-8-20-6-7-1-10-13-2-15-14-18-4-11-17

3.3.2. Perhitungan Algoritma Genetika Generasi Kedua

a. Membangkitkan Populasi Awal

Populasi awal yang digunakan pada generasi kedua adalah hasil populasi baru setelah proses *elitism*. Sehingga populasi yang didapatkan adalah :

Tabel 3.10 Populasi Awal Generasi Kedua

Kromosom 1	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20
Kromosom 2	8-17-16-10-2-3-12-19-20-11-13-15-18-14-4-7-5-6-9-1
Kromosom 3	4-11-6-8-12-16-19-10-3-18-15-2-13-14-5-7-17-1-9-20
Kromosom 4	9-12-17-11-1-13-7-2-20-15-4-3-6-10-18-19-5-14-16-8
Kromosom 5	8-17-16-10-2-3-12-19-20-11-13-15-18-14-4-7-5-6-9-1
Kromosom 6	7-13-10-17-15-4-1-16-6-3-12-9-5-8-19-20-14-11-18-2

Kromosom 7	12-1-16-11-9-17-4-13-15-6-3-7-20-2-8-18-19-14-5-10
Kromosom 8	15-8-16-17-20-11-7-2-5-3-10-9-14-12-19-1-4-13-18-6
Kromosom 9	13-11-20-1-3-16-10-15-17-9-6-14-4-5-7-8-12-2-18-19
Kromosom 10	19-9-16-5-3-12-8-20-6-7-1-10-13-2-15-14-18-4-11-17

b. Menentukan Nilai *Fitness*

Hitung jarak rute kromosom berdasarkan **Tabel 3.10**, sehingga :

$$\begin{aligned} \text{Kromosom 1} &= (1 \rightarrow 2) + (2 \rightarrow 3) + (3 \rightarrow 4) + (4 \rightarrow 5) + (5 \rightarrow 6) + \\ &(6 \rightarrow 7) + (7 \rightarrow 8) + (8 \rightarrow 9) + \dots + (20 \rightarrow 1) \\ &= 130,2913 \end{aligned}$$

$$\begin{aligned} \text{Fitness Kromosom 1} &= \frac{1}{130,2913} \\ &= 0,00767 \end{aligned}$$

$$\begin{aligned} \text{Kromosom 2} &= (3 \rightarrow 5) + (5 \rightarrow 16) + (16 \rightarrow 17) + (17 \rightarrow 9) + (9 \rightarrow 2) \\ &+ (2 \rightarrow 8) + (8 \rightarrow 1) + (1 \rightarrow 13) + \dots + (19 \rightarrow 3) \\ &= 172,7382 \end{aligned}$$

$$\begin{aligned} \text{Fitness Kromosom 2} &= \frac{1}{172,7382} \\ &= 0,00578 \end{aligned}$$

Lakukan perhitungan nilai *fitness* semua kromosom lainnya sehingga *fitness* semua kromosom adalah :

Tabel 3.11 Nilai Kromosom Generasi Kedua

Kromosom	Total jarak	<i>Fitness</i>
1	130,2913	0,00767511
2	124,3184	0,00804386
3	150,1243	0,00666115
4	143,3452	0,00697617
5	124,3184	0,00804386
6	168,8343	0,00592297
7	160,5408	0,00622894

8	161,3478	0,00619779
9	152,7117	0,00654829
10	145,5479	0,00687059
Total <i>Fitness</i>		0,06916873

Kromosom 2 dan 5 mempunyai nilai *fitness* tertinggi adalah 0,008043 dan jarak rute yang didapatkan adalah 124,3184. Kemudian kromosom ini disimpan dan disalin langsung ke populasi berikutnya untuk proses *elitism*.

c. Proses Seleksi

Total *fitness* semua kromosom dalam populasi adalah 0,069168. Kemudian menghitung nilai *fitness* relatif pada tiap kromosomnya. Berikut perhitungannya :

- Mencari probabilitas *fitness* tiap kromosom (p_i) dengan menggunakan rumus dari **persamaan 2.2**.

$$p_i = \frac{f_i}{\sum_{i=1}^{10} f_i}$$

$$p_1 = \frac{0,00767}{0,06916} = 0,1109$$

$$p_2 = \frac{0,00804}{0,06916} = 0,1162$$

$$p_3 = \frac{0,00666}{0,06916} = 0,0963$$

$$p_4 = \frac{0,00697}{0,06916} = 0,1008$$

$$p_5 = \frac{0,00804}{0,06916} = 0,1162$$

$$p_6 = \frac{0,00592}{0,06916} = 0,0856$$

$$p_7 = \frac{0,00622}{0,06916} = 0,09005$$

$$p_8 = \frac{0,00619}{0,06916} = 0,0896$$

$$p_9 = \frac{0,00654}{0,06916} = 0,0946$$

$$p_{10} = \frac{0,00687}{0,06916} = 0,0993$$

Kemudian menghitung *fitness* kumulatif setiap *fitness* relatif yang telah diperoleh :

$$c_1 = 0,1109$$

$$c_2 = 0,1109 + 0,1162 = 0,2272$$

$$c_3 = 0,2272 + 0,0963 = 0,3235$$

$$c_4 = 0,3235 + 0,10085 = 0,4244$$

$$c_5 = 0,4244 + 0,1162 = 0,5407$$

$$c_6 = 0,5407 + 0,0856 = 0,6263$$

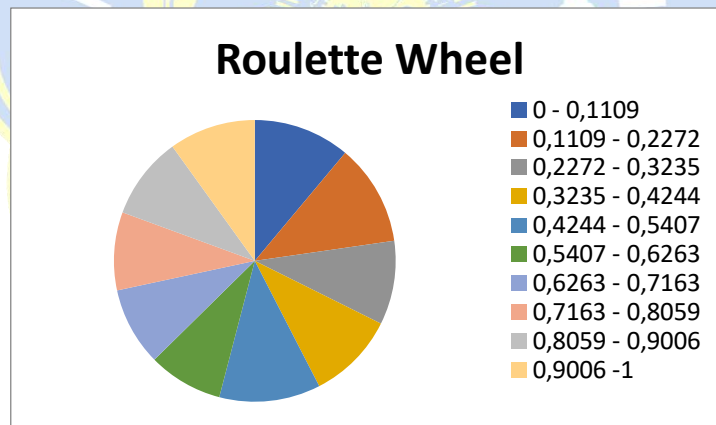
$$c_7 = 0,6263 + 0,09005 = 0,7163$$

$$c_8 = 0,7163 + 0,0896 = 0,8059$$

$$c_9 = 0,8059 + 0,0946 = 0,9006$$

$$c_{10} = 0,9006 + 0,0993 = 1$$

Dari hasil probabilitas kumulatif tersebut jika digambarkan dengan *Roulette Wheel* adalah sebagai berikut :



Gambar 3.6 Probabilitas Kumulatif dalam *Roulette Wheel* Generasi Kedua

Bilangan acak R dibangkitkan sebanyak jumlah kromosom dalam satu populasi yaitu 10 bilangan. Perintah untuk membangkitkan 10 bilangan acak pada excel dengan perintah =rand(). Hasilnya adalah :

$$\begin{aligned} R1 &= 0,6261 \\ R2 &= 0,3564 \\ R3 &= 0,2836 \\ R4 &= 0,4325 \end{aligned}$$

R5 = 0,7492
 R6 = 0,7995
 R7 = 0,9816
 R8 = 0,0739
 R9 = 0,8875
 R10 = 0,3268

Kromosom orangtua yang terpilih untuk proses reproduksi adalah :

Tabel 3.12 Kromosom Terpilih sebagai Induk Generasi Kedua

Induk	Ri	K. Terpilih	Susunan Gen
1	0,6261	K.6	7-13-10-17-15-4-1-16-6-3-12-9-5-8-19-20-14-11-18-2
2	0,3564	K.4	9-12-17-11-1-13-7-2-20-15-4-3-6-10-18-19-5-14-16-8
3	0,2836	K.3	4-11-6-8-12-16-19-10-3-18-15-2-13-14-5-7-17-1-9-20
4	0,4325	K.5	8-17-16-10-2-3-12-19-20-11-13-15-18-14-4-7-5-6-9-1
5	0,7492	K.8	15-8-16-17-20-11-7-2-5-3-10-9-14-12-19-1-4-13-18-6
6	0,7995	K.8	15-8-16-17-20-11-7-2-5-3-10-9-14-12-19-1-4-13-18-6
7	0,9816	K.10	19-9-16-5-3-12-8-20-6-7-1-10-13-2-15-14-18-4-11-17
8	0,0739	K.1	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20
9	0,8875	K.9	13-11-20-1-3-16-10-15-17-9-6-14-4-5-7-8-12-2-18-19
10	0,3268	K.4	9-12-17-11-1-13-7-2-20-15-4-3-6-10-18-19-5-14-16-8

d. Crossover

Hasil bilangan acak yang dihasilkan adalah :

R1 = 0,6652
 R2 = 0,8926
 R3 = 0,8043
R4 = 0,2405
 R5 = 0,8676

Di antara bilangan acak yang dihasilkan terdapat nilai yang lebih kecil dari nilai P_c yaitu R_4 . Pasangan pertama yang berpotongan adalah *parent 3* dan *parent 9* sisanya masuk ke tahap berikutnya tanpa pindah silang. Berikut proses *order crossover* untuk pasangan orangtua terpilih :

- a. Salin gen yang dipilih dari *parent* ke *child*. Susunan gen yang telah dipilih disalin langsung ke offspring. Misal pada *parent 1* akan disalin ke *child 1* dan *parent 2* akan disalin ke *child 2*.

Parent 1

4	11	6	8	12	16	19	10	3	18
15	2	13	14	5	7	17	1	9	20

Parent 2

13	11	20	1	3	16	10	15	17	9
6	14	4	5	7	8	12	2	18	19

Child 1

4	11	6	8						
					7	17	1	9	20

Child 2

13	11	20	1						
					8	12	1	18	19

- b. Salin gen yang lain. Pada *parent* pasangannya, hilangkan gen-gen yang sudah disalin dan salin sisa gen ke *child* sesuai dengan urutan.

Parent 2

13				3	16	10	15		
	14		5			12	2	18	19

Child 1

4	11	6	8						
					7	17	1	9	20

Salin sisa gen dari *parent 2* ke *child 1*

4	11	6	8	13	3	16	10	15	14
5	12	2	18	19	7	17	1	9	20

Parent 1

4		6			16		10	3	
15			14	5	7	17		9	

Child 2

13	11	20	1						
					8	12	1	18	19

Salin sisa gen dari *parent 1* ke *child 2*

13	11	20	1	4	6	16	10	3	15
14	5	7	17	9	8	12	1	18	19

Setelah pindah silang dilakukan, maka akan mendapatkan kromosom baru, yaitu *child 1* dan *child 2*.

Tabel 3.13 Hasil Pindah Silang Generasi Kedua

Induk 1	7-13-10-17-15-4-1-16-6-3-12-9-5-8-19-20-14-11-18-2
Induk 2	9-12-17-11-1-13-7-2-20-15-4-3-6-10-18-19-5-14-16-8
Offspring 3	4-11-6-8-13-3-16-10-15-14-5-12-2-18-19-7-17-1-9-20
Induk 4	8-17-16-10-2-3-12-19-20-11-13-15-18-14-4-7-5-6-9-1
Induk 5	15-8-16-17-20-11-7-2-5-3-10-9-14-12-19-1-4-13-18-6
Induk 6	15-8-16-17-20-11-7-2-5-3-10-9-14-12-19-1-4-13-18-6

Induk 7	19-9-16-5-3-12-8-20-6-7-1-10-13-2-15-14-18-4-11-17
Induk 8	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20
Offspring 9	13-11-20-1-4-6-16-10-3-15-14-5-7-17-9-8-12-2-18-19
Induk 10	9-12-17-11-1-13-7-2-20-15-4-3-6-10-18-19-5-14-16-8

e. Mutasi

Nilai probabilitas mutasi (P_m) adalah 0,1. Lalu dihasilkan bilangan acak menggunakan excel dengan perintah `=rand()` R_i antara 0 sampai 1 sebanyak jumlah populasi. Misal bilangan acak yang telah dihasilkan adalah :

- R1 = 0,3994
- R2 = 0,3319
- R3 = 0,02367**
- R4 = 0,8316
- R5 = 0,9489
- R6 = 0,4931
- R7 = 0,895
- R8 = 0,8388
- R9 = 0,3258
- R10 = 0,5723

Berdasarkan bilangan acak yang dihasilkan, diketahui R3 saja yang mempunyai nilai lebih kecil dari P_m , sehingga kromosom ke-3 saja yang dipilih untuk mutasi sebagai berikut :

4	11	6	8	13	3	16	10	15	4
5	12	2	18	19	7	17	1	9	20

Akan dipilih gen yang akan dimutasi yaitu gen ke-4 dan gen ke-5. Sebagai contoh gen akan melakukan pembalikan dalam kromosom tersebut akan menjadi seperti berikut.

4	11	6	13	8	3	16	10	15	4
5	12	2	18	19	7	17	1	9	20

Terlihat pada perubahan bahwa gen ke-4 saat ini menempati urutan gen ke-5 dan gen ke-5 saat ini menempati urutan gen ke-4. Setelah dilakukan mutasi, kromosom-kromosom yang tidak mengalami mutasi membentuk populasi baru. Populasi baru yang dihasilkan dari mutasi adalah :

Tabel 3.14 Populasi Baru Hasil Mutasi Generasi Kedua

Kromosom 1	7-13-10-17-15-4-1-16-6-3-12-9-5-8-19-20-14-11-18-2
Kromosom 2	9-12-17-11-1-13-7-2-20-15-4-3-6-10-18-19-5-14-16-8
Kromosom 3	4-11-6-13-8-3-16-10-15-14-5-12-2-18-19-7-17-1-9-20
Kromosom 4	8-17-16-10-2-3-12-19-20-11-13-15-18-14-4-7-5-6-9-1
Kromosom 5	15-8-16-17-20-11-7-2-5-3-10-9-14-12-19-1-4-13-18-6
Kromosom 6	15-8-16-17-20-11-7-2-5-3-10-9-14-12-19-1-4-13-18-6
Kromosom 7	19-9-16-5-3-12-8-20-6-7-1-10-13-2-15-14-18-4-11-17
Kromosom 8	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20
Kromosom 9	13-11-20-1-4-6-16-10-3-15-14-5-7-17-9-8-12-2-18-19
Kromosom 10	9-12-17-11-1-13-7-2-20-15-4-3-6-10-18-19-5-14-16-8

f. Elitism

Proses *elitism* ditambahkan karena kromosom terbaik dari generasi sebelumnya mungkin tidak masuk dalam populasi baru atau nilai *fitness* kromosom terbaik sudah menurun karena persilangan dan mutasi. Kromosom paling baik dari evaluasi sebelumnya disalin untuk menggantikan kromosom dengan nilai *fitness* terendah dalam populasi baru. Berikut adalah nilai *fitness* populasi baru :

Tabel 3.15 Nilai *Fitness* Kromosom Populasi Baru Generasi Kedua

Kromosom	Total Jarak	<i>fitness</i>	
1	168,8343	0,005922967	
2	143,3452	0,006976167	
3	171,7516	0,005822361	0,008043
4	124,3184	0,008043862	
5	161,3478	0,006197791	
6	161,3478	0,006197791	
7	145,5479	0,00687059	
8	130,2913	0,00767511	
9	151,0467	0,006620468	
10	143,3452	0,006976167	

Tabel 3.15 menunjukkan kromosom ke-3 mempunyai nilai *fitness* terendah dan digantikan oleh kromosom terbaik yang disimpan dari evaluasi sebelumnya. Jadi, populasi baru yang dihasilkan adalah :

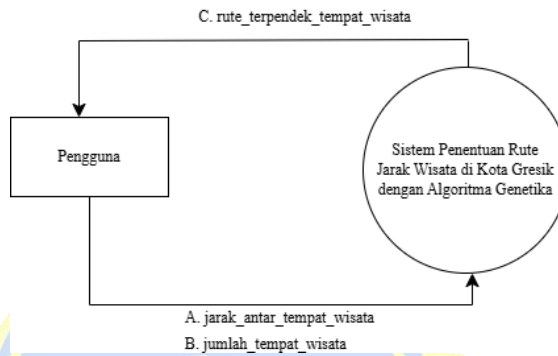
Tabel 3.16 Hasil Populasi Baru Generasi Kedua

Kromosom 1	7-13-10-17-15-4-1-16-6-3-12-9-5-8-19-20-14-11-18-2
Kromosom 2	9-12-17-11-1-13-7-2-20-15-4-3-6-10-18-19-5-14-16-8
Kromosom 3	8-17-16-10-2-3-12-19-20-11-13-15-18-14-4-7-5-6-9-1
Kromosom 4	8-17-16-10-2-3-12-19-20-11-13-15-18-14-4-7-5-6-9-1
Kromosom 5	15-8-16-17-20-11-7-2-5-3-10-9-14-12-19-1-4-13-18-6
Kromosom 6	15-8-16-17-20-11-7-2-5-3-10-9-14-12-19-1-4-13-18-6
Kromosom 7	19-9-16-5-3-12-8-20-6-7-1-10-13-2-15-14-18-4-11-17
Kromosom 8	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20
Kromosom 9	13-11-20-1-4-6-16-10-3-15-14-5-7-17-9-8-12-2-18-19
Kromosom 10	9-12-17-11-1-13-7-2-20-15-4-3-6-10-18-19-5-14-16-8

3.4. PERANCANGAN SISTEM

3.4.1. Diagram Konteks Sistem

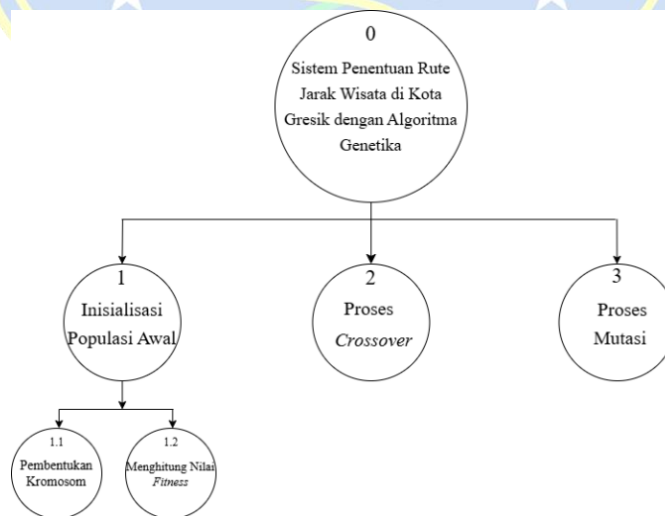
Diagram ini menunjukkan entitas di luar atau terlibat dalam sistem, termasuk aktivitas dari data yang menghubungkan entitas sistem.



Gambar 3.7 Diagram Konteks Sistem Penentuan Rute Jarak Wisata di Kota Gresik

Gambar 3.7 menggambarkan bagaimana Sistem Penentuan Jarak Wisata Kota Gresik bekerja. Pengguna adalah bagian dari sistem tersebut. Dimulai dengan memasukkan jarak dan jumlah tempat wisata. Selanjutnya akan didapatkan hasil rute terpendek antar tempat wisata.

3.4.2. Diagram Jenjang Sistem



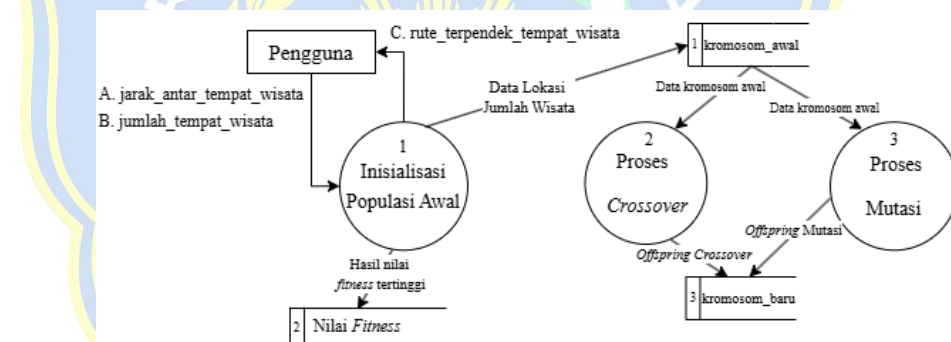
Gambar 3.8 Diagram Jenjang Sistem Penentuan Rute Jarak Wisata di Kota Gresik

Pada **Gambar 3.8** menggambarkan diagram berjenjang sistem yang terdiri dari:

1. Top level : Sistem Penentuan Rute Jarak Wisata di Kota Gresik dengan Algoritma Genetika
2. Level 1 : Hasil dari pembagian proses aplikasi sistem menjadi beberapa subproses yang meliputi Inisialisasi Populasi Awal, proses *Crossover* dan proses Mutasi.
3. Level 2 : Bagian dari proses yang ada pada Level 1 Inisialisasi Populasi mencakup langkah-langkah yaitu pembentukan kromosom dan menghitung nilai *fitness* .

3.4.3. Data Flow Diagram (DFD) Sistem

3.4.3.1. DFD Level 0 Pada Sistem Penentuan Rute Jarak Wisata di Kota Gresik

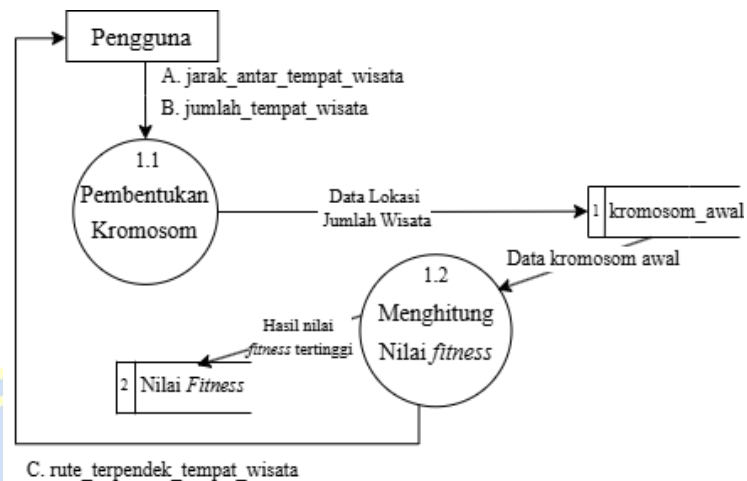


Gambar 3.9 DFD Level 0 Pada Sistem Penentuan Rute Jarak Wisata di Kota Gresik

Aktivitas DFD level 0 terdapat beberapa proses, seperti melakukan inisialisasi populasi awal yang menghasilkan kromosom awal dan nilai *fitness*. Inisialisasi populasi awal dengan data lokasi jumlah wisata untuk kromosom awal kemudian menghitung nilai *fitness* sampai dihasilkan nilai *fitness* tertinggi, data lokasi jumlah wisata akan menghasilkan kromosom awal, kemudian data kromosom awal tersebut akan dipakai di proses *crossover* dan mutasi sampai

dihasilkan *Offspring* dari masing-masing proses dan membentuk kromosom baru.

3.4.3.2. DFD Level 1 Pada Pada Sistem Penentuan Rute Jarak Wisata di Kota Gresik



Gambar 3.10 DFD Level 1 Pada Sistem Penentuan Rute Jarak Wisata di Kota Gresik

Aktivitas DFD level 1 pada Sistem Penentuan Rute Jarak Wisata di Kota Gresik dengan Algoritma Genetika terdapat beberapa proses antara lain yaitu pembentukan kromosom, menghitung nilai *fitness*. Pembentukan kromosom dengan data lokasi jumlah wisata untuk kromosom awal kemudian menghitung nilai *fitness* sampai dihasilkan nilai *fitness* tertinggi.

3.5. PERANCANGAN BASIS DATA

ERD dapat dilihat sebagai tempat penyimpanan data pada Sistem Penentuan Rute Jarak Wisata di Kota Gresik, dibutuhkan adanya sebuah database yang terdiri dari tabel user, lokasi dan parameter.

3.5.1. Tabel User

Tabel user menyimpan data pengguna dan memungkinkan pengguna mengakses aplikasi.

Tabel 3.17 Tabel User

#	Name	Type	Length	Key
1	id	int	20	PK
2	name	varchar	191	
3	email	varchar	191	
4	password	varchar	191	

3.5.2. Tabel Lokasi

Tabel Lokasi berisikan data lokasi tempat wisata di Kota Gresik.

Tabel 3.18 Tabel Lokasi

#	Name	Type	Length	Key
1	id	int	11	PK
2	kode	varchar	50	
3	nama_lokasi	varchar	255	
4	latitude	varchar	255	
5	longitude	varchar	255	
6	jarak	varchar	50	
7	url_image	varchar	255	
8	desc_image	text		

3.5.3. Tabel Parameter

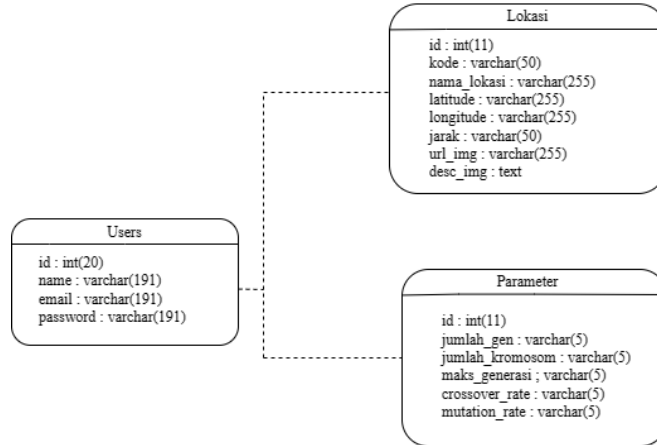
Tabel parameter adalah tabel yang berisi input proses perhitungan Algoritma Genetika.

Tabel 3.19 Tabel Parameter

#	Name	Type	Length	Key
1	id	int	11	PK
2	jumlah_gen	varchar	5	
3	jumlah_kromosom	varchar	5	
4	maks_generasi	varchar	5	
5	crossover_rate	varchar	5	
6	mutation_rate	varchar	5	

Sistem ini memiliki *database* yang menyimpan data-data yang diperlukan. Tabel user berhubungan dengan tabel lokasi dan tabel

parameter, tipe relasi dari tabel user ke tabel lokasi dan tabel parameter adalah *one to many*.

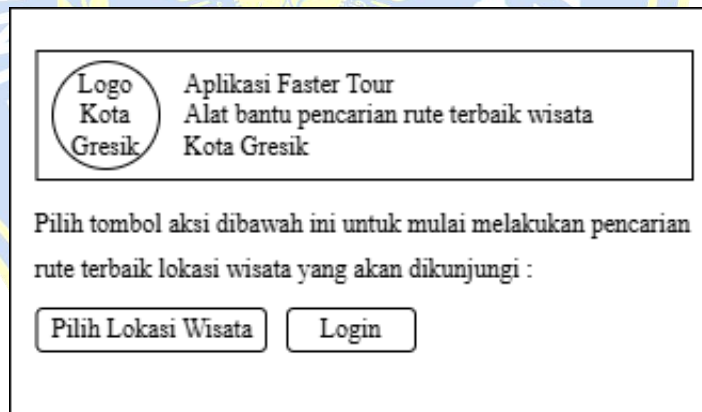


Gambar 3.11 ERD Sistem Penentuan Rute Jarak Wisata di Kota Gresik

3.6. PERANCANGAN ANTARMUKA SISTEM

3.6.1. Halaman Tampilan Utama

Halaman ini merupakan tampilan utama sistem dimana pengguna diharuskan memilih pilihan sesuai dengan yang diinginkan.

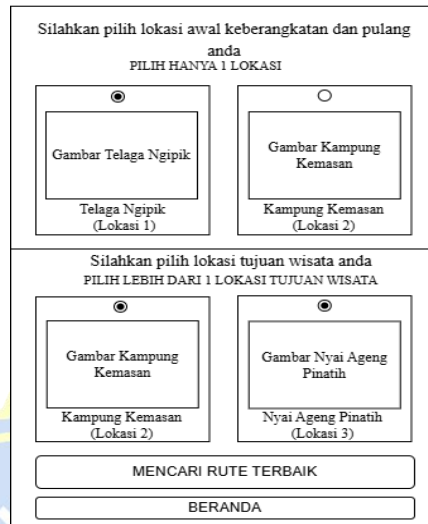


Gambar 3.12 Halaman Tampilan Utama

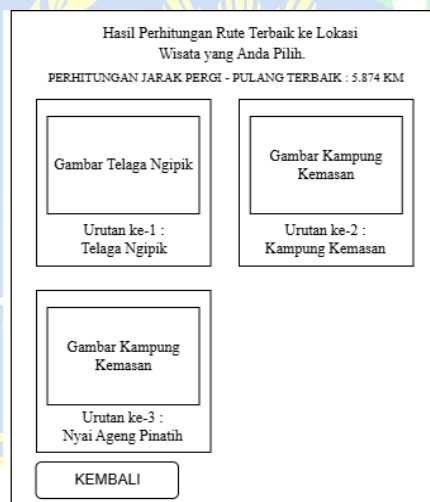
3.6.2. Halaman Pilih Lokasi Wisata

Halaman ini berisikan pemilihan lokasi tempat awal sampai tujuan wisata oleh pengguna. Pada **Gambar 3.13** pengguna memilih hanya 1 lokasi awal keberangkatan dan pulang tempat lokasi wisata, setelah itu memilih tujuan wisata lebih dari 1 lokasi yang ingin dituju. Jika semua telah

terisi klik “Mencari Rute Terbaik” maka akan muncul tampilan seperti pada **Gambar 3.14** yang berisi tampilan hasil perhitungan jarak pergi – pulang terbaik beserta gambar dan urutan tempat wisata yang ingin dituju.



Gambar 3.13 Halaman Pilih Lokasi Wisata

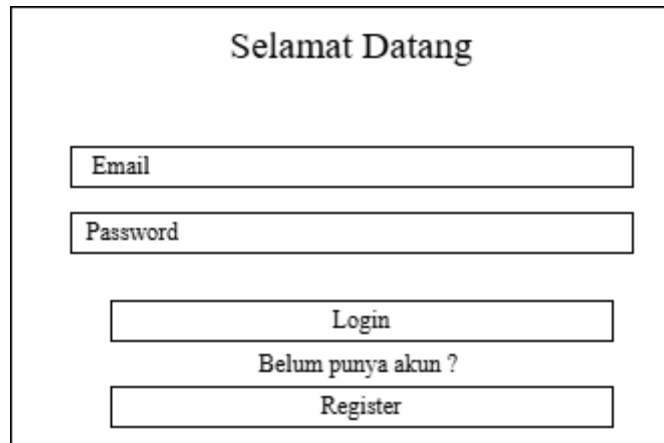


Gambar 3.14 Hasil Pilih Lokasi Wisata

3.6.3. Halaman *Login*

Halaman login digunakan apabila pengguna ingin menambahkan lokasi yang belum ada pada halaman pilih lokasi wisata yang dituju. Pengguna harus masukkan *username* dan *password* yang tepat. Jika informasi yang masuk tidak cocok dengan data yang tersimpan, sistem akan

memberikan peringatan yang menyatakan bahwa *username* dan *password* yang dimasukkan salah. Apabila pengguna sebelumnya tidak memiliki akun maka bisa klik “Register” untuk membuat akun.



Selamat Datang

Email

Password

Login

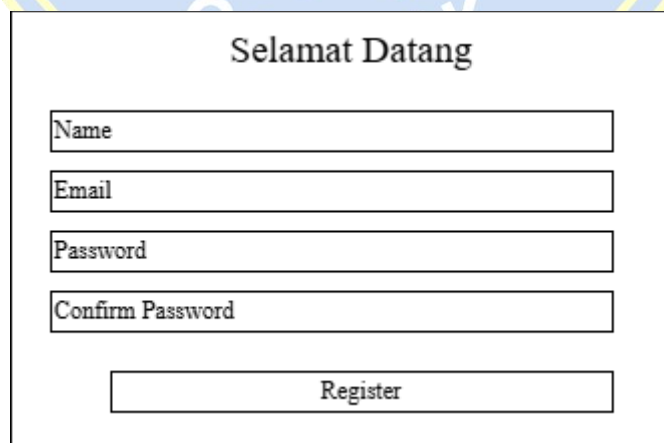
Belum punya akun ?

Register

Gambar 3.15 Halaman Login

3.6.4. Halaman Registrasi

Halaman registrasi merupakan halaman pendaftaran akun agar pengguna dapat masuk ke sistem apabila belum memiliki akun. Pengguna harus memasukkan nama, *username*, *password* yang sesuai agar dapat mendaftarkan akun ke sistem. Apabila pengguna memasukkan *username* yang sudah digunakan, maka sistem akan memberi peringatan bahwa *username* telah digunakan, silahkan gunakan *username* lain.



Selamat Datang

Name

Email

Password

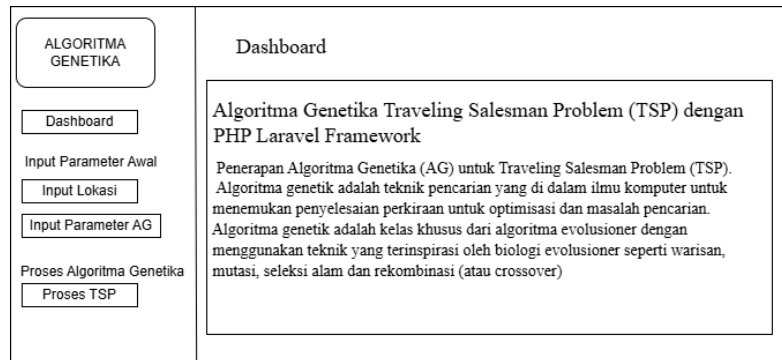
Confirm Password

Register

Gambar 3.16 Halaman Registrasi

3.6.5. Halaman Dashboard

Halaman dashboard merupakan tampilan awal setelah pengguna berhasil masuk ke dalam sistem. Halaman ini berisikan informasi atau penjelasan mengenai pengertian dan komponen dalam Algoritma Genetika *Traveling Salesman Problem*.



Gambar 3.17 Halaman Dashboard

3.6.6. Halaman Input Lokasi

Halaman input lokasi digunakan untuk menambahkan data lokasi ke sistem. Input data lokasi dapat dimasukkan secara manual. Mulai dari input kode lokasi, nama lokasi, upload gambar dan apabila lokasi tersebut merupakan lokasi awal maka pilih Ya atau sebaliknya. Untuk latitude dan longitude sudah terisi otomatis apabila kita meletakkan lokasi melalui Maps atau satelite, setelah semua sudah terisi klik “Simpan” maka hasil inputan lokasi akan muncul seperti pada **Gambar 3.19**. Ada fitur edit dan hapus yang digunakan untuk mengedit dan menghapus seluruh data yang masuk kedalam sistem. TSP juga memiliki tombol proses, yang digunakan untuk menjalankan algoritma genetika. Tombol ini hanya dapat digunakan setelah memasukkan data lokasi jika tidak dilakukan, tombol tersebut tidak akan berfungsi.

Gambar 3.18 Halaman Input Lokasi

Gambar 3.19 Halaman Daftar Lokasi

3.6.7. Halaman Input Parameter

Halaman input parameter digunakan untuk menambahkan inputan parameter dari algoritma genetika. Jumlah gen dibangkitkan akan terisi otomatis sesuai dengan jumlah inputan lokasi sedangkan jumlah kromosom, jumlah generasi dibangkitkan, *crossover rate* dan *mutation rate* diisi secara manual. Setelah semua parameter sudah terisi maka tombol simpan bisa digunakan.

<p>ALGORITMA GENETIKA</p> <p>Input Parameter Awal</p> <p>Input Lokasi</p> <p>Input Parameter AG</p> <p>Proses Algoritma Genetika</p> <p>Proses TSP</p>	<p>Input Parameter Algoritma Genetika</p>	
	<p>Jumlah Gen Dibangkitkan (Jumlah Lokasi)</p> <input type="text"/>	<p>KETERANGAN</p> <p>Membangkitkan populasi awal adalah proses membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu.</p> <p>Jumlah maksimal kromosom yang dibangkitkan dalam satu generasi</p> <p>Jumlah maksimal generasi yang dibangkitkan dalam satu populasi.</p> <p>Crossover (perkawinan silang) merupakan proses mengkombinasikan dua individu untuk memperoleh individu-individu baru yang diharapkan mempunyai fitness lebih baik.</p> <p>Mutation gen adalah proses penggantian gen dengan nilai inversinya, gen 0 menjadi 1 dan gen 1 menjadi 0. Proses ini dilakukan secara acak pada posisi gen tertentu pada individu-individu yang terpilih untuk dimutasikan.</p>
	<p>Jumlah Kromosom Dibangkitkan (Jumlah Populasi dalam Generasi)</p> <input type="text"/> <p>Silahkan masukan antara 1 - 100</p>	
	<p>Jumlah Generasi Dibangkitkan</p> <input type="text"/> <p>Silahkan masukan antara 1 - 1000</p>	
	<p>% Crossover Rate</p> <input type="text"/> <p>Silahkan masukan antara 1 - 100</p>	
	<p>% Mutation Rate</p> <input type="text"/> <p>Silahkan masukan antara 1 - 100</p>	
<p>Silahkan masukan antara 1 - 100</p> <p>Simpan</p>		

Gambar 3.20 Halaman Input Parameter

3.6.8. Halaman Proses TSP

Halaman ini untuk memulai proses TSP. Hasil dari input parameter akan muncul pada proses TSP mulai dari lokasi awal, jumlah lokasi sampai dengan *mutation rate*. Klik tombol "Lakukan Perhitungan Algoritma Genetika TSP" akan muncul hasil perhitungan seperti *initial distance*, *fitness* dengan satuan Km dan rute terbaik.

<p>ALGORITMA GENETIKA</p> <p>Input Parameter Awal</p> <p>Input Lokasi</p> <p>Input Parameter AG</p> <p>Proses Algoritma Genetika</p> <p>Proses TSP</p>	<p>Proses Traveling Salesman Problem</p>	
	<p>Lokasi Awal Telaga Ngipik (Lokasi 1)</p>	<p>HASIL PERHITUNGAN</p>
	<p>Jumlah Lokasi 20 Titik</p>	<p>Initial Distance 152.3477 Km</p>
	<p>Jumlah Populasi dalam Generasi 100 kali</p>	<p>Fitness (Final Distance) 118.4453 Km</p>
	<p>Jumlah Iterasi (Generasi Dibangkitkan) 100 kali</p>	<p>Rute Terbaik :</p> <p>Lokasi 9 - Lokasi 12 - Lokasi 17 - Lokasi 11 - Lokasi 1 - Lokasi 2 - Lokasi 3 - Lokasi 4 - Lokasi 6 - Lokasi 7 - Lokasi 10 - Lokasi 13 - Lokasi 15 - Lokasi 18 - Lokasi 20 - Lokasi 19 - Lokasi 5 - Lokasi 14 - Lokasi 16 - Lokasi 8</p>
	<p>% Crossover Rate 50 %</p>	
<p>% Mutation Rate 1%</p>		
<p>Lakukan Perhitungan Algoritma Genetika TSP</p>		

Gambar 3.21 Halaman Proses TSP Menggunakan Algoritma Genetika

3.7. PERENCANAAN PENGUJIAN SISTEM

Pada perencanaan pengujian sistem yaitu untuk mencari nilai *fitness* terbaik dengan menggunakan cara sebagai berikut :

