

# Perancangan Kombinasi Algoritma Genetika dan Tabu Search untuk Penyelesaian Travelling Salesman Problem

*by* John Doe

---

**Submission date:** 24-Nov-2020 09:22AM (UTC+0700)

**Submission ID:** 1453852662

**File name:** paper\_kolokium.pdf (585.84K)

**Word count:** 4071

**Character count:** 25618

# Perancangan Kombinasi Algoritma Genetika dan Tabu Search untuk Penyelesaian *Travelling Salesman Problem*

xxx<sup>1</sup>, yyy<sup>2</sup>

<sup>1</sup>Program Studi Informatika Program Sarjana

<sup>2</sup>Jurusan Informatika

Universitas Islam Indonesia

Yogyakarta, Indonesia

<sup>1</sup>[xxx@students.uii.ac.id](mailto:xxx@students.uii.ac.id), <sup>2</sup>[yyy@uui.ac.id](mailto:yyy@uui.ac.id)

**Abstrak**—Saat melakukan kunjungan ke beberapa tempat, seringkali seseorang kebingungan dalam menentukan urutan titik atau tempat yang akan dikunjungi agar seluruh tempat tersebut dapat dilalui dan jarak yang ditempuh sekecil mungkin. Permasalahan ini dikenal dengan *Travelling Salesman Problem (TSP)*. TSP adalah sebuah persoalan klasik seorang *salesman* saat mencari rute atau jalur terpendek. Banyak algoritma yang dapat digunakan dalam penyelesaian TSP, namun setiap algoritma juga memiliki kelebihan dan kekurangan masing-masing. Tujuan dari makalah ini yaitu merancang kombinasi Algoritma Genetika dan Tabu Search (GA-TS) dalam menyelesaikan kasus TSP. Diharapkan dengan menggunakan kombinasi GA-TS, hasil akhir dari penyelesaian TSP yang didapatkan lebih optimal dibandingkan dengan Algoritma Genetika. Kombinasi GA-TS memiliki beberapa tahapan yaitu inialisasi kromosom, evaluasi kromosom, seleksi kromosom, pengecekan tabulist, *crossover*, mutasi, regenerasi, dan penentuan kromosom terbaik. Tahapan selanjutnya setelah perancangan yaitu implementasi sistem. Sistem diharapkan dapat menghasilkan sebuah program yang mampu menyelesaikan beberapa kasus TSP yang berbeda dan mendapatkan hasil yang optimal.

**Kata Kunci**—*Travelling Salesman Problem, Algoritma Genetika, Tabu Search, Tabulist*

## I. LATAR BELAKANG

Permasalahan optimasi klasik dalam mencari rute kunjungan terpendek dikenal dengan istilah *Travelling Salesman Problem (TSP)* [1][2]. Seorang *salesman* terlibat dalam TSP, yang mana dia wajib mengunjungi beberapa kota untuk menawarkan hasil produksinya. Dalam melakukan kunjungan, *salesman* harus merencanakan suatu rute agar kota-kota tersebut hanya dilalui satu kali saja, kemudian *salesman* kembali ke kota asal. Untuk mendapatkan rute terpendek, beberapa algoritma yang memecahkan persoalan TSP mewajibkan memperhitungkan semua probabilitas rute yang dapat diperoleh [3].

TSP merupakan salah satu persoalan penting di dunia matematika dan informatika. Persoalan tersebut telah banyak diterapkan di berbagai kasus di dunia nyata [4][5]. Adapun contoh kasus yang mampu dipecahkan dengan TSP pada dunia nyata antara lain [2]: mencari rute bis agar dapat

mengantarkan para siswa tepat waktu, mencari rute truk untuk mengantar parcel, serta mencari rute pengambilan tagihan telepon secara efisien.

Menurut [6], terdapat dua jenis kasus TSP. Yang pertama yaitu TSP asimetris. Pada kasus TSP asimetris, dana atau biaya dari kota 1 ke kota 2 berbeda dengan dana atau biaya dari kota 2 ke kota 1. Jenis yang kedua yaitu TSP simetris. Pada kasus TSP simetris, dana atau biaya dari kota 1 ke kota 2 setara dengan dana atau biaya dari kota 2 ke kota 1.

Banyak cara penyelesaian yang dapat diterapkan untuk mendapatkan rute terpendek dalam TSP, di antaranya yaitu menggunakan Algoritma *Artificial Bee Colony*, *Tabu Search*, Algoritma *Cheapest Insertion Heuristics*, Algoritma Genetika, dan yang lainnya. Setiap algoritma mempunyai keunggulan dan kekurangannya sendiri. Hasil dari sebuah algoritma pun berbeda-beda, dikarenakan belum pasti sebuah algoritma yang nilai optimasinya besar untuk sebuah masalah memiliki nilai optimasi yang besar pula untuk masalah lainnya [7].

Pembahasan yang akan dilakukan pada makalah ini ialah perancangan kombinasi Algoritma Genetika dan Tabu Search (GA-TS) untuk penyelesaian *Travelling Salesman Problem* dengan kasus simetris. Sebelum ini sudah pernah dilakukan penelitian dengan kasus penjadwalan yang menggunakan GA-TS [8]. Dan hasil akhir membuktikan bahwa GA-TS lebih efisien dibandingkan Algoritma Genetika. Sehingga diharapkan pada penelitian ini, penggunaan GA-TS lebih optimal dibandingkan Algoritma Genetika.

## II. PENELITIAN SEJENIS

Dalam TSP banyak sekali algoritma yang dapat diterapkan. Beberapa di antaranya yaitu Algoritma *Artificial Bee Colony*, *Tabu Search*, Algoritma *Cheapest Insertion Heuristics*, Algoritma *Greedy*, Algoritma Genetika, dan masih banyak lagi. Terdapat banyak penelitian mengenai penerapan ataupun penyelesaian menggunakan algoritma-algoritma tersebut pada kasus TSP. Berikut ini akan disajikan pemaparannya.

Algoritma Greedy termasuk ke dalam jenis algoritma heuristik, dimana algoritma ini mencari *local optimal* dan

mengoptimalkannya sehingga mendapatkan *global optima*. Berdasarkan penelitian oleh [9], walaupun Algoritma Greedy memiliki iterasi yang lebih banyak dibandingkan Algoritma Genetika dan *Nearest Neighbor*, hasil yang didapatkan pada Algoritma Greedy ialah yang paling mendekati hasil optimal. Sedangkan menurut [10], nilai yang diperoleh tidak terus-menerus optimal, dikarenakan masih terpicat dalam *local optimal*. Sehingga Algoritma Greedy kurang tepat digunakan untuk jumlah kota yang besar.

*Artificial Bee Colony* termasuk ke dalam *swarm intelligence* yaitu Algoritma *Evolutionary Computation* (salah satu algoritma optimasi berbasis probabilistik). Algoritma ini terinspirasi oleh perilaku lebah yang hidupnya berkoloni. Seekor lebah mampu menjangkau sumber makanan sambil mengingat arah, tata letak, dan jarak dari sumber makanan. Berdasarkan hasil diskusi [1][11], *Artificial Bee Colony* ialah algoritma fleksibel, sederhana, dan juga mempunyai keahlian agar bebas dari *local minimum* dan secara kemampuan dimanfaatkan untuk multivariabel optimasi fungsi. Akan tetapi, apabila jumlah kota yang diproses semakin banyak maka tingkat *error* akan semakin besar.

*Cheapest Insertion Heuristics* (CIH) dan *Nearest Neighbor Heuristics* (NNH) merupakan metode heuristik yang cukup efektif memecahkan permasalahan TSP. Namun metode heuristik ini tidak selalu bisa memecahkan permasalahan, namun kerap memecahkan masalah dengan cukup baik karena pada metode ini bertujuan untuk mengurangi jumlah pencarian solusi. Menurut [2], *runtime* CIH jauh lebih cepat dibandingkan NNH. Kemudian pada penelitian [9], menyatakan bahwa ketika persoalan TSP memiliki kompleksitas besar dan jumlah iterasi besar, NNH mampu menemukan rute terpendek.

Algoritma Genetika adalah sebuah metode yang meniru proses evolusi alam dan algoritma ini dikenal dalam menyelesaikan masalah kompleks. Waktu komputasi yang diperlukan pada Algoritma Genetika cenderung stabil. Meskipun jumlah kota besar jika dibandingkan dengan algoritma lainnya, Algoritma Genetika sanggup memberikan rute terdekat. Walaupun demikian, tingkat keberhasilan Algoritma Genetika sangat bergantung pada saat menetapkan parameter masukan, yaitu ukuran populasi, jumlah generasi, probabilitas *crossover*, dan probabilitas mutasi [9][12].

*Tabu Search* termasuk ke dalam metode optimasi dengan teknik pencarian lokal. Dimana tujuan dari metode ini ialah agar tidak terjadi perulangan serta dididapkannya hasil yang serupa pada suatu iterasi dimana iterasi tersebut akan digunakan pada iterasi berikutnya [13]. Metode optimasi ini pun teruji efektif dan kerap digunakan untuk menyelesaikan permasalahan optimasi berskala besar. Berdasarkan penelitian oleh [14], ia melakukan sebuah penelitian dengan menyelesaikan TSP menggunakan *Firefly Algorithm* namun terdapat kelemahan dalam penyelesaian optimasi berskala besar maka diperlukannya sebuah kombinasi dengan *Tabu Search* agar solusi yang didapatkan lebih akurat. Dan hasilnya pun terbukti bahwa dengan kombinasi tersebut dapat meningkatkan hasil akurasi permasalahan TSP.

*Held-karp* oleh [15] membuktikan bahwa hasil *run time* *Held-karp* jauh lebih cepat dibandingkan dengan *brute force*. Walaupun terjadi peningkatan *runtime* setiap penambahan destinasi namun berjalan cukup stabil. Dan dapat dikatakan bahwa hasil penyelesaian menggunakan *Held-karp* lebih baik

dari *brute force*. Kemudian [16] mengatakan bahwa dengan intensitas yang besar atau kota yang banyak tidaklah memungkinkan menggunakan algoritma ini dalam memecahkan masalah TSP.

*Ant Colony Optimization* (ACO) meniru kelakuan semut saat mencari makanan dari sarangnya ke tempat sumber makanan dengan rute terpendek. Prinsip dasar ACO ialah pada rute yang dilalui saat perjalanan, semut selalu meninggalkan suatu feromon. Feromon tersebut menjadi penunjuk jalan untuk semut yang lain dalam menyelesaikan perjalanan. Pada penelitian [17], mengatakan bahwa ACO tidak dapat memperhitungkan jumlah iterasi yang baik, dikarenakan solusi yang dihasilkan tidak akan akurat bila jumlah iterasi terlalu kecil. Sedangkan solusi akan semakin besar jika jumlah iterasi terlalu besar. Dan ACO tepat untuk jumlah simpul yang sangat banyak, dikarenakan ACO meletakkan semut pada setiap simpul, maka pencarian semakin cepet.

*Branch and Bound* adalah sebuah algoritma yang memiliki peranan untuk menyelesaikan permasalahan optimasi. Pendekatan yang digunakan dalam *branch and bound* ialah pendekatan numerisasi dengan menyingkirkan *search space* yang tidak tertuju pada pemecahan masalah. Hasil optimasi yang didapatkan menggunakan algoritma ini bergantung pada keakuratan pemilihan fungsi pembatas yang dipilih. Metode yang dipilih berdasarkan naluri dan pengetahuan maka seringkali hasil yang didapatkan tidak optimal. Akan tetapi pada aspek waktu, *branch and bound* dapat menjadi pilihan dalam menyelesaikan masalah optimasi kombinatorial [18].

Berdasarkan penelitian-penelitian sebelumnya pada kasus TSP yang diselesaikan menggunakan berbagai macam algoritma, hasil yang didapatkan setiap algoritma berbeda-beda yaitu mempunyai kelebihan dan kekurangannya masing-masing. Selain kasus TSP, algoritma-algoritma tersebut dapat menyelesaikan kasus lain contohnya ialah kasus penjadwalan. Pada penelitian [8] menggabungkan dua algoritma yaitu Algoritma Genetika dan *Tabu Search* (GA-TS), ia membuktikan bahwa kombinasi GA-TS hasilnya lebih efektif dibandingkan Algoritma Genetika pada kasus penjadwalan. Hasil pengujian membuktikan bahwa kombinasi Algoritma Genetika dan *Tabu Search* ketika jumlah generasi ke-98 sudah mendapatkan satu jadwal optimal, sedangkan dengan Algoritma Genetika jadwal optimal didapatkan pada generasi ke-298 [8]. Sehingga berdasarkan penelitian tersebut untuk penelitian kali ini menggunakan kombinasi GA-TS dalam penyelesaian TSP dan diharapkan dapat lebih optimal dibandingkan Algoritma Genetika.

### III. LANDASAN TEORI

#### A. *Travelling Salesman Problem*

Pertama kali ide mengenai *Travelling Salesman Problem* disampaikan pada tahun 1800 oleh matematikawan Irlandia William Roman Hamilton dan matematikawan Inggris Thomas Penyngton [14]. Dimulai pada tahun 1930 para matematikawan mempelajari bentuk umum dari TSP. Karl Menger di Vienna dan Harvard mengawalinya, setelah itu permasalahan TSP dipublikasikan oleh Hassler Whitney dan Merrill Flood di Princeton [4]. Pada tahun 1985 Hoffman dan Wolfe menjelaskan bahwa TSP ialah masalah yang harus



diselesaikan oleh seorang *salesman* dalam menjalankan perjalanannya.

TSP merupakan sebuah persoalan dimana seorang *salesman* harus menjajakan produknya ke kota-kota namun setiap kota hanya bisa dikunjungi satu kali saja dan kembali ke kota awal keberangkatan. Hal tersebut sering dikenal dengan nama sirkuit *hamilton*. Meskipun bernama *Travelling Salesman Problem*, namun penerapannya tidak selalu berhubungan dengan *salesman* atau pedagang [13]. Dan penyelesaian dari kasus TSP ialah menemukan jarak terpendek dari kota atau tempat yang akan dituju. Cara penyelesaiannya ialah dengan menghitung seluruh peluang rute, setelah itu menentukan salah satu rute yang terpendek. Maka dari itu bila terdapat  $n$  kota yang harus dituju, maka terdapat  $n!$  kombinasi kota yang akan dibandingkan jaraknya. *Runtime* yang diperlukan tentu bertambah bersamaan dengan meningkatnya jumlah kota yang harus dituju [10].

### B. Algoritma Genetika

Algoritma genetika (GA) termasuk ke dalam algoritma evolusi, dimana teknik optimasinya meniru proses evolusi biologi. GA merupakan tipe algoritma evolusi yang paling terkenal dan banyak diimplementasikan pada permasalahan yang kompleks. Dalam bidang biologi, sosiologi, fisika, ekonomi, Algoritma Genetika banyak digunakan hingga masalah optimasi yang model matematikanya sangat kompleks [19]. Algoritma Genetika memiliki beberapa langkah, yaitu:

#### 1) Inisialisasi Kromosom

Inisialisasi dibuat secara acak untuk membangkitkan himpunan solusi baru yang terdiri dari sejumlah string kromosom dan disebut sebagai populasi. Kromosom mengandung informasi solusi dari sekian banyak kemungkinan solusi masalah yang dihadapi [12].

#### 2) Evaluasi Kromosom

Melakukan perhitungan nilai fitness terhadap kromosom. Setelah nilai fitness didapatkan, nilai tersebut akan digunakan untuk proses seleksi kromosom.

#### 3) Seleksi Kromosom

Proses ini berguna agar dapat memilih kualitas kromosom yang baik sehingga dapat meneruskan proses *crossover*. Ada berbagai macam teknik seleksi yang dapat digunakan, di antaranya adalah *Roulette Wheel Selection*, *Elitism*, *Rank Base Selection*, dan *Steady State Selection*.

#### 4) Crossover

Crossover atau kawin silang menghasilkan dua kromosom anak dengan cara menukar informasi dari dua induk kromosom [20]. Proses ini akan memilih dua kromosom induk yang memiliki nilai fitness terbaik kemudian akan mengalami proses kawin silang atau *crossover* secara acak dan menghasilkan kromosom anak. Metode *crossover* ada beberapa macam, di antaranya yaitu *crossover* banyak titik, *crossover* satu titik, *crossover* aritmatika, dan *crossover* untuk representasi kromosom permutasi. Pada *crossover* untuk representasi kromosom permutasi ada beberapa metode, seperti *order-based crossover*, *order crossover*, *cycle crossover*, *position-based crossover*, *partial mapped crossover*, *heuristic cross over*, dan lain-lain [21].

#### 5) Mutasi

Melakukan modifikasi satu atau lebih gen dalam individu merupakan tahapan dari mutasi yang akan menghasilkan individu baru. Mutasi akan membuat populasi menjadi bervariasi dengan menukar gen yang hilang dari populasi selama proses seleksi serta menyimpan gen yang tidak ada dalam populasi awal [22].

#### 6) Regenerasi

Setelah melewati tahap mutasi, hasil dari proses tersebut ditempatkan pada populasi yang baru atau generasi baru. Kemudian populasi baru ini akan diproses kembali dari tahapan ke-2 (menghitung nilai *fitness*) hingga kromosom mencapai iterasi atau perulangan yang telah ditentukan.

### C. Tabu Search

Pada tahun 1970-an, Glover memperkenalkan Tabu Search (TS) pertama kalinya. Menurut Glover, TS adalah salah satu metode *metaheuristic* tingkat tinggi untuk penyelesaian permasalahan optimasi kombinatorial. TS menyimpan solusi terbaik dan tetap mencari berdasarkan solusi terakhir, untuk menjaga agar solusi terbaik tidak hilang. Selain itu algoritma ini mampu mengingat sebagian solusi yang pernah didapatkan dan mencegah menggunakan solusi yang telah ditelusuri untuk menghindari pengulangan yang sia-sia.

Adapun tahapan-tahapan dalam Tabu Search sebagai berikut [23]:

- 1) Menentukan titik awal.
- 2) Menentukan solusi alternatif dengan menggabungkan dua titik berdasarkan jarak terkecil.
- 3) Mengevaluasi solusi-solusi alternatif dengan *tabu list* untuk melihat apakah solusi tersebut sudah ada pada *tabu list*. Untuk menghindari perulangan, solusi tidak akan dievaluasi jika sudah ada pada *tabu list* dan begitu sebaliknya.
- 4) Mendapatkan solusi terbaik dan memutuskan sebagai solusi optimum baru.
- 5) *Tabu list* diperbarui.
- 6) Ulangi langkah-langkah tersebut hingga kriteria sudah terpenuhi.

## IV. RANCANGAN SISTEM

### A. Pengambilan Data

Data yang digunakan pada sistem ini adalah dataset TSP dengan kasus simetris yang memiliki koordinat  $x, y$ . Data yang diambil bervariasi yaitu dengan ukuran yang berbeda-beda dari kecil hingga besar. Pada data yang digunakan, sudah terdapat solusi optimumnya sehingga pada tahap evaluasi, hasil yang didapatkan dari algoritma yang diusulkan dapat dibandingkan dengan solusi yang optimum.

Dataset yang digunakan diambil melalui internet pada tautan <http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/> atau <https://people.sc.fsu.edu/~jburkardt/datasets/tsp/tsp.html>.

Gambar 1 merupakan contoh data yang akan digunakan. Pada kolom pertama merupakan kolom kota/titik, kolom kedua ialah koordinat titik  $x$  dan kolom terakhir ialah koordinat titik

y. Sebagai contoh, pada kota 1, koordinat titik  $x$  ialah 6734 sedangkan koordinat titik  $y$  ialah 1453.

Kota	X	Y
1	6734	1453
2	2233	10
3	5530	1424
4	401	841
5	3082	1644
6	7608	4458

Gambar 1. Contoh Data TSP

### B. Analisis Kebutuhan

Dalam sistem yang akan dibuat diperlukan sebuah masukan berupa banyaknya iterasi yang akan dijalankan sistem, jumlah populasi yang akan dibuat, dan probabilitas mutasi. Adapun kebutuhan proses yang akan dilakukan pada sistem ini ialah:

- Proses konversi data koordinat titik  $x, y$  menjadi jarak antar titik atau kota.
- Proses pembuatan inialisasi kromosom atau rute secara random.
- Proses menghitung nilai fitness (evaluasi) kromosom atau rute.
- Proses seleksi kromosom atau rute.
- Proses pengecekan tabulist.
- Proses *crossover*.
- Proses mutasi.
- Proses menghasilkan kromosom atau rute terbaik.
- Proses menghitung fitness dan jarak akhir.

Dari seluruh proses kombinasi GA-TS tersebut keluaran yang diharapkan dalam sistem ini nantinya ialah dapat menampilkan hasil dari persoalan TSP yaitu berupa rute terbaik.

### C. Flowchart Rancangan Sistem

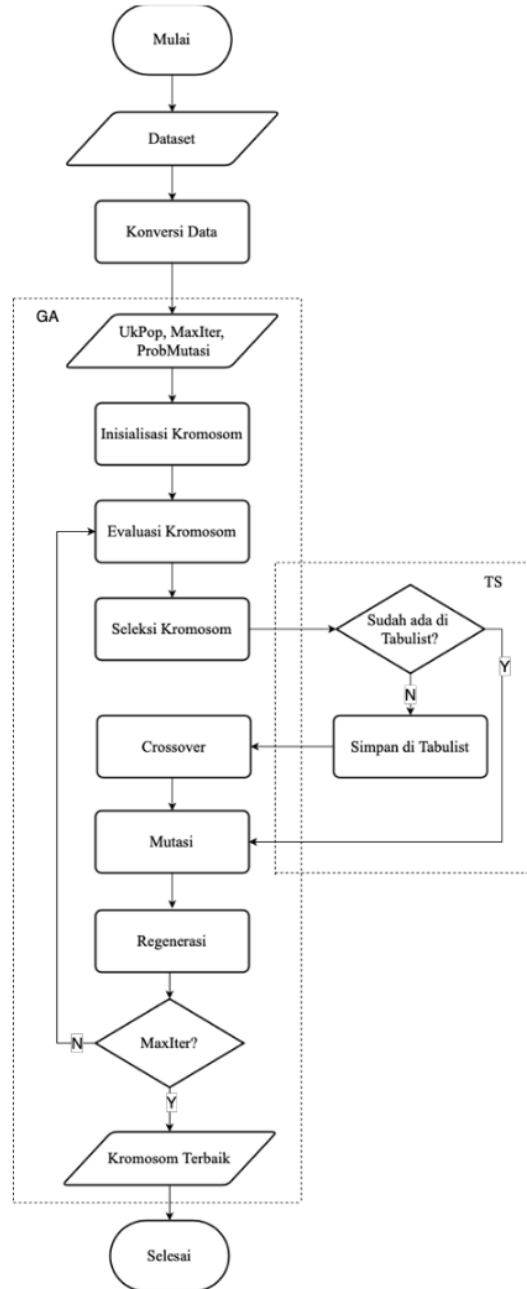
6 Pada makalah ini, rancangan utama dari kombinasi Algoritma Genetika dan Tabu Search untuk penyelesaian TSP akan disusun dalam bentuk diagram *flowchart*. Selanjutnya akan dijelaskan lebih detil mengenai setiap proses yang terdapat pada *flowchart*. Gambar 2 merupakan *flowchart* rancangan sistem. Terdapat dua blok utama pada rancangan sistem, yaitu blok GA dan TS. Dari *flowchart* dapat dilihat bahwa blok TS merupakan bagian dari blok GA. Hal ini menunjukkan bahwa proses kombinasi antara GA dan TS dilakukan dengan menjadikan TS sebagai salah satu tahapan di dalam GA, yang mana TS berperan dalam proses seleksi kromosom.

Langkah pertama yang dilakukan dalam sistem ini ialah memasukkan data kemudian mengkonversi data yang awalnya berupa koordinat  $x, y$  menjadi data jarak antar kota atau titik. Cara mengkonversi data yaitu dengan menggunakan rumus pada Persamaan 1 dengan  $x_1, y_1$  ialah koordinat kota/titik awal dan  $x_2, y_2$  koordinat kota/titik akhir [12]. Tabel 1 menunjukkan contoh gambaran sebuah hasil

dari konversi data koordinat titik  $x, y$  menjadi jarak antar kota atau titik.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

Kemudian setelah jarak didapatkan, maka *user* memasukkan ukuran populasi (UkPop), maksimum iterasi (MaxIter), dan probabilitas mutasi (ProbMutasi).



Gambar 2. Flowchart Rancangan Sistem

Tabel 1. Contoh Hasil Konversi Data

	1	2	3	4	5	6
1	0	1450	47	643	223	62
2	1450	0	1417	833	1634	4451
3	47	1417	0	605	233	3034
4	643	833	605	0	807	3624
5	223	1634	233	807	0	2817
6	62	4451	3034	3624	2817	0

Setelah itu, langkah-langkah yang dilakukan sesuai dengan urutan berikut:

1) *Inisialisasi Kromosom*

Kromosom mengandung informasi solusi dari sekian banyak kemungkinan solusi masalah yang dihadapi. Dalam permasalahan TSP, kromosom ditunjukkan sebagai rute perjalanan atau lebih jelasnya ialah dalam satu individu terdiri dari beberapa gen yang direpresentasikan dalam bentuk angka-angka. Dimana setiap angka tersebut mewakili titik-titik (tempat) tujuan yang akan dilalui.

2	3	1	5	4	6
---	---	---	---	---	---

Gambar 3. Contoh Representasi Kromosom

Gambar 3 merupakan contoh representasi kromosom. Dapat dilihat bahwa rute yang akan dilalui ialah dari titik 2-3-1-5-4-6. Tabel 2 merupakan contoh inisialisasi kromosom dengan ukuran populasi yaitu sebesar 5.

Tabel 2. Inisialisasi Kromosom

Kromosom
Kromosom 1 (5-1-2-4-3-6)
Kromosom 2 (1-4-5-3-6-2)
Kromosom 3 (3-1-2-5-6-4)
Kromosom 4 (3-5-2-1-4-6)
Kromosom 5 (2-1-3-5-4-6)

2) *Evaluasi Kromosom*

Setelah kromosom terbentuk maka setiap kromosom dilakukan perhitungan nilai fitness menggunakan Persamaan 2 [13]. *TotalJarak* merupakan total jarak yang ditempuh dari titik awal hingga akhir setiap kromosom. Pada Tabel 3 merupakan contoh perhitungan *fitness* setiap kromosom.

$$fitness = \frac{1}{TotalJarak} \quad (2)$$

Tabel 3. Evaluasi Kromosom

Kromosom	Jarak	Fitness
Kromosom 1 (5-1-2-4-3-6)	6145	0,00016273
Kromosom 2 (1-4-5-3-6-2)	9168	0,00010908
Kromosom 3 (3-1-2-5-6-4)	9572	0,00010447
Kromosom 4 (3-5-2-1-4-6)	7584	0,00013186
Kromosom 5 (2-1-3-5-4-6)	6161	0,00016231

3) *Seleksi Kromosom*

Seleksi yang dipilih dalam penelitian ini ialah *Rank Base Selection*. Nilai fitness yang telah didapatkan akan diurutkan berdasarkan nilai terbesar hingga terkecil, kemudian urutan yang telah didapat akan diseleksi menggunakan *elitism* yaitu menyimpan 2 kromosom (*parent*) dengan nilai yang baik. Kromosom tersebut akan digunakan untuk proses selanjutnya. Tabel 4 merupakan contoh hasil seleksi kromosom.

Tabel 4. Contoh Hasil Seleksi Kromosom

Induk 1	5	1	2	4	3	6
Induk 2	2	1	3	5	4	6

4) *Cek Tabulist*

Cek tabulist memiliki peranan penting dalam kombinasi GA-TS yaitu agar tidak didapatkannya solusi yang sama dikarenakan oleh perulangan. Dan solusi tersebut akan digunakan pada tahap selanjutnya. Setelah hasil seleksi didapatkan maka untuk iterasi pertama kromosom akan disimpan pada *tabulist*, dan untuk iterasi selanjutnya kromosom akan dicek terlebih dahulu menggunakan *tabulist*. Apabila kromosom sudah terdapat pada *tabulist*, maka kromosom akan dimutasi untuk menghindari perulangan, jika belum ada maka akan disimpan dan dilakukan *crossover*. Tabel 5 merupakan contoh ilustrasi penyimpanan kromosom pada *tabulist*.

Tabel 5. Ilustrasi Tabulist

Tabulist
5-1-2-4-3-6
2-1-3-5-4-6

5) *Crossover*

Pada *crossover*, dua kromosom induk akan mengalami proses kawin silang dengan probabilitas *crossover*. Pada tahap ini kromosom tidak dapat memiliki gen atau kota yang sama. Oleh karena itu, dapat digunakan metode *ordered crossover* dengan memilih secara acak beberapa gen di induk

1, kemudian sisa gen pada induk 1 akan diisi oleh induk 2 dimana gen tidak boleh sama. Proses *crossover* akan menghasilkan *offspring/children* baru. Gambar 4 merupakan contoh *ordered crossover*.

Induk 1	5	1	2	4	3	6
Induk 2	2	1	3	5	4	6
Children	3	1	2	4	5	6

Gambar 4. Contoh Ordered Crossover

#### 6) Mutasi

Setelah terjadinya *crossover* ataupun pengecekan *tabu list*, kromosom akan dimutasi sesuai dengan probabilitas mutasi menggunakan metode *reciprocal exchange mutation* yaitu dengan memilih dua posisi secara acak kemudian menukarkan kedua nilai tersebut. Gambar 5 merupakan contoh *reciprocal exchange mutation*.

		XP1 ↓		XP2 ↓		
Children	3	1	2	4	5	6
Mutasi	3	5	2	4	1	6

Gambar 5. Contoh Reciprocal Exchange Mutation

#### 7) Regenerasi

Setelah proses mutasi selesai, maka hasil dari proses tersebut menjadi populasi yang baru atau generasi baru. Kemudian generasi tersebut akan diproses kembali dari tahapan pertama hingga akhir. Pada Tabel 6 merupakan contoh populasi baru atau generasi baru.

Tabel 6. Populasi Baru

Kromosom	Jarak	Fitness
Kromosom 1 (5-1-2-4-3-6)	6145	0,00016273
Kromosom 2 (1-4-5-3-6-2)	9168	0,00010908
Kromosom 3 (3-5-2-4-1-6)	3405	0,00029369
Kromosom 4 (3-5-2-1-4-6)	7584	0,00013186
Kromosom 5 (2-1-3-5-4-6)	6161	0,00016231

#### 8) Kromosom Terbaik

Proses akan berhenti, jika iterasi yang ditetapkan telah selesai.

#### D. Rancangan Pengujian

Skenario pengujian yang akan diterapkan nantinya pada penelitian ini ialah pengujian performa sistem. Pengujian performa yang dilakukan adalah analisis hasil waktu pemrosesan berdasarkan jumlah iterasi, analisis hasil akhir berupa jarak berdasarkan jumlah iterasi, dan analisis

perbandingan antara GA-TS dan Algoritma Genetika (tanpa TS).

## V. DISKUSI

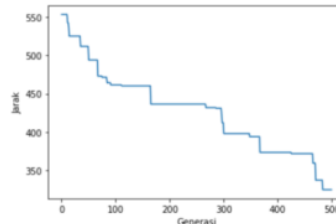
### A. Prototype

Setelah perancangan sistem selesai, tahap berikutnya ialah tahapan implementasi sistem. Untuk implementasi sistem, akan digunakan bahasa pemrograman Python. Perangkat lunak yang akan digunakan yaitu Google Colaboratory (Colab) dan *browser* Google Chrome.

Pada tahapan awal implementasi, dicari data yang sesuai dengan kriteria, yaitu sudah memiliki nilai optimal agar dapat menjadi patokan keberhasilan sistem ini. Selain itu, data yang diambil harus bervariasi dari data yang ukurannya kecil hingga besar kemudian isi data mencakup koordinat titik  $x,y$ . Setelah data sudah didapatkan, maka setiap langkah yang ada di *flowchart* sistem (sesuai dengan yang ada pada Gambar 2) diimplementasikan.

Keluaran dari sistem diharapkan dapat menampilkan solusi atau rute terbaik, jarak tempuh terbaik, nilai *fitness* terbaik, dan diagram garis. Diagram garis menunjukkan nilai jarak tempuh terbaik di setiap generasi. Pada Gambar 6 dapat dilihat contoh keluaran yang diharapkan dari hasil implementasi sistem. Untuk tahapan implementasi sudah dilakukan namun belum sepenuhnya selesai dibuat sehingga tidak dicantumkan pada makalah ini.

INDIVIDU TERBAIK = ([15, 9, 5, 7, 3, 12, 14, 8, 6, 4, 11, 1, 13, 2, 10])  
 JARAK TERBAIK = 324.7994634240415  
 FITNESS TERBAIK = 0.00307882282026572



Gambar 6. Rancangan Hasil Implementasi

### B. Kontribusi

Makalah ini memiliki kontribusi pada kasus TSP yaitu melakukan modifikasi pada Algoritma Genetika dengan melakukan kombinasi terhadap Tabu Search. Proses modifikasi dilakukan dengan mengkolaborasikan salah satu langkah yang ada pada Algoritma Genetika dengan Tabu Search. Langkah tersebut adalah proses seleksi kromosom. Proses seleksi kromosom dengan memanfaatkan Tabu Search memiliki peranan penting dalam penyelesaian kasus TSP yaitu agar terhindar dari perulangan agar hasil tidak sama, sehingga diharapkan solusi optimal dapat lebih cepat diperoleh.

## VI. KESIMPULAN

Makalah ini telah membuat sebuah rancangan sistem yang akan diimplementasikan untuk menyelesaikan *Travelling Salesman Problem* dengan kasus simetris. Algoritma yang digunakan ialah kombinasi Algoritma Genetika dan Tabu Search dengan harapan hasil dapat lebih optimal dibanding dengan Algoritma Genetika. Hasil akhir



pada sistem berupa rute terbaik, jarak tempuh terbaik, fitness terbaik, dan diagram garis. Kemudian hasil tersebut akan dilakukan pengujian performa yang membandingkan Algoritma Genetika dan kombinasi GA-TS.

#### REFERENSI

- [1] Andri, Suyandi, and WinWin, "Aplikasi Travelling Salesman Problem dengan Metode Artificial Bee Colony," vol. 14, no. 1, pp. 59–68, 2013.
- [2] Kusriani and J. E. Istiyanto, "Penyelesaian Travelling Salesman Problem Dengan Algoritma Cheapest Insertion Heuristics Dan Basis Data," *J. Inform.*, vol. 8, no. 2, pp. 114–114, 2007, doi: 10.9744/informatika.8.2.pp.109-114.
- [3] S. Puspitorini, "Penyelesaian Masalah Traveling Salesman Problem dengan Jaringan Saraf Self Organizing," *Media Inform.*, vol. 6, no. 1, pp. 39–55, 2008, doi: 10.20885/informatika.vol6.iss1.art3.
- [4] U. Rafflesia, "Travelling Salesperson Problem dengan Pendekatan Heuristik," *J. Gradien*, vol. 12, no. 2, pp. 1171–1174, 2016.
- [5] S. N., "Studi Metode Program Dinamik Dalam Mencari Solusi Optimal Pada Persoalan Travelling Salesman Problem (TSP)," pp. 1–12, 1993.
- [6] D. T. Wiyanti, "Algoritma Optimasi Untuk Penyelesaian Travelling Salesman Problem," *J. Transform.*, vol. 11, no. 1, p. 1, 2013, doi: 10.26623/transformatika.v1i1i.76.
- [7] G. Aristi, "Perbandingan Algoritma Greedy, Algoritma Cheapest Insertion Heuristics Dan Dynamic Programming Dalam Penyelesaian Travelling Salesman Problem," *Paradigma*, vol. XVI, no. 2, pp. 52–58, 2014.
- [8] E. D. Sari, "Optimasi Penjadwalan Personalia Rumah sakit Berbasis AGENT Menggunakan Kombinasi Algoritma Genetika dan Tabu Search," *Optimasi Penjadwalan Pers. Rumah sakit Berbas. AGENT Menggunakan Komb. Algoritm. Genet. dan Tabu Search*, vol. 1, 2013.
- [9] H. A. Abdulkarim and I. F. Alshammari, "Comparison of Algorithms for Solving Traveling Salesman Problem," *Int. J. Eng. Adv. Technol.*, vol. 4, no. 6, pp. 76–79, 2015.
- [10] A. Lukman, R. AR, and Nurhayati, "Penyelesaian Travelling Salesman Problem dengan Algoritma Greedy," vol. 04, no. December 2011, pp. 1–5, 2011.
- [11] F. Amri, E. B. Nababan, and M. F. Syahputra, "Artificial Bee Colony Algorithm untuk Menyelesaikan Travelling Salesman Problem," *J. Dunia Teknol. Inf.*, vol. 1, no. 1, pp. 8–13, 2012.
- [12] S. Lukas, T. Anwar, and W. Yuliani, "Penerapan Algoritma Genetika Untuk Traveling Salesman Problem Dengan Menggunakan Metode Order Crossover Dan Insertion Mutation," *Semin. Nas. Apl. dan Teknol. Inf. (SNATI 2005)*, vol. 2005, no. Snati, pp. 1–5, 2005.
- [13] Fatmawati, B. Prihandono, and E. Noviani, "Penyelesaian Travelling Salesman Problem Dengan Metode Tabu Search," *Bul. Ilm. Mat. Stat. Dan Ter.*, vol. 04 no. 1, no. 1, pp. 17–24, 2015.
- [14] R. N. Hay's, "Implementasi Firefly Algorithm Tabu Search Untuk Penyelesaian Traveling Salesman Problem," *J. Online Inform.*, vol. 2, no. 1, pp. 42–48, 2017, [Online]. Available: <http://join.if.uinsgd.ac.id/index.php/join/article/view/v2i18/51>.
- [15] S. Rani, Y. A. Kurnia, S. N. Huda, and S. A. S. Ekamas, "Smart travel itinerary planning application using held-karp algorithm and balanced clustering approach," *ACM Int. Conf. Proceeding Ser.*, pp. 106–110, 2019, doi: 10.1145/3377817.3377847.
- [16] C. Nilsson, "heuristics for the traveling salesman problem," *Math. Program.*, vol. 19, no. 1, pp. 111–114, 2003, doi: 10.1007/BF01581633.
- [17] A. Maria, E. Y. Sinaga, and M. H. Iwo, "Penyelesaian Masalah Travelling Salesman Problem Menggunakan Ant Colony Optimization (ACO) Departemen Teknik Informatika , Institut Teknologi Bandung."
- [18] Y. D. Prasetyo, "Penyelesaian Travelling Salesman Problem dengan Algoritma Branch and Bound," vol. 04, no. Maret 2017, pp. 1–5, 2017.
- [19] W. Firdaus Mahmudy, "Algoritma Evolusi," no. September, 2013.
- [20] N. S. Dini, "Produk Air Minum Kemasan Galon Menggunakan Kombinasi Algoritma Genetika Dan Pencarian Tabu Di Depot Air Minum Isi Ulang Banyu Belik," 2015.
- [21] J. Melorose, R. Perroy, and S. Careas, "Bab 7 Algoritma Genetika," *Statew. Agric. L. Use Baseline 2015, 1.*, 2015, [Online]. Available: <https://doi.org/10.1017/CBO9781107415324.004>.
- [22] N. D. Priandani and W. F. Mahmudy, "Optimasi Travelling Salesman Problem With Time Windows (Tsp-Tw) Pada Penjadwalan Paket Rute Wisata," *Semin. Nas. Sist. Inf. Indones.*, no. November, pp. 2–3, 2015.
- [23] E. Fernando, "Kombinasi Algoritma Genetik Dan Tabu Search," vol. 6, no. 1, 2012.



# Perancangan Kombinasi Algoritma Genetika dan Tabu Search untuk Penyelesaian Travelling Salesman Problem

## ORIGINALITY REPORT

15%

SIMILARITY INDEX

14%

INTERNET SOURCES

5%

PUBLICATIONS

0%

STUDENT PAPERS

## PRIMARY SOURCES

1	<a href="http://docplayer.info">docplayer.info</a> Internet Source	2%
2	<a href="http://id.123dok.com">id.123dok.com</a> Internet Source	2%
3	<a href="http://ejournal.bsi.ac.id">ejournal.bsi.ac.id</a> Internet Source	1%
4	<a href="http://etheses.uin-malang.ac.id">etheses.uin-malang.ac.id</a> Internet Source	1%
5	<a href="http://www.ejurnal.stmik-budidarma.ac.id">www.ejurnal.stmik-budidarma.ac.id</a> Internet Source	1%
6	<a href="http://repository.unair.ac.id">repository.unair.ac.id</a> Internet Source	1%
7	<a href="http://eprints.uty.ac.id">eprints.uty.ac.id</a> Internet Source	<1%
8	<a href="http://repositori.usu.ac.id">repositori.usu.ac.id</a> Internet Source	<1%
9	<a href="http://eprints.upnjatim.ac.id">eprints.upnjatim.ac.id</a>	

10

Xiaopeng Yuan, Tianyu Yang, Yulin Hu, Jie Xu, Anke Schmeink. "Trajectory Design for UAV-Enabled Multiuser Wireless Power Transfer with Nonlinear Energy Harvesting", IEEE Transactions on Wireless Communications, 2020

Publication

<1%

11

[eprints.umm.ac.id](http://eprints.umm.ac.id)

Internet Source

<1%

12

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

Internet Source

<1%

13

[www.neliti.com](http://www.neliti.com)

Internet Source

<1%

14

[journals.usm.ac.id](http://journals.usm.ac.id)

Internet Source

<1%

15

[fmipa.unmul.ac.id](http://fmipa.unmul.ac.id)

Internet Source

<1%

16

[onesearch.id](http://onesearch.id)

Internet Source

<1%

17

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

<1%

18

Syafiul Muzid. "An Adaptive Approach to Controlling Parameters and Population Size of Evolutionary Algorithm", Journal of Physics: Conference Series, 2020

Publication

<1%

---

19

123dok.com

Internet Source

<1%

---

20

www.scribd.com

Internet Source

<1%

---

21

join.if.uinsgd.ac.id

Internet Source

<1%

---

22

unisbank.ac.id

Internet Source

<1%

---

23

www.astesj.com

Internet Source

<1%

---

24

text-id.123dok.com

Internet Source

<1%

---

25

boris.unibe.ch

Internet Source

<1%

---

26

James Sparkes, Chris Holland. "The rheological properties of native sericin", Acta Biomaterialia, 2018

Publication

<1%

---

27 [ejurnal.its.ac.id](http://ejurnal.its.ac.id)  
Internet Source

<1%

28 [pt.scribd.com](http://pt.scribd.com)  
Internet Source

<1%

29 Erwin Yulianto, Awan Setiawan. "OPTIMASI RUTE SALES COVERAGE MENGGUNAKAN ALGORITMA CHEAPEST INSERTION HEURISTIC DAN LAYANAN GOOGLE MAPS API", INTERNAL (Information System Journal), 2018  
Publication

<1%

Exclude quotes On

Exclude matches Off

Exclude bibliography On