

**LAPORAN MINI PROJECT**  
**KASUS 2 : SIMETRIC TRAVELLING SALESMAN PROBLEM WITH**  
**GENETIC ALGORITHM**



Mata Kuliah : Teknik Optimalisasi

Dosen Pengampun :  
Hindriyanto Dwi Purnomo, ST., MIT., Ph.D

Anggota :

Andrew C Handoko ( 672019250 )

Arya Damar Pratama ( 672019227 )

Deffa Ferdian Alif Utama ( 672019163 )

**TEKNIK INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS KRISTEN SATYA WACANA**  
**2022**

# DAFTAR ISI

<b>DAFTAR ISI</b>	<b>1</b>
<b>METODE METAHEURISTIK</b>	<b>2</b>
<b>HASIL EKSPERIMEN DAN ANALISA</b>	<b>3</b>
1.1 Dataset berlin52	3
1.2 Dataset kroA100	5
1.3 Dataset ch150	7

## METODE METAHEURISTIK

Kelompok kami mengambil kasus 2 sebagai eksperimen, dimana metode metaheuristik yang digunakan adalah metode algoritma genetik. Algoritma genetik merupakan salah satu metode metaheuristik yang menggunakan metafora proses evolusi dari makhluk hidup, terkhusus teori evolusi Darwin. Metode ini diperkenalkan oleh John Hollan (1975) dan selanjutnya dikembangkan oleh David. E Goldberg (1989). Algoritma ini kami pilih karena memiliki performa yang baik untuk berbagai macam jenis permasalahan optimisasi.

Algoritma genetik ini akan kami gunakan untuk melakukan optimisasi pada permasalahan TSP (*Travelling Salesman Problem*). Dataset yang kami gunakan diperoleh dari internet, dengan nama dataset yaitu berlin52, ch150 dan kroA100. Tiap dataset akan dicari masing-masing nilai optimumnya. Proses optimisasi ini kami lakukan dengan menggunakan bantuan *compiler* berupa *Goggle Collaboratory* dan menggunakan bahasa pemrograman *Python*.

Tahapan awal optimisasi yang kami lakukan yaitu dengan mengimpor beberapa *library* seperti *numpy*, *random* dan *matplotlib*. Lalu setelah itu dataset yang kami peroleh akan kami masukkan ke dalam *collab* untuk dibaca isinya. Dataset ini memiliki format .tsp, sehingga harus diubah dahulu. Kami mengubah isi dari file .tsp masing-masing dataset menjadi sebuah array. Lalu dari array tersebut akan diurutkan ke bawah sesuai arraynya, agar data dapat terlihat.

Langkah selanjutnya adalah melakukan inisialisasi populasi pada masing-masing dataset. Dimana tiap dataset kami beri populasi awal dengan jumlah yang berbeda, yaitu berlin52 dengan inisialisasi populasi 10, lalu chh150 dan kroA100 dengan inisialisasi 50. Dimana nantinya individu yang bangkit dari populasi tersebut akan dijabarkan sesuai no kotanya. Inisialisasi ini dilakukan secara random, dimana individu yang dibangkitkan akan berbeda-beda.

Lalu membuat fungsi objektif untuk mengukur jarak euclidean. Jarak euclidean ini akan menghitung total jarak dari tiap individu yang dibangkitkan di tiap populasi. Dari situ juga nilai fitness dari tiap individu akan didapat dari evaluasi yang dilakukan. Nilai fitness ini digunakan untuk melihat kualitas dari tiap individu yang bangkit. Setelah itu akan dilakukan seleksi individu yang akan digunakan sebagai induk. Seleksi yang dilakukan ini menggunakan metode turnamen. Dimana induk yang dipilih hanya 2 saja, dan induk tersebut juga dipilih secara acak.

Lalu akan masuk ke dalam proses reproduksi, dimana pada proses ini ada 2 operasi utama yang akan dilakukan, yaitu *crossover* dan *mutasi*. Proses reproduksi pertama adalah *crossover*, pada operasi ini dipilih metode *order crossover* (ox) dan jumlah anak yang dihasilkan adalah 2. Dimana metode ini digunakan untuk mewariskan sifat baik dari induk kepada anaknya. Sehingga iterasi selanjutnya akan mendapat nilai yang lebih baik lagi. Selanjutnya akan dilakukan operasi mutasi dengan swap. Dimana individu yang dipilih harus sesuai dengan probabilitas mutasi yang ditentukan, yaitu 0,01. Dan indeks lokasi yang di swap pada 2 individu adalah sama.

Setelah didapatkan individu dengan nilai fitness dan rute terbaik pada iterasi tersebut, maka selanjutnya akan dilakukan *replacement*. *Replacement* ini digunakan untuk mengganti individu yang kurang baik pada populasi sebelumnya. Jenis yang dipakai adalah *steady-state*

Setelah itu akan masuk ke bagian kode untuk melakukan prosedur algoritma genetik. Pada bagian ini, akan digabungkan ke semua prosedur dari inisialisasi, seleksi, reproduksi dan elitism. Dimana akan dilakukan perulangan sebanyak maksimal 5, 10 dan 975 generasi sesuai dataset, yang pada akhirnya akan didapatkan nilai paling optimum dari fungsi tersebut.

```

[28 38 0 37 6 22 16 7 4 43 9 18 21 25 29 24 36 10 46 14 17 5 42 1
45 13 49 40 8 44 41 35 30 2 39 34 11 48 50 31 20 19 26 23 27 51 33 47
3 12 32 15]
[39 49 32 36 3 2 40 8 4 7 14 28 19 43 38 17 10 48 18 41 5 11 44 15
25 51 27 50 12 31 30 22 42 9 20 34 35 47 24 37 46 33 16 45 23 29 6 26
21 13 1 0]
[36 37 5 22 15 0 23 19 31 8 18 41 3 10 17 28 26 1 6 30 35 47 45 50
11 9 39 2 33 7 27 42 14 20 43 24 16 32 51 12 13 34 38 25 49 44 48 4
46 29 40 21]
[24 19 11 45 16 6 17 27 49 2 28 44 35 30 29 20 23 5 48 25 22 3 10 34
7 33 9 39 41 13 37 40 12 47 26 42 14 0 21 36 18 8 15 4 51 32 46 38
31 1 43 50]
[48 47 16 35 2 42 31 12 27 8 45 29 10 46 7 30 34 26 23 28 32 4 1 25
43 36 21 17 49 14 13 0 15 19 24 22 20 39 9 37 11 18 3 50 51 33 41 38
40 5 6 44]]
fitness populasi awal : [27962.86967612 30456.62549053 30830.44086736 27633.22701318
32831.24260459 28121.48448995 27575.41188164 25988.26593216
30261.01144201 30738.75112276]
fitness terbaik awal : 25988.26593216359
rute awal : [36 37 5 22 15 0 23 19 31 8 18 41 3 10 17 28 26 1 6 30 35 47 45 50
11 9 39 2 33 7 27 42 14 20 43 24 16 32 51 12 13 34 38 25 49 44 48 4
46 29 40 21]
fitness populasi akhir : [27575.41188164 27744.0075385 27765.45578492 27633.22701318
27912.07874254 27575.41188164 27575.41188164 25988.26593216
27575.41188164 27575.41188164]
fitness akhir : 25988.26593216359
rute akhir : [36 37 5 22 15 0 23 19 31 8 18 41 3 10 17 28 26 1 6 30 35 47 45 50
11 9 39 2 33 7 27 42 14 20 43 24 16 32 51 12 13 34 38 25 49 44 48 4
46 29 40 21]

```

**Gambar 1.2.** Hasil dari Berlin52

Untuk hasil run dari proses fungsi ini kami lakukan sebanyak 10 kali. Dimana tiap run menghasilkan nilai optimum atau nilai terbaik yang berbeda-beda. Berikut hasil dari tiap fungsi yang di run, beserta nilai rata-rata dari semuanya dan standar deviasinya.

No	Nilai Terbaik
1	25.988,26593216
2	27.827,85051514
3	26.734,33316777
4	27.016,42455418
5	27.152,09161882
6	26.316,20160480
7	27.136,82322958



```

162833.11384855 164814.3055898 163884.8399903 178471.9060168
171130.43225229 168308.30733083 175673.09653567 175852.50285543
173823.84259519 175446.66816472 150671.21859478 169713.3820521
162934.77946396 167650.1941644 ]
fitness terbaik awal : 149091.7789087042
route awal : [24 98 86 45 4 39 18 69 10 37 15 78 83 67 99 41 34 1 75 82 47 2 13 57
73 94 22 9 20 51 81 48 26 55 72 5 30 65 35 61 16 66 92 31 3 79 6 59
11 19 0 71 21 25 68 63 42 56 36 95 87 62 50 93 88 74 33 85 27 77 64 8
70 44 43 89 17 23 54 53 80 29 46 28 52 12 40 38 90 14 91 97 96 76 60 58
7 32 84 49]
fitness populasi akhir : [124200.00811086 124200.00811086 124200.00811086 124200.00811086
124200.00811086 124200.00811086 124200.00811086 124200.00811086
124200.00811086 124200.00811086 124200.00811086 124200.00811086
124200.00811086 124200.00811086 124200.00811086 124200.00811086
124200.00811086 124200.00811086 124200.00811086 124200.00811086
124200.00811086 124200.00811086 124200.00811086 124200.00811086
124200.00811086 124200.00811086 124200.00811086 124200.00811086
124200.00811086 124200.00811086 124200.00811086 124200.00811086
124200.00811086 124200.00811086 124200.00811086 124200.00811086
124200.00811086 124200.00811086]
fitness akhir : 124200.00811086455
route akhir : [74 69 52 93 34 50 51 16 10 37 60 5 83 67 99 41 1 75 82 47 2 13 57 97
94 22 54 26 92 62 73 3 66 89 72 43 95 46 17 87 58 53 25 0 23 91 48 32
39 7 84 49 12 42 4 38 79 14 27 36 31 21 96 30 29 80 59 19 55 35 71 56
77 28 40 70 8 6 63 68 88 64 98 86 18 15 78 9 20 81 24 45 33 61 44 76
85 11 90 65]

```

**Gambar 2.2.** Hasil dari KroA100

No	Nilai Terbaik
1	124.200,008110864
2	121.896,196623727
3	130.222,283718356
4	115.160,616979084
5	126.719,236402606
6	124.904,009270077
7	131.290,025631889
8	133.360,190402652
9	119.318,708501580
10	132.645,566490847
<b>Rata-rata</b>	125.971,68421317
<b>Standar Deviasi</b>	5727,752962

**Tabel 2.1.** Hasil dari KroA100

Dari Percobaan yang telah Kami lakukan, didapatkan bahwa hasil dari file kroA100 dengan ketentuan maksimum fungsi objektif sebanyak 2000 dan setiap fungsi dijalankan sebanyak 10 kali dengan mencari nilai terbaik, nilai rata-rata dan standar deviasi, pada Gambar dan Tabel diatas. Untuk Tahapannya yaitu :

1. Membaca dan menampilkan Data File
2. Menginisialisasikan isi Data File
3. Membuat fungsi objektif
4. Seleksi Populasi dengan metode Tournamen
5. mendapatkan Jumlah Anak yang dihasilkan
6. Melakukan Mutasi
7. Melakukan Proses Elitism
8. Prosedur Algoritma Genetik

Keterangan :

P0 = Populasi

Fitness Populasi awal

Fitness Terbaik awal

Rute awal

Fitness Populasi akhir

Fitness Terbaik akhir

Rute akhir

Analisa yang didapatkan,

Pada tahapan kedelapan, dari Populasi yang telah kita tentukan sebanyak 50 dan maksimal generasi sebanyak 975, diketahui Fitness Populasi awal, Fitness Terbaik awal dan Rute awal Menghasilkan Fitness Populasi akhir, Fitness Terbaik akhir dan Rute akhir nilai yang terbaik. Bisa Kita lihat pada **Gambar 3.1**. Nilai tersebut cukup kecil dari nilai awal, yang dimana semakin kecil atau sedikit, maka hasilnya akan terbaik. Lalu total hasil percobaan jalankan sebanyak 10 kali tersebut, kami catat untuk mencari Nilai rata rata dan Nilai Deviasinya. didapatkanlah hasil Nilai Rata-rata senilai 125.971,68421317 dan Standar Deviasinya senilai 5727,752962 pada **Tabel 3.1**.

### 1.3 Dataset ch150

Untuk hasil eksperimen yang pertama dengan ketentuan jumlah maksimum evaluasi fungsi objective adalah 2.000 dan yang kedua setiap fungsi di run 10 kali, kemudian tentukan nilai terbaik, nilai rata-rata dan standar deviasinya seperti berikut :



```

print("fitness akhir :",fitP.min())
print("rute akhir :",P[fitP.argmin()])

P0 : [[ 70  24  54 ... 128  72  18]
      [138 149 127 ...  55 134 110]
      [ 42  64   0 ...  26  57 149]
      ...
      [ 37  86 120 ...   0 135   8]
      [ 13 106 147 ...  64  34  16]
      [ 82  75  58 ...  24   7 138]]
fitness populasi awal : [52789.76280673 53725.78224121 54177.99675913 54147.80534886
50366.31091393 53529.22485545 53357.19500849 55527.24842987
55669.22127254 55151.37248175 54799.32102385 54338.95323386
50009.46557177 50190.70304468 55239.11422835 50704.13168236
51466.26561434 53394.73712085 52574.86678671 53965.11951878
52968.06061897 54509.74649231 56975.71196099 54629.98641057
52201.49328091 51909.89851669 55892.21111134 53443.59499091
51251.02342425 53918.65605716 52128.72356528 50381.7734086
52289.44492353 51642.56598083 53948.89701346 57096.42112883
58504.67141895 53001.70264836 56955.08518132 54337.88969142
53321.20422649 54085.20594983 52070.57996437 52630.34136631
53601.43203518 54506.78768183 53510.68982809 55770.44642715
52740.27107432 52353.87299964]
fitness terbaik awal : 50009.465571774715
rute awal : [121  23 144 135 147   3 139 128 141  47  85  54 136  76  97  15  51  31
  43  94 129  28 131  74  84   0  89  90  81  67   4  71  73  79  17  88
 116  13 115  41 109 111 125   7  96 134   5  40  44  42  35  34   8 117
 114 110 113  50  24 101  36 140  93  21  92  20   2 137  27  46  53  19
  70 122  86 102 149  98  72 148  14  33  12  10  57 112 130 138  59  55
 133 118  64  48 123  66  52  49  18 132  11  99 146 100 124  30  26  38
  62 108  45   6 104  78  25  83  91  87   1  39 142 119 127  29  60  56
 107 143   9 120  63  58 106 103  61  37 105  95  65 126  16  77  68  69
   75  22  32 145  80  82]
fitness populasi akhir : [52789.76280673 53725.78224121 54177.99675913 54147.80534886

```

**Gambar 3.1.** Hasil dari ch150

```

fitness terbaik awal : 50009.465571774715
rute awal : [121 23 144 135 147 3 139 128 141 47 85 54 136 76 97 15 51 31
43 94 129 28 131 74 84 0 89 90 81 67 4 71 73 79 17 88
116 13 115 41 109 111 125 7 96 134 5 40 44 42 35 34 8 117
114 110 113 50 24 101 36 140 93 21 92 20 2 137 27 46 53 19
70 122 86 102 149 98 72 148 14 33 12 10 57 112 130 138 59 55
133 118 64 48 123 66 52 49 18 132 11 99 146 100 124 30 26 38
62 108 45 6 104 78 25 83 91 87 1 39 142 119 127 29 60 56
107 143 9 120 63 58 106 103 61 37 105 95 65 126 16 77 68 69
75 22 32 145 80 82]
fitness populasi akhir : [52789.76280673 53725.78224121 54177.99675913 54147.80534886
50366.31091393 53529.22485545 53357.19500849 52741.64157747
52731.89414018 50799.76910402 52529.11291529 50157.11812058
50009.46557177 50190.70304468 53806.70568403 50704.13168236
51466.26561434 53394.73712085 52574.86678671 53965.11951878
52968.06061897 51118.11668344 52016.50918541 53634.29201004
52201.49328091 51909.89851669 51289.17563808 53443.59499091
51251.02342425 53918.65605716 52128.72356528 50381.7734086
52289.44492353 51642.56598083 53948.89701346 52311.80162065
50777.60133805 53001.70264836 52596.16694187 49649.69152296
53321.20422649 54085.20594983 52070.57996437 52630.34136631
53601.43203518 52292.64961348 53510.68982809 51202.97061365
52740.27107432 52353.87299964]
fitness akhir : 49649.69152295931
rute akhir : [ 40 36 54 32 117 129 11 18 65 37 49 136 5 20 48 14 146 145
69 45 19 56 85 17 47 107 119 71 99 106 94 33 66 128 64 121
90 134 8 112 31 139 58 116 79 144 88 61 80 21 4 110 142 93
147 16 111 143 115 3 104 43 62 108 57 81 123 97 9 109 138 141
13 70 60 50 101 126 63 2 102 52 87 125 23 114 51 77 6 120
67 124 25 73 27 113 29 44 39 135 149 100 53 140 22 30 86 55
78 96 127 98 137 92 26 35 46 76 1 103 118 38 0 7 105 72
75 130 89 122 12 91 41 131 133 74 28 84 42 34 132 68 95 83
82 10 148 24 59 15]

```

**Gambar 3.2.** Hasil dari ch150

Berdasarkan gambar tersebut diketahui populasi awalnya lalu kualitas awalnya tiap individunya yang memiliki nilai fitness terbaiknya adalah 50222.81219019731 yang merupakan rute awal nilai terbaiknya dari populasi awal kemudian diketahui nilai fitness populasi akhir yang nilai individunya berbeda dari nilai populasi awal. Didalam nilai fitness populasi akhir memiliki nilai terbaiknya adalah 49144,28101897165 sehingga dinilai tersebut lebih baik daripada nilai fitness terbaik awal sehingga menghasilkan rute yang bagus.

Untuk mencari nilai terbaiknya diambil dari nilai fitness akhir yaitu 49144,28 lalu untuk nilai rata-rata dan deviasinya didapat dengan mengambil beberapa nilai terbaiknya sebanyak 10 kali

Yang hasilnya run 10 x seperti tabel berikut :

No	Nilai Terbaik
1	49144,28101897165
2	49649,6915229593
3	49548,3969320703
4	49046,0369165041
5	49213,3054592156

<b>6</b>	50079,7740571782
<b>7</b>	48804,8631427115
<b>8</b>	49646,553908288
<b>9</b>	47088,4725704668
<b>10</b>	49037,7629808962
<b>Rata-rata</b>	49.125,91385093
<b>Standar Deviasi</b>	768,772817

**Tabel 3.1.** Hasil dari ch150

Dari hasil tersebut diketahui rata-ratanya adalah 49.125,91385093 berasal dari jumlah nilai terbaik seluruhnya dibagi 10 sedangkan deviasi diperoleh dari penyimpangan setiap data dari rata-ratanya.dengan cara mengurangi nilai rata-rata Kemudian, simpangan setiap nilai data kuadratkan lalu kita bagi dengan nilai rata-rata data. Nilai yang dihasilkan disebut varians. Kemudian akar kuadratkan nilai varian tersebut yang ditemukan hasil nilainya 768,772817.