

OPTIMALISASI PENCARIAN LINTASAN TRAVELING SALESMAN PROBLEM MENGGUNAKAN ALGORITMA BACKTRACKING

Hari Murti¹, Edy Supriyanto², Sugiyamta³

^{1,2,3}Program Studi Sistem Informasi, Universitas Stikubank

email: ¹harimurti@edu.unisbank.ac.id, ²edy_supriyanto@edu.unisbank.ac.id,

³sugiyamtagik@edu.unisbank.ac.id

Abstrak

Tujuan pemecahan masalah penjual atau wira niaga keliling (*traveling salesman problem*, TSP) adalah menentukan lintasan atau rute dengan total jarak atau biaya yang paling minimum. Penelitian ini berkenaan dengan proses optimalisasi pencarian alternatif lintasan atau rute pada masalah TSP. Penjual mencari rute atau lintasan untuk mengunjungi semua kota yang ada sebanyak satu kali kunjungan. Proses optimalisasi pencarian alternatif menggunakan algoritma backtracking.

Terdapat lima buah kasus TSP yang akan digunakan yaitu TSP1 memiliki 4 buah kota dengan 6 lintasan (jalur), TSP2 memiliki 5 buah kota dengan 10 lintasan, TSP3 memiliki 6 buah kota dengan 15 lintasan, TSP4 memiliki 7 buah kota dengan 21 lintasan, dan TSP5 memiliki 8 buah kota dengan 28 lintasan. Dari hasil pengujian diperoleh algoritma runut-balik (backtracking) dapat digunakan untuk menghasilkan jumlah alternatif lintasan yang lebih sedikit jika dibandingkan dengan total kombinasi alternatif lintasan. Semakin besar jumlah node (kota) dan jumlah lintasan (jalur) maka prosentase pengurangan variasi (alternatif) akan lintasan semakin besar.

Kata Kunci: *traveling salesman problem*, algoritma runut-balik, *backtracking*.

1. PENDAHULUAN

a. Latar Belakang Masalah

Penelitian ini berkenaan dengan proses optimalisasi pencarian alternatif lintasan atau rute pada masalah penjual atau wira niaga keliling (*traveling salesman problem*, TSP). Wira niaga mencari rute atau lintasan untuk mengunjungi semua kota yang ada sebanyak satu kali kunjungan. Proses optimalisasi pencarian alternatif menggunakan algoritma *backtracking*. Beberapa penelitian yang pernah dilakukan yang berhubungan dengan hal tersebut di atas, antara lain:

- 1) Penelitian yang dilakukan oleh Erdiwansyah dan Gani (2016). Masalah optimasi pada *Traveling Salesman Problem* (TSP) sangat terkenal dan telah menjadi standar untuk mencoba algoritma yang komputasional. *Traveling Salesman Problem* merupakan sebuah permasalahan optimasi yang dapat diterapkan pada berbagai kegiatan seperti pendistribusian barang, pengambilan tagihan listrik dan penjadwalan. Algoritma *Local Search* merupakan metode pencarian solusi berdasarkan *neighborhood* dari solusi awal. Metode ini dikenal juga dengan nama *iterative improvement*. Algoritma *Population Based* merupakan pencarian secara global. Hasil menunjukkan algoritma *local search* lebih baik dari algoritma *population based* dalam pencarian nilai yang optimal.
- 2) Penelitian yang dilakukan oleh Widodo dan Mahmudy (2010). Pelancong tidak hanya berfikir tentang tempat apa yang layak dikunjungi namun juga memikirkan makanan apa yang dapat dinikmati beserta keunikan dan kenyamanannya. Perhatian masyarakat terhadap wisata kuliner mempengaruhi kecenderungan dan gaya dalam berwisata. Kondisi ini membawa perubahan pertimbangan seorang pelancong melakukan pilihan terhadap tempat kunjungan wisata. Rekomendasi tentang faktor pilihan wisata, kesesuaian selera tempat makan, dan waktu perjalanan menjadi hal yang penting. Dalam penelitian ini dilakukan pendekatan terhadap masalah tersebut dengan menganalogikan pada kasus pemilihan rute manakah yang memiliki biaya paling murah untuk dilalui seorang pelancong yang harus mengunjungi sejumlah m daerah tujuan wisata ($m > 1$). Tiap daerah tujuan harus dikunjungi tepat satu kali dan kemudian kembali lagi ke tempat semula. Masalah tersebut sering dikenal dengan nama *Traveling Salesman Problem* (TSP) yaitu masalah

kombinatorial yang solusi optimumnya didapat dengan cara mencoba semua kemungkinan sehingga memerlukan waktu komputasi yang cukup tinggi. Berbeda dengan fungsi TSP konvensional yang tujuannya hanya meminimalkan jarak, kasus rekomendasi ini harus mencocokkan pilihan wisata, kesesuaian selera tempat makan, dan waktu perjalanan. Algoritma Genetika (GA) terbukti sesuai untuk menyelesaikan masalah multi obyektif, sehingga dapat diterapkan untuk sistem rekomendasi wisata kuliner metode crossover dengan satu titik potong dan mutasi dengan pergeseran gen pada GA mampu menyelesaikan masalah.

- 3) Penelitian yang dilakukan oleh Murti, Soelistijadi, dan Sugiyanto (2017). Permasalahan *traveling salesman problem* (TSP) adalah seorang penjual yang harus mengunjungi semua kota sebanyak satu kali saja dimana dia harus mengawali dan mengakhiri perjalanan di kota yang sama. Tujuan TSP adalah menentukan lintasan atau rute dengan total jarak atau biaya yang paling minimum. Penelitian ini bertujuan untuk mengetahui kebutuhan waktu pada proses penyelesaian TSP menggunakan teknik pencarian buta dan pencarian terbimbing. Metode yang digunakan adalah pencarian melebar pertama (*breadth-first search*), pencarian mendalam pertama (*depth-first search*), pembangkitan dan pengujian (*generate and test*), dan pencarian terbaik pertama (*best first search*). Terdapat tiga buah ilustrasi kasus TSP yang digunakan yaitu TSP1 memiliki 4 buah kota dengan 6 jalan penghubung, TSP2 memiliki 5 buah kota dengan 10 jalan penghubung, dan TSP3 memiliki 8 buah kota dengan 28 jalan penghubung. Dari hasil pengujian diperoleh pencarian mendalam pertama (*depth first search*) membutuhkan waktu paling sedikit (tercepat) apabila dibandingkan dengan metode pencarian yang lain.
- 4) Penelitian yang berjudul “Perbandingan Algoritma Greedy, Algoritma Cheapest Insertion Heuristics dan Dynamic Programming Dalam Penyelesaian Travelling Salesman Problem” (Aristi, 2014). Pada penelitian ini dibandingkan hasil dari perhitungan menggunakan algoritma *greedy*, algoritma *cheapest insertion heuristics*, dan *dynamic programming*, sehingga diketahui algoritma mana yang lebih baik untuk menyelesaikan kasus TSP. Hasil penelitian menunjukkan algoritma *greedy* lebih sederhana cara penyelesaiannya untuk jumlah kota yang masih sedikit sedangkan pada jumlah kota yang lebih banyak maka lebih baik menggunakan algoritma *cheapest insertion heuristics*. Melalui penelitian ini dapat disimpulkan jumlah kota yang dikunjungi berpengaruh terhadap pemilihan algoritma.
- 5) Penelitian oleh Fatmawati dan kawan-kawan pada tahun 2015 (Fatmawati, dkk., 2015). *Tabu Search* merupakan salah satu metode heuristik yang berbasis pada pencarian lokal. Proses kinerjanya bergerak dari satu solusi ke solusi berikutnya dengan cara memilih solusi terbaik. Tujuan utama metode ini adalah mencegah proses pencarian agar tidak melakukan pencarian ulang pada ruang solusi yang sudah pernah ditelusuri. Metode ini menggunakan *Tabu List* untuk menyimpan sekumpulan solusi yang baru saja dievaluasi, hasilnya akan disesuaikan terlebih dahulu dengan isi pada *Tabu List* untuk melihat apakah solusi tersebut sudah ada atau tidak. Jika solusi tersebut sudah ada maka solusi tersebut tidak akan dievaluasi lagi pada iterasi berikutnya. Pada penelitian ini, metode *Tabu Search* diterapkan untuk menyelesaikan *travelling salesman problem* (TSP) pada contoh kasus salesman PT. XX dalam mengatur rute perjalanannya.
- 6) Penelitian oleh Yunus, Helmi, dan Martha pada tahun 2015 (Yunus, dkk., 2015). Program Dinamis merupakan salah satu metode yang dapat digunakan untuk menyelesaikan Traveling Salesman Problem (TSP). Program Dinamis adalah metode penyelesaian dengan menguraikan solusi menjadi beberapa tahap atau iterasi sedemikian hingga solusi dari persoalan tersebut dapat dipandang sebagai serangkaian keputusan yang saling berkaitan. Traveling Salesman Problem adalah membentuk sebuah sirkuit untuk melewati semua simpul dengan total bobot dari sisi pembentuk sirkuit minimum. Penelitian ini mengkaji penggunaan Program Dinamis pada penyelesaian TSP untuk menentukan sirkuit dengan bobot minimum dan melewati semua simpul dari graf. Berdasarkan arah dan bobotnya jika diambarkan dalam bentuk graf, TSP simetris merupakan jenis graf tidak berarah dan berbobot, sedangkan TSP asimetris adalah graf berarah dan berbobot. Penyelesaian TSP dapat menggunakan Program Dinamis rekursif maju dan rekursif mundur karena mempunyai solusi optimal yang sama. Hasil penyelesaian TSP menggunakan Program Dinamis dengan rekursif maju diperoleh berdasarkan solusi optimal dari iterasi ke-1 sampai iterasi ke-t.

Penelitian yang akan dilakukan oleh penulis adalah melakukan optimalisasi pencarian alternatif pada lintasan *traveling salesman problem* (TSP). Hal ini mengingat penelitian tentang optimalisasi pencarian alternatif pada lintasan TSP masih belum banyak dilakukan. Beberapa contoh kasus TSP diperoleh dari data set (www.cc.gatech.edu/~bdilkina/CSE6140-2014fa/ProjectTSP.htm) dan buku teks (Russell dan Norvig, 2010; Kusumadewi dan Purnomo, 2005; Kusumadewi, 2003).

b. Perumusan Masalah

Penelitian ini berkenaan dengan proses optimalisasi pencarian alternatif lintasan atau rute pada masalah penjual atau wira niaga keliling (*traveling salesman problem*, TSP). Wira niaga mencari rute atau lintasan untuk mengunjungi semua kota yang ada sebanyak satu kali kunjungan. Proses optimalisasi pencarian alternatif menggunakan algoritma backtracking.

c. Batasan Masalah

Terdapat lima buah kasus *traveling salesman problem* (TSP) yang akan digunakan yaitu TSP1 memiliki 4 buah kota dengan 6 lintasan (jalur), TSP2 memiliki 5 buah kota dengan 10 lintasan, TSP3 memiliki 6 buah kota dengan 15 lintasan, TSP4 memiliki 7 buah kota dengan 21 lintasan, dan TSP5 memiliki 8 buah kota dengan 28 lintasan.

2. TUJUAN DAN MANFAAT PENELITIAN

a. Tujuan dan Manfaat Penelitian

Tujuan penelitian ini yaitu untuk memperoleh alternatif lintasan pada proses penyelesaian *traveling salesman problem* (TSP) dan optimalisasi pencarian alternatif pada lintasan tersebut. Manfaat dari penelitian ini adalah menghasilkan alternatif lintasan yang optimal pada penyelesaian beberapa kasus TSP.

3. TINJAUAN PUSTAKA

a. Pencarian dan Pencocokan

Masalah pencarian merupakan pencarian solusi yang direncanakan yang mencari lintasan dari keadaan awal atau keadaan sekarang sampai dengan keadaan tujuan. Pencarian adalah suatu proses mencari solusi dari suatu permasalahan melalui sekumpulan kemungkinan ruang keadaan (*state space*). Ruang keadaan merupakan suatu ruang yang berisi semua keadaan yang mungkin. Kondisi suatu pencarian meliputi:

- 1) Keadaan awal atau keadaan sekarang.
- 2) Keadaan tujuan atau sasaran yang ingin dicapai.
- 3) Biaya atau nilai yang diperoleh dari solusi.

Solusi adalah suatu lintasan dari keadaan awal sampai keadaan tujuan (Desiani dan Arhami, 2006). Proses pencarian solusi dapat dilakukan dengan cara sebagai berikut:

- 1) Memeriksa keadaan awal atau keadaan sekarang.
- 2) Mengeksekusi aksi yang dibolehkan dan berpindah ke keadaan berikutnya (keadaan baru).
- 3) Memeriksa keadaan baru. Jika keadaan baru bukan solusi maka keadaan baru menjadi keadaan sekarang dan kembali ke proses a, jika sebaliknya yaitu keadaan baru merupakan solusi maka proses pencarian selesai.

Proses di atas dikerjakan sampai dengan solusi ditemukan atau ruang keadaan telah dieksekusi semua. Kesuksesan dalam pencarian dan pencocokan (*search and matching*) akan menentukan tingkat keberhasilan sistem yang bekerja berdasarkan kecerdasan buatan. Terdapat dua kelompok utama teknik pencarian dan pencocokan, yaitu pencarian buta (*blind search*) dan pencarian terbimbing (*heuristic search*) (Kusumadewi, 2003).

b. Pencarian mendalam pertama (depth-first search)

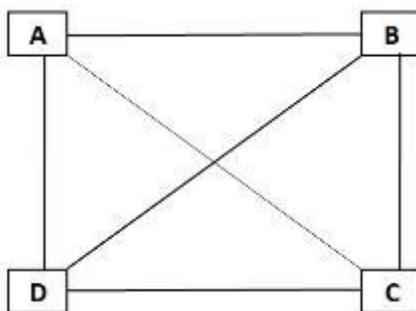
Metode pencarian buta akan mencari semua kemungkinan penyelesaian yang dimiliki. Salah satu metode pencarian buta (*blind search*) yaitu Pencarian mendalam pertama (*depth-first search*). Pencarian ini dimulai dari node akar dan dilanjutkan ke tingkat yang lebih tinggi. Proses ini diulang

sampai dengan ditemukan solusi. Keuntungan metode ini adalah membutuhkan memori yang relatif lebih kecil apabila dibandingkan dengan metode Pencarian melebar pertama. Disini hanya node-node pada lintasan yang aktif yang disimpan. Pada metode ini dimungkinkan untuk menemukan solusi tanpa harus menguji semua node (Kusumadewi, 2003).

Pada kasus terbaik, metode ini akan mencapai tujuan pada tingkat pertama, sehingga dibutuhkan waktu pencarian yang cukup singkat. Pada kasus terburuk, metode ini akan mencapai tujuan pada tingkat terakhir dengan node terakhir.

c. Traveling Salesman Problem (TSP)

Studi kasus *traveling salesman problem* (TSP) dapat diilustrasikan seorang penjual yang harus mengunjungi seluruh kota dimana masing-masing kota dikunjungi sebanyak satu kali. Dia harus mengawali dan mengakhiri perjalanan di kota yang sama. Tujuan TSP adalah menentukan lintasan atau rute yang ditempuh penjual agar memperoleh total jarak atau biaya paling minimum.



Gambar 3.1 Ilustrasi denah kasus TSP

Pada Gambar 3.1. dapat dilihat sebuah ilustrasi denah kasus TSP, terdapat empat buah kota yaitu A, B, C, dan D, masing-masing dengan jarak tempuh diantara keempat kota tersebut.

d. Algoritma Backtracking

Algoritma runut-balik (*backtracking*) pertama kali diperkenalkan oleh D.H Lehmer pada tahun 1950. Algoritma ini cukup handal digunakan untuk beberapa penyelesaian masalah dan untuk memberikan kecerdasan buatan dalam game. Algoritma runut-balik berbasis pada pencarian mendalam pertama (*depth-first search*, DFS). Aturan pencariannya akan mengikut kepada aturan pencarian DFS yaitu dengan mencari solusi dari akar ke daun dengan pencarian mendalam.

Algoritma ini secara sistematis akan mencari solusi berdasarkan ruang solusi yang ada secara sistematis namun tidak semua ruang solusi akan diperiksa, hanya pencarian yang mengarah kepada solusi yang akan diproses. Untuk memfasilitasi pencarian ini, maka ruang solusi diorganisasikan ke dalam struktur pohon. Lintasan dari akar ke daun menyatakan solusi yang mungkin (Paliyus, R., dkk., 2015).

4. METODOLOGI PENELITIAN

a. Metode Penelitian

Pada penelitian ini menggunakan metode penelitian studi pustaka. Studi pustaka bahan referensi meliputi pencarian buta (*blind search*) dengan metode pencarian mendalam pertama (*depth-first search*) serta algoritma runut-balik (*backtracking*) untuk penyelesaian persoalan *traveling salesman problem* (TSP).

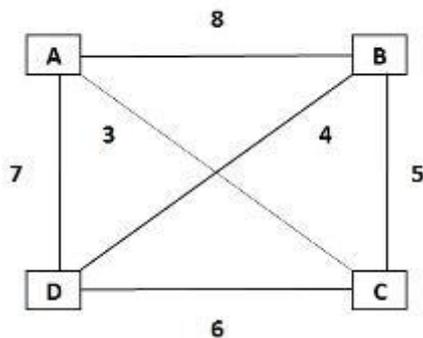
5. HASIL DAN ANALISA

a. Ilustrasi kasus TSP

Pada penelitian ini menggunakan lima buah ilustrasi kasus TSP seperti dapat dilihat pada Gambar 5.1, Gambar 5.2, Gambar 5.3, Gambar 5.4, dan Gambar 5.5. Pada TSP1 memiliki 4 buah kota dengan 6 lintasan (jalur), TSP2 memiliki 5 buah kota dengan 10 lintasan, TSP3 memiliki 6 buah kota dengan

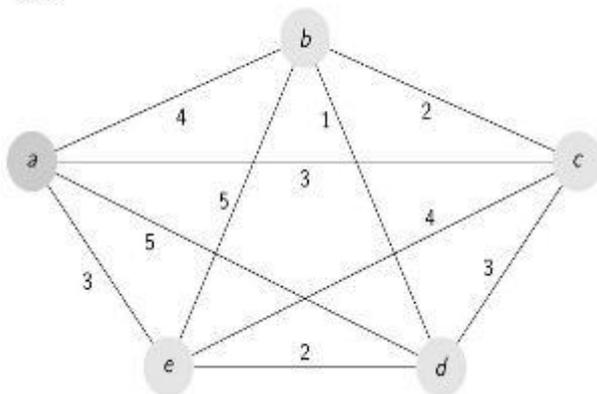
15 lintasan, TSP4 memiliki 7 buah kota dengan 21 lintasan, dan TSP5 memiliki 8 buah kota dengan 28 lintasan. Rangkuman lima buah kasus TSP ini dapat dilihat pada Tabel 1.

TSP 1

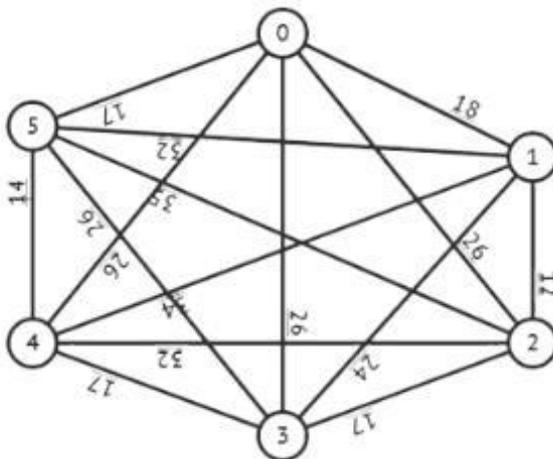


Gambar 5.1 Ilustrasi kasus TSP1

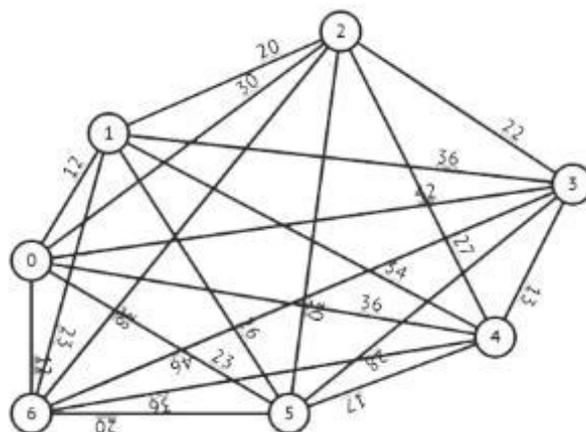
TSP2



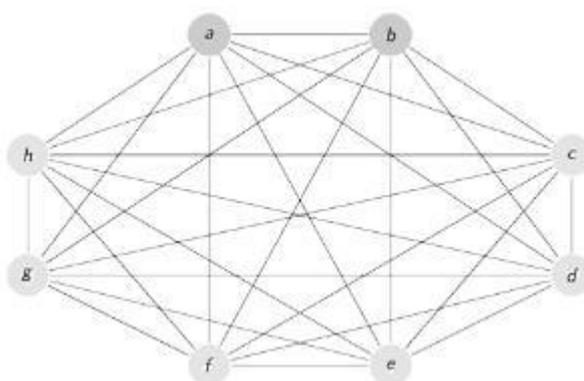
Gambar 5.2 Ilustrasi kasus TSP2



Gambar 5.3 Ilustrasi kasus TSP3



Gambar 5.4 Ilustrasi kasus TSP4



Gambar 5.5 Ilustrasi kasus TSP5

Tabel 5.1 Rangkuman lima buah kasus TSP

No	Studi kasus	Jumlah kota	Jumlah lintasan (jalur)
1	TSP1	4	6
2	TSP2	5	10
3	TSP3	6	15
4	TSP4	7	21
5	TSP5	8	28

Pada Tabel 5.2. dapat dilihat seluruh alternatif lintasan yang memungkinkan beserta panjang lintasan untuk menyelesaikan kasus TSP1 seperti pada Gambar 5.1. Diperoleh sebanyak 24 buah alternatif lintasan yang merupakan kombinasi dari urutan kunjungan ke kota A, B, C, dan D. Panjang lintasan terpilih yang terkecil adalah 12 dengan urutan lintasan terpilih adalah ACBD atau DBCA. Dua lintasan tersebut dapat dipilih salah satunya sebagai solusi dari kasus TSP karena memiliki panjang lintasan terkecil yang sama.

Tabel 5.2 Alternatif Lintasan Traveling Salesman Problem studi kasus TSP1

Pencarian ke	Lintasan	Panjang Lintasan	Lintasan Terplih	Panjang Lintasan Terpilih
1	ABCD	19	ABCD	19
2	ABDC	18	ABDC	18
3	ACBD	12	ACBD	12

4	ACDB	13	ACBD	12
5	ADBC	16	ACBD	12
6	ADCB	18	ACBD	12
7	BACD	17	ACBD	12
8	BADC	21	ACBD	12
9	BCAD	15	ACBD	12
10	BCDA	18	ACBD	12
11	BDAC	14	ACBD	12
12	BDCA	13	ACBD	12
13	CABD	15	ACBD	12
14	CADB	14	ACBD	12
15	CBAD	20	ACBD	12
16	CBDA	16	ACBD	12
17	CDAB	21	ACBD	12
18	CDBA	18	ACBD	12
19	DABC	20	ACBD	12
20	DACB	15	ACBD	12
21	DBAC	15	ACBD	12
22	DBCA	12	ACBD atau DBCA	12
23	DCAB	17	ACBD atau DBCA	12
24	DCBA	19	ACBD atau DBCA	12

b. Optimalisasi pencarian alternatif lintasan

Hasil optimalisasi pencarian alternatif lintasan atau rute pada Gambar 5.1 atau studi kasus TSP1 dapat dilihat pada Tabel 5.3. Terdapat 12 alternatif lintasan yang berbeda dari sebelumnya yaitu sebanyak 24 alternatif lintasan, sehingga variasi lintasan berkurang sebesar 50%. Prosentase ini diperoleh dari:

$$= (\text{total kombinasi alternatif lintasan} - \text{jumlah alternatif lintasan hasil optimalisasi}) / \text{total kombinasi alternatif lintasan} * 100\%$$

$$= (24 - 12) / 24 * 100\%$$

$$= 50\%$$

Tabel 5.3. Hasil optimalisasi pencarian alternatif lintasan untuk studi kasus TSP1

Nomor	Pencarian ke n (pada tabel 1)	Lintasan	Panjang Lintasan	Lintasan Terpilih	Panjang Lintasan Terpilih
1	1	ABCD	19	ABCD	19
2	2	ABDC	18	ABDC	18
3	3	ACBD	12	ACBD	12
4	4	ACDB	13	ACBD	12
5	5	ADBC	16	ACBD	12
6	6	ADCB	18	ACBD	12
7	7	BACD	17	ACBD	12
8	8	BADC	21	ACBD	12
9	9	BCAD	15	ACBD	12
10	11	BDAC	14	ACBD	12
11	13	CABD	15	ACBD	12
12	15	CBAD	20	ACBD	12

Pada Gambar 5.2 atau studi kasus TSP2 terdapat 120 buah alternatif lintasan yang merupakan kombinasi dari kunjungan ke kota A, B, C, D, dan E. Seluruh kombinasi lintasan dapat dilihat pada Tabel 5.4. Hasil optimalisasi pencarian alternatif lintasan atau rute pada Gambar 5.2 dapat dilihat pada

Tabel 5.5. Hasil optimalisasi pencarian alternatif lintasan untuk studi kasus TSP2

Nomor	Lintasan	Panjang Lintasan	Lintasan Terpilih	Panjang Lintasan Terpilih
1	AEDBC	8	AEDBC	8
2	ACBDE	8	AEDBC atau ACBDE	8
3	BDECA	10	AEDBC atau ACBDE	8
4	BDEAC	9	AEDBC atau ACBDE	8
5	CBDAE	11	AEDBC atau ACBDE	8
6	CBAED	11	AEDBC atau ACBDE	8
7	DBCAE	9	AEDBC atau ACBDE	8
8	DBCEA	10	AEDBC atau ACBDE	8
9	EDBAC	10	AEDBC atau ACBDE	8
10	EDCBA	11	AEDBC atau ACBDE	8

Panjang lintasan pada kasus TSP5 dapat dilihat pada Tabel 5.6, terdapat dua puluh delapan lintasan yang menghubungkan delapan kota. Pada Tabel 5.7 dapat dilihat rekapitulasi prosentase pengurangan variasi (alternatif) lintasan untuk seluruh kasus TSP. Dari hasil pengujian diperoleh algoritma runut-balik (*backtracking*) dapat digunakan untuk menghasilkan jumlah alternatif lintasan yang lebih sedikit jika dibandingkan dengan total kombinasi alternatif lintasan. Untuk seluruh kombinasi lintasan pada kasus TSP5 dapat dilihat pada Tabel 5.8.

Tabel 5.6. Panjang Lintasan setiap kota pada TSP5

Nomor	Kota Asal	Kota Tujuan	Panjang Lintasan
1	A	B	6
2	A	C	4
3	A	D	9
4	A	E	9
5	A	F	5
6	A	G	5
7	A	H	8
8	B	C	7
9	B	D	7
10	B	E	6
11	B	F	9
12	B	G	9
13	B	H	3
14	C	D	9
15	C	E	4
16	C	F	3
17	C	G	7
18	C	H	6
19	D	E	9
20	D	F	9
21	D	G	3
22	D	H	9
23	E	F	4
24	E	G	3
25	E	H	4
26	F	G	5
27	F	H	3

Tabel 5.7 Rekapitulasi prosentase pengurangan variasi (alternatif) lintasan

No	Studi kasus	Jumlah kota	Jumlah lintasan (jalur)	Total kombinasi alternatif lintasan	Jumlah alternatif lintasan hasil optimalisasi	Prosentase pengurangan variasi (alternatif) lintasan
1	TSP1	4	6	24	12	50
2	TSP2	5	10	120	10	91.66666667
3	TSP3	6	15	720	18	97.5
4	TSP4	7	21	5040	28	99.44444444
5	TSP5	8	28	40320	40	99.90079365

Tabel 5.8 Seluruh kombinasi lintasan untuk studi kasus TSP5

No	Kombinasi lintasan	No	Kombinasi lintasan	No	Kombinasi lintasan
1	(A, B, C, D, E, F)	31	(A, C, D, B, E, F)	61	(A, D, E, B, C, F)
2	(A, B, C, D, E, F)	32	(A, C, D, B, E, F)	62	(A, D, E, B, C, F)
3	(A, B, C, D, E, F)	33	(A, C, D, B, E, F)	63	(A, D, E, B, C, F)
4	(A, B, C, D, E, F)	34	(A, C, D, B, E, F)	64	(A, D, E, B, C, F)
5	(A, B, C, D, E, F)	35	(A, C, D, B, E, F)	65	(A, D, E, B, C, F)
6	(A, B, C, D, E, F)	36	(A, C, D, B, E, F)	66	(A, D, E, B, C, F)
7	(A, B, C, D, E, F)	37	(A, C, D, B, E, F)	67	(A, D, E, B, C, F)
8	(A, B, C, D, E, F)	38	(A, C, D, B, E, F)	68	(A, D, E, B, C, F)
9	(A, B, C, D, E, F)	39	(A, C, D, B, E, F)	69	(A, D, E, B, C, F)
10	(A, B, C, D, E, F)	40	(A, C, D, B, E, F)	70	(A, D, E, B, C, F)
11	(A, B, C, D, E, F)	41	(A, C, D, B, E, F)	71	(A, D, E, B, C, F)
12	(A, B, C, D, E, F)	42	(A, C, D, B, E, F)	72	(A, D, E, B, C, F)
13	(A, B, C, D, E, F)	43	(A, C, D, B, E, F)	73	(A, D, E, B, C, F)
14	(A, B, C, D, E, F)	44	(A, C, D, B, E, F)	74	(A, D, E, B, C, F)
15	(A, B, C, D, E, F)	45	(A, C, D, B, E, F)	75	(A, D, E, B, C, F)
16	(A, B, C, D, E, F)	46	(A, C, D, B, E, F)	76	(A, D, E, B, C, F)
17	(A, B, C, D, E, F)	47	(A, C, D, B, E, F)	77	(A, D, E, B, C, F)
18	(A, B, C, D, E, F)	48	(A, C, D, B, E, F)	78	(A, D, E, B, C, F)
19	(A, B, C, D, E, F)	49	(A, C, D, B, E, F)	79	(A, D, E, B, C, F)
20	(A, B, C, D, E, F)	50	(A, C, D, B, E, F)	80	(A, D, E, B, C, F)
21	(A, B, C, D, E, F)	51	(A, C, D, B, E, F)	81	(A, D, E, B, C, F)
22	(A, B, C, D, E, F)	52	(A, C, D, B, E, F)	82	(A, D, E, B, C, F)
23	(A, B, C, D, E, F)	53	(A, C, D, B, E, F)	83	(A, D, E, B, C, F)
24	(A, B, C, D, E, F)	54	(A, C, D, B, E, F)	84	(A, D, E, B, C, F)
25	(A, B, C, D, E, F)	55	(A, C, D, B, E, F)	85	(A, D, E, B, C, F)
26	(A, B, C, D, E, F)	56	(A, C, D, B, E, F)	86	(A, D, E, B, C, F)
27	(A, B, C, D, E, F)	57	(A, C, D, B, E, F)	87	(A, D, E, B, C, F)
28	(A, B, C, D, E, F)	58	(A, C, D, B, E, F)	88	(A, D, E, B, C, F)
29	(A, B, C, D, E, F)	59	(A, C, D, B, E, F)	89	(A, D, E, B, C, F)
30	(A, B, C, D, E, F)	60	(A, C, D, B, E, F)	90	(A, D, E, B, C, F)
181	(B, D, E, A, C, F)	211	(B, E, F, A, C, D)	241	(C, A, B, D, E, F)
182	(B, D, E, A, C, F)	212	(B, E, F, A, C, D)	242	(C, A, B, D, E, F)
183	(B, D, E, A, C, F)	213	(B, E, F, A, C, D)	243	(C, A, B, D, E, F)
184	(B, D, E, A, C, F)	214	(B, E, F, A, C, D)	244	(C, A, B, D, E, F)
185	(B, D, E, A, C, F)	215	(B, E, F, A, C, D)	245	(C, A, B, D, E, F)
186	(B, D, E, A, C, F)	216	(B, E, F, A, C, D)	246	(C, A, B, D, E, F)
187	(B, D, E, A, C, F)	217	(B, E, F, A, C, D)	247	(C, A, B, D, E, F)
188	(B, D, E, A, C, F)	218	(B, E, F, A, C, D)	248	(C, A, B, D, E, F)
189	(B, D, E, A, C, F)	219	(B, E, F, A, C, D)	249	(C, A, B, D, E, F)
190	(B, D, E, A, C, F)	220	(B, E, F, A, C, D)	250	(C, A, B, D, E, F)
191	(B, D, E, A, C, F)	221	(B, E, F, A, C, D)	251	(C, A, B, D, E, F)
192	(B, D, E, A, C, F)	222	(B, E, F, A, C, D)	252	(C, A, B, D, E, F)
193	(B, D, E, A, C, F)	223	(B, E, F, A, C, D)	253	(C, A, B, D, E, F)
194	(B, D, E, A, C, F)	224	(B, E, F, A, C, D)	254	(C, A, B, D, E, F)
195	(B, D, E, A, C, F)	225	(B, E, F, A, C, D)	255	(C, A, B, D, E, F)
196	(B, D, E, A, C, F)	226	(B, E, F, A, C, D)	256	(C, A, B, D, E, F)
197	(B, D, E, A, C, F)	227	(B, E, F, A, C, D)	257	(C, A, B, D, E, F)
198	(B, D, E, A, C, F)	228	(B, E, F, A, C, D)	258	(C, A, B, D, E, F)
199	(B, D, E, A, C, F)	229	(B, E, F, A, C, D)	259	(C, A, B, D, E, F)
200	(B, D, E, A, C, F)	230	(B, E, F, A, C, D)	260	(C, A, B, D, E, F)
201	(B, D, E, A, C, F)	231	(B, E, F, A, C, D)	261	(C, A, B, D, E, F)
202	(B, D, E, A, C, F)	232	(B, E, F, A, C, D)	262	(C, A, B, D, E, F)
203	(B, D, E, A, C, F)	233	(B, E, F, A, C, D)	263	(C, A, B, D, E, F)
204	(B, D, E, A, C, F)	234	(B, E, F, A, C, D)	264	(C, A, B, D, E, F)
205	(B, D, E, A, C, F)	235	(B, E, F, A, C, D)	265	(C, A, B, D, E, F)
206	(B, D, E, A, C, F)	236	(B, E, F, A, C, D)	266	(C, A, B, D, E, F)
207	(B, D, E, A, C, F)	237	(B, E, F, A, C, D)	267	(C, A, B, D, E, F)
208	(B, D, E, A, C, F)	238	(B, E, F, A, C, D)	268	(C, A, B, D, E, F)
209	(B, D, E, A, C, F)	239	(B, E, F, A, C, D)	269	(C, A, B, D, E, F)
210	(B, D, E, A, C, F)	240	(B, E, F, A, C, D)	270	(C, A, B, D, E, F)

No	Kombinasi lintasan	No	Kombinasi lintasan	No	Kombinasi lintasan
271	{C', 'B', 'D', 'A', 'E', 'F'}	301	{C', 'D', 'E', 'A', 'B', 'F'}	331	{C', 'E', 'F', 'A', 'B', 'D'}
272	{C', 'B', 'D', 'A', 'E', 'F'}	302	{C', 'D', 'E', 'A', 'F', 'B'}	332	{C', 'E', 'F', 'A', 'D', 'B'}
273	{C', 'B', 'D', 'E', 'A', 'F'}	303	{C', 'D', 'E', 'B', 'A', 'F'}	333	{C', 'E', 'F', 'B', 'A', 'D'}
274	{C', 'B', 'D', 'E', 'F', 'A'}	304	{C', 'D', 'E', 'B', 'F', 'A'}	334	{C', 'E', 'F', 'B', 'D', 'A'}
275	{C', 'B', 'D', 'E', 'F', 'A', 'E'}	305	{C', 'D', 'E', 'F', 'A', 'B'}	335	{C', 'E', 'F', 'D', 'A', 'B'}
276	{C', 'B', 'D', 'E', 'F', 'E', 'A'}	306	{C', 'D', 'E', 'F', 'B', 'A'}	336	{C', 'E', 'F', 'D', 'B', 'A'}
277	{C', 'B', 'D', 'E', 'A', 'D', 'E'}	307	{C', 'D', 'E', 'F', 'A', 'B', 'E'}	337	{C', 'E', 'F', 'A', 'D', 'E'}
278	{C', 'B', 'D', 'E', 'A', 'E', 'D'}	308	{C', 'D', 'E', 'F', 'A', 'E', 'B'}	338	{C', 'E', 'F', 'A', 'D', 'E', 'D'}
279	{C', 'B', 'D', 'E', 'A', 'E', 'F'}	309	{C', 'D', 'E', 'F', 'B', 'A', 'E'}	339	{C', 'E', 'F', 'A', 'D', 'B', 'E'}
280	{C', 'B', 'E', 'D', 'F', 'A'}	310	{C', 'D', 'E', 'F', 'B', 'E', 'A'}	340	{C', 'E', 'F', 'A', 'D', 'E', 'B'}
281	{C', 'B', 'E', 'F', 'A', 'D'}	311	{C', 'D', 'E', 'F', 'E', 'A', 'B'}	341	{C', 'E', 'F', 'A', 'E', 'B', 'D'}
282	{C', 'B', 'E', 'F', 'B', 'D', 'A'}	312	{C', 'D', 'E', 'F', 'E', 'B', 'A'}	342	{C', 'E', 'F', 'A', 'E', 'B', 'E'}
283	{C', 'B', 'E', 'F', 'A', 'D', 'E'}	313	{C', 'D', 'E', 'A', 'B', 'D', 'F'}	343	{C', 'E', 'F', 'A', 'D', 'E', 'E'}
284	{C', 'B', 'E', 'F', 'A', 'E', 'D'}	314	{C', 'D', 'E', 'A', 'B', 'F', 'D'}	344	{C', 'E', 'F', 'A', 'D', 'E', 'D'}
285	{C', 'B', 'E', 'F', 'D', 'A', 'E'}	315	{C', 'D', 'E', 'A', 'D', 'B', 'F'}	345	{C', 'E', 'F', 'B', 'D', 'A', 'E'}
286	{C', 'B', 'E', 'F', 'D', 'E', 'A'}	316	{C', 'D', 'E', 'A', 'D', 'F', 'B'}	346	{C', 'E', 'F', 'B', 'D', 'E', 'A'}
287	{C', 'B', 'E', 'F', 'E', 'A', 'D'}	317	{C', 'D', 'E', 'A', 'F', 'B', 'D'}	347	{C', 'E', 'F', 'B', 'D', 'E', 'D'}
288	{C', 'B', 'E', 'F', 'E', 'D', 'A'}	318	{C', 'D', 'E', 'A', 'F', 'D', 'B'}	348	{C', 'E', 'F', 'B', 'D', 'E', 'E'}
289	{C', 'D', 'A', 'B', 'E', 'F'}	319	{C', 'D', 'E', 'A', 'D', 'E', 'F'}	349	{C', 'E', 'F', 'B', 'A', 'E', 'D'}
290	{C', 'D', 'A', 'B', 'E', 'F', 'E'}	320	{C', 'D', 'E', 'B', 'A', 'F', 'D'}	350	{C', 'E', 'F', 'D', 'A', 'E', 'B'}
291	{C', 'D', 'A', 'E', 'B', 'F'}	321	{C', 'D', 'E', 'B', 'D', 'A', 'F'}	351	{C', 'E', 'F', 'D', 'B', 'A', 'E'}
292	{C', 'D', 'A', 'E', 'F', 'B'}	322	{C', 'D', 'E', 'B', 'D', 'F', 'A'}	352	{C', 'E', 'F', 'D', 'B', 'E', 'A'}
293	{C', 'D', 'A', 'E', 'F', 'B', 'E'}	323	{C', 'D', 'E', 'B', 'E', 'A', 'D'}	353	{C', 'E', 'F', 'D', 'E', 'A', 'B'}
294	{C', 'D', 'A', 'E', 'F', 'E', 'B'}	324	{C', 'D', 'E', 'B', 'E', 'D', 'A'}	354	{C', 'E', 'F', 'D', 'E', 'B', 'A'}
295	{C', 'D', 'A', 'E', 'F', 'E', 'F'}	325	{C', 'D', 'E', 'D', 'A', 'B', 'F'}	355	{C', 'E', 'F', 'D', 'E', 'B', 'D'}
296	{C', 'D', 'A', 'E', 'F', 'E', 'E'}	326	{C', 'D', 'E', 'D', 'A', 'E', 'B'}	356	{C', 'E', 'F', 'D', 'E', 'B', 'E'}
297	{C', 'D', 'A', 'E', 'F', 'A', 'E'}	327	{C', 'D', 'E', 'D', 'B', 'A', 'F'}	357	{C', 'E', 'F', 'D', 'E', 'B', 'D'}
298	{C', 'D', 'A', 'E', 'F', 'E', 'A'}	328	{C', 'D', 'E', 'D', 'B', 'F', 'A'}	358	{C', 'E', 'F', 'E', 'B', 'D', 'A'}
299	{C', 'D', 'A', 'E', 'F', 'A', 'E'}	329	{C', 'D', 'E', 'D', 'E', 'A', 'B'}	359	{C', 'E', 'F', 'E', 'B', 'D', 'B'}
300	{C', 'D', 'A', 'E', 'F', 'E', 'A'}	330	{C', 'D', 'E', 'D', 'E', 'B', 'A'}	360	{C', 'E', 'F', 'E', 'D', 'B', 'A'}

No	Kombinasi lintasan	No	Kombinasi lintasan	No	Kombinasi lintasan
361	{D', 'A', 'B', 'C', 'E', 'F'}	391	{D', 'B', 'C', 'A', 'E', 'F'}	421	{D', 'C', 'E', 'A', 'B', 'F'}
362	{D', 'A', 'B', 'C', 'E', 'F'}	392	{D', 'B', 'C', 'A', 'E', 'F'}	422	{D', 'C', 'E', 'A', 'F', 'B'}
363	{D', 'A', 'B', 'C', 'E', 'F'}	393	{D', 'B', 'C', 'E', 'A', 'F'}	423	{D', 'C', 'E', 'B', 'A', 'F'}
364	{D', 'A', 'B', 'E', 'C', 'F'}	394	{D', 'B', 'C', 'E', 'F', 'A'}	424	{D', 'C', 'E', 'B', 'E', 'A'}
365	{D', 'A', 'B', 'E', 'C', 'E', 'F'}	395	{D', 'B', 'C', 'E', 'A', 'E'}	425	{D', 'C', 'E', 'E', 'F', 'A', 'D'}
366	{D', 'A', 'B', 'E', 'F', 'C', 'E'}	396	{D', 'B', 'C', 'E', 'E', 'A', 'D'}	426	{D', 'C', 'E', 'E', 'F', 'B', 'A'}
367	{D', 'A', 'C', 'B', 'E', 'F'}	397	{D', 'B', 'E', 'A', 'C', 'F'}	427	{D', 'C', 'E', 'F', 'A', 'B', 'E'}
368	{D', 'A', 'C', 'B', 'E', 'F', 'E'}	398	{D', 'B', 'E', 'A', 'F', 'C'}	428	{D', 'C', 'E', 'F', 'A', 'E', 'B'}
369	{D', 'A', 'C', 'E', 'B', 'F'}	399	{D', 'B', 'E', 'C', 'A', 'F'}	429	{D', 'C', 'E', 'F', 'B', 'A', 'E'}
370	{D', 'A', 'C', 'E', 'F', 'B'}	400	{D', 'B', 'E', 'C', 'E', 'A'}	430	{D', 'C', 'E', 'F', 'B', 'E', 'A'}
371	{D', 'A', 'C', 'E', 'F', 'B', 'E'}	401	{D', 'B', 'E', 'E', 'A', 'C'}	431	{D', 'C', 'E', 'F', 'E', 'A', 'B'}
372	{D', 'A', 'C', 'E', 'F', 'E', 'B'}	402	{D', 'B', 'E', 'E', 'C', 'A'}	432	{D', 'C', 'E', 'F', 'E', 'B', 'A'}
373	{D', 'A', 'C', 'E', 'F', 'C', 'B'}	403	{D', 'B', 'E', 'F', 'A', 'C', 'E'}	433	{D', 'C', 'E', 'F', 'A', 'B', 'C', 'E'}
374	{D', 'A', 'C', 'E', 'F', 'C', 'E'}	404	{D', 'B', 'E', 'F', 'A', 'E', 'C'}	434	{D', 'C', 'E', 'F', 'A', 'E', 'C', 'E'}
375	{D', 'A', 'C', 'E', 'F', 'C', 'E', 'F'}	405	{D', 'B', 'E', 'F', 'C', 'A', 'E'}	435	{D', 'C', 'E', 'F', 'A', 'C', 'B', 'E'}
376	{D', 'A', 'C', 'E', 'F', 'C', 'E', 'B'}	406	{D', 'B', 'E', 'F', 'C', 'E', 'A'}	436	{D', 'C', 'E', 'F', 'A', 'C', 'F', 'B'}
377	{D', 'A', 'C', 'E', 'F', 'B', 'C'}	407	{D', 'B', 'E', 'F', 'E', 'A', 'C'}	437	{D', 'C', 'E', 'F', 'A', 'E', 'B', 'C'}
378	{D', 'A', 'C', 'E', 'F', 'C', 'B', 'E'}	408	{D', 'B', 'E', 'F', 'E', 'C', 'A'}	438	{D', 'C', 'E', 'F', 'A', 'E', 'C', 'B'}
379	{D', 'A', 'C', 'E', 'F', 'B', 'C', 'E'}	409	{D', 'C', 'A', 'B', 'E', 'F'}	439	{D', 'C', 'E', 'F', 'A', 'C', 'F'}
380	{D', 'A', 'C', 'E', 'F', 'B', 'C', 'E'}	410	{D', 'C', 'A', 'B', 'E', 'E'}	440	{D', 'C', 'E', 'F', 'A', 'C', 'E'}
381	{D', 'A', 'C', 'E', 'F', 'C', 'B', 'E'}	411	{D', 'C', 'A', 'E', 'B', 'F'}	441	{D', 'C', 'E', 'F', 'A', 'C', 'E'}
382	{D', 'A', 'C', 'E', 'F', 'C', 'E', 'B'}	412	{D', 'C', 'A', 'E', 'E', 'B'}	442	{D', 'C', 'E', 'F', 'A', 'C', 'E', 'A'}
383	{D', 'A', 'C', 'E', 'F', 'E', 'B', 'C'}	413	{D', 'C', 'A', 'E', 'D', 'E'}	443	{D', 'C', 'E', 'F', 'A', 'C', 'E'}
384	{D', 'A', 'C', 'E', 'F', 'C', 'B', 'E'}	414	{D', 'C', 'A', 'E', 'E', 'B'}	444	{D', 'C', 'E', 'F', 'A', 'C', 'E', 'A'}
385	{D', 'A', 'C', 'E', 'F', 'C', 'E', 'B'}	415	{D', 'C', 'A', 'B', 'A', 'E', 'F'}	445	{D', 'C', 'E', 'F', 'A', 'C', 'E', 'B'}
386	{D', 'A', 'C', 'E', 'F', 'C', 'E', 'E'}	416	{D', 'C', 'A', 'B', 'A', 'E', 'E'}	446	{D', 'C', 'E', 'F', 'A', 'C', 'E', 'B'}
387	{D', 'A', 'C', 'E', 'F', 'C', 'E', 'E'}	417	{D', 'C', 'A', 'B', 'A', 'E', 'E'}	447	{D', 'C', 'E', 'F', 'A', 'C', 'E', 'E'}
388	{D', 'A', 'C', 'E', 'F', 'C', 'E', 'E'}	418	{D', 'C', 'A', 'B', 'E', 'F', 'A'}	448	{D', 'C', 'E', 'F', 'A', 'C', 'E', 'A'}
389	{D', 'A', 'C', 'E', 'F', 'C', 'E', 'E'}	419	{D', 'C', 'A', 'B', 'E', 'A', 'E'}	449	{D', 'C', 'E', 'F', 'A', 'C', 'E', 'B'}
390	{D', 'A', 'C', 'E', 'F', 'C', 'E', 'E'}	420	{D', 'C', 'A', 'B', 'E', 'E', 'A'}	450	{D', 'C', 'E', 'F', 'A', 'C', 'E', 'B'}

No	Kombinasi lintasan	No	Kombinasi lintasan	No	Kombinasi lintasan
451	{E', 'A', 'B', 'C', 'D', 'F'}	481	{E', 'A', 'B', 'C', 'D', 'F'}	511	{E', 'B', 'C', 'A', 'D', 'F'}
452	{E', 'A', 'B', 'C', 'D', 'F'}	482	{E', 'A', 'B', 'C', 'D', 'F'}	512	{E', 'B', 'C', 'A', 'D', 'F'}
453	{E', 'A', 'B', 'C', 'D', 'F'}	483	{E', 'A', 'B', 'C', 'D', 'F'}	513	{E', 'B', 'C', 'D', 'A', 'F'}
454	{E', 'A', 'B', 'C', 'D', 'F'}	484	{E', 'A', 'B', 'D', 'C', 'F'}	514	{E', 'B', 'C', 'D', 'A', 'F'}
455	{E', 'A', 'B', 'C', 'D', 'F'}	485	{E', 'A', 'B', 'E', 'C', 'D'}	515	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
456	{E', 'A', 'B', 'C', 'D', 'F'}	486	{E', 'A', 'B', 'E', 'D', 'C'}	516	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
457	{E', 'A', 'B', 'C', 'D', 'F'}	487	{E', 'A', 'C', 'B', 'D', 'F'}	517	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
458	{E', 'A', 'C', 'B', 'D', 'F'}	488	{E', 'A', 'C', 'B', 'E', 'D'}	518	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
459	{E', 'A', 'C', 'B', 'D', 'F'}	489	{E', 'A', 'C', 'D', 'B', 'E'}	519	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
460	{E', 'A', 'C', 'B', 'D', 'F'}	490	{E', 'A', 'C', 'D', 'E', 'B'}	520	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
461	{E', 'A', 'C', 'B', 'D', 'F'}	491	{E', 'A', 'C', 'E', 'B', 'D'}	521	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
462	{E', 'A', 'C', 'B', 'D', 'F'}	492	{E', 'A', 'C', 'E', 'D', 'B'}	522	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
463	{E', 'A', 'C', 'B', 'D', 'F'}	493	{E', 'A', 'D', 'B', 'C', 'F'}	523	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
464	{E', 'A', 'C', 'B', 'D', 'F'}	494	{E', 'A', 'D', 'B', 'E', 'C'}	524	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
465	{E', 'A', 'C', 'B', 'D', 'F'}	495	{E', 'A', 'D', 'C', 'B', 'F'}	525	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
466	{E', 'A', 'C', 'B', 'D', 'F'}	496	{E', 'A', 'D', 'C', 'E', 'B'}	526	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
467	{E', 'A', 'C', 'B', 'D', 'F'}	497	{E', 'A', 'D', 'E', 'B', 'C'}	527	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
468	{E', 'A', 'C', 'B', 'D', 'F'}	498	{E', 'A', 'D', 'E', 'C', 'B'}	528	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
469	{E', 'A', 'C', 'B', 'D', 'F'}	499	{E', 'A', 'E', 'B', 'C', 'D'}	529	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
470	{E', 'A', 'C', 'B', 'D', 'F'}	500	{E', 'A', 'E', 'B', 'D', 'C'}	530	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
471	{E', 'A', 'C', 'B', 'D', 'F'}	501	{E', 'A', 'E', 'C', 'B', 'D'}	531	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
472	{E', 'A', 'C', 'B', 'D', 'F'}	502	{E', 'A', 'E', 'C', 'D', 'B'}	532	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
473	{E', 'A', 'C', 'B', 'D', 'F'}	503	{E', 'A', 'E', 'D', 'B', 'C'}	533	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
474	{E', 'A', 'C', 'B', 'D', 'F'}	504	{E', 'A', 'E', 'D', 'C', 'B'}	534	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
475	{E', 'A', 'C', 'B', 'D', 'F'}	505	{E', 'A', 'E', 'D', 'E', 'B'}	535	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
476	{E', 'A', 'C', 'B', 'D', 'F'}	506	{E', 'A', 'E', 'D', 'E', 'B'}	536	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
477	{E', 'A', 'C', 'B', 'D', 'F'}	507	{E', 'A', 'E', 'D', 'E', 'B'}	537	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
478	{E', 'A', 'C', 'B', 'D', 'F'}	508	{E', 'A', 'E', 'D', 'E', 'B'}	538	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
479	{E', 'A', 'C', 'B', 'D', 'F'}	509	{E', 'A', 'E', 'D', 'E', 'B'}	539	{E', 'B', 'C', 'D', 'E', 'A', 'F'}
480	{E', 'A', 'C', 'B', 'D', 'F'}	510	{E', 'A', 'E', 'D', 'E', 'B'}	540	{E', 'B', 'C', 'D', 'E', 'A', 'F'}

No	Kombinasi lintasan	No	Kombinasi lintasan	No	Kombinasi lintasan
541	('E', 'C', 'D', 'A', 'B', 'F')	571	('E', 'D', 'F', 'A', 'B', 'C')	601	('F', 'A', 'B', 'C', 'D', 'E')
542	('E', 'C', 'D', 'A', 'F', 'B')	572	('E', 'D', 'F', 'A', 'C', 'B')	602	('F', 'A', 'B', 'C', 'E', 'D')
543	('E', 'C', 'D', 'B', 'A', 'F')	573	('E', 'D', 'F', 'B', 'A', 'C')	603	('F', 'A', 'B', 'D', 'C', 'E')
544	('E', 'C', 'D', 'B', 'F', 'A')	574	('E', 'D', 'F', 'B', 'C', 'A')	604	('F', 'A', 'B', 'D', 'E', 'C')
545	('E', 'C', 'D', 'F', 'A', 'B')	575	('E', 'D', 'F', 'C', 'A', 'B')	605	('F', 'A', 'B', 'E', 'C', 'D')
546	('E', 'C', 'D', 'F', 'B', 'A')	576	('E', 'D', 'F', 'C', 'B', 'A')	606	('F', 'A', 'B', 'E', 'D', 'C')
547	('E', 'C', 'F', 'A', 'B', 'D')	577	('E', 'F', 'A', 'B', 'C', 'D')	607	('F', 'A', 'C', 'B', 'D', 'E')
548	('E', 'C', 'F', 'A', 'D', 'B')	578	('E', 'F', 'A', 'B', 'D', 'C')	608	('F', 'A', 'C', 'B', 'E', 'D')
549	('E', 'C', 'F', 'B', 'A', 'D')	579	('E', 'F', 'A', 'C', 'B', 'D')	609	('F', 'A', 'C', 'D', 'B', 'E')
550	('E', 'C', 'F', 'B', 'D', 'A')	580	('E', 'F', 'A', 'C', 'D', 'B')	610	('F', 'A', 'C', 'D', 'E', 'B')
551	('E', 'C', 'F', 'D', 'A', 'B')	581	('E', 'F', 'A', 'D', 'B', 'C')	611	('F', 'A', 'C', 'E', 'B', 'D')
552	('E', 'C', 'F', 'D', 'B', 'A')	582	('E', 'F', 'A', 'D', 'C', 'B')	612	('F', 'A', 'C', 'E', 'D', 'B')
553	('E', 'D', 'A', 'B', 'C', 'F')	583	('E', 'F', 'B', 'A', 'C', 'D')	613	('F', 'A', 'D', 'B', 'C', 'E')
554	('E', 'D', 'A', 'B', 'F', 'C')	584	('E', 'F', 'B', 'A', 'D', 'C')	614	('F', 'A', 'D', 'B', 'E', 'C')
555	('E', 'D', 'A', 'C', 'B', 'F')	585	('E', 'F', 'B', 'C', 'A', 'D')	615	('F', 'A', 'D', 'C', 'B', 'E')
556	('E', 'D', 'A', 'C', 'F', 'B')	586	('E', 'F', 'B', 'C', 'D', 'A')	616	('F', 'A', 'D', 'C', 'E', 'B')
557	('E', 'D', 'A', 'E', 'B', 'C')	587	('E', 'F', 'B', 'D', 'A', 'C')	617	('F', 'A', 'D', 'E', 'B', 'C')
558	('E', 'D', 'A', 'F', 'C', 'B')	588	('E', 'F', 'B', 'D', 'C', 'A')	618	('F', 'A', 'D', 'E', 'C', 'B')
559	('E', 'D', 'B', 'A', 'C', 'F')	589	('E', 'F', 'C', 'A', 'B', 'D')	619	('F', 'A', 'E', 'B', 'C', 'D')
560	('E', 'D', 'B', 'A', 'F', 'C')	590	('E', 'F', 'C', 'A', 'D', 'B')	620	('F', 'A', 'E', 'B', 'D', 'C')
561	('E', 'D', 'B', 'C', 'A', 'F')	591	('E', 'F', 'C', 'B', 'A', 'D')	621	('F', 'A', 'E', 'C', 'B', 'D')
562	('E', 'D', 'B', 'C', 'F', 'A')	592	('E', 'F', 'C', 'B', 'D', 'A')	622	('F', 'A', 'E', 'C', 'D', 'B')
563	('E', 'D', 'B', 'F', 'A', 'C')	593	('E', 'F', 'C', 'D', 'A', 'B')	623	('F', 'A', 'E', 'D', 'B', 'C')
564	('E', 'D', 'B', 'F', 'C', 'A')	594	('E', 'F', 'C', 'D', 'B', 'A')	624	('F', 'A', 'E', 'D', 'C', 'B')
565	('E', 'D', 'C', 'A', 'B', 'F')	595	('E', 'F', 'D', 'A', 'B', 'C')	625	('F', 'B', 'A', 'C', 'D', 'E')
566	('E', 'D', 'C', 'A', 'F', 'B')	596	('E', 'F', 'D', 'A', 'C', 'B')	626	('F', 'B', 'A', 'C', 'E', 'D')
567	('E', 'D', 'C', 'B', 'A', 'F')	597	('E', 'F', 'D', 'B', 'A', 'C')	627	('F', 'B', 'A', 'D', 'C', 'E')
568	('E', 'D', 'C', 'B', 'F', 'A')	598	('E', 'F', 'D', 'B', 'C', 'A')	628	('F', 'B', 'A', 'D', 'E', 'C')
569	('E', 'D', 'C', 'F', 'A', 'B')	599	('E', 'F', 'D', 'C', 'A', 'B')	629	('F', 'B', 'A', 'E', 'C', 'D')
570	('E', 'D', 'C', 'F', 'B', 'A')	600	('E', 'F', 'D', 'C', 'B', 'A')	630	('F', 'B', 'A', 'E', 'D', 'C')
631	('F', 'B', 'C', 'A', 'D', 'E')	661	('F', 'C', 'D', 'A', 'B', 'E')	691	('F', 'D', 'E', 'A', 'B', 'C')
632	('F', 'B', 'C', 'A', 'E', 'D')	662	('F', 'C', 'D', 'A', 'E', 'B')	692	('F', 'D', 'E', 'A', 'C', 'B')
633	('F', 'B', 'C', 'D', 'A', 'E')	663	('F', 'C', 'D', 'B', 'A', 'E')	693	('F', 'D', 'E', 'B', 'A', 'C')
634	('F', 'B', 'C', 'D', 'E', 'A')	664	('F', 'C', 'D', 'B', 'E', 'A')	694	('F', 'D', 'E', 'B', 'C', 'A')
635	('F', 'B', 'C', 'E', 'A', 'D')	665	('F', 'C', 'D', 'E', 'A', 'B')	695	('F', 'D', 'E', 'C', 'A', 'B')
636	('F', 'B', 'C', 'E', 'D', 'A')	666	('F', 'C', 'D', 'E', 'B', 'A')	696	('F', 'D', 'E', 'C', 'B', 'A')
637	('F', 'B', 'D', 'A', 'C', 'E')	667	('F', 'C', 'E', 'A', 'B', 'D')	697	('F', 'E', 'A', 'B', 'C', 'D')
638	('F', 'B', 'D', 'A', 'E', 'C')	668	('F', 'C', 'E', 'A', 'D', 'B')	698	('F', 'E', 'A', 'B', 'D', 'C')
639	('F', 'B', 'D', 'C', 'A', 'E')	669	('F', 'C', 'E', 'B', 'A', 'D')	699	('F', 'E', 'A', 'C', 'B', 'D')
640	('F', 'B', 'D', 'C', 'E', 'A')	670	('F', 'C', 'E', 'B', 'D', 'A')	700	('F', 'E', 'A', 'C', 'D', 'B')
641	('F', 'B', 'D', 'E', 'A', 'C')	671	('F', 'C', 'E', 'D', 'A', 'B')	701	('F', 'E', 'A', 'D', 'B', 'C')
642	('F', 'B', 'D', 'E', 'C', 'A')	672	('F', 'C', 'E', 'D', 'B', 'A')	702	('F', 'E', 'A', 'D', 'C', 'B')
643	('F', 'B', 'E', 'A', 'C', 'D')	673	('F', 'D', 'A', 'B', 'C', 'E')	703	('F', 'E', 'B', 'A', 'C', 'D')
644	('F', 'B', 'E', 'A', 'D', 'C')	674	('F', 'D', 'A', 'B', 'E', 'C')	704	('F', 'E', 'B', 'A', 'D', 'C')
645	('F', 'B', 'E', 'C', 'A', 'D')	675	('F', 'D', 'A', 'C', 'B', 'E')	705	('F', 'E', 'B', 'C', 'A', 'D')
646	('F', 'B', 'E', 'C', 'D', 'A')	676	('F', 'D', 'A', 'C', 'E', 'B')	706	('F', 'E', 'B', 'C', 'D', 'A')
647	('F', 'B', 'E', 'D', 'A', 'C')	677	('F', 'D', 'A', 'E', 'B', 'C')	707	('F', 'E', 'B', 'D', 'A', 'C')
648	('F', 'B', 'E', 'D', 'C', 'A')	678	('F', 'D', 'A', 'E', 'C', 'B')	708	('F', 'E', 'B', 'D', 'C', 'A')
649	('F', 'C', 'A', 'B', 'D', 'E')	679	('F', 'D', 'B', 'A', 'C', 'E')	709	('F', 'E', 'C', 'A', 'B', 'D')
650	('F', 'C', 'A', 'B', 'E', 'D')	680	('F', 'D', 'B', 'A', 'E', 'C')	710	('F', 'E', 'C', 'A', 'D', 'B')
651	('F', 'C', 'A', 'D', 'B', 'E')	681	('F', 'D', 'B', 'C', 'A', 'E')	711	('F', 'E', 'C', 'A', 'D', 'C')
652	('F', 'C', 'A', 'D', 'E', 'B')	682	('F', 'D', 'B', 'C', 'E', 'A')	712	('F', 'E', 'C', 'B', 'D', 'A')
653	('F', 'C', 'A', 'E', 'B', 'D')	683	('F', 'D', 'B', 'E', 'A', 'C')	713	('F', 'E', 'C', 'D', 'B', 'A')
654	('F', 'C', 'A', 'E', 'D', 'B')	684	('F', 'D', 'B', 'E', 'C', 'A')	714	('F', 'E', 'C', 'D', 'B', 'C')
655	('F', 'C', 'B', 'A', 'D', 'E')	685	('F', 'D', 'C', 'A', 'B', 'E')	715	('F', 'E', 'D', 'A', 'B', 'C')
656	('F', 'C', 'B', 'A', 'E', 'D')	686	('F', 'D', 'C', 'A', 'E', 'B')	716	('F', 'E', 'D', 'A', 'C', 'B')
657	('F', 'C', 'B', 'D', 'A', 'E')	687	('F', 'D', 'C', 'B', 'A', 'E')	717	('F', 'E', 'D', 'B', 'A', 'C')
658	('F', 'C', 'B', 'D', 'E', 'A')	688	('F', 'D', 'C', 'B', 'E', 'A')	718	('F', 'E', 'D', 'B', 'C', 'A')
659	('F', 'C', 'B', 'E', 'A', 'D')	689	('F', 'D', 'C', 'E', 'A', 'B')	719	('F', 'E', 'D', 'C', 'A', 'B')
660	('F', 'C', 'B', 'E', 'D', 'A')	690	('F', 'D', 'C', 'E', 'B', 'A')	720	('F', 'E', 'D', 'C', 'B', 'A')

6. KESIMPULAN DAN SARAN

a. Kesimpulan

Dari hasil pengujian diperoleh algoritma runut-balik (*backtracking*) dapat digunakan untuk menghasilkan jumlah alternatif lintasan yang lebih sedikit jika dibandingkan dengan total kombinasi alternatif lintasan. Semakin besar jumlah *node* (kota) dan jumlah lintasan (jalur) maka prosentase pengurangan variasi (alternatif) akan lintasan semakin besar.

b. Saran

Saran untuk penelitian selanjutnya adalah menambah variasi jumlah *node* (kota) dan jumlah lintasan (jalur) serta menampilkan visualisasi rute yang dipilih.

DAFTAR PUSTAKA

[1] Aristi, G., 2014, Perbandingan Algoritma Greedy, Algoritma Cheapest Insertion Heuristics dan Dynamic Programming Dalam Penyelesaian Travelling Salesman Problem, Jurnal Paradigma, Vol. XVI, No. 2, Hal. 52 - 58, September 2014.
 [2] Desiani, A. dan Arhami, M., 2006, Konsep Kecerdasan Buatan, Penerbit Andi, Yogyakarta.

- [3] Erdiwansyah, Gani, T. A., 2016, Analisis Perbandingan Metode Local Search dan Population Based Dalam Algoritma Berevolusi untuk Penyelesaian Travelling Salesman Problem (TSP), Jurnal Serambi Engineering, Volume I, Nomor 1, Hal. 111, Agustus 2016.
- [4] Fatmawati, Prihandono, B., Noviani, E., 2015, Penyelesaian Travelling Salesman Problem Dengan Metode Tabu Search, Buletin Ilmiah Mat. Stat. Dan Terapannya (Bimaster), Vol. 04, No.1, hal 17 – 24, (2015).
- [5] Kusumadewi, S., 2003, Artificial Intelligence (Teknik dan Aplikasinya), Graha Ilmu, Yogyakarta.
- [6] Kusumadewi, S. dan Purnomo, H., 2005, Penyelesaian Masalah Optimasi dengan Teknik-teknik Heuristik, Graha Ilmu, Y ogyakarta.
- [7] Murti, H., Soelistijadi, R., Sugiyamto, 2017, Analisa Kebutuhan Waktu pada Proses Penyelesaian Traveling Salesman Problem, Prosiding SINTAK 2017 Hal. 74 - 78, Universitas Stikubank Semarang.
- [8] Paliyus, R., Muhammad Danial, M., Udjulawa, D.,, 2015, Implementasi Backtracking Dalam Permainan Cari Kata untuk Mengenalkan Nama dan Fungsi Anggota Tubuh, <http://eprints.mdp.ac.id/1681/>, STMIK MDP GI Palembang.
- [9] Russell, S., Norvig, P., 2010, Artificial Intelligence A Modern Approach Third Edition, Pearson, USA.
- [10] Widodo, A.W., Mahmudy, W.F., 2010, Penerapan Algoritma Genetika pada Sistem Rekomendasi Wisata Kuliner, Jurnal Ilmiah KURSOR Vol. 5, No. 4, hlm. 205-211, Juli 2010.
- [11] Yunus, H., Helmi, Martha, S., 2015, Metode Program Dinamis pada Penyelesaian Traveling Salesman Problem, Buletin Ilmiah Mat. Stat. dan Terapannya, Volume 04, No. 3 (2015), hal 329 – 336.
- [12] <https://visualgo.net/bn/tsp>