

Solusi Optimal Model Optimisasi *Robust* untuk Masalah *Traveling Salesman* dengan Ketidaktentuan Kotak dan Pendekatan Metode *Branch and Bound*

Poppy Amriyati¹, Diah Chaerani^{1*}, Eman Lesmana¹

Abstract: Traveling Salesman Problem (TSP) is route search technique that starts from a starting point, every city should be visited once and return to place of origin so that the total distance or travel time is minimum. To overcome the uncertainty of distance or travel time, it is necessary to develop a model of TSP. Optimization with uncertainty can be handle using Robust Optimization. In this paper some applications of Robust Optimization on TSP (RTSP) is discussed using what so-called Box Uncertainty approach and is solved via Branch and Bound Method. Numerical simulations on Maple application software is presented for some real cases related to application of RTSP such as construction management problem, determining the distance for traveling in Java Island and finding the optimal route for Mandiri Fun Run Route.

Keywords: Traveling salesman problem, robust optimization, branch and bound method, box uncertainty.

Pendahuluan

Traveling Salesman Problem (TSP) merupakan pencarian sebuah rute dari seorang *salesman* yang dimulai dari titik awal, mengunjungi beberapa kota yang ditentukan dan kembali ke tempat asal sedemikian sehingga total jarak perjalanan adalah minimum dan setiap kota dikunjungi hanya sekali (lihat Punnen [1]). Tonggak penyelesaian untuk TSP simetris dicapai oleh Dantzig, Fulkerson, dan Johnson pada tahun 1954 yang meneliti 42 kota hingga Applegate, Bixby, Chvatal, dan Cook kembali meneliti pada tahun 1998 dengan 13.509 kota (lihat Schrijver, [2]). Telah ditunjukkan dalam masalah TSP, jika terdapat $(n - 1)$ kota yang harus dikunjungi, maka jumlah rute yang mungkin terjadi adalah sebanyak $(n - 1)!$. Untuk jumlah kota yang sedikit, rute dapat didefinisikan secara eksplisit. Namun dengan peningkatan jumlah kota yang ingin dikunjungi, maka terjadi peningkatan yang tajam pula pada jumlah kemungkinan rute (Filip dan Otakar, [3]). Tak hanya mengenai jumlah kota yang menjadi objek penelitian dalam TSP, teknik perhitungan atau metode penyelesaian yang digunakan pun menjadi hal menarik untuk dilakukan penelitian lebih lanjut. Pendekatan Pemrograman Dinamik dilakukan oleh Bellman pada tahun 1962 dan Held-Karp pada tahun 1962. Teknik Relaksasi Lagrange yang diperkenalkan oleh Christofides pada tahun 1970 and Held and Karp pada tahun 1970

hingga 1971. Untuk Metode Held-Karp telah diimplementasikan dan diperluas oleh Helbig Hansen and Krarup pada tahun 1974, Smith and Thompson pada tahun 1977, and Volgenant and Jonker.

pada tahun 1982 hingga 1983 (Schrijver, [2]). Pada tahun 1966, Lawler dan Wood [4] melakukan penelitian mengenai hal penting dalam Metode *Branch and Bound*, dan beberapa aplikasinya seperti Pemrograman Bilangan Bulat, Pemrograman Non Linier, *Traveling Salesman Problem* dan *Quadratic Assignment Problem* (Lawler dan Wood, [4]). Untuk kasus TSP, Metode *Branch and Bound* digunakan pada paper yang dibuat oleh Tompkins pada tahun 1956, Rossman and Twery pada tahun 1958, and Eastman pada tahun 1959 (Schrijver, [2]). Selain itu, pada kasus meminimumkan *robust shortest path* oleh R. Montemanni, L.M. Gambardella, dan Donati pada tahun 2006 (Montemanni *et al.*, [5]).

Memperkirakan jarak atau waktu perjalanan secara pasti biasanya menjadi sebuah tugas yang sulit, karena bergantung pada beberapa faktor yang sulit diprediksi seperti kondisi lalu lintas dan kondisi cuaca. Salah satu bidang Optimisasi yang mampu menyelesaikan permasalahan terkait ketidakpastian adalah Optimisasi *Robust*. Seperti dipaparkan oleh Chaerani dan Roos [6], metode yang digunakan untuk mengatasi ketidaktentuan data dalam masalah optimisasi adalah Metodologi Robust Counterpart (RC) yang diperkenalkan oleh Ben-Tal dan Nemirovskii [7].

Tantangan utama dalam metodologi RC ini adalah menentukan bagaimana dan kapan suatu formulasi

¹ Fakultas Matematika dan Ilmu Pengetahuan Alam, Program Studi Matematika, Universitas Padjajaran, Jl. Raya Bandung Sumedang Km. 21 Jatinangor 45363.
Email: poppy.amriyati@gmail.com, d.chaerani@unpad.ac.id, man.msie@gmail.com

* Penulis korespondensi

RC dari masalah optimisasi taktentu dapat diformulasikan kembali sebagai suatu masalah optimisasi yang *computationally tractable* atau paling tidak formulasi RC yang diperoleh dapat didekati dengan suatu masalah yang tractable. Ini berarti bahwa penentuan formulasi RC sangat bergantung kepada bagaimana himpunan data yang melibatkan ketidaktentuan yang digunakan, sebut U , dipilih. Sehingga tantangan yang diberikan oleh RC dapat dipenuhi jika himpunan taktentu U dapat ditentukan melalui suatu cara yang tepat. Pembahasan perkembangan keilmuan Optimisasi *Robust* secara komprehensif dibahas oleh Gabrel [8].

Pembahasan TSP yang melibatkan data data tak tentu diawali oleh Montemanni *et al* [5] yang memperhitungkan waktu tempuh taktentu yang dispesifikasi sebagai rentang dari waktu tempuh. Montemanni *et al* [5] memodelkan RTSP dengan mempertimbangkan kriteria deviasi robust pada data yang dinyatakan dalam interval dari masalah RTSP yang diperoleh. Selain itu, Montemanni *et al* pun membahas Algoritma Exact dan Heuristic untuk menyelesaikan RTSP yang disajikan dalam paper Montemanni *et al* [9] dan [10]. Chassein dan Goerigk [11] membahas TSP taktentu dimana jarak antara dua titik tidak diketahui secara tepat dan mengusulkan Model RTSP yang *recoverable*, yang berarti i Model RTSP yang diusulkan oleh Chassein dan Goerigk [11] mengijinkan sebuah tour untuk mengubah sejumlah sisi terbatasnya ketika suatu skenario diketahui. Model ini memiliki jumlah kendala dan variabel yang eksponensial. Selanjutnya Wang *et al* [12] membahas mengenai TSP taktentu dengan fungsi tujuan yang multi-objektif dan variabel taktentu pada sisi yang menghubungkan titik.

Dalam makalah ini, pendekatan pemodelan RTSP yang berbeda diperkenalkan, yaitu menggunakan Metode RC dari Ben-Tal dan Nemirovskii [7]. Tujuan dari makalah ini adalah memperoleh formulasi Robust Counterpart (RC) untuk Uncertain TSP dengan melibatkan data jarak atau waktu tempuh antar dua titik merupakan data taktentu yang dapat direpresentasikan sebagai ketidaktentuan kotak (*box uncertainty*). Selain data jarak atau waktu tempuh yang taktentu, variabel keputusan yang harus diselesaikan merupakan variabel biner, sehingga penentuan solusi optimal diselesaikan dengan menggunakan Metode *Branch and Bound*. Hasil perhitungan menunjukkan bahwa RSTP memberikan solusi optimal yang robust terhadap ketidaktentuan yang yang mewakili kemungkinan terburuk yang mungkin terjadi.

Metode Penelitian

Optimisasi Robust

Metode yang digunakan untuk memperhitungkan ketidakpastian dalam penelitian ini adalah Opti-

misasi *Robust* (Chaerani [6], Ben-Tal [7]). Pada Optimisasi *Robust*, sebuah masalah Pemrograman Linier dengan ketidakpastian didefinisikan dalam model matematika pada persamaan berikut.

$$\begin{aligned} \min & c^T x, \\ \text{s.t.} & Ax = b, \\ & x \geq 0, \\ & \forall (c, A, b) \in U \end{aligned} \quad (1)$$

dimana $c \in R^n$ adalah koefisien fungsi objektif, $A \in R^{m \times n}$ adalah matriks fungsi-fungsi kendala, $b \in R^m$ adalah vektor ruas kanan, dan $x \in R^n$ adalah vektor variabel keputusan yang harus ditentukan. Perhatikan bahwa dalam masalah optimisasi taktentu parameter ini (c, A, b) diasumsikan berada dalam himpunan taktentu U .

Dalam lingkungan pengambilan keputusan, solusi yang berarti untuk masalah ketidakpastian (*uncertain*) adalah solusi *Robust Feasible*. Hal ini mengingatkan bagaimana memutuskan untuk mendapatkan nilai objektif (yang dapat berupa ketidakpastian), sebagai suatu solusi. Seperti halnya yang diterapkan pada objektif, filosofi orientasi kemungkinan terburuk mempermudah untuk mengukur kualitas pada solusi *Robust Feasible* x dengan dijamin oleh nilai asli fungsi objektif, yaitu dengan nilai terbesar dari $\sup\{c^T x : (c, A, b) \in U\}$.

Dengan demikian, kemungkinan terbaik dari solusi *Robust Feasible* adalah salah satu yang dapat memecahkan masalah optimisasi berikut:

$$\begin{aligned} \min & \sup c^T x, \\ \text{s.t.} & Ax = b, \\ & x \geq 0, \\ & (c, A, b) \in U. \end{aligned} \quad (2)$$

Unsur ketidakpastian yang berada pada fungsi objektif menyebabkan fungsi objektif harus dibatasi oleh suatu nilai misalkan ℓ . Sehingga persamaan (2) dapat ditulis kembali menjadi persamaan berikut:

$$\begin{aligned} \min & \ell, \\ \text{s.t.} & c^T x \leq \ell, \\ & Ax = b, \\ & x \geq 0, \\ & (c, A, b) \in U. \end{aligned} \quad (3)$$

Nilai ℓ menjadi batas atas yang membatasi fungsi objektif, hal ini hanya dapat terjadi karena tujuan dari masalah tersebut adalah meminimumkan fungsi objektif. Masalah (5) disebut Metodologi *Robust Counterpart* (RC) dan solusi yang didapat dinamakan *Robust Feasible* / *Robust Optimal Solution*.

Metode Branch and Bound

Merujuk pada Hillier dan Lieberman [13], dapat disarikan bahwa Metode *Branch and Bound* sangat efektif dalam menyelesaikan masalah Pemrograman Bilangan Bulat campuran baik yang linier maupun nonlinier. Dalam penyelesaian Metode *Branch and Bound*, masalah dibagi menjadi 3 bagian, yaitu *branching* (percabangan) sehingga terbentuk sebuah struktur pohon pencarian (*search tree*), *bounding* (pembatasan), dan *fathoming* (penghilangan). Langkah-langkah penerapan Metode *Branch and Bound* pada penelitian ini, sebagai berikut:

1. Tahap inisialisasi

Bentuk model matematika dari kasus yang akan diselesaikan. Bentuk PL relaksasi dari model yang didapatkan, yaitu dengan menghapus satu set kendala yang membuat kasus tersebut menjadi sulit diselesaikan. Dalam kasus ini, kendala yang dihapus adalah solusi berupa sebuah *tour*. Tetapkan $Z^* = \infty$, kemudian selesaikan masalah PL relaksasi dengan metode simpleks. Terapkan langkah *bounding*, *fathoming*, dan uji optimalitas yang dijelaskan di bawah terhadap keseluruhan kasus. Apabila tidak dihilangkan, klasifikasikan kasus ini sebagai satu submasalah yang tersisa untuk melakukan iterasi lengkap berikutnya.

2. Tahap iterasi, tahap ini terbagi menjadi 3:

a. *Branching* (percabangan)

Konstruksi submasalah baru dari masalah sebelumnya dengan cara menetapkan nilai tiap jalur pada *subtour* terkecil dengan nilai 0. Misalkan *subtour* terkecil yang didapatkan adalah $(a - b - c)$, sehingga submasalah baru yang pertama memiliki fungsi kendala tambahan yaitu $x_{ab} = 0$ dan submasalah baru yang kedua memiliki fungsi kendala tambahan yaitu $x_{bc} = 0$.

b. *Bounding* (pembatasan)

Setelah dilakukan *branching* (percabangan atau pembagian), setiap submasalah perlu dicari batas mengenai seberapa jauh solusi layak terbaik dapat dicapai dengan cara menyelesaikan PL relaksasi dari setiap submasalah tersebut menggunakan metode simpleks. Jika solusi layak telah diperoleh, hitung nilai dari fungsi objektif (Z). Kemudian hitung batas = $\lfloor Z \rfloor$.

c. *Fathoming* (penghilangan)

Suatu submasalah dapat dihilangkan dari pertimbangan lebih lanjut jika memenuhi salah satu kondisi berikut.

Uji 1: Jika batasnya $\leq Z^*$. Di mana Z^* adalah nilai Z untuk *incumbent* saat ini.

Uji 2: PL relaksasi tidak memiliki solusi yang layak.

Uji 3: Jika solusi optimal untuk PL relaksasinya adalah sebuah *tour*. Apabila solusi ini lebih baik dari pada *incumbent*, maka solusi ini menjadi *incumbent* baru, dan uji 1 diterapkan kembali pada semua submasalah yang tidak dihilangkan, dengan Z^* baru.

3. Uji optimalitas

Uji optimalitas dilakukan pada setiap akhir iterasi. Berhenti ketika tidak ada lagi submasalah yang tersisa. *Incumbent* (calon solusi optimal) yang berlaku adalah optimal. Jika tidak, kembali lakukan iterasi lainnya.

Hasil dan Pembahasan

Model Optimisasi *Robust* TSP untuk Kasus Pencarian Rute

Pada sub bagian ini, untuk mendapatkan *Robust Counterpart* digunakan *Box Uncertainty* dimana himpunan unsur tak pasti diasumsikan berbentuk *box*. Dalam penelitian ini, diasumsikan bahwa parameter data yang tidak pasti adalah koefisien pada fungsi objektif. Permasalahan tersebut dapat diformulasikan sebagai TSP tak tentu seperti di bawah ini:

$$\begin{aligned} \min & d^T x, \\ \text{s.t.} & \sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n \\ & \sum_{i=1}^n x_{ij} = 1, \quad j = 1, 2, \dots, n \\ & x_{ij} = \{0, 1\}, \\ & d \in \mathcal{U}, \\ & \text{solusi membentuk sebuah tour (tertutup)}. \end{aligned}$$

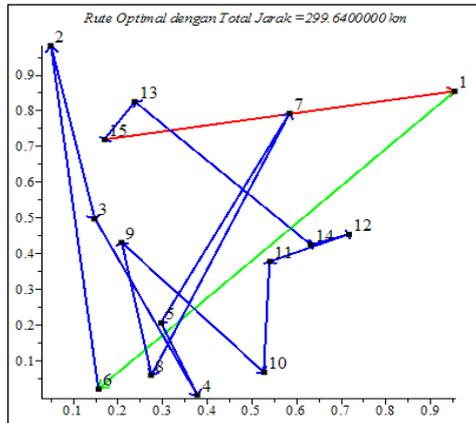
dimana d adalah vektor kolom yang merepresentasikan jarak atau waktu tempuh perjalanan dari kota i ke kota j , x adalah vektor kolom yang memuat variabel keputusan perjalanan dari kota i ke kota j dilakukan atau tidak. Fungsi objektif pada persamaan (4) memuat unsur tidak tentu maka fungsi objektif haruslah dikonstruksi menjadi sebuah fungsi objektif tentu dengan cara menghilangkan parameter tak tentu dari fungsi objektif dan menyajikan dalam bentuk fungsi variabel tunggal σ , sehingga diperoleh

$$\begin{aligned} \min & \sigma, \\ \text{s.t.} & d^T x \leq \sigma, \\ & \sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n \\ & \sum_{i=1}^n x_{ij} = 1, \quad j = 1, 2, \dots, n \\ & x_{ij} = \{0, 1\}, \\ & d \in \mathcal{U}, \\ & \sigma \geq 0, \\ & \text{solusi membentuk sebuah tour (tertutup)}. \end{aligned} \tag{5}$$

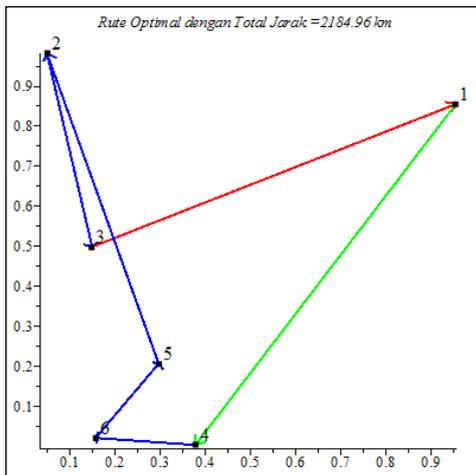
b. Model Optimisasi *Robust* TSP (RTSP)

Berdasarkan persamaan (8) yang telah disubstitusikan data dari Klanšek [14] diperoleh:

- (i) Nilai Fungsi Objektif (Z) = 299,64, artinya jarak optimal yang harus ditempuh untuk mengunjungi seluruh tempat pembangunan adalah 299,64 km
- (ii) $x_{1,6} = x_{2,3} = x_{3,4} = x_{4,5} = x_{5,7} = x_{6,2} = x_{7,8} = x_{8,9} = x_{9,10} = x_{10,11} = x_{11,12} = x_{12,14} = x_{13,15} = x_{14,13} = x_{15,1} = 1$, artinya rute perjalanan yang harus ditempuh oleh *supervisor* adalah (1) Maribor - (6) Ložane - (2) Sladki Vrh - (3) Murska Sobota - (4) Mačkovci - (5) Moravske Toplice - (7) Ljutomer - (8) Ormož - (9) Ptuj - (10) Slovenska Bistrica - (11) Kopivnik - (12) Hotinja vas - (14) Rogoza - (13) Hoče - (15) Ruše - (1) Maribor.
- (iii) Jumlah iterasi sebanyak 8 dengan total sub-masalah sebanyak 135.



Gambar 2. Rute optimal RTSP untuk kasus manajemen konstruksi



Gambar 3. Rute optimal TSP untuk kasus kunjungan ibukota provinsi se-Pulau Jawa

B. Pencarian Rute Optimal untuk Kasus Kunjungan Ibukota Provinsi se-Pulau Jawa.

Data untuk simulasi numerik pada kasus ini didapatkan dari Google Maps [15], yaitu seorang *traveler* akan mengunjungi seluruh ibu kota provinsi di Pulau Jawa, yaitu : (1) Serang, (2) Bandung, (3) Jakarta, (4) Semarang, (5) Yogyakarta, (6) Surabaya, dengan titik awal kunjungan adalah Serang.

a. Model Optimisasi TSP

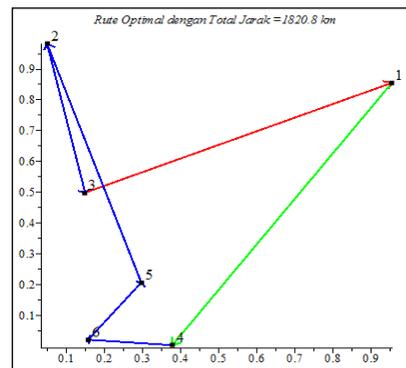
Berdasarkan persamaan (4) yang telah disubstitusikan data dari Google Maps [15] diperoleh :

- (i) Nilai Fungsi Objektif (Z) = 1820,8, artinya jarak optimal yang harus ditempuh untuk mengunjungi seluruh ibukota provinsi se-Pulau Jawa adalah 1820,8 km.
- (ii) $x_{1,4} = x_{4,6} = x_{6,5} = x_{5,2} = x_{2,3} = x_{3,1} = 1$, artinya rute perjalanan yang harus ditempuh oleh *traveler* adalah (1) Serang - (4) Semarang - (6) Surabaya - (5) Yogyakarta - (2) Bandung - (3) Jakarta - (1) Serang.
- (iii) Jumlah iterasi sebanyak 4 dengan total sub-masalah sebanyak 34.

b. Model Optimisasi *Robust* TSP (RTSP)

Berdasarkan persamaan (8) yang telah disubstitusikan data dari Google Maps [15] diperoleh:

- (i) Nilai Fungsi Objektif (Z) = 2184,96, artinya jarak optimal yang harus ditempuh untuk mengunjungi seluruh ibukota provinsi se-Pulau Jawa adalah 2184,96 km.
- (ii) $x_{1,4} = x_{4,6} = x_{6,5} = x_{5,2} = x_{2,3} = x_{3,1} = 1$, artinya rute perjalanan yang harus ditempuh oleh *traveler* adalah (1) Serang - (4) Semarang - (6) Surabaya - (5) Yogyakarta - (2) Bandung - (3) Jakarta - (1) Serang.
- (iii) Jumlah iterasi sebanyak 4 dengan total sub-masalah sebanyak 34.



Gambar 4. Rute optimal RTSP untuk kasus kunjungan ibukota provinsi se-Pulau Jawa

C. Pencarian Rute Optimal untuk Kasus Mandiri *Run* Bandung 5 KM

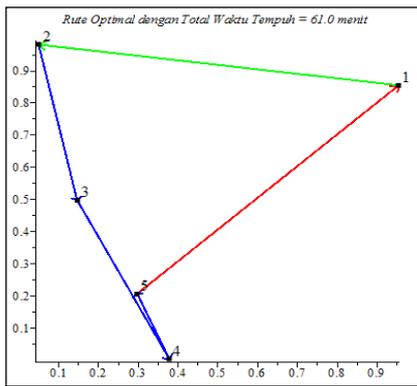
Data untuk simulasi numerik pada kasus ini didapatkan dari Google Maps [15], yaitu seorang peserta Mandiri *Run* Bandung akan mengunjungi

seluruh tempat kilometer *marker* yang dimulai dari tempat start/finish, yaitu : (1) *Start/Finish*, (2) KM 1, (3) KM 2, (4) KM 3, (5) KM 4, dengan titik awal kunjungan adalah *Start/Finish*.

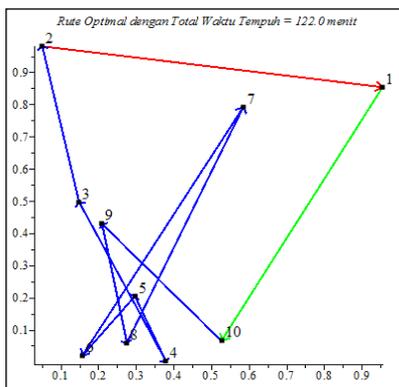
a. Model Optimisasi TSP

Berdasarkan persamaan (4) yang telah disubstitusikan data dari Google Maps [15] diperoleh :

- (i) Nilai Fungsi Objektif (Z) = 61,0, artinya waktu tempuh optimal yang harus ditempuh untuk seluruh tempat kilometer *marker* pada Mandiri *Run* Bandung 5 KM adalah 61,0 menit.
- (ii) ($x_{1,2} = x_{2,3} = x_{3,4} = x_{4,5} = x_{5,1} = 1$, artinya rute perjalanan yang harus ditempuh oleh peserta Mandiri *Run* Bandung adalah (1) *Start/Finish* - (2) KM 1 - (3) KM 2 - (4) KM 3 - (5) KM 4 - (1) *Start/Finish*).
- (iii) Jumlah iterasi sebanyak 0 dengan total sub-masalah sebanyak 0 (penyelesaian hanya sampai tahap inisialisasi, karena rute yang dihasilkan pada tahap inisialisasi sudah merupakan rute tertutup yang tidak memiliki *sub tour*)



Gambar 5. Rute optimal TSP untuk kasus mandiri *run* Bandung 5 KM



Gambar 6. Rute optimal RTSP untuk kasus mandiri *run* Bandung 5 KM

b. Model Optimisasi *Robust* TSP (RTSP)

Berdasarkan persamaan (8) yang telah disubstitusikan data dari Google Maps [15] diperoleh:

- (i) Nilai Fungsi Objektif (Z) = 73,19, artinya waktu tempuh optimal yang harus ditempuh untuk seluruh tempat kilometer *marker* pada Mandiri *Run* Bandung 5 KM adalah 73,19 menit.
- (ii) $x_{1,5} = x_{5,4} = x_{4,3} = x_{3,2} = x_{2,1} = 1$, artinya rute perjalanan yang harus ditempuh oleh peserta Mandiri *Run* Bandung adalah (1) *Start/Finish* - (5) KM 4 - (4) KM 3 - (3) KM 2 - (2) KM 1 - (1) *Start/Finish*.
- (iii) Jumlah iterasi sebanyak 0 dengan total sub-masalah sebanyak 0 (penyelesaian hanya sampai tahap inisialisasi karena rute yang dihasilkan pada tahap inisialisasi sudah merupakan rute tertutup yang tidak memiliki *sub tour*)

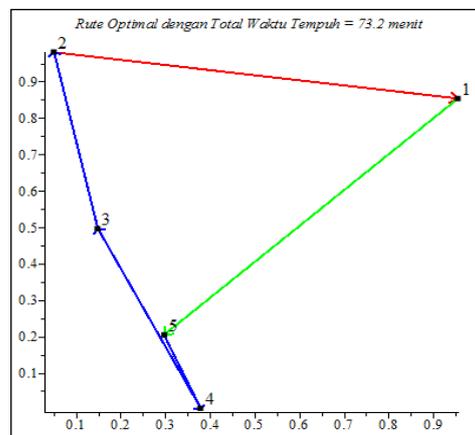
D. Pencarian Rute Optimal untuk Kasus Mandiri *Run* Bandung 10 KM

Data untuk simulasi numerik pada kasus ini didapatkan dari Google Maps [15], yaitu seorang peserta Mandiri *Run* Bandung akan mengunjungi seluruh tempat kilometer *marker* yang dimulai dari tempat start/finish, yaitu: (1) *Start/Finish*, (2) KM 1, (3) KM 2, (4) KM 3, (5) KM 4, (6) KM 5, (7) KM 6, (8) KM 7, (9) KM 8, (10) KM 9, dengan titik awal kunjungan adalah *Start/Finish*.

a. Model Optimisasi TSP

Berdasarkan persamaan (4) yang telah disubstitusikan data dari Google Maps [15], maka didapatkan hasil sebagai berikut.

- (i) Nilai Fungsi Objektif (Z) = 122,0, artinya waktu tempuh optimal yang harus ditempuh untuk seluruh tempat kilometer *marker* pada Mandiri *Run* Bandung 10 KM adalah 122 menit.
- (ii) $x_{1,10} = x_{10,9} = x_{9,8} = x_{8,7} = x_{7,6} = x_{6,5} = x_{5,4} = x_{4,3} = x_{3,2} = x_{2,1} = 1$, artinya rute perjalanan yang harus ditempuh oleh peserta Mandiri *Run*



Gambar 7. Rute optimal TSP untuk kasus mandiri *run* Bandung 10 KM

Bandung adalah (1) *Start/Finish* - (10) KM 9 - (9) KM 8 - (8) KM 7 - (7) KM 6 - (6) KM 5 - (5)

KM 4 - (4) KM 3 - (3) KM 2 - (2) KM 1 - (1) *Start/Finish*.

(iii) Jumlah iterasi sebanyak 2 dengan total submasalah sebanyak 6.

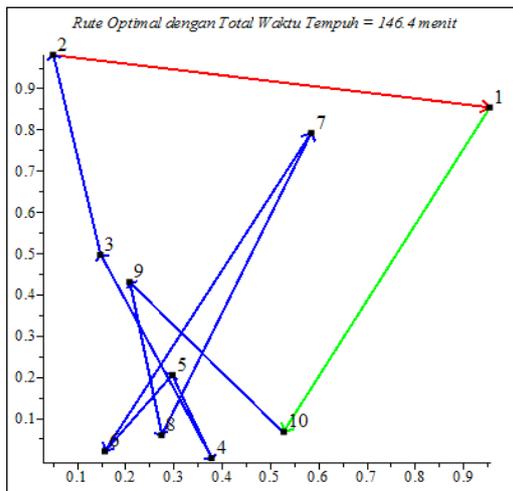
b. Model Optimisasi *Robust* TSP (RTSP)

Berdasarkan persamaan (8) yang telah disubstitusikan data dari Google Maps [15], maka didapatkan hasil sebagai berikut :

(i) Nilai Fungsi Objektif (Z) = 146,39, artinya waktu tempuh optimal yang harus ditempuh untuk seluruh tempat kilometer *marker* pada Mandiri *Run* Bandung 10 KM adalah 146,39 menit.

(ii) $x_{1,10} = x_{10,9} = x_{9,8} = x_{8,7} = x_{7,6} = x_{6,5} = x_{5,4} = x_{4,3} = x_{3,2} = x_{2,1} = 1$, artinya rute perjalanan yang harus ditempuh oleh peserta Mandiri *Run* Bandung adalah (1) *Start/Finish* - (10) KM 9 - (9) KM 8 - (8) KM 7 - (7) KM 6 - (6) KM 5 - (5) KM 4 - (4) KM 3 - (3) KM 2 - (2) KM 1 - (1) *Start/Finish*.

(iii) Jumlah iterasi sebanyak 3 dengan total submasalah sebanyak 18



Gambar 8. Rute optimal RTSP untuk kasus mandiri *run* Bandung 10 KM

Analisis Hasil Perhitungan Simulasi Numerik

Tabel 1 menyajikan nilai fungsi objektif dari solusi optimal, jumlah iterasi dan percabangan, serta waktu yang dibutuhkan CPU untuk menyelesaikan perhitungan yang didapat dari Model Optimisasi TSP untuk masalah pencarian rute optimal.

Tabel 2. menyajikan nilai fungsi objektif dari solusi optimal, jumlah iterasi dan percabangan, serta waktu yang dibutuhkan CPU untuk menyelesaikan perhitungan yang didapat dari Model Optimisasi *Robust* TSP (RTSP) untuk masalah pencarian rute optimal.

Tabel 1. Nilai fungsi objektif dari solusi optimal, jumlah iterasi dan percabangan, serta waktu yang dibutuhkan CPU untuk Model TSP

Kasus	Jumlah Titik	Rute Optimal	Nilai Fungsi Objektif	Letak Solusi Optimal	CPU Times
Manajemen konstruksi	15	(1-6-2-3-4-5-7-8-9-10-11-12-14-13-15-1)	249,7 Km	Iterasi ke 3 dan submasalah ke-13	22,308 s
Kunjungan Ibukota Provinsi se Pulau Jawa Mandiri	6	(1-4-6-5-2-3-1)	1820,8 Km	Iterasi ke 3 dan submasalah ke-13	6,142 s
Run Bandung 5 Km	5	(1-2-3-4-5-1)	61 Menit	Inisialisasi	0,172 s
Mandiri Run Bandung 10 Km	10	(1-10-9-8-7-6-5-4-3-2-1)	122 menit	Iterasi ke 2 dan submasalah ke-3	1,233 s

Tabel 2. Nilai fungsi objektif dari solusi optimal, jumlah iterasi dan percabangan, serta waktu yang dibutuhkan CPU untuk Model RTSP

Kasus	Jumlah Titik	Rute Optimal	Nilai Fungsi Objektif	Letak Solusi Optimal	CPU Times
Manajemen konstruksi	15	(1-6-2-3-4-5-7-8-9-10-11-12-14-13-15-1)	299,631 Km	Iterasi ke 3 dan submasalah ke-13	22,547 s
Kunjungan Ibukota Provinsi se Pulau Jawa Mandiri Run Bandung 5 Km	6	(1-4-6-5-2-3-1)	2184,95 Km	Iterasi ke 3 dan submasalah ke-13	5,745 s
Run Bandung 5 Km	5	(1-5-4-3-2-1)	73,19 Menit	Inisialisasi	0,188 s
Mandiri Run Bandung 10 Km	10	(1-10-9-8-7-6-5-4-3-2-1)	146,3 menit	Iterasi ke 3 dan submasalah ke-7	2,655 s

Penentuan solusi optimal dari RTSP dengan *Box Uncertainty* menggunakan Metode *Branch and Bound* dan kemudian dilakukan simulasi numerik yang menunjukkan bahwa diperoleh hasil *robust* optimal yang menyajikan *the best worst case solution* untuk Model Optimisasi *Robust* TSP. Solusi optimal yang didapat dari Model Optimisasi *Robust* TSP pada beberapa contoh kasus menghasilkan nilai fungsi objektif yang lebih besar dari solusi optimal yang didapat dari Model Optimisasi TSP. Hal ini terjadi karena pada Optimisasi *Robust* memperhitungkan faktor tak tentu yang mewakili kemungkinan terburuk yang mungkin terjadi, seperti kemacetan, kondisi jalan yang tidak baik, atau pun kemungkinan lainnya.

Simpulan

Model RTSP merupakan Pemrograman Bilangan Bulat Campuran sehingga dapat dijamin bahwa RTSP dengan *Box Uncertainty* merupakan formulasi masalah Optimisasi yang dapat diselesaikan

secara komputasi. Oleh karena itu, penerapan Optimisasi *Robust* pada kasus TSP tak tentu menghasilkan solusi optimal *robust* global

Daftar Pustaka

1. Punnen, A. P. and Gutin, G., *The Traveling Salesman Problem and It's Variations*. Dordrecht: Kluwer Academic Publishers, 2002.
2. Schrijver, A., *Combinatorial Optimization*. Heiderberg: Springer-Verlag Berlin, 2004.
3. Filip, E., and Otakar, M., The Travelling Salesman Problem and its Application in Logistic, *WSEAS Transactions on Business And Economics*, 8(4), 2011, pp 164-173.
4. Lawler, E.L., and Wood, D.E., Branch and Bound Method : A Survey, *Operations Research*, 14 (4), 1966, pp. 699–719.
5. Montemanni, R., Barta, J., Mastrolilli, M., and Gambardella, L.M., The Robust Traveling Salesman Problem with Interval Data, *Transportation Science* 41(3), 2007, pp. 366-381.
6. Chaerani, D., and Roos., C., Handling Uncertain Optimization Problem via Robust Counterpart Methodology, *Jurnal Teknik Industri.*, 15(2), 2013, Desember 2013, pp. 111-118.
7. Ben-Tal, A., and Nemirovskii, A., Robust Optimization—Methodology and Applications, *Mathematical Programming*, 92 (3), 2002, pp. 453-480.
8. Gabrel, V., Murat, C., and Thiele, A., Recent Advances in Robust Optimization: An Overview, *European Journal of Operational Research*, 235(3), 2014, pp. 471–483.
9. Montemanni, R., Barta, J., and Gambardella, L.M., An Exact Algorithm for the Robust Traveling Salesman Problem with Interval Data, 2006, <http://people.idsia.ch/~roberto/papers/Mon-BarGam.pdf>, diakses pada 15 Oktober 2015.
10. Montemanni, R., Barta, J., Mastrolilli, M., and Gambardella, L.M., Heuristic Algorithms for the Robust Traveling Salesman Problem with Interval Data, 2006, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.110.7537&rep=rep1&type=pd>, diakses pada 15 Oktober 2015.
11. Chassein, A and Goerigk, M., *On the Recoverable Robust Traveling Salesman Problem*, Technische Universität Kaiserslautern, Fachbereich Mathematik, 2014. <http://kluedo.ub.uni-kl.de/files/3898/paper.pdf>., diakses 21 Oktober 2015.
12. Wang, Z., Guo, J., Zheng, M., and Wang, Y., Uncertain Multi-objective Traveling Salesman Problem, *European Journal of Operational Research*, 241(2), 2015, pp. 478-489.
13. Hillier, F., and Lieberman., *Introduction to Operation Research*. New York: McGraw-Hill, 2008.
14. Klanšek, U., Using the TSP Solution for Optimal Route Scheduling in Construction Management., *Organization, Technology & Management in Construction: An International Journal*, 658(05), 2011, pp. 243-249.
15. Google Maps, 2015. *Data Peta*. [Online] Available at: <https://www.google.co.id/maps?source=tldsi&hl=id>., diakses pada 28 Maret 2015.