PERBANDINGAN ALGORITMA *BRANCH AND BOUND* DAN *PARTICLE SWARM OPTIMIZATION* PADA PENYELESAIAN *TRAVELING SALESMAN PROBLEM*

SKRIPSI

EKO SURA' KAPUANGAN

H011201032

PROGRAM STUDI MATEMATIKA

DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

MEI 2024

PERBANDINGAN ALGORITMA *BRANCH AND BOUND* DAN *PARTICLE SWARM OPTIMIZATION* PADA PENYELESAIAN *TRAVELING SALESMAN PROBLEM*

SKRIPSI

Diajukan sebagai salah satu syrarat untuk memperoleh gelar Sarjana Sains pada Program Studi Matematika Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin

EKO SURA' KAPUANGAN

H011201032

PROGRAM STUDI MATEMATIKA

DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

MEI 2024

LEMBAR PERNYATAAN KEOTENTIKAN

Saya yang bertanda tangan dibawah ini menyatakan dengan sungguh-sungguh bawha skripsi yang saya buat dengan judul:

Perbandingan Algoritma *Branch and Bound* dan *Particle Swarm Optimization* pada Penyelesaian *Traveling Salesman Problem*

Adalah benar hasil karya saya sendiri, bukan hasil plagiat dan belum pernah dipublikasikan dalam bentuk apapun.

Makassar, 20 Mei 2024

Eko Sura' Kapuangan

Nim. H011201032

PERBANDINGAN ALGORITMA *BRANCH AND BOUND* DAN *PARTICLE SWARM OPTIMIZATION* PADA PENYELESAIAN *TRAVELING SALESMAN PROBLEM*

Disetujui oleh:

Pembimbing Utama

Dr. Khaeruddin, M. Sc.

NIP. 196509141991031001

Pada 20 Mei 2024

HALAMAN PENGESAHAN

PERBANDINGAN ALGORITMA *BRANCH AND BOUND* DAN *PARTICLE SWARM OPTIMIZATION* PADA PENYELESAIAN *TRAVELING SALESMAN PROBLEM*

Disusun dan diajukan oleh:

Eko Sura' Kapuangan

H011201032

Telah dipertahankan dihadapan Panitia Ujian yang dibentuk dalam rangka penyelesaian Studi Program Sarjana Program Studi Matematika Fakultas Matematika dan Ilmu Pengetahuam Alam Universitas Hasanuddin pada tanggal 13 Mei 2024 dan dinyatakan telah memenuhi syarat kelulusan.

Menyetujui

Pembimbing Utama

Dr. Khaeruddin, M. Sc.

NIP. 196509141991031001

Ketua Program Studi

Dr. Firman, S. Si, M. Si.

NIP. 196804292002121001

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama : Eko Sura' Kapuangan

NIM : H011201032

Program Studi : Matematika

Judul Skrpsi : Perbandingan Algoritma *Branch and Bound* dan *Particle Swarm Optimization* pada Penyelesaian *Traveling Salesman Problem*

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian dari persyaratan yang diperluhkan untuk memperoleh gelar Sarjana Sains pada Program Studi Matematika dan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

DEWAN PENGUJI

1. Ketua : Dr. Khaeruddin, M. Sc. ( )
2. Anggota : Prof. Dr. Jeffry Kusuma Ph. D. ( )
3. Anggota : Dr. Firman, S. Si, M. Si. ( )

Ditetapkan di: Makassar

Tanggal: 20 Mei 2024

PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Hasanuddin, saya yang bertanda tangan dibawah ini:

Nama	: Eko Sura' Kapuangan
NIM	: H011201032
Program Studi	: Matematika
Departemen	: Matematika
Fakultas	: Matematika dan Ilmu Pengetahuan Alam
Jenis Karya	: Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty-Free Right) atas karya ilmiah saya yang berjudul:

"Perbandingan *Algoritma Branch and Bound* dan *Particle Swarm Optimization* pada *Penyelesaian Traveling Salesman Problem*"

Beserta perangkat yang ada (jika diperlukan). Terkait dengan hal diatas, maka dipihak universitas berhak menyimpan, mengalih-meda/format-kan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan tugas akhir saya selama tetap mncantumkan nama saya sebagai penulist/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya

Dibuat di Makassar Pada 20 Mei 2024

Yang menyatakan

Eko Sura' Kapuangan

ABSTRAK

Penelitian ini mengevaluasi penyelesaian masalah *Traveling Salesman Problem* (TSP) menggunakan dua pendekatan: metode heuristik *Branch and Bound* dan metode metaheuristik *Particle Swarm Optimization* (PSO). Dalam studi ini, data dihasilkan dengan jumlah kota yang bervariasi, yaitu 5, 10, 15, dan 20, untuk kasus simetri dan asimetri. Hasil penelitian menunjukkan bahwa metode *Branch and Bound* menghasilkan rute yang lebih optimal untuk jumlah kota yang kecil, tetapi PSO lebih efektif dalam hal efisiensi waktu komputasi. Selain itu, PSO memiliki kemampuan untuk menangani data berukuran lebih besar dibandingkan *Branch and Bound*, menjadikannya lebih fleksibel untuk masalah TSP berskala besar. Oleh karena itu, penelitian ini memberikan wawasan tentang kelebihan dan keterbatasan masing-masing metode, serta mengidentifikasi skenario aplikasi yang paling cocok untuk masing-masing pendekatan.

Kata kunci: *Traveling Salesman Problem, Particle Swarm Optimization, Branch and Bound, Heuristic, Metaheuristic,* Optimasi.

Judul	: Perbandingan Algoritma *Branch and Bound* dan *Particle Swarm Optimization* pada Penyelesaian *Traveling Salesman Problem*
Nama	: Eko Sura' Kapuangan
NIM	: H011201032
Program Studi	: Matematika

ABSTRACT

This study evaluates the solution to the Traveling Salesman Problem (TSP) using two approaches: the heuristic Branch and Bound (BB) method and the metaheuristic Particle Swarm Optimization (PSO) method. This study generated data with varying numbers of cities (5, 10, 15, and 20) for symmetric and asymmetric cases. The results show that the Branch and Bound method produces a more optimal route for a smaller number of cities, but PSO is more efficient in terms of time. Additionally, PSO can handle larger data than Branch and Bound, making it more flexible for large-scale TSP problems. Therefore, this research provides insights into the strengths and limitations of each method and identifies the most suitable application scenarios for each approach.

Key Words*: Traveling Salesman Problem, Particle Swarm Optimization, Branch and Bound, Heuristic, Metaheuristic, Optimization*

Title	: *Comparison of Branch and Bound & Particle Swarm Optimization on Solving the Traveling Salesman Problem*
Name	: Eko Sura' Kapuangan
NIM	: H011201032
Study Program	: *Mathematics*

KATA PENGANTAR

Segala puji bagi Tuhan YME yang telah melimpahkan memberikan berkat-Nya kepada penulis sehingga skripsi dengan judul “Perbandingan Algoritma *Branch and Bound* dan *ParticleSwarm Optimization* pada *Traveling Salesman Problem*. Skripsi ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Sains pada Program Studi Matematika Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

Penulis mengucapkan terima kasih khususnya kepada kedua orang tua Bapak **Simon Petrus** dan Ibu **Elisabeth** yang telah membesarkan dan mendidik penulis dengan penuh kesabaran serta selalu mencurahkan kasih sayang yang tak pernah putus, memberikan dukungan dan doa sehingga penulis dapat menyelesaikan penulisan skripsi ini. Begitu pula kepada adik penulis **Elva Sura’ Kapuangan** yang telah memberikan dukungan pada penulis. Terima kasih telah menjadi sumber motivasi dan kekuatan bagi penulis selama perjalanan akademik ini.

Penulis menyadari bahwa penyelesaian skripsi ini tidak lepas dari dukungan dan bantuan banyak pihak baik secara langsung maupun tidak langsung. Untuk itu pada kesempatan ini penulis mengucapkan terima kasih dan penghargaan yang setinggi-tingginya kepada:

1. Bapak **Prof. Dr. Ir. Jamaluddin Jompa, M.Sc.**, selaku Rektor Universitas Hasanuddin.

2. Bapak **Dr. Eng. Amiruddin, S.Si., M.Si.**, selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

3. Bapak **Dr. Khaeruddin., M.Sc.**, selaku pembimbing untuk segala ilmu, nasihat, dan kesabaran dalam membimbing dan mengarahkan penulis, serta bersedia meluangkan waktunya untuk mendampingi penulis sehingga skripsi ini dapat diselesaikan.

4. Bapak **Prof. Dr. Jeffry Kusuma Ph. D**., dan Bapak **Dr. Firman, S. Si, M. Si.**, selaku penguji dan penasihat akademik penulis yang telah bersedia meluangkan waktunya untuk memberikann saran dan arahan kepada penulis dalam penulisan skripsi ini.

5. Bapak/Ibu Dosen Departemen Matematika yang telah membagikan ilmu dan pengalamannya, serta Staf Departemen Matematika atas segala bantuannya.
6. Teman-teman Matematika 2020 atas segala dukungan, kebersamaan, dan kerjasamanya selama ini.
7. Teman-teman KKNTG-110 Posko Baji Mangngai atas segala dukungan, kebersamaan, serta pengalaman baru yang diperoleh selama proses KKN hingga saat ini.
8. Serta segala pihak yang tidak dapat disebutkan satu persatu yang telah membantu penulis dalam menyelesaikan skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, kritik dan saran yang membangun diharapkan oleh penulis untuk perbaikan dan pengembangan penelitian lebih lanjut. Akhir kata, semoga hasil penelitian ini dapat bermanfaat bagi kalangan akademisi, praktisi, dan semua pihak.

Makassar, 20 Mei 2024

Eko Sura' Kapuangan

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang

Optimasi merupakan bagian dari penerapan ilmu Matematika dalam menentukan solusi yang paling sesuai untuk suatu masalah yang diberikan oleh suatu keadaan. Definisi dari optimasi adalah suatu tindakan untuk mencapai hasil terbaik dari suatu keadaan tertentu yang mencari nilai minimum atau maximum. Optimasi dapat dijumpai di banyak bidang antara lain design, construction, maintenance dan lain-lain (J. Yi dkk 2009). Optimasi memegang peran penting dalam mendesain suatu sistem, dengan optimasi, bisa menghasilkan ongkos yang lebih murah atau profit yang tinggi. Pentingnya optimasi dapat dilihat dari dampaknya di sektor bisnis dan industri. Dengan menggunakan teknik optimasi, perusahaan dapat mengelola sumber daya mereka secara lebih efisien, mengurangi biaya produksi, dan meningkatkan keuntungan.

Masalah optimasi muncul di semua disiplin ilmu kuantitatif mulai dari ilmu komputer dan teknik hingga riset operasi dan ekonomi. Salah satu masalah optimasi yang cukup populer adalah *Traveling Salesman Problem* (TSP), dimana masalah TSP ini sering digambarkan sebagai seorang salesman yang harus mengunjungi sejumlah kota, masing-masing hanya sekali, dan kembali ke kota asal. Pada TSP ini terdapat beberapa kendala, kendala yang pertama yaitu memastikan bahwa dalam tur (rute) perjalanan seorang salesman, setiap kota diikuti oleh tepat satu kota. Demikian pula, kendala menentukan bahwa sebuah kota unik dikunjungi langsung sebelum kota lainnya. Kendala yang tersisa disebut sebagai kendala eliminasi subtur.

Berbagai metode telah berkembang untuk mengatasi masalah optimasi TSP, mulai dari metode deterministik, metode heuristik hingga metode metaheuristik. Metode deterministik adalah jenis metode yang memastikan bahwa solusi yang dihasilkan adalah solusi optimal dengan pendekatan tradisional, namun dalam persoalan TSP Ketika jumlah kota semakin besar maka metode deterministik sudah tidak efektif lagi (M. Sulkifly, dkk 2016), sehingga untuk mengeksplorasi TSP dengan jumlah kota yang lebih besar metode heuristik dan metode metaheuristik lebih disarankan. Lalu metode heuristik adalah jenis metode yang menggunakan aturan praktis atau pendekatan kasar untuk menemukan solusi yang memadai, tetapi tidak selalu optimal, salah satu contoh metode heuristik adalah *Branch and Bound.* Selanjutnya metode metaheuristik yaitu metode yang menggabungkan metode heuristik dasar untuk menjelajahi ruang pencarian, salah satu contoh metode metaheuristik adalah *Particle Swarm Optimization* (PSO). Setiap metode memiliki pendekatan yang berbeda-beda untuk mencapai Solusi TSP yang diinginkan (Desale, dkk 2015).

Metode *Branch and Bound*, berbagai penelitian telah dilakukan mengenai metode ini, beberapa diantaranya adalah *A Java Implementation of the Branch and Bound Algorithm: The Asymmetric Traveling Salesman Problem* oleh Pawel pada yang membahas implementasi java dalam penyelesaian masalah TSP Asimetrik dengan metode *branch and bound*. *Solving The Traveling Salesman Problem Using the Branch and Bound Method* oleh (Mataija, dkk 2016) tujuan dari paper ini untuk mengoptimalkan secara acak di kota Rijeka Dimana masalah ini berbentuk TSP. Kemudian *Implementation of Branch and Bound Algorithm to Solve the Travelling Salesman Problem at* PT Jasa Harapan Barat oleh A. A. S. Fetrisia dan N. Normalina, mengenai Penerapan *Branch and Bound* untuk memperoleh biaya distrbusi yang lebih murah untuk setiap pembagian minuman cap badak. Lalu penelitian *Branch and Bound Methods for the Traveling Salesman Problem* oleh Balas dkk, penelitian ini mengulas metode solusi enumeratif terkini untuk masalah TSP. Dan penelitian *Solving the Travelling Salesman Problem (TSP)* Using *Branch and Bound Method* (*Case Study at Company of XYZ*) oleh Fauzi dan Anwar, penelitian ini untuk mengetahui rute terbaik dengan jarak pengiriman terpendek, menggunakan *Branch and Bound* Metode penyelesaian *Traveling Salesman Problem*; dari distributor ke pengecer dan kembali ke distributor, setiap pengecer dapat dilewati sekali saja.

Penelitian mengenai penyelesaian masalah TSP menggunakan metode PSO, beberapa penelitian yang telah dilakukan sebelumnya diantaranya; penelitian *Discrete Particle Swarm Optimization for TSP*: *Theoretical Results and Experimental Evaluations* oleh Matthias Hoffmann dkk yang membahas analisis teoritis dari algoritma PSO diskrit. Lalu *Population size in Particle Swarm Optimization* yaitu penelitian untuk mendapatkan ukuran populasi yang ideal untuk permasalahan yang lebih kompleks, penelitian ini dilakukan oleh Piotrowski dkk. Selanjutnya terdapat penelitian *Particle Swarm Optimization An overview* oleh R. Poli dkk, pada penelitian ini terdapat ulasan mengenai temuan-temuan baru pada pengembangan PSO, dalam penelitian ini juga dibahas mengenai studi teoritis tentang efek berbagai parameter PSO, dari aspek algoritma juga terdapat sudut pandang penulis terhadap arus penelitian terkait PSO.

Selanjutnya terdapat juga berbagai penelitian yang membandingkan PSO dengan algoritma lain dalam optimasi TSP. seperti penelitian yang berjudul *Solve Traveling Salesman Problem Using Particle Swarm Optimization Algorithm*, pada penelitian ini oleh Xuesong Yan dkk membandingkan metode PSO dengan *Genetic Algorithm* untuk menunjukkan keefektifan algoritma PSO. Selanjutnya untuk TSP yang juga memberikan wawasan tentang perilaku konvergensi populasi. Lalu penelitian *Performance Comparison of Genetic Algorithm, Particle Swarm Optimization and Simulated Annealing Applied to TSP* yang dilakukan oleh Madhumita Panda, dimana ia membandingkan kinerja ketiga metode terrsebut dan mendapatkan kesimpulan bahwa PSO lebih efektif. Lalu penelitian mengenai *A Comparative Study of the Improvement of Performance Using A PSO modifed by ACO applied to TSP* oleh Elloumi dkk, pada penelitian ini menyajikan pendekatan baru dengan memperkenalkan PSO, yang dimodifikasi oleh algoritma ACO untuk meningkatkan kinerja. Metode hybrid baru (PSO-ACO) divalidasi menggunakan tolok ukur TSP dan hasil empiris dengan mempertimbangkan waktu penyelesaian dan durasi terbaik.

Setelah membaca berbagai penelitian mengenai Penyelesaian masalah TSP menggunakan pendekatan metode *Branch and Bound* maupun PSO, dari banyak penelitian yang telah dilakukan, belum ada penelitian yang membandingkan kinerja antara Metode *Branch and Bound* dengan metode PSO. Hal ini melatarbelakangi peneliti untuk membandingkan bagaimana kinerja dari kedua metode dalam optimasi TSP untuk menghasilkan Solusi yang lebih baik. Dengan demikian penulis mengambil judul penelitian “Perbandingan Algoritma *Branch and Bound* dan *Particle Swarm Optimization* pada Penyelesaian *Traveling Salesman Problem*”.

1.2 Rumusan Masalah

Rumusan masalah yang dibahas pada penelitian ini adalah solusi permasalahan Travelling Salesman Problem diperoleh menggunakan algoritma *Particle Swarm Optimization* dengan perbandingan menggunakan metode *Branch and Bound.* Secara lebih spesifik, penelitian ini mencakup

1. Bagaimana mengimplementasikan algoritma *Branch and Bound* untuk mengatasi persoalan *Traveling Salesman Problem*?
2. Bagaimana mengimplementasikan algoritma *Particle Swarm Optimization* untuk mengatasi persoalan *Traveling Salesman Problem*?

1.3 Batasan Masalah

Pembahasan pada penelitian ini terbatas pada:

1. Metode yang digunakan adalah *Particle Swarm Optimization* dan *Branch and Bound*,
2. Varian *Traveling Salesman Problem* yang akan dibahas Traveling Salesman Problem Simetri dan Traveling Salesman Problem Simetri Asimetri,
3. Metrik evaluasi yang digunakan berupa total jarak minimum yang diperoleh berdasarkan penerapan kedua algoritma dan efektifitas algoritma berdasarkan stabilitas dan waktu komputasi yang dibutuhkan.

1.4 Tujuan Penelitian

Berdasarkan Rumusan masalah, penelitian ini bertujuan untuk memproleh hasil perbandingan algoritma *Branch and Bound* dan *Particle Swarm Optimization* untuk mengatasi persoalan optimasi *Traveling Salesman Problem*

1.5 Manfaat Penelitian

Dengan adanya penelitian ini diharapkan dapat memberi beberapa manfaat positif dengan:

1. Menambah wawasan dan memberi pemahaman terkait penerapan algoritma *Branch and Bound* dan *Particle Swarm optimization* pada *Traveling Salesman Problem*
2. Mengetahui perbandingan kinerja antara kedua metode sehingga dapat menjadi potensi dalam pengembangan penelitian.
3. Sebagai bahan Pustaka di Universitas Hasanuddin

BAB II
TINJAUAN PUSTAKA

2.1 Traveling Salesman Problem

Permasalahan tentang *Traveling Salesman Problem* (TSP) dikemukakan pada tahun 1800 oleh matematikawan Irlandia William Rowan Hamilton dan matematikawan Inggris Thomas Penyngton. Traveling Salesman Problem ini adalah representatif dari sejumlah besar masalah yang dikenal sebagai masalah optimisasi kombinatorial. Masalah TSP ini sering digambarkan sebagai seorang salesman yang harus mengunjungi sejumlah kota, masing-masing hanya sekali, dan kembali ke kota asal. TSP juga dikenal sebagai suatu permasalah optimasi yang bersifat klasik dan *Non- Deterministic Polynomial-time Complete* (NPC), dimana tidak ada penyelesaian yang paling optimal selain mencoba seluruh kemungkinan penyelesaian yang ada sehingga tidak ada algoritma yang dapat menjamin solusi optimal dalam waktu yang wajar (Madona dkk, 2013)

Secara umum, TSP mencakup dua jenis yang berbeda, yaitu TSP Simetris (STSP) dan TSP Asimetris (ATSP). Dalam bentuk simetris yang dikenal sebagai STSP, hanya ada satu cara antara dua kota yang bersebelahan, yaitu jarak antara kota A dan B sama dengan jarak antara kota B dan A. Tetapi dalam ATSP (TSP Asimetris), tidak ada simetri seperti itu dan mungkin ada dua biaya atau jarak yang berbeda antara dua kota. Oleh karena itu, jumlah tur dalam ATSP dan STSP pada n titik (kota) masing-masing adalah $(\mathrm{n}-1)!$ dan $\frac{(\mathrm{n}-1)!}{2}$ (Greco 2008).

Tabel 2. 1 Contoh TSP Simetri

		Ke Kota				
		A	B	C	D	E
Dari Kota	A	0	2	1	4	4
	B	2	0	3	3	2
	C	1	3	0	6	1
	D	4	3	6	0	1
	E	4	2	1	1	0

Tabel 2. 2 Contoh TSP Asimetri

		Ke Kota				
		A	B	C	D	E
Dari Kota	A	0	1	7	4	3
	B	2	0	6	3	4
	C	1	6	0	2	1
	D	1	5	4	0	6
	E	7	5	4	5	0

Terdapat beberapa formulasi Matematis untuk TSP, salah satunya adalah due to Miller, Tuker, and Zemlin pada tahun 1960. Formulasi ini dimulai dengan memberi nomor kota dari 1 hingga n, dengan kota 1 ditetapkan sebagai kota asal.

dengan set indeks $I = \{i : i = 1,2,3, \dots, n\}$. Misalkan $x \in R^n$ adalah variabel keputusan yang didefenisikan sebagai berikut.

$$c_{ij} = \begin{cases} 1, jika\ kota\ j\ dikunjungi\ langsung\ setelah\ kota\ i \\ 0, \qquad jika\ tidak \end{cases} \qquad (2,1)$$

Misalkan x_{ij} menunjukkan jarak tempuh antara kota i dan j. Model optimasi kombinatorial dari TSP dapat di formulasikan sebagai berikut:

Minimalkan $\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$ (2.2)

fungsi tujuan: $\sum_{j=1, j \neq i}^{n} c_{i,j} = 1, i \in I, \sum_{i=1, i \neq j}^{n} c_{i,j} = 1, j \in I$ (2.3)

$t_i - t_j + n x_{ij} \leq n - 1,\ i, j \in I,\ t_i$ dan t_j adalah sembarang bilangan real.

2.2 *Branch and Bound*

Pemecahan masalah optimasi *Traveling Salesman Problem* merupakan pekerjaan yang membutuhkan algoritma yang efisien dan algoritma *Branch and Bound* merupakan salah satu metode untuk memecahkan masalah tersebut. Metode *Branch and Bound* ini dapat dengan mudah diterapkan pada TSP baik pada TSP Asimetris (ATSP) atau TSP Simetris (STSP) (Suyudi, dkk 2019).

Strategi *Branch and Bound* membagi masalah yang akan diselesaikan menjadi beberapa sub-masalah. Proses membagi masalah menjadi sub-sub masalah tersebut disebut branching dan untuk menyelesaikannya diperlukan batas (*bound*). Yang dimaksud batas adalah nilai fungsi tujuan, dalam masalah maksimalisasi menggunakan batas atas dan masalah minimalisasi menggunakan batas bawah. Karena ada nilai yang menjadi batas bawah sehingga dalam metode ini tidak perlu menghitung semua kemungkinannya. Selanjutnya pemecahan masalah dilakukan berdasarkan urutan sub-masalah, dimana masing-masing memiliki beberapa solusi yang mungkin dan di mana solusi yang dipilih untuk satu sub-masalah dapat memengaruhi solusi yang mungkin dari sub-masalah berikutnya. Kemudian proses percabangan akan berakhir jika sub-masalah tidak memiliki solusi layak atau penyelesaian sub-masalah tidak lebih baik dari solusi sebelumnya.

Untuk Langkah-langkah penyelesaiannya secara lebih lengkap akan dibahas pada sub bab 2.3.

2.3 Langkah-langkah dan Contoh Penyelesaian TSP dengan Metode Branch and Bound

Pada bagian ini membahas langkah-langkah metode *Branch and Bound* untuk menyelesaikan suatu permasalahan TSP. Langkah-langkah metode Branch and Bound ini disusun berdasarkan penelitian Mataija dkk pada tahun 2016 mengenai "*Solving the Traveling Salesman Problem Uing the Branch and Bound Method*".

Permasalahan yang dibahas menggunakan contoh permasalahan TSP pada tabel 2.2 yang akan diselesaikan menggunakan metode *branch and bound*. Pada Permasalahan TSP tersebut akan dicari rute dengan jarak minimum.

Tahap 1: Membuat tabel jarak, Jarak antar kota dinotasikan x_{ij}, x_{ij} berarti jarak antar kota i ke kota j . Tabel jarak yang digunakan berdasarkan tabel 2.1, namun kota yang tidak saling terhubung $x_{ij} = \infty$ sehingga $x_{ii} = \infty$ diperoleh tabel 2.3 seperti berikut

Tabel 2. 3 Jarak antar Kota

		Ke Kota				
		A	B	C	D	E
Dari Kota	A	∞	1	7	4	3
	B	2	∞	6	3	4
	C	1	6	∞	2	1
	D	1	5	4	∞	6
	E	7	5	4	5	∞

Tahap 2: berikutnya akan dicari nilai terkecil (r_i) pada setiap baris pada tabel 2.3

Tabel 2. 4 Pencarian (r_i)

		Ke Kota					r_i
		A	B	C	D	E	
Dari Kota	A	∞	1	7	4	3	1
	B	2	∞	6	3	4	2
	C	1	6	∞	2	1	1
	D	1	5	4	∞	6	1
	E	7	5	4	5	∞	4

Kemudian setiap baris (x_{ij}) pada tabel dikurangi dengan r_i, lalu dicari nilai terkecil setiap kolomnya, diperoleh hasil tabel 2.5:

Tabel 2. 5 $x''_{ij} = x_{ij} - r_i$

		Ke Kota				
		A	B	C	D	E
Dari Kota	A	∞	0	6	3	2
	B	0	∞	4	1	2
	C	0	5	∞	1	0
	D	0	4	3	∞	5
	E	3	1	0	4	∞
c_j		0	0	0	1	0

Selanjutnya setiap kolom (x'_{ij}) pada tabel dikurangkan dengan c_j, diperoleh tabel x''_{ij} sebagai berikut:

Tabel 2. 6 $x''_{ij} = x_{ij} - r_i - c_j$

		Ke Kota				
		A	B	C	D	E
Dari Kota	A	∞	0	6	2	2
	B	0	∞	4	0	2
	C	0	5	∞	0	0
	D	0	4	3	∞	5
	E	3	1	0	3	∞

Tahap 3: Menghitung batas bawah (b). Untuk menghitung batas bawah jarak untuk rute perjalanan menggunakan formula

$$b = \sum_{i=1}^{n} r_i + \sum_{j=1}^{n} c_j = 9 + 1 = 10$$

Menghitung batas bawah dilakukan untuk menghindari perhitungan tidak perlu pada tahap selanjutnya.

Tahap 4: Membuat percabangan (*branching*), Selanjutnya dilakukan pengecekan terhadap baris dan kolom untuk memperoleh rutenya. Pengecekan dilakukan berdasarkan persamaan

$\pi_{ij} = min_{j,}x''_{ij} + min_{i,}x''_{ij}$,

$min_{j,}x''_{ij}$ = nilai terkecil x''_{ij} terhadap kolom *j*

$min_{j,}x''_{ij}$ = nilai terkecil x''_{ij} terhadap baris *i*

Tabel 2. 7 Sub Problem 1

		Ke Kota				
		A	B	C	D	E
Dari Kota	A	∞	0	6	2	2
	B	0	∞	4	0	2
	C	0	5	∞	0	0
	D	0	4	3	∞	5
	E	3	1	0	3	∞

diperoleh solusi $z = x_{AB} + x_{BD} + x_{DA} + x_{CE} + x_{EC} = 1 + 3 + 1 + 1 + 4 = 10$ dengan rute yang dilalui yaitu A-B-D-A dan C-E-C. Walaupun $z \leq 10$ *(*memenuhi batas bawah), Jelas bahwa solusi yang diperoleh memiliki dua lintasan sehingga ini bukan bentuk solusi yang diharapkan. Kemudian lintasan C-E-C akan dipecahkan menjadi sub problem 2 (untuk lintasan C-E) dan *sub problem* 3 (untuk lintasan E-C) dengan $\mathrm{x_{CE}} = \mathrm{x_{EC}} = \infty$.

Tahap 5: Selanjutnya pada *sub problem* 2 dan 3 dilakukan langkah yang sama pada tahap 1 sampai 4:

Tabel 2. 8 Sub Problem 2

		Ke Kota				
		A	B	C	D	E
Dari Kota	A	∞	0	6	2	0
	B	0	∞	4	0	0
	C	0	5	∞	0	∞
	D	0	4	3	∞	3
	E	3	1	0	3	∞

untuk sub problem 2 solusi total jarak yang diperoleh yaitu $z = x_{DA} + x_{AB} + x_{BE} + x_{EC} + x_{CD} = 1 + 4 + 2 + 1 + 4 = 12$ dengan rute D-A-B-E-C-D

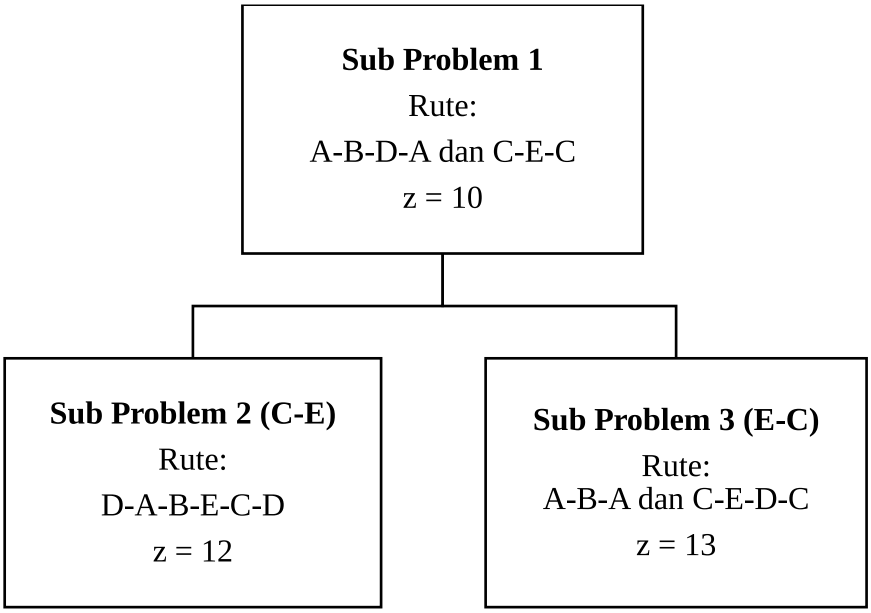
Tabel 2. 9 Sub Problem 2

		Ke Kota				
		A	B	C	D	E
Dari Kota	A	∞	0	3	3	2
	B	0	∞	1	1	2
	C	0	5	∞	1	0
	D	0	4	0	∞	5
	E	2	0	∞	0	∞

Selanjutnya pada tabel 2.9 untuk sub problem 3 dengan solusi total jarak yang diperoleh yaitu $z = x_{AB} + x_{BA} + x_{CE} + x_{ED} + x_{DC} = 1 + 2 + 1 + 4 + 5 = 13$ dengan rute perjalanan yang dilalui yaitu A-B-A dan C-E-D-C

Pada tahap ini dilakukan seleksi berdasarkan total jarak yang diperoleh dan rrute perjalanan yang memenuhi, yang lebih lengkap dibahas pada tahap 6

Tahap 6:



Gambar 2. 1 Pohon Pencarian

Karena solusi yang diperoleh pada sub problem 3 lebih besar dari sub problem dua dan lintasannya saling lepas (rutenya melalui semua kota) maka solusi yang diambil adalah sub problem

2.4 *Particle Swarm Optimization* (PSO)

Algoritma *Particle Swarm Optimization* (PSO) adalah algoritma pencarian berbasis populasi yang didasarkan pada simulasi perilaku sosial burung dalam kelompoknya. Niat awal dari konsep *Particle Swarm* adalah untuk mensimulasikan secara grafis koreografi yang anggun dan tidak terduga dari populasi burung, dengan tujuan untuk menemukan pola-pola yang mengatur kemampuan burung untuk terbang secara bersamaan, dan untuk tiba-tiba mengubah arah dengan pengelompokan dalam formasi optimal. Dari tujuan awal ini, konsep tersebut berkembang menjadi algoritma optimisasi yang sederhana dan efisien (Engelbrecht, 2007).

Dalam algoritma ini, kita memiliki populasi (*swarm*) yang sepenuhnya terhubung, yang berarti bahwa semua partikel berbagi informasi, setiap partikel mengetahui posisi terbaik yang pernah dikunjungi oleh setiap partikel dalam populasi. Setiap partikel memiliki posisi dan kecepatan yang dihitung berdasarkan persamaan 2.4 dam 2.5

$$v_{ij}^{(t+1)} = v_{ij}^{(t)} + c_1 r_{1j}^{(t)} \left[P_{best,j}^{(t)} - x_{ij}^{(t)} \right] + c_2 r_{2j}^{(t)} \left[G_{best,j}^{(t)} - x_{ij}^{(t)} \right], \quad (2.4)$$

$$x_{ij}^{(t+1)} = x_{ij}^{(t)} + v_{ij}^{(t+1)} \quad (2.5)$$

$v_{ij}^{(t)}$ = kecepatan partikel i pada dimensi j saat iterasi t

$v_{ij}^{(t+1)}$ = kecepatan partikel i pada dimensi j saat iterasi $t+1$

$x_{ij}^{(t)}$ = posisi partikel i pada dimensi j saat iterasi t

$x_{ij}^{(t+1)}$ = kecepatan partikel i pada dimensi j saat iterasi $t+1$

$P_{best,j}^{(t)}$ = posisi terbaik partikel pada iterasi t

$G_{best,j}^{(t)}$ = posisi terbaik yang pernah dicapi oleh partikel selama iterasi t

c_1, c_2 = koefisien percepatan

r_1, r_2 = koefisien learnitas

Konsep pada persamaan 2.4 kemudian mengalami perkembangan dengan pengunaan beban inersia. Konsep penggunaan beban inersia ini diperkenalkan oleh Shi dan Eberhart (1998) sebagai mekanisme untuk mengendalikan kemampuan eksplorasi dan eksploitasi dari populasi, dan sebagai mekanisme untuk menghilangkan kebutuhan untuk pelatuk kecepatan. Persamaan baru yang diperoleh dengan penggunaan inersia adalah

$$v_{ij}^{(t+1)} = w v_{ij}^{(t)} + c_1 r_{1j}^{(t)} \left[P_{best,j}^{(t)} - x_{ij}^{(t)} \right] + c_2 r_{2j}^{(t)} \left[G_{best,j}^{(t)} - x_{ij}^{(t)} \right] \quad (2.6)$$

w = beban inersia

Untuk mendapatkan rasio yang baik antara peningkatan kinerja dan keberhasilan algoritma dalam menemukan solusi yang diinginkan, nilai w

diaanjurkan berada di antara [0.9, 1.2]. Pengujian lebih lanjut menunjukkan bahwa nilai w yang berubah-ubah bahkan lebih meningkatkan rasio ini, dengan kasus pengujian dilakukan dengan penurunan linear dari w. Pembaruan inersia menggunakan persamaan(Poli, dkk. 2007):

$$w^t = w^{t-1} \times \alpha \tag{2.7}$$

w^t = beban inersia pada saat iterasi t

w^{t-1} = beban inersia pada saat iterasi $t-1$

α = decrement faktor

2.5 Particle Dasar dalam PSO

Dalam algoritma PSO, terdapat beberapa parameter yang dapat memengaruhi kinerjanya. Untuk masalah optimisasi tertentu, beberapa nilai dan pilihan parameter ini memiliki dampak besar pada efisiensi metode PSO, sementara parameter lainnya memiliki dampak kecil atau tidak berpengaruh sama sekali. Bagian ini membahas parameter-parameter tersebut ((Carlisle & Dozier, 2001)).

1. Ukuran populasi (Swarm Size)

Ukuran *swarm* atau ukuran populasi adalah jumlah partikel dalam kelompok yang dinotasikan sebagai n. ukuran *swarm* yang besar akan menghasilkan lebih banyak bagian dari ruang pencarian yang harus dicakup setiap iterasi. Jumlah partikel yang banyak dapat mengurangi jumlah iterasi yang diperlukan untuk mendapatkan hasil optimisasi yang baik. Namun, sebaliknya, jumlah partikel yang sangat besar akan meningkatkan kompleksitas komputasi setiap iterasi dan memakan lebih banyak waktu. Ukuran populasi yang dianjurkan adalah 20-50 untuk masalah yang sederhana, sedangkan masalah yang lebih kompleks dianjurkan mengunakan 70-500 partikel (Piotrowski, dkk. 2020).

2. Iterasi

Jumlah iterasi yang diperlukan untuk mendapatkan hasil yang baik juga tergantung pada masalah yang sedang dihadapi. Jumlah iterasi yang terlalu rendah dapat menghentikan proses pencarian terlalu dini, sedangkan jumlah iterasi yang terlalu besar akan mengakibatkan penambahan kompleksitas komputasi yang tidak perlu dan lebih banyak waktu yang dibutuhkan.

3. Komponen Kecepatan

Komponen kecepatan sangat penting dalam memperbarui kecepatan partikel. Terdapat tiga istilah dalam persamaan (2.6) yang menyusun kecepatan partikel: Istilah w disebut sebagai komponen inersia. Komponen ini mewakili momentum yang mencegah perubahan arah partikel secara drastis dan mengarah ke arah saat ini. Selanjutnnya $c_1 r_{1j}(t)\left[P_{best,j}(t) - x_{ij}(t)\right]$ disebut sebagai komponen kognitif yang mengukur kinerja partikel relatif terhadap kinerja masa lalu. Komponen ini mirip dengan ingatan individu terhadap posisi terbaik yang pernah dicapai oleh partikel. Efek dari komponen kognitif ini mewakili kecenderungan individu untuk kembali ke posisi yang ideal mereka di masa lalu. Lalu $c_2 r_{2j}(t)\left[G_{best,j}(t) - x_{ij}(t)\right]$ disebut sebagai komponen sosial yang mengukur kinerja partikel relatif terhadap sekelompok partikel atau tetangga. Efek dari komponen sosial ini adalah bahwa setiap partikel terbang menuju posisi terbaik yang ditemukan oleh tetangga partikel tersebut (Engelbrecht, 2007).

4. Koefisien Percepatan

Koefisien percepatan c_1 dan c_2, bersama dengan nilai-nilai acak r_1 dan r_2, menjaga pengaruh stokastik dari komponen kognitif dan sosial dari kecepatan partikel tersebut, masing. Konstan c_1 mengungkapkan sejauh mana kepercayaan yang dimiliki oleh partikel pada dirinya sendiri, sementara c_2 mengungkapkan sejauh mana kepercayaan yang dimiliki oleh partikel pada tetangganya (Engelbrecht, 2007). Terdapat beberapa ketentuan dari c_1 dan c_2:

1) Ketika $c_1 = c_2 = 0$, maka semua partikel akan terus bergerak dengan kecepatan saat ini hingga mencapai batas ruang pencarian. Oleh karena itu, dari persamaan (2.6), persamaan pembaruan kecepatan dihitung sebagai berikut:

$$v_{ij}(t+1) = v_{ij} \tag{2.8}$$

2) Ketika $c_1 > 0$ dan $c_2 = 0$ maka semua partikel saling independen. persamaan pembaruan kecepatan dihitung sebagai berikut:

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)\left[P_{best,j}(t) - x_{ij}(t)\right] \qquad (2.9)$$

3) Sebaliknya, ketika c_1 dan c_2 memiliki nilai yang besar, semua partikel akan tertarik ke satu titik tunggal dalam seluruh kelompok dan persamaan pembaruan kecepatan akan menjadi seperti berikut:

$$v_{ij}(t+1) = v_{ij}(t) + +c_2 r_{2j}(t)\left[G_{best,j}(t) - x_{ij}(t)\right] \qquad (2.10)$$

4) Ketika $c_1 = c_2 = 0$, Semua partikel tertarik menuju rata-rata. $P_{best,j}(t)$ dan G_{best}.
5) Ketika $c_1 \gg c_2$, setiap partikel lebih kuat dipengaruhi oleh posisi terbaik pribadinya, yang mengakibatkan pergerakan yang berlebihan. Sebaliknya, ketika $c_2 \gg c_1$ besar, maka semua partikel lebih dipengaruhi oleh posisi terbaik global, yang menyebabkan semua partikel bergerak terlalu cepat menuju optimum. Biasanya, c_1 dan c_2 bersifat statis, dengan nilai-nilai teroptimasi yang ditemukan secara empiris. Inisialisasi yang salah dari c_1 dan c_2 dapat mengakibatkan perilaku yang divergen atau siklik. Berdasarkan berbagai penelitian empiris, telah diusulkan bahwa dua konstanta percepatan tersebut sebaiknya memiliki nilai yang adalah $c_1 = c_2 = 2$.

5. Stop Kriteria

Kriteria penghentian digunakan untuk menghentikan proses pencarian berulang. Beberapa keriteria penghentian pada PSO seperti penghentian algoritma ketika jumlah iterasi maksimum atau evaluasi fungsi telah tercapai algoritma dihentikan bila tidak ada perbaikan yang signifikan selama beberapa iterasi. selain itu, penghentian algoritma juga dapat berasarkan jumlah maksimum iterasi, dimana algoritma dihentikan apabila iterasi iterasi maksimum telah tercapai.

Pada penelitian ini akan menggunakan maksimum iterasi sebagai stop kriteria implementasi algoritma.

2.6 Langkah-langkah dan Contoh Penyelesaian TSP dengan Metode PSO

Berikut adalah pennyelesaian contoh TSP pada tabel 2 dengan menggunakan metode PSO

Tahap 1: Inisialisasi parameter

Pada tahap awal dilakukan inisialisasi ukuran swarm (n), posisi awal, kecepatan awal c_1, c_2, r_1, r_2 dan w. Untuk menyelesaikan TSP pada tabel 2, mengikuti parameter $n = 5$, $w = 0{,}9$, $c_1 = c_2 = 2$, dan karena r_1 dan r_2 bilangan acak dalam rentang (0,1), maka pada contoh ini ditentukan $r_1 = 0{,}3$, $r_2 = 0{,}5$. Lalu menginisialisasi posisi awal dan kecepatan awal partikel.

Tabel 2. 10 Posisi Awal

		Dimensi				
		1	2	3	4	5
Particle	1	0,37	0,60	0,46	0,15	0,20
	2	0,13	0,89	0,86	0,30	0,97
	3	0,37	0,27	0,54	0,51	0,35
	4	0,76	0,46	0,09	0,89	0,73
	5	0,19	0,54	0,41	0,84	0,03

Posisi awal setiap partikel dibangkitkan secara acak sedangkan kecepatan awal setiap partikel diasumsikan 0,1.

Tabel 2. 11 Kecepatan Awal

		Dimensi				
		1	2	3	4	5
Particle	1	0,10	0,10	0,10	0,10	0,10
	2	0,10	0,10	0,10	0,10	0,10
	3	0,10	0,10	0,10	0,10	0,10
	4	0,10	0,10	0,10	0,10	0,10
	5	0,10	0,10	0,10	0,10	0,10

Tahap 2: penentuan Fitness dan penentuan $\mathrm{P}_{\mathrm{best,j}}(\mathrm{t})$ dan $\mathrm{G}_{\mathrm{best,j}}$, fungsi fitness pada TSP adalah total jarak yang dilalui pada setiap rute. Setelah menghitung total jarak selanjutnya ditentukan $\mathrm{P}_{\mathrm{best,j}}(\mathrm{t})$ setiap partikel kemudian particle dengan total jarak terkecil akan menjadi $\mathrm{G}_{\mathrm{best,j}}$

Tabel 2. 12 Evaluasi Fitness

Particle	Rute	Total jarak		
1	3-5-4-1-2-3	14	$\mathrm{P_{best,1}}(0)$	$\mathrm{G_{best}}(0)$
2	1-4-3-2-5-1	25	$\mathrm{P_{best,2}}(0)$	
3	3-1-5-4-2-3	20	$\mathrm{P_{best,3}}(0)$	
4	4-2-1-5-3-4	16	$\mathrm{P_{best,4}}(0)$	
5	2-4-3-5-1-2	16	$\mathrm{P_{best,5}}(0)$	

Tahap 3: update kecepatan dan update posisi, berdasarkan hasil yang diperoleh pada tahap 1 dan tahap 2, selanjutnya dilakukan update kecepatan dan update posisi, tahap ini adalah menjadi proses iterasi (1)

Iterasi (1)

Update kecepatan particle 1:

$$v_{ij}(t+1) = mv_{ij}(t) + c_1 r_{1j}(t)\left[P_{best,j}(t) - x_{ij}(t)\right] + c_2 r_{2j}(t)\left[G_{best,j}(t) - x_{ij}(t)\right]$$

$$v_{11}(1) = mv_{00}(0) + c_1 r_{1j}(0)\left[P_{best,1}(0) - x_{ij}(0)\right] + c_2 r_{2j}(0)\left[G_{best,1}(0) - x_{ij}(0)\right]$$

$$v_{11}(1) = (0{,}9)(0{,}1) + (1)(0{,}3)[0{,}15 - 0{,}37] + (1)(0{,}5)[0{,}15 - 0{,}37]$$

$$v_{11}(1) = -0{,}14$$

$$v_{12}(1) = (0{,}9)(0{,}1) + (1)(0{,}3)[0{,}20 - 0{,}60] + (1)(0{,}5)[0{,}20 - 0{,}60]$$

$$v_{12}(1) = -0{,}12$$

$$v_{13}(1) = (0{,}9)(0{,}1) + (1)(0{,}3)[0{,}37 - 0{,}46] + (1)(0{,}5)[0{,}37 - 0{,}46]$$

$$v_{13}(1) = 0{,}18$$

$$v_{14}(1) = (0{,}9)(0{,}1) + (1)(0{,}3)[0{,}46 - 0{,}15] + (1)(0{,}5)[0{,}46 - 0{,}15]$$

$$v_{14}(1) = 0{,}47$$

$$v_{15}(1) = (0{,}9)(0{,}1) + (1)(0{,}3)[0{,}60 - 0{,}20] + (1)(0{,}5)[0{,}60 - 0{,}20]$$

$$v_{15}(1) = 0{,}54$$

Update posisi partikel 1:

$$x_{ij}(1) = v_{ij}(1) + x_{ij}(0)$$

$$x_{11}(1) = v_{11}(1) + x_{11}(0) = -0{,}14 + 0{,}37$$

$$x_{11}(1) = 0{,}23$$

$$x_{12}(1) = v_{12}(1) + x_{12}(0) = -0{,}12 + 0{,}60$$

$$x_{12}(1) = 0{,}48$$

$$x_{13}(1) = v_{13}(1) + x_{13}(0) = 0{,}18 + 0{,}46$$

$$x_{13}(1) = 0{,}64$$

$$x_{14}(1) = v_{14}(1) + x_{14}(0) = 0{,}47 + 0{,}15$$

$$x_{14}(1) = 0{,}62$$

$$x_{15}(1) = v_{15}(1) + x_{15}(0) = 0{,}54 + 0{,}20$$

$$x_{15}(1) = 0{,}73$$

Untuk partikel 2 sampai particle 5 dilakukan update kecapatan dan update posisi dengan cara yang sama pada particle 1, diperoleh hasil pada tabbel 2.13 dan 2.14:

Tabel 2. 13 $x_{ij}(1)$

		Dimensi				
		1	2	3	4	5
Particle	1	0,23	0,48	0,64	0,62	0,73
	2	0,18	0,57	0,87	0,78	1,00
	3	0,27	0,46	0,65	0,71	0,75
	4	0,29	0,53	0,67	0,86	0,93
	5	0,16	0,46	0,64	0,40	0,77

Tabel 2. 14 $v_{ij}(1)$

		Dimensi				
		1	2	3	4	5
Particle	1	-0,14	-0,12	0,18	0,47	0,54
	2	0,05	-0,32	0,00	0,48	0,03
	3	-0,11	0,19	0,11	0,19	0,40
	4	-0,47	0,07	0,58	-0,03	0,20
	5	-0,03	-0,08	0,23	-0,44	0,74

Tahap 4: evaluasi Fitness, dari tabel yang diperoleh pada tahap ke 3 dilakukan evaluasi fitness dengan menghitung total jarak pada rute yang diperoleh (seperti pada tahap ke 2) lalu tentukan Pbest dan Gbest nya:

Tabel 2. 15 Evaluasi Fitness Iterasi 1

Particle	Rute	Total jarak		
1	1-2-4-3-5-1	16	$\mathrm{P}_{\mathrm{best},1}(1)$	$\mathrm{G}_{\mathrm{best}}(0)$
2	1-2-4-3-5-1	16	$\mathrm{P}_{\mathrm{best},2}(1)$	
3	1-2-3-4-5-1	22	$\mathrm{P}_{\mathrm{best},3}(1)$	
4	1-2-3-4-5-1	22	$\mathrm{P}_{\mathrm{best},4}(1)$	
5	1-3-4-2-5-1	25	$\mathrm{P}_{\mathrm{best},5}(1)$	

Pada tahap ini di evaluasi apakah *Gbest* yang diperoleh merupakan hasil minimum, namun hasil yang diperoleh ternyata lebih besar dari sebelumnya sehingga

Tahap 5: Pada tahap ini di evaluasi apakah Gbest yang diperoleh merupakan hasil minimum, namun hasil yang diperoleh ternyata lebih besar dari sebelumnya sehingga dilakukan update kecepatan dan posisi pada iterasi ke 2.

Iterasi (2)

Dengan melakukan cara yang sama pada tahap 3 dan tahap 4 diperoleh:

Tabel 2. 16 Evaluasi Fitness Iterasi 2

Particle	Rute	Total jarak		
1	1-2-3-4-5-1	22	$\mathrm{P}_{\mathrm{best},1}(2)$	
2	1-2-3-5-4-1	14	$\mathrm{P}_{\mathrm{best},2}(2)$	
3	1-2-3-4-5-1	22	$\mathrm{P}_{\mathrm{best},3}(2)$	
4	1-2-5-3-4-1	12	$\mathrm{P}_{\mathrm{best},4}(2)$	$\mathrm{G}_{\mathrm{best}}(2)$
5	1-3-4-2-5-1	25	$\mathrm{P}_{\mathrm{best},5}(2)$	

Berdasarkan tabel 2.16 diperoleh Gbest dengan total jarak 12 Sehingga dari 2 iterasi PSO diperoleh total jarak minimum 12 dengan rute 1-2-5-3-4-1