

# Algoritma Evolutionary

*by* Tedy Setiadi

---

**Submission date:** 02-Nov-2020 05:45AM (UTC+0700)

**Submission ID:** 1432993152

**File name:** Aplikasi\_Algoritma\_Evolutionary\_dan\_Analisis\_Komputasinya.docx (444.89K)

**Word count:** 1794

**Character count:** 11657

# APLIKASI ALGORITMA EVOLUTIONARY DAN ANALISIS KOMPUTASINYA UNTUK PENYELESAIAN *TRAVELLING SALESMAN PROBLEM*

Tedy Setiadi<sup>1</sup>, Eka Satya Wijaya<sup>2</sup>

Program Studi Informatika Universitas Ahmad Dahlan, Yogyakarta

[tedz68@yahoo.com](mailto:tedz68@yahoo.com), [ekasty@yahoo.com](mailto:ekasty@yahoo.com)

## Abstrak

*Travelling Salesman Problem* (TSP) merupakan persoalan penentuan rute optimal dari beberapa kota yang dikunjungi wiraniaga tanpa harus mendatangi kota yang sama lebih dari sekali. Telah banyak penelitian yang dilakukan namun sampai saat ini solusinya masih memerlukan waktu eksponensial. Pada penelitian ini dikembangkan aplikasi algoritma evolutionary untuk solusi TSP serta analisis waktu komputasinya. Algoritma ini merupakan metode pencarian stokastik yang mencoba untuk menirukan sifat evolusi biologis dari makhluk hidup, dengan prinsip suatu individu yang baik akan bertahan dan menghasilkan individu-individu yang semakin baik pada setiap generasi. Perangkat lunak dikembangkan dengan Borland Delphi 6.0 serta MathLab untuk menganalisis waktu komputasinya. Fitur aplikasi yang dimiliki mampu mensimulasikan dan memvisualisasikan proses penemuan rute optimal serta waktu komputasinya. Hasil uji coba beberapa kasus dapat dihipotesiskan hubungan jumlah kota dengan waktu komputasinya adalah  $y = 2,00332x^2 - 55,9171x + 331,9833$  serta hubungan banyaknya kota terhadap jumlah generasi yang dibangkitkan adalah  $y = 0,0012x^2 - 0,1300x + 69,322$

**Kata Kunci:** rute optimal, algoritma *evolutionary*, analisis komputasi

## 1. Pendahuluan

Salah satu strategi pemasaran tradisional yang paling terkenal dalam dunia industri adalah penawaran produk melalui penjualan keliling (*travelling Salesman*) [2]. Persoalan untuk mendapatkan hasil yang optimal dalam strategi pemasaran tersebut disebut sebagai *travelling Salesman Problem* (TSP), yaitu pencarian rute terpendek yang bisa dilalui seorang yang ingin mengunjungi beberapa kota tanpa harus mendatangi kota yang sama lebih dari satu kali. Jika jumlah kota yang harus didatangi hanya sedikit, misalnya hanya ada 5 kota, permasalahan ini dapat dipecahkan dengan sangat mudah, bahkan tidak memerlukan komputer untuk menghitungnya. Tetapi, masalahnya jadi rumit jika kota lebih dari 20 kota yang harus didatangi. Ada begitu banyak kemungkinan yang harus dicoba dan diuji untuk menemukan jawabannya. [1]

Permasalahan TSP merupakan salah satu dari kelompok problem yang belum ada solusinya dan bahkan tidak mungkin diselesaikan meskipun dibantu oleh *super hyper* komputer yang tercanggih saat ini [3]. Dalam beberapa dasawarsa terakhir banyak yang berkembang penelitian yang berkaitan dengan TSP. Beberapa metode baru yang dapat diaplikasikan pada TSP antara lain adalah

metode *Simulated Annealing*, *Tabu Search*, *Genetik Algorithm* dan *Evolutionary Algorithm* [4]. Metode metode tersebut berkembang karena adanya tuntutan untuk mendapatkan urutan rute yang dapat ditempuh pada perjalanan wiraniaga sehingga dapat menghasilkan jarak tempuh yang efisien dan mendekati optimal dengan jangka waktu pencarian rute yang lebih singkat.

Salah satu metode yang dapat digunakan untuk solusi problem pencarian nilai optimal (optimasi) adalah algoritma *evolutionary*. Algoritma ini dikembangkan pertama kali oleh Burdett dan Kozan pada Tahun 2000 [1]. Algoritma ini merupakan metode pencarian stokastik yang mencoba untuk menirukan sifat evolusi biologis dari makhluk hidup. Algoritma *Evolutionary* bekerja dengan prinsip bahwa suatu individu yang baik akan bertahan dan menghasilkan individu-individu yang semakin baik pada setiap generasi.

## 2. Algoritma Evolutionary

Algoritma *Evolutionary* merupakan metode pencarian stokastik yang mencoba untuk menirukan sifat evolusi biologis dari makhluk hidup.

*Algorithm* bekerja dengan prinsip bahwa suatu individu yang baik akan bertahan dan menghasilkan individu-individu yang semakin baik pada setiap generasi. Elemen-elemen yang ada pada metode ini diambil dari proses alam seperti seleksi, kombinasi, mutasi, migrasi, dan perpindahan secara lokal.

## 2.1 Operator dalam Algoritma *Evolutionary*

Proses utama pada algoritma *evolutionary* yaitu proses perhitungan nilai fungsi tujuan dari permasalahan yang dihadapi yang juga disebut dengan *fitness value* [5]. Dengan adanya perhitungan *fitness value* maka akan terjadi proses perulangan hingga nilai yang diinginkan tercapai. Selain itu ada beberapa operator yang umumnya diterapkan dalam algoritma *evolutionary* untuk penyelesaian masalah optimasi, yang juga terdapat pada sifat evolusi biologis dari makhluk hidup, yaitu:

### 1. Seleksi

Seleksi berfungsi sebagai proses pemilihan individu dimana individu yang memiliki kemampuan untuk bertahan hidup (*fitness*) tinggi akan bertahan.

### 2. Kombinasi

Kombinasi berfungsi sebagai proses pertukaran yang akan menghasilkan individu baru yang berbeda dari induknya.

### 3. Mutasi

Mutasi berperan dalam melakukan perubahan (*genotype*) sifat dari suatu individu karena proses pertukaran pada individu tersebut yang memungkinkan makhluk hidup melakukan penyesuaian dengan lingkungan barunya tidak sesuai dengan induknya semula

Algoritma ini berbeda dengan metode optimasi klasik yang lainnya dalam beberapa hal, antara lain:

- Metode ini merupakan metode *nondeterministic* yang akan menghasilkan penyelesaian-penyelesaian yang berbeda meskipun model awalnya tidak dirubah, dikarenakan adanya pemakaian *random sampling* dalam algoritma ini.

- Algoritma ini mempunyai populasi yang berisi calon-calon penyelesain.
- Dalam pengaplikasiannya, *Evolutionary Algorithm* mencoba untuk menggabungkan elemen-elemen dari solusi-solusi yang telah ada untuk menciptakan solusi baru dengan mewarisi ciri-ciri yang dipunyai oleh tiap orangtua.

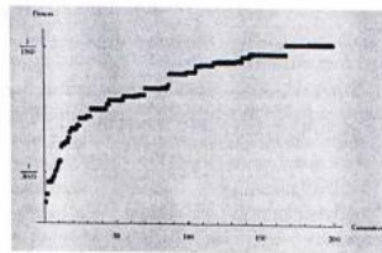
## 2.2 Kinerja Algoritma *Evolutionary*

Secara teoritis, dalam algoritma *evolutionary*, semakin banyak jumlah generasi yang berevolusi pada sebuah populasi akan meningkatkan nilai *fitness*. Dan dengan adanya peningkatan nilai *fitness* yang semakin tinggi, maka akan didapatkan solusi pemecahan TSP yang semakin optimal.

Misalkan dalam sebuah populasi dari 15 individu, masing-masing memproduksi dua keturunan yang termutasi setiap generasinya, dan dengan cepat mengalami peningkatan nilai *fitness*, dimana hal itu menggambarkan panjang jarak yang ditempuh dalam sebuah perjalanan keliling berkurang dari 3000 menjadi 1500 dalam 200 generasi yang dihasilkan dari individu yang berevolusi [1].

$$\text{Nilai fitness } (f) = 1 / (g) \quad (1)$$

*Genotip* (*g*) = panjang jarak tempuh dari setiap hasil pencarian.



**Gambar 1.** Proses peningkatan nilai *fitness* (kemampuan untuk bertahan hidup) dalam sebuah populasi.

Dari gambar 1 dapat dilihat nilai *fitness* yang terkadang tetap dan diselingi dengan lompatan kenaikan yang mendadak, hal ini merupakan karakteristik dari algoritma *evolutionary*.

## 3. Model Penyelesaian

### 3.1. Individu

individu dinyatakan dengan rangkaian urutan rute pada setiap perjalanan. Untuk kasus ini dapat dimasukkan titik-titik sebagai jumlah Kota yang dikunjungi sebanyak mungkin. Misalkan pada suatu perjalanan urutan susunan perjalanannya adalah {1,2,3,10,7,9,8,4,6,5,1}.

### 3.2. Solusi Awal

Solusi awal dibangun dengan mencari urutan rute secara random. Setiap urutan rute yang terbentuk, dikelompokkan dalam suatu populasi dan diurutkan sesuai dengan nilai panjang jarak tempuh masing-masing urutan rute perjalanan.

### 3.3. Seleksi, Kombinasi dan Mutasi

Untuk memperkecil pengaruh probabilitas kombinasi dan mutasi maka digunakan apa yang disebut dengan *Reproduction Plan*. Dalam penelitian ini digunakan dua buah *Reproduction Plan* sebagai berikut:

#### 1. *Reproduction Plain 1*

Memastikan bahwa setiap anggota paling sedikit sekali disilangkan dengan memilih pasangan dari populasi secara acak. Keturunan yang baru akan secara langsung dimasukan ke dalam populasi apabila solusi tersebut tidak ada sebelumnya. Karena populasi adalah daftar yang berurutan, maka anggota yang terbaik akan berada di tempat yang pertama dan yang terburuk pada posisi yang terakhir. Anggota terakhir dipindahkan setiap kali terjadi penambahan keturunan baru. Muatsi dalam bentuk pencarian terbatas dilakukan pada tiap anggota populasi dari tiap generasi. Operator mutasi yang dapat digunakan antara lain:

1. *Inversion/reversion*, yaitu melakukan pembalikan pada suatu urutan yang dipilih dalam suatu rangkaian.
2. *Transport*, yaitu memilih suatu urutan dan menyisipkannya ke posisi yang lain dalam rangkaian tersebut.
3. *Insertion*, yaitu memilih suatu elemen, memindahkannya dan kemudian menyisipkannya ke posisi lain dalam urutan.
4. *Exchange*, yaitu memilih dua elemen dan menukar posisi mereka dalam rangkaian tersebut.

#### 2. *Reproduction Plan 2*

Memastikan bahwa semua kemungkinan kombinasi antara setiap anggota populasi

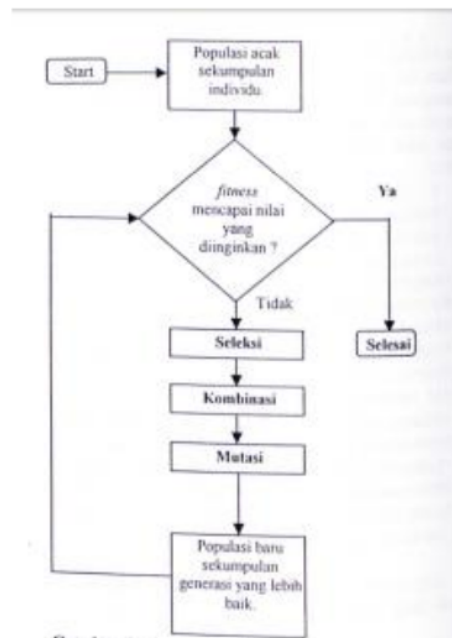
dilakukan pada setiap generasi. Apabila terdapat  $p$  anggota dalam populasi, maka akan ada  $p(p-1)$  kemungkinan persilangan, sebab kombinasi antara *parent 1* dan *parent 2* akan menghasilkan *offspring* yang berbeda dengan *offspring* yang dihasilkan dari kombinasi *parent 2* dengan *parent 1*. Setiap keturunan baru yang belum ada dalam populasi secara otomatis akan dimasukan ke dalam populasi dan anggota terburuk dihilangkan untuk menjaga agar populasi tetap pada ukuran semula. *Reproduction plan* ini tidak melakukan mutasi.

### 3.4. Kriteria Berhenti

Kriteria berhenti ditetapkan hingga sejumlah generasi tertentu, yang dalam penelitian ini user dapat memasukkan jumlah generasi sesuai yang diinginkan.

## 4. Pengembangan Aplikasi

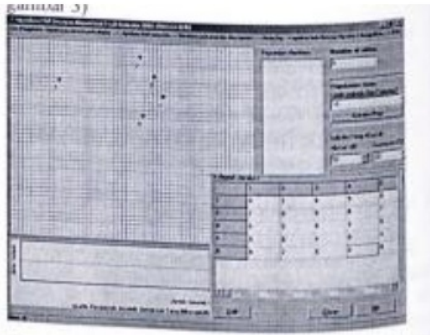
Adapun skema global algoritma evolutionary terlihat pada gambar 2



**Gambar 2.** Skema algoritma evolutionary

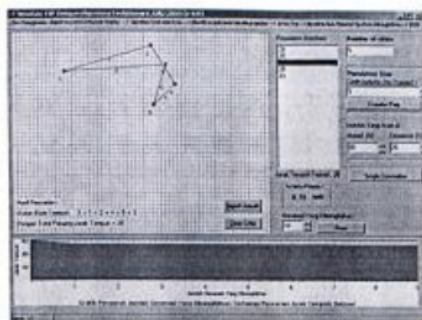
Program aplikasi yang dikembangkan akan membaca variabel input graf berbobot

brupa kota yang dikunjungi serta jaraknya, kemudian parameter-parameter yaitu jumlah populasi, mutasi, crossover, serta generasi yang dibangkitkan (terlihat gambar 3)



Gambar 3. Form input

Sedangkan output yang dihasilkan adalah rute optimal serta waktu komputasinya (terlihat gambar 4)



Gambar 4. Tampilan Output

## 5. Analisis Hasil

Beberapa parameter dalam aplikasi yang dibuat yaitu banyaknya simpul graph yang menggambarkan banyaknya kota yang dikunjungi dalam TSP terhadap waktu proses serta terhadap solusi TSP. Pada pengujian ini disediakan 10 kasus uji, dengan variasi banyaknya kota adalah 5, 10,15,....50.

Dari hasil pengujian tersebut, diperoleh

1. Pengaruh banyaknya kota terhadap waktu proses komputasi, terlihat pada tabel di bawah ini..

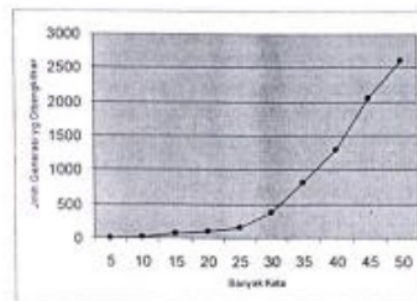
**Tabel 1.** Pengaruh banyaknya kota terhadap lama waktu proses.

No	banyak kota	waktu proses (detik)
1	5	0,125
2	10	1,55
3	15	3,65
4	20	8,8
5	25	11
6	30	31
7	35	47,3
8	40	65,82
9	45	122,66
10	50	314,48

Dari data tersebut, dapat diketahui bahwa pengaruh banyaknya kota terhadap lama waktu proses bersifat eksponensial. Namun dengan mengambil sampel hasil uji dengan banyak kota 5 hingga 50 didapatkan pendekatan regresi polynomial derajat 2, dengan persamaan

$$Y = 0,02622 x^2 - 9,421x + 97,3847 \quad (2)$$

Berikut adalah grafik perbandingan antara banyaknya simpul/ kota dengan waktu proses, seperti pada gambar.



Gambar 5. Pengaruh banyaknya kota terhadap waktu proses.

2. Pengaruh banyaknya kota terhadap jumlah generasi yang dibangkitkan untuk mendapatkan solusi optimal, seperti ditunjukkan pada tabel di bawah ini.

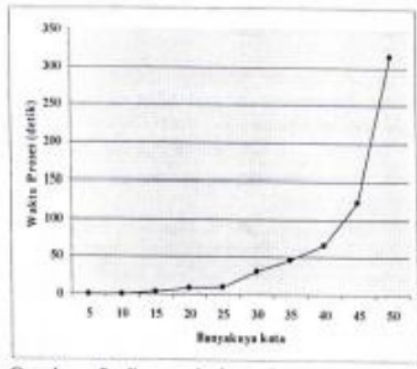
**Tabel 2.** Tabel Pengaruh banyaknya kota terhadap jumlah generasi yang dibangkitkan.

No	Ukuran Populasi	Waktu Proses (detik)
1	10	68
2	20	67,5
3	30	66,34
4	40	65,82
5	50	65,81

Dengan regresi polinomial derajat 2, diperoleh persamaan.

$$y = 2,0332 x^2 - 55,9171x + 331,9833 \quad (3)$$

Berikut adalah grafik perbandingan antara banyaknya kota terhadap jumlah generasi yang dibangkitkan, seperti pada gambar.



**Gambar 6.** Pengaruh banyak kota terhadap jumlah generasi yang dibangkitkan.

3. Pengaruh ukuran populasi terhadap waktu proses komputasi di dapat hasil ditunjukkan pada tabel di bawah ini.

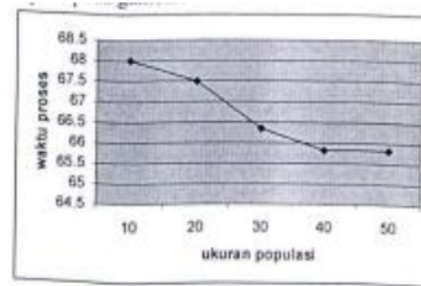
**Tabel 3.** Pengaruh banyaknya ukuran populasi terhadap waktu proses.

No	Ukuran Populasi	Waktu Proses (detik)
1	10	68
2	20	67,5
3	30	66,34
4	40	65,82
5	50	65,81

Dengan regresi polinomial derajat 2, diperoleh persamaan

$$y = 0,0012 x^2 - 0,1300 x + 69,322 \quad (4)$$

berikut adalah grafik perbandingan antara banyaknya ukuran populasi terhadap waktu proses, seperti pada gambar.



**Gambar 7.** Pengaruh banyaknya ukuran populasi terhadap waktu proses.

5 Berdasarkan hasil pengujian dan analisa algoritma yang telah dilakukan, maka dapat disimpulkan sebagai berikut :

1. Dari hasil pengujian terhadap waktu proses algoritma menunjukkan bahwa semakin banyak kota yang dikunjungi maka waktu proses semakin lala, dengan penekanan polinomial derajat 2 Didapat persamaan

$$y = 0,02622 x^2 - 9,421x + 97,3847$$

2. Semakin banyak jumlah ukuran populasi yang dibangkitkan, maka waktu proses akan semakin cepat dengan persamaan

$$y = 2,0332 x^2 - 55,9171x + 331,9833$$

3. Banyaknya kota juga berpengaruh terhadap jumlah generasi yang dibangkitkan dengan persamaan

$$y = 0,0012 x^2 - 0,1300 x + 69,322$$

4. Aplikasi yang dihasilkan dapat digunakan sebagai pengambilan suatu keputusan untuk pencarian urutan rute dan panjang jarak optimal yang ditempuh oleh wiraniaga yang melakukan perjalanan (*Tour*) ke beberapa kota

**Daftar Pustaka**

[1] Brandt, Hannelore., 1999, *Correlation Anysis of Fines Landscape*, International Institute for Applied System Analysis, Schlossplatz, Luxenburg, Austria.  
 [2] Febrian, Lala dan Widyadana, I Gede Agus, 2001, *Penerapan Evaluatory*

*Alogarhm Pada Penjadwalan Produksi*,  
Jurnal, Universitas Kristen Petra, Surabaya.

- [3] Hemert, jano van., 2004, *why Solving the travelling salesman problem can be difficult: a hunt for hand instance*, Centre of Emergent Computing, Napier University, Edinburgh.
- [4] Rustamaji, Heru Cahya dan Kodong, Frans Richard., 2003, “ *Aplikasi Koloni Semut Untuk Mneyelesaikan Masalah Perjalanan Wiraniaga (Travelling Salesman Problem)*”, *Prosiding Seminar Nasional Informa 3 II UAD*, 24-29
- [5] Soper, A.J.C. Walshaw And M. Cross., 2003, *A Combined Evolutionary Search and Multilevel Optimisation Approach to Graph- Partitioning*, *Journal of Global Optimization*, Kluwer Academic Publishers, Netherlands.

# Algoritma Evolutionary

---

## ORIGINALITY REPORT

---

8%

SIMILARITY INDEX

8%

INTERNET SOURCES

3%

PUBLICATIONS

2%

STUDENT PAPERS

---

## PRIMARY SOURCES

---

1

[pt.scribd.com](https://pt.scribd.com)

Internet Source

3%

2

[ejournal.unisbablitar.ac.id](http://ejournal.unisbablitar.ac.id)

Internet Source

1%

3

[www.emse.fr](http://www.emse.fr)

Internet Source

1%

4

[vdocuments.mx](http://vdocuments.mx)

Internet Source

1%

5

[repository.its.ac.id](http://repository.its.ac.id)

Internet Source

1%

6

Dian Tri Wiyanti. "ALGORITMA OPTIMASI  
UNTUK PENYELESAIAN TRAVELLING  
SALESMAN PROBLEM", Jurnal Transformatika,  
2013

Publication

1%

---

Exclude quotes Off

Exclude bibliography Off

Exclude matches Off



