

# Penerapan Algoritma Genetika untuk Menentukan Rute Perjalanan Dalam Sistem Rekomendasi Pariwisata

## *Application of Genetic Algorithm for Determining Travel Route in Tourism Recommendation System*

Fajar Hendra, Dr. Kemas Muslim L, Dr. Z.K Abdurahman Baizal.

<sup>123</sup> Prodi S1 Teknik Informatika, Fakultas Informatika, Universitas Telkom.

<sup>1</sup>[fajarahendra@students.telkomuniversity.ac.id](mailto:fajarahendra@students.telkomuniversity.ac.id), <sup>2</sup>[kemasmuslim@telkomuniversity.ac.id](mailto:kemasmuslim@telkomuniversity.ac.id),

<sup>3</sup>[baizal@telkomuniversity.ac.id](mailto:baizal@telkomuniversity.ac.id)

---

### Abstrak

*Travelling Salesman Problem (TSP) merupakan penggambaran kasus untuk optimasi jarak atau waktu dalam teori ilmu komputer, TSP dapat digunakan untuk optimasi jarak terpendek atau waktu tercepat, pada penelitian ini TSP digunakan untuk penentuan rute pariwisata yang optimal sekitar Bandung Raya. Dalam penelitian ini variabel optimal adalah rute terpendek (berdasarkan waktu) dan jumlah wisata per hari yang dapat dihasilkan. Untuk permasalahan TSP akan menggunakan Algoritma Genetika (AG), AG salah satu algoritma yang sudah banyak diketahui dan dipelajari dan dapat menyelesaikan permasalahan Travelling Salesman Problem. AG dapat menemukan solusi terbaik dari banyak kemungkinan rute-rute yang ada dengan running time yang cepat, berbeda dengan algoritma bruteforce yang pasti menemukan solusi terbaik namun memiliki running time yang sangat lama. Dengan alasan yang telah dijelaskan, menggunakan AG untuk penelitian ini diharapkan dapat mempermudah pembaca untuk memahami penelitian ini karena pembaca dapat fokus pada pengembangan penggunaan AG pada penelitian ini. Pada tahap awal user diharuskan memilih tempat penginapan karena akan berpengaruh terhadap hasil perutean, setelah itu user akan menentukan destinasi-destinasi wisata yang diinginkan dan selanjutnya akan diproses dengan AG. Proses penjadwalan dan penentuan rute akan melewati beberapa tahapan encoding, fitness, selection, crossover, mutation, survivor selection, dengan output yang diharapkan sistem dapat membuat penjadwalan rute wisata untuk user dan dapat memaksimalkan banyaknya wisata untuk setiap harinya. Terdapat dua pengujian dengan pengujian pertama perbandingan hasil GA dengan GA bertingkat, dan percobaan kedua perbandingan GA bertingkat dengan bruteforce. Hasil evaluasi untuk percobaan GA bertingkat didapatkan hasil terbaik pengurangan waktu pada hari pertama sebesar 108 menit. Untuk pengujian perbandingan GA bertingkat dengan bruteforce didapatkan perbedaan waktu total wisata hanya 1 menit.*

**Kata kunci :** pariwisata, tsp, algoritma genetika, sistem rekomendasi, penjadwalan.

---

### Abstract

Traveling Salesman Problem (TSP) represents the case for distance or time optimization in computer science theory, TSP can be used for shortest distance optimization or fastest time, in this research TSP is used for determination of optimal tourism route around Bandung Raya. In this study the optimal variable is the shortest route (based on time) and the number of tours per day that can be generated. For TSP problems will use Genetic Algorithm (AG), AG one of the algorithms that have been widely known and studied and can solve problems Traveling Salesman Problem. AG can find the best solution of many possible routes with fast running times, unlike the bruteforce algorithms that definitely find the best solution but have a very long running time. For that reason, using AG for this study is expected to facilitate the reader to understand this research because the reader can focus on developing the use of AG in this study. In the early stages the user is required to choose a lodging because it will affect the routing results, after which the user will determine the desired tourist destinations and will then be processed with AG. The process of scheduling and determining the route will pass through several stages of encoding, fitness, selection, crossover, mutation, survivor selection, with the expected output of the system making scheduling routes for the user and maximizing the number of tours for each day. There are two tests with the first test of comparison of GA results with multilevel GA, and a second experiment of stratified GA comparisons with bruteforce. Evaluation results for the multilevel GA experiment obtained the best result of reduction of time on the first day by 108 minutes. For GA-level comparison test with bruteforce was found that the total time difference was only 1 minute.

**Keywords:** tourism, tsp, genetic algorithm, recommended system, scheduling.

---

## 1. Pendahuluan

## 1.1 Latar Belakang

Kota Bandung memiliki beragam tempat pariwisata dengan keunikan masing-masing, berdasarkan data kunjungan wisata yang didapatkan wisatawan yang menghabiskan waktu di kota Bandung setiap tahun mengalami kenaikan. Wisatawan yang berkunjung ke kota Bandung berasal dari dalam dan luar negeri dengan rata-rata kunjungan selama 2 hari [1, 2]. Karena itu pengembangan suatu aplikasi untuk sektor ini akan sangat membantu wisatawan untuk memberikan informasi serta memaksimalkan kunjungan wisata di kota Bandung.

Travelling Salesman Problem (TSP) adalah kasus yang sering dijumpai untuk menyelesaikan permasalahan-permasalahan yang ada sehari-hari seperti pengiriman barang, penjadwalan mata pelajaran, jalur pemasangan kabel atau pipa gas dan dalam penelitian ini membahas tentang penjadwalan rute wisata untuk wilayah Bandung Raya.

Algoritma yang digunakan untuk kasus TSP ini adalah Algoritma Genetika (AG), algoritma yang memiliki metode perhitungan yang tidak sulit, tetapi memiliki kapabilitas dalam pencarian data. Dalam hal ini AG dapat digunakan dalam penyelesaian permasalahan Travelling Salesman Problem karena permasalahan TSP membutuhkan pencarian solusi (dalam hal ini pencarian rute pariwisata) berdasarkan tujuan atau tempat wisata yang telah ditentukan oleh user tanpa harus mencoba keseluruhan kemungkinan solusi yang ada. Selain itu AG dapat menentukan rute optimal dari tujuan pariwisata dengan masing – masing tempat wisata hanya akan dikunjungi sekali, dan waktu kunjungan pada setiap tempat wisata juga sangat diperhitungkan, untuk itu pemilihan GA menjadi salah satu solusi terbaik untuk masalah ini [5, 8, 11].

Pembuatan sistem bertujuan untuk memberikan penjadwalan rute pariwisata yang optimal, dalam hal ini nilai optimal yang dimaksud adalah aplikasi akan memaksimalkan jumlah wisata yang dikunjungi pada awal hari wisatawan mencari rute terpendek setiap harinya. Total hari kunjungan tempat-tempat wisata maksimal 3 hari, jam 8 malam adalah waktu yang untuk user sudah kembali berada di hotel.

## 1.2 Topik dan Batasannya

Terkait permasalahan yang dipaparkan, dirumuskan masalah yang diangkat dalam penelitian ini, yang pertama adalah bagaimana mendesain penjadwalan rute pariwisata yang optimal menggunakan algoritma genetika? Dan bagaimana mendesain sistem rekomendasi untuk menentukan penjadwalan rute pariwisata?

## 1.3 Tujuan

Tujuan yang dicapai dalam penelitian ini, yaitu Mendesain penjadwalan rute pariwisata yang optimal menggunakan algoritma genetika. Mendesain sistem rekomendasi untuk menentukan penjadwalan rute pariwisata.

## 1.4 Metoda Penelitian

Dalam pengerjaan penelitian ini terdapat beberapa metodologi penyelesaian masalah, yaitu :

1. Studi Literatur Proses pengumpulan, pencarian dan pemahaman literatur yang terkait dengan Algoritma Genetika, Travelling Salesman Problem, Google Maps API, Bahasa Pemrograman python, dan pemrograman web. Pengumpulan Informasi literatur berasal dari paper, buku, dan situs web.
2. Identifikasi Kebutuhan Mengidentifikasi perangkat keras, perangkat lunak dan kebutuhan sistem untuk membangun aplikasi pada tugas akhir ini.
3. Pengumpulan Data Pengumpulan data berasal dari survei yang dilakukan pada objek-objek wisata yang ada pada wilayah Bandung Raya, dan data yang berasal dari google API.
4. Perancangan dan Pembuatan Sistem Membangun database yang berisi data-data wisata Bandung raya dan menggunakan algoritma genetika dalam penjadwalan rute wisata, aplikasi yang dibangun dapat menerima input user berupa tempat-tempat tujuan wisata dan tempat penginapan yang disediakan oleh aplikasi. Seluruh tempat wisata yang terpilih akan melewati proses algoritma genetika untuk mencari jalur terpendek dan penjadwalan rute wisata dengan maksimal 3 hari perjalanan wisata.
5. Implementasi Mengimplementasikan seluruh rancangan pada tahap sebelumnya untuk membuat aplikasi tugas akhir.
6. Pengujian dan Analisis Menguji aplikasi yang telah dibangun untuk mengetahui apakah aplikasi yang telah dibangun telah sesuai dengan dengan rancangan yaitu apakah rute yang dihasilkan telah optimal.
7. Pembuatan Laporan dan Dokumentasi Membuat laporan dan dokumentasi dari hasil pembangunan aplikasi pada tugas akhir ini.

## 2. Dasar Teori dan Metodologi

### 2.1 Traveling Salesman Problem

*Traveling Salesman Problem (TSP)* merupakan permasalahan yang bertujuan untuk menemukan jalur atau rute yang optimal dengan syarat bahwa setiap tempat hanya dapat dikunjungi satu kali dan rute diharuskan

kembali ke tempat awal . Permasalahan yang dapat ditangani oleh TSP diantaranya pada bidang pengiriman logistik dan surat kabar (POS), TSP juga dapat menangani masalah dalam bidang teknologi seperti bidang robotik, games, pemasangan saluran pipa dan lain-lain [3].

TSP merupakan salah satu persoalan optimasi kombinatorial, jika terdapat sejumlah kota (atau tempat) dan Biaya perjalanan dari satu kota ke kota lain, mendeskripsikan persoalannya adalah bagaimana menemukan rute perjalanan paling dekat dari suatu kota dan mengunjungi semua kota lainnya. Kombinasi dari semua rute perjalanan yang ada adalah faktorial dari jumlah kota. Biaya perjalanan bisa berupa jarak, waktu, bahan bakar, kenyamanan, dan sebagainya.

## **2.2 Algoritma Genetika (AG)**

AG Dapat dikategorikan sebagai teknik pencarian yang telah dioptimasi berdasarkan prinsip rekayasa genetika, dalam hal ini AG dapat digunakan dalam penyelesaian permasalahan Travelling Salesman Problem, menentukan rute optimal dari tujuan pariwisata dengan masing – masing tempat wisata hanya dikunjungi sekali, dan waktu kunjungan pada setiap tempat wisata juga sangat diperhitungkan, untuk itu pemilihan AG menjadi salah satu solusi terbaik untuk masalah ini [8].

## **2.3 Seleksi Induk**

Pemilihan induk direpresentasikan sebagai mesin Roulette Wheel dimana setiap kromosom pada 1 generasi memiliki bagian-bagian pada papan berbentuk lingkaran, besarnya 1 bagian kromosom tergantung dari besarnya nilai Fitness, lalu cara kerjanya adalah dengan memutar papan lingkaran tersebut, bagi kromosom yang terpilih (yang berhenti pada 1 titik) akan di pilih menjadi induk, begitu seterusnya untuk kromosom lainnya.

## **2.4 Crossover**

Partial-Mapped Crossover adalah algoritma Crossover yang juga menggunakan Multi Point Crossover, algoritma ini digunakan untuk permasalahan yang mengharuskan gen dari kromosom tidak boleh sama, cara kerja algoritma sama seperti Multi Point Crossover namun setelah ke 2 kromosom saling bertukar gen, yang harus dilakukan selanjutnya adalah melihat kembali isi dari setiap kromosom apakah terdapat gen yang bernilai sama dan gen yang hilang. Jika salah satu atau kedua kromosom memiliki gen yang bernilai sama dan gen yang hilang maka setiap kromosom baru (telah ditukar) harus melihat kembali isi kromosom sebelum ditukar.

## **2.5 Mutasi**

Swap Mutasi Algoritma ini membangkitkan 2 bilangan random, yang nantinya digunakan untuk memilih gen-gen berdasarkan bilangan random tersebut, setelah dipilih maka gen ditukar.

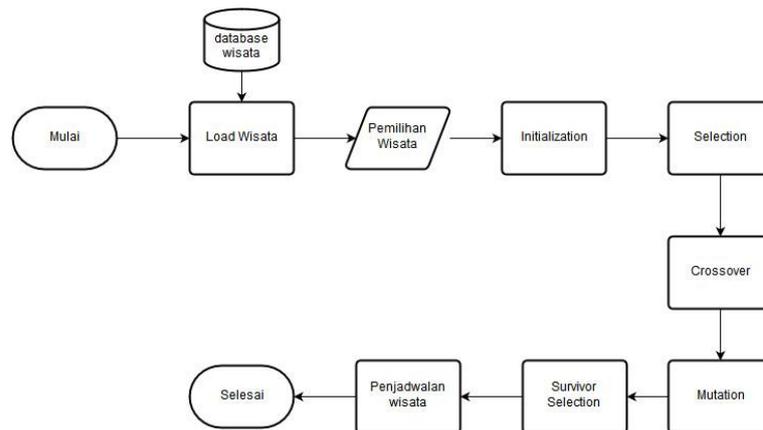
## **2.6 Seleksi Survivor**

Algoritma General Replacement menyimpan 1 atau 2 jumlah kromosom dari populasi lama yang memiliki nilai Fitness terbesar, kromosom yang terpilih ini akan menggantikan sejumlah kromosom yang telah terpilih dipopulasi baru dengan Fitness terendah

# **3. Pembahasan**

## **3.1 Gambaran Umum Sistem**

Aplikasi yang dibangun berupa aplikasi website bertujuan dapat membantu wisatawan untuk berwisata dikota bandung, aplikasi akan menangani permasalahan penentuan rute wisata dengan output yang dapat membuat penjadwalan rute wisata. Aplikasi dapat menangani kondisi saat user ingin mengunjungi tempat-tempat wisata dari tempat penginapan dan kembali lagi ke tempat penginapan yang sama. Sistem yang dibangun akan memiliki lima tahapan utama yaitu Initialization, Selection, Crossover, Mutation, Survivor Selection Proses sistem secara umum akan dijelaskan sebagai berikut,



Gambar 1 Gambaran Umum Sistem

Gambar 2

### 3.2 Parameter Pengujian

Pengujian dilakukan dengan parameter nKromosom = 50, nGenerasi = 150, pc = 0.7, pm = 0.3 parameter yang digunakan berasal dari data pengujian yang terlampir, dan pengujian untuk bab IV bertujuan untuk mengetahui seberapa baik performasi dari algoritma genetika dalam menentukan rute wisata yang optimal seperti yang telah dijelaskan pada BAB I. Tujuan dari pengujian ini adalah menguji kemampuan algoritma genetika untuk menentukan penjadwalan rute wisata dan mengetahui seberapa optimal solusi yang dihasilkan algoritma genetika dibandingkan dengan hasil algoritma bruteforce.

### 3.3 Skenario 1 (Proses GA bertingkat)

Proses yang kita ketahui berdasarkan penjelasan sebelumnya, setelah menentukan tempat-tempat wisata adalah ke tahap pemrosesan oleh GA mulai dari initialization hingga pemilihan rute wisata, dan kita ketahui pada tahap pemilihan rute wisata terdapat variabel h(n) yaitu rute wisata pada hari ke-n, lalu bagaimana jika setiap harinya h(1) hingga h(3) kita proses kembali dengan GA dari initialization hingga Survivor Selection. Tujuan dari skenario pertama adalah mengetahui seberapa besar perbedaan waktu total wisata dan seberapa besar perbedaan running time pada proses GA tanpa menggunakan proses GA kembali untuk setiap harinya(h(1) hingga h(3)) dengan proses GA yang menggunakan proses GA kembali untuk h(1) hingga h(3). Pengujian pertama menggunakan 15 data tempat wisata dan hotel yang dipilih oleh penulis.

Percobaan	Hari ke-1		Hari ke-2		Hari ke-3	
	H1 a	H1 b	H2 a	H2 b	H3 a	H3 b
1	0.603059	0.603284	0.504217	0.504477	0.209014	0.209014
2	0.605699	0.606108	0.502975	0.504238	0.304913	0.305076
3	0.503964	0.504560	0.503547	0.504806	0.307054	0.307165
4	0.602959	0.603075	0.504189	0.504614	0.205183	0.205183
5	0.605165	0.606518	0.502760	0.503487	0.303452	0.303481
6	0.603874	0.604208	0.404410	0.404775	0.307226	0.307226
7	0.603737	0.605894	0.504280	0.504560	0.303840	0.303840
8	0.604162	0.604601	0.502414	0.502556	0.307833	0.307833
9	0.503435	0.503617	0.504129	0.504336	0.307526	0.307526
10	0.605715	0.605749	0.504089	0.504530	0.303480	0.303536
11	0.603938	0.604178	0.502858	0.503228	0.208748	0.208774
12	0.504336	0.504730	0.504593	0.504612	0.304895	0.307545
13	0.603842	0.604088	0.504512	0.504537	0.209168	0.209168
14	0.603087	0.603230	0.504130	0.504612	0.308565	0.308565
15	0.504252	0.504357	0.504294	0.504656	0.406498	0.407328

Gambar 3 Hasil Fitness 15 kali percobaan

Untuk tabel diatas variabel yang berakhiran (a) adalah data H1, H2 atau H3 sebelum dilakukan proses GA, dan variabel yang berakhiran (b) adalah data dari H1, H2 atau H3 yang telah melewati proses GA kembali. Ditandai dengan kolom yang berwarna menunjukkan peningkatan nilai fitness setelah dilakukan proses GA. Dapat disimpulkan bahwa dari 15 kali percobaan menghasilkan jadwal rute wisata selama 3 hari dan hari ke-1 dan hari ke-2 selalu mengalami kenaikan fitness dan pada hari ke-3 hanya sebagian yang mengalami kenaikan fitness. Menaikannya nilai fitness berarti waktu yang dibutuhkan untuk wisata setiap harinya berkurang, pengurangan waktu perharinya dari 15 kali percobaan berbeda-beda, berikut tabel perbedaan waktu hasil dari tabel Hasil fitness 15 kali percobaan.

Percobaan	Hari ke-1	Hari ke-2	Hari ke-3
1	22 menit	14 menit	0 menit
2	12 menit	100 menit	6 menit
3	33 menit	73 menit	2 menit
4	12 menit	22 menit	0 menit
5	40 menit	75 menit	2 menit
6	21 menit	17 menit	0 menit
7	108 menit	4 menit	0 menit
8	23 menit	23 menit	0 menit
9	15 menit	8 menit	0 menit
10	1 menit	24 menit	5 menit
11	14 menit	40 menit	1 menit
12	19 menit	1 menit	72 menit
13	10 menit	1 menit	0 menit
14	14 menit	26 menit	0 menit
15	6 menit	18 menit	17 menit

**Gambar 4 Hasil perbedaan waktu perhari dari 15 kali percobaan**

Berdasarkan tabel hasil perbedaan waktu tersebut, pengurangan waktu terbesar selama percobaan mencapai 108 menit pada 1 hari hasil terdapat pada percobaan ke-7.

### 3.3.2 Skenario 2 (Perbandingan GA bertingkat dengan Bruteforce)

Pengujian skenario ini adalah untuk mengetahui perbandingan hasil rute wisata antara bruteforce dan GA pada skenario 1 (yang menggunakan proses GA kembali untuk meningkatkan hasil pada setiap harinya) yang dipilih berdasarkan fitness terbesar dari 20 kali running program. Hasil dari pengujian ini untuk mengetahui perbedaan fitness GA dengan bruteforce, mengetahui perbedaan waktu kunjungan wisata dari hasil GA dengan bruteforce dan jumlah wisata yang didapatkan dari ke dua algoritma tersebut. Untuk pengujian menggunakan 10 data tempat wisata yang dipilih oleh penulis dan hotel yang akan dipakai sebagai tempat penginapan.

**Table 1 Tabel fitness dari hasil 20 kali percobaan GA**

Pengujian ke	Hasil Fitness GA
1	0.902084057
2	0.902016061
3	0.902096949
4	0.902005281
5	0.902120966
6	0.902126076
7	0.902086231
8	0.902076556
9	0.902109037
10	0.902062564
11	0.902014978
12	0.902212553
13	0.902082827
14	0.902188423
15	0.90221664
16	0.90210926
17	0.902118869
18	0.902016671
19	0.901987808
20	0.902188503
<i>fitness terbesar</i>	0.90221664
Rata-rata	0.902096015
<i>Fitness Bruteforce</i>	0.902329916

Dari hasil GA dan Bruteforce memiliki total jumlah wisata yang sama yaitu 9 tempat wisata walaupun pada hari ke-1 bruteforce memiliki solusi yang lebih baik memiliki 6 tempat wisata. Kemudian waktu total kunjungan wisata untuk 2 hari dari hasil bruteforce adalah 12 Jam 16 Menit untuk hari ke-1 dan 9 Jam 45 Menit untuk hari ke-2 dengan total waktu perjalanan selama 2 hari adalah 22 Jam 1 Menit, sedangkan waktu total kunjungan wisata untuk 2 hari dari hasil GA no.15 adalah 11 Jam 14 Menit untuk hari ke-1 dan 10 Jam 46 Menit untuk hari

ke-2 dengan total waktu perjalanan selama 2 hari adalah 22 Jam 2 Menit. Lalu untuk running time bruteforce dengan 10 tempat wisata membutuhkan waktu 20 Menit sedangkan proses GA bertingkat (memproses GA kembali setiap harinya) hanya membutuhkan waktu rata-rata 50 detik.

#### 4. Kesimpulan

Berdasarkan hasil pengujian dan analisis pada penelitian ini, maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Skenario pertama menunjukkan bahwa penerapan algoritma genetika dalam menentukan rute wisata dengan banyak kromosom 50 dan banyak generasi 150 menghasilkan solusi yang optimal. Hasil rata-rata tempat wisata yang didapatkan dari AG adalah 14 wisata dari 15 wisata, jumlah kunjungan wisata hari pertama akan lebih atau sama dengan hari kedua, begitupun dengan jumlah kunjungan wisata hari kedua akan lebih atau sama dengan hari ketiga. Dengan dilakukannya pengujian skenario pertama membuktikan bahwa pengurangan waktu kunjungan wisata perharinya selalu terjadi pada hari ke 1 dan hari ke 2 (untuk kasus 3 hari penjadwalan wisata) dengan pengurangan waktu yang berbeda-beda (waktu pengurangan terkecil 1 menit hingga terbesar 108 menit). Namun semakin banyak wisata yang diproses semakin bertambah pula running time yang diperlukan akibat pengulangan proses AG.
2. Berdasarkan 20 kali percobaan dan dari hasil percobaan no.15 pada skenario kedua menunjukkan bahwa algoritma genetika dapat menghasilkan waktu total penjadwalan wisata selama 2 hari yaitu 22 jam 2 menit hampir sama dengan waktu total yang dihasilkan oleh bruteforce yaitu 22 jam 1 menit memiliki perbedaan hasil sekitar 1 menit. Kunjungan wisata hasil algoritma genetika dan bruteforce memiliki perbedaan, dalam hal ini bruteforce memiliki hasil lebih optimal karena pada hari ke 1 menghasilkan 6 kunjungan wisata sedangkan algoritma genetika menghasilkan 5 kunjungan wisata.

#### 5.2 Saran

Untuk penelitian lebih lanjut, saran yang dapat diberikan dan dipertimbangkan adalah

1. Menemukan algoritma atau cara untuk mengurangi running time akibat proses AG yang berulang.
2. Dalam pembuatan penjadwalan rute wisata memiliki penjadwalan yang lebih kompleks seperti, terdapat waktu jeda sholat (untuk muslim), makan siang, istirahat, dan lain-lain beserta lokasinya.

#### Daftar Pustaka

- [1] <https://ppid.bandung.go.id/knowledgebase/data-kunjungan-wisatawanyang-datang-ke-kota-bandung-tahun-2010-2015/>.
- [2] <http://www.disparbud.jabarprov.go.id/applications/frontend/index.php?mod=statistikwisatawan>.
- [3] Alves, R. M., and Lopes, C. R. Using genetic algorithms to minimize the distance and balance the routes for the multiple traveling salesman problem. In Evolutionary Computation (CEC), 2015 IEEE Congress on (2015), IEEE, pp. 3171–3178.
- [4] Chen, S.-H., and Chen, M.-C. Operators of the two-part encoding genetic algorithm in solving the multiple traveling salesmen problem. In Technologies and Applications of Artificial Intelligence (TAAI), 2011 International Conference on (2011), IEEE, pp. 331–336.
- [5] Chen, X., Li, Y., and Hu, T. Solving the supermarket shopping route planning problem based on genetic algorithm. In Computer and Information Science (ICIS), 2015 IEEE/ACIS 14th International Conference on (2015), IEEE, pp. 529–533.
- [6] Le, Q. T., and Pishva, D. Application of web scraping and google api service to optimize convenience stores' distribution. In 2015 17th International Conference on Advanced Communication Technology (ICACT) (2015), IEEE, pp. 478–482.
- [7] Liu-ai, W., and Wen-Qing, F. A parameter model of genetic algorithm regulating ant colony algorithm. In e-Business Engineering (ICEBE), 2012 IEEE Ninth International Conference on (2012), IEEE, pp. 50–54.
- [8] Muniandy, M. A., Mee, L. K., and Ooi, L. K. Efficient route planning for travelling salesman problem. In Open Systems (ICOS), 2014 IEEE Conference on (2014), IEEE, pp. 24–29.
- [9] Serpell, M., and Smith, J. E. Self-adaptation of mutation operator and probability for permutation representations in genetic algorithms. *Evolutionary Computation* 18, 3 (2010), 491–514.
- [10] Suyanto, A. I. Searching-reasoning-planning-learning (edisi revisi). Penerbit Informatika, Feb (2011).
- [11] Vahdati, G., Yaghoubi, M., Poostchi, M., et al. A new approach to solve traveling salesman problem using genetic algorithm based on heuristic crossover and mutation operator. In Soft Computing and Pattern Recognition, 2009. SOCPAR'09. International Conference of (2009), IEEE, pp. 112–116.
- [12] Widodo, A. W., and Mahmudy, W. F. Penerapan algoritma genetika pada sistem rekomendasi wisata kuliner. *Genetika (GA)* 3, 4 (2010), 5.

- [13] Yan, L., and Kongyu, Y. Immunity genetic algorithm based on elitist strategy and its application to the tsp problem. In Intelligent Information Technology Application Workshops, 2008. IITAW'08. International Symposium on (2008), IEEE, pp. 3–6.
- [14] Yu, Y., Chen, Y., and Li, T. A new design of genetic algorithm for solving tsp. In Computational Sciences and Optimization (CSO), 2011 Fourth International Joint Conference on (2011), IEEE, pp. 309–313.

