ISSN :2685-2373

PENERAPAN ALGORITMA *BRANCH AND BOUND* PADA PERSOALAN PEDAGANG KELILING (*TRAVELLING SALESMAN PROBLEM*)

Justin Eduardo Simarmata[1]*, Elly Rosmaini, Normalina Napitupulu
Universitas Timor, Universitas Sumatera Utara, Universitas Sumatera Utara
*justinesimarmata@unimor.ac.id
Dikirim: 02 Oktober 2019. Diterima: 28 Oktober 2019. Dipublikasikan: 31 Januari 2020

ABSTRAK

Persoalan pedagang kelilingmerupakan persoalan optimasi untuk mencari perjalanan terpendek bagi pedagang keliling yang ingin berkunjung ke beberapa kota, dan kembali ke kota asal keberangkatan. Beberapa metode telah digunakan untuk memecahkan persoalan TSP. Namun, pada zaman yang serba praktis sekarang ini dibutuhkan algoritma yang dapat menyelesaikan TSP dengan cepat sehingga diperoleh solusi yang mendekati. Penelitianinimembahas tentang algoritma *Branch and Bound* dalam menyelesaikan persoalan TSP. Dengan menerapkan algoritma *Branch and Bound* pada contoh soal persoalan pedagang keliling (*Travelling Salesman Problem*) dalam penelitian ini maka diperoleh rute perjalanan terpendekyaitu A → D → B → E → C → A. Dengan total biaya perjalanan sebesar 17 rupee.Dengan menggunakan algoritma *Branch and Bound* pada Persoalan Pedagang Keliling (*Travelling Salesman Problem*) diperoleh sebanyak 5 percabangan (*Branch*) yang menghasilkan biaya minimum.

Kata kunci: Algoritma*Branch and Bound*, Persoalanpedagangkeliling, TSP

ABSTRACT

Travelling Salesman Problem (TSP) is an optimization problem to find the shortest trip out for travelling traders who want to visit several citiesand return toorigin city. Several methods have been used to solve the TSP problem. Nowadays, it is needed algorithm that can solve TSPefficientlythereforeapproximate solution is obtained. This study discusses the Branch and Bound algorithm in solving TSP problems. By applying the Branch and Bound algorithm to the TSP, the shortest travel route with the most minimal cost is obtained.
Keywords: Branch and Bound Algorithm, Travelling Salesman Problem, TSP

Pendahuluan

Persoalan pedagang keliling (*Travelling Salesman Problem*-TSP) merupakan persoalan optimasi untuk mencari perjalanan terpendek bagi pedagang keliling yang ingin berkunjung ke beberapa kota, dan kembali ke kota asal keberangkatan. TSP merupakan persoalan yang sulit bila dipandang dari sudut komputasinya. Nama persoalan ini diilhami oleh masalah seorang pedagang yang berkeliling mengunjungi sejumlah kota. Deskripsi persoalannya adalah bagaimana menemukan rute perjalanan paling murah dari suatu kota dan mengunjungi semua kota lainnya, masing-masing kota hanya dikunjungi satu kali, dan harus kembali ke kota asal keberangkatan. Cara termudah untuk menyelesaikan TSP yaitu dengan mencoba semua kemungkinan rute dan mencari rute yang terpendek. Namun, pada zaman yang serba praktis sekarang ini dibutuhkan algoritma yang dapat menyelesaikan TSP dengan cepat sehingga diperoleh solusi yang mendekati solusi optimal. Oleh karena itu digunakan algoritma *branch and bound* untuk menentukan perjalanan terpendek yang melalui kota lainnya hanya sekali dan kembali ke kota asal keberangkatan.

Penelitian ini bertujuan untuk menerapkan Algoritma *Branch and Bound* dalam menyelesaikan persoalan pedagang keliling (*Travelling Salesman Problem*) dengan menggunakan Algoritma *Branch and Bound.*

MetodePenelitian

Langkah-langkah yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Studi Literatur
   Penelitian ini diawali dengan mempelajari dan mengenal lebih dalam tentang Algoritma *Branch and Bound* dan *Travelling Salesman Problem* (TSP). Penulis membaca dan mempelajari beberapa buku dan jurnal yang berkaitan persoalan *Travelling Salesman Problem*.
2. Menjelaskan definisi Persoalan Pedagang Keliling (*Travelling Salesman Problem*) dan Algoritma *Branch and Bound*.
3. Membahas serta memahami konsep Persoalan Pedagang Keliling (*Travelling Salesman Problem*) dan Algoritma *Branch and Bound*.
4. Menjelaskan contoh penyelesaian Persoalan Pedagang Keliling (*Travelling Salesman Problem*) dan Algoritma *Branch and Bound*.
5. Memaparkan serta menjelaskan dalam penyelesaian Persoalan Pedagang Keliling (*Travelling Salesman Problem*) dan Algoritma *Branch and Bound*.
6. Menarik kesimpulan
   Menyimpulkan hasil dan informasi dari penyelesaian permasalahan yang telah diselesaikan.

HasilPenelitiandanPembahasan

Branch and Bound adalah suatu yang membagi (*divide*) dan memilih yang tepat mengurangi masalah asli menjadi satu problema lebih kecil dari subpersoalan dan kemudian secara berulang memecahkan masalah subpersoalan tersebut.

Ada tigakomponendalamalgoritmaini, yaitu:

1. FungsiPembatas (*Bounding*): fungsi yang disediakan*subspace*dariruangsolusidengan batas rendahuntuknilaisolusiterbaik yang diperolehdalam*subspace*.
   Metode dalam *bounding* ada dua, yaitu:
   a) Metode *Upper bounding*: suatu metode untuk menentukan suatu batas atas pada solusi optimal.
   b) Metode *lower bounding*: suatu metode untuk menentukan suatu batas bawah dari fungsi objektif.
2. Strategi Pencarian: suatu strategi untuk menyeleksi tiap-tiap node yang dihasilkan dan mendapatkan node yang optimum.
3. Metode Percabangan (*Branching*): suatu metode yang diaplikasikan jika *subspace* setelah diperiksa tidak dapat dibatalkan, karena itu pembagian *subspace* kedalam dua atau lebih *subspace* untuk diperiksa dalam sub rangkaian iterasi.

PenerapanAlgoritma*Branch and Bound*dalamPenyelesaian*Travelling Salesman Problem* (TSP)

Metode *Branch and Bound* sebenarnya bukan merupakan metode yang mutlak untuk menyelesaikan permasalahan *Travelling Salesman Problem* (TSP), metode ini merupakan kumpulan dari berbagai masalah (*Class of Solving Problem*), hanya saja persamaan karakteristik cara-cara tersebut yang disebut *Branch and Bound*. Berikutpersoalan *Travelling Salesman Problem* (TSP) yang diselesaikan dengan menggunakan algoritma *Branch and Bound*.

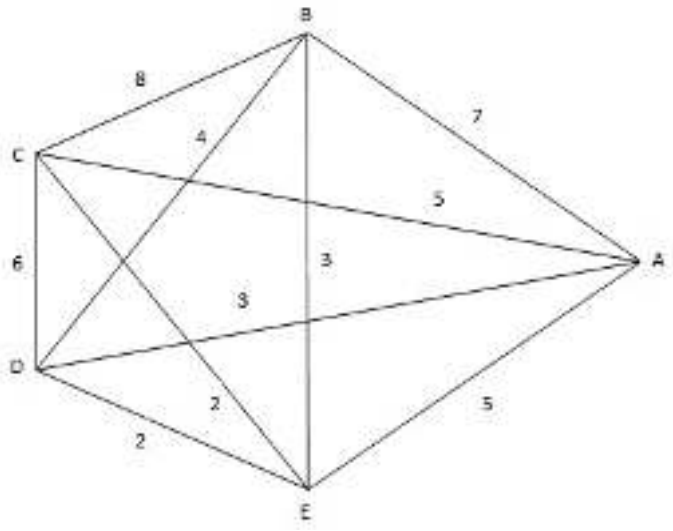
Seorang *Salesman* telah merencanakan untuk mengunjungi 5 kota. Dia akan memulai perjalanan dari kota tertentu, mengunjungi setiap kota hanya sekali dan kembali ke kota awal keberangkatan. Biaya perjalanan *Salesman* dalam rupee diberikan dalam tabel di bawah ini. Tentukan biaya rute perjalanan paling murah.

ISSN :2685-2373

Tabel1. Biayaperjalanan *Salesman* dalam rupee

Dari kota \ Ke kota	A	B	C	D	E
A	∞	7	5	3	5
B	7	∞	8	4	3
C	5	8	∞	6	2
D	3	4	6	∞	2
E	5	3	2	2	∞

Persoalan di atas dapat digambarkan dalam bentuk sebuah graf tak berarah (*undirected graph*) yang dapat dilihat pada gambar1 berikut:



Gambar 1. Graf tak berarah (*undirected graph*)

Langkah 1:

i. Berangkat dari A → A, B → B, dan seterusnya adalah tidak diijinkan sehingga diberi tanda ∞ untuk sel dalam biaya matriks. Menetapkan biaya *penalty* terbesar, yang di mana disimbolkan dengan S. Tabel 2 di bawah ini menunjukkan *cost matrix* (C_{ij}).

Tabel 2. *Cost matrix* (C_{ij})

Dari kota \ Ke kota	A	B	C	D	E
A	∞	7	5	3	5
B	7	∞	8	4	3
C	5	8	∞	6	2
D	3	4	6	∞	2
E	5	3	2	2	∞

ii. Kurangkan matriks: untuk kasus minimalisasi, mencari biaya terkecil untuk setiap baris, dan kemudian menggunakan biaya terkecil tersebut untuk mengurangi semua biaya yang ada pada baris yang sama.

ISSN :2685-2373

Tabel 3. Reduksibaris

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

iii. Memastikan semua baris dan kolom sudah memiliki nilai nol. Apabila masih ada kolom yang belum memiliki nilai nol, maka dicari nilai terkecil pada kolom tersebut untuk selanjutnya digunakan untuk mengurangi semua nilai yang ada pada kolom tersebut. Hasil matriks yang dikurangkan elemen terkecil disetiap baris dan setiap kolom disebut sebagai matriks (C'_{ij}). Semua angka yang digunakan untuk mengurangi tiap baris atau kolom tersebut kemudian dijumlahkan. Hasil dari penjumlahan inilah yang kemudian dijadikan sebagai *r* (*root*) atau nilai ongkos dari simpul awal atau akar. Hal ini juga berarti bahwa solusi pada persoalan TSP tersebut paling tidak memiliki bobot minimum sebesar nilai *r* (*root*) yang diperoleh tersebut.

Tabel 4. *Cost matrix* (C'_{ij}) dengan 0 disetiap baris dan setiap kolom

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

Dari tabel 3 dan tabel 4 diperoleh $r = (3 + 3 + 2 + 2 + 2) + (1 + 1) = 14$.

Langkah 2:

Menentukan *penalty* pada angka 0 yang ada dalam sel (C'_{ij}). Tetapkan bahwa apabila menggunakan jalur (*h,k*), harus digunakan elemen di baris *h* dan beberapa elemen di kolom *k*. Dengan demikian *(h,k)* dijumlahkan dengan elemen yang paling kecil dari baris *h* dan elemen yang paling kecil dari kolom *k*. Hasil *penalty* ini akan dicatat di sudut atas sebelah kanan di dalam sel angka 0 seperti dalam tabel 5 berikut.

Tabel 5. Hasilpenghitungan*penalty*

Dari kota \ Ke kota	A	B	C	D	E
A	∞	3	2	0^{2}	2
B	3	∞	5	1	0^{1}
C	2	5	∞	4	0^{2}
D	0^{2}	1	4	∞	0^{0}
E	2	0^{1}	0^{2}	0^{0}	∞

Langkah 3:

Misalkan (*h,k*) merupakan entri 0 dengan *penalty* yang paling besar. Sekarang, S dibagi menjadi dua bagian yaitu: S(*h,k*) yang di mana melalui jalur (*h,k*) dan S($\overline{h,k}$), yang dimana tidak melalui jalur (*h,k*). Selanjutnya menghitung *lower bound* pada harga semua rute pada masing-masing bagian.

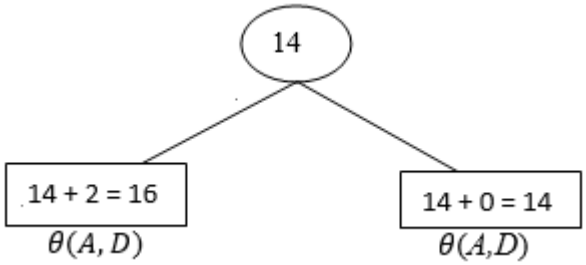
ISSN :2685-2373

Jika (h,k) tidak digunakan, di dalam penjumlahan pengurangan r, ada biaya yang paling minimal P_{hk}. Oleh karena itu, *lower bound*$\theta(\overline{h,k})$ diberikan sebagai berikut:

$$\theta(\overline{h,k}) = \text{r} + P_{hk}$$

Dalam hal ini diperoleh r = 14 dan P_{ad} = 2
Maka $\theta(\overline{A,D})$ = 14 + 2 = 16.
Total biaya dari rute yang tidak berisikan jalur yang melalui *penalty* yang paling besar adalah = $\theta(\overline{A,D})$ = 16.

Langkah 4:
Menentukan *lower bound* untuk S(h,k). Dalam persoalan di atas $C'_{DA} = \infty$ dan menghapus baris A dan kolom D. Hasil matriks dapat dilihat pada tabel 3.6. hasil langkah 3 dan langkah 4 dapat dibuat seperti gambar 2.



Gambar 2. Hasillangkah 3 danlangkah 4

Tabel setelah menghapus baris A dan kolom D dapat diliihat pada tabel 6 berikut:

Tabel 6. Penghapusan baris A dan kolom D

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

Langkah 5:
Dengan mengetahui $\theta(\overline{A,D})$ dan $\theta(A,D)$, salah satu yang lebih rendah untuk dipartisi lanjut sebagai subset S (h,k). Jika $\theta(A,D)$ lebih rendah, kembali ke langkah 2 dan ulangi menghitung *cost* matriks yang diperoleh pada langkah 4 (Tabel 3.7). Jika $\theta(\overline{A,D})$ adalah yang dipilih, kembali ke bentuk semula yaitu mengurangi matriks (langkah 1), letakkan pada sel (A,D) = ∞ dan ulangi langkah-langkah dari 2 ke depan. Dalam persoalan yang diselesaikan $\theta(A,D)$ adalah batas yang lebih rendah. Dengan menerapkan langkah 2 maka diperoleh tabel 7 yang di mana *penalty* telah dicatat di sudut kanan atas pada sel yang berisi entri 0.

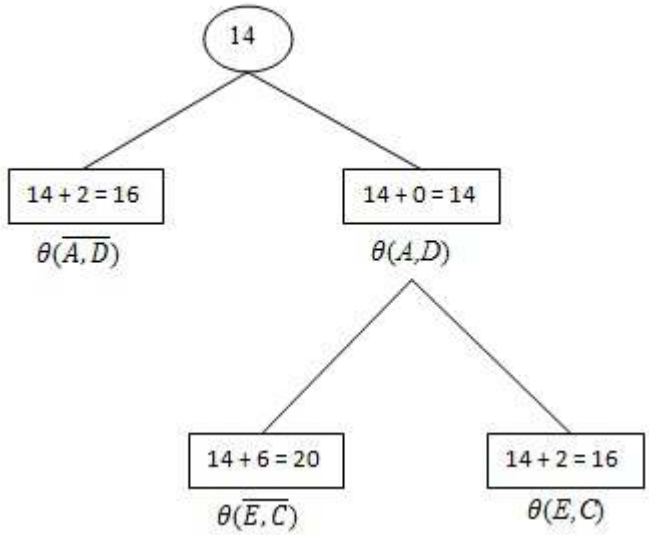
Tabel 7. Hasil penghitungan *penalty*

ISSN :2685-2373

Dari kota \ Ke kota	A	B	C	E
B	1	∞	5	0 [1]
C	0 [0]	5	∞	0 [0]
D	∞	1	4	0 [1]
E	0 [0]	0 [1]	0 [4]	∞

Langkah 6:
Sel (*E,C*) memiliki nilai *penalty* yang paling besar yaitu 4. Sekarang, bagi himpunan S (*h,k*) kedalam dua bagian, yaitu (*E,C*) dan yang lain yang tidak melalui (*E,C*) dan hitung batas yang lebih rendah (*lower bounds*) pada semua rute di masing-masing bagian.
Total biaya rute yang melalui *penalty* paling besar $= \theta(\overline{E,C}) = 14 + 6 = 20$.

Langkah 7:
Menentukan total rute yang melalui *penalty* yang paling besar. Ini dilakukan dengan ketentuan bahwa jika digunakan rute (E,C) maka jalur (C,E) tidak boleh digunakan kembali. Dengan demikian letakkan tanda $C'_{CE} = \infty$ dan menghapus baris E dan kolom C. Dari matriks yang dipilih, ambil satu elemen matriks yang paling kecil dari masing-masing baris dan kolom dan kemudian jumlahkan dengan matriks yang batas bawah yang dipilih.
Hasil yang diperoleh dari langkah 6 dan langkah 7 dimuat dalam gambar 3 dan tabel 8.



Gambar 3. Hasil yang diperolehdarilangkah 6 danlangkah 7

Tabel 8. Setelah penghapusan baris E dan kolom C

Dari kota \ Ke kota	A	B	E
B	1	∞	0
C	0	5	∞
D	∞	1	0

Langkah 8:

ISSN :2685-2373

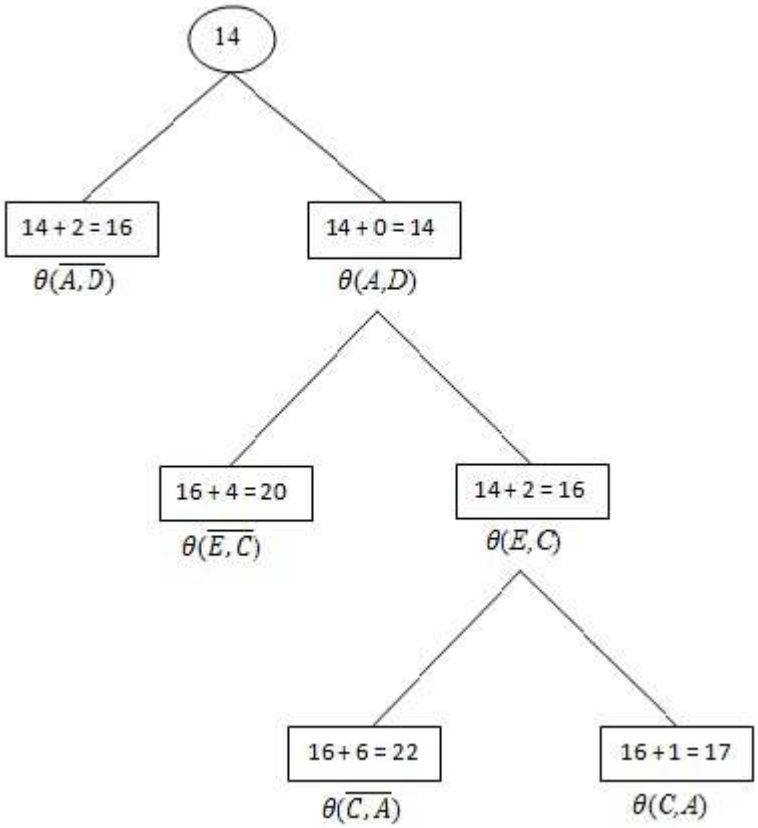
Hasil dari $\theta(\overline{E,C})$ dan $\theta(E,C)$, pilih salah satu yang memiliki nilai lebih rendah. Karena $\theta(E,C)$ adalah lebih rendah, maka ulangi langkah 2 dan mengulanginya dengan menggunakan *cost matrix* yang diperoleh pada langkah 7 (Tabel 8). Setelah mengurangi dengan elemen matriks yang paling kecil sehingga setiap baris dan kolom memiliki entri 0 (Tabel 9) pada setiap sel. Nilai *penalty* dicatat pada sudut kanan atas dari sel yang memiliki entri 0 yang dapat dilihat pada tabel 9.

Tabel 9. Hasilpenghitungan*penalty*

Dari kota \ Ke kota	A	B	E
B	1	∞	0 [1]
C	0 [5]	4	∞
D	∞	0 [4]	0 [0]

Langkah 9:
Sel (C,A) memiliki nilai *penalty* yang paling besar yaitu 5.
Biaya rute yang tidak melalui sel (C,A); $\theta(\overline{C,A})$ = 16 + 6 = 22.

Langkah 10:
Jika digunakan jalur (C,A) maka jalur (D,E) harus diberi tanda $C'_{DE} = \infty$. Dan menghapus baris C dan kolom A.
Biaya rute yang tidak melalui sel (C,A); θ(C,A) = 16 + 1 = 17.



Gambar 4. Hasil yang diperolehdarilangkah 9 danlangkah 10

ISSN :2685-2373

Langkah 11:
Karena (D,E) tidak boleh dilalui maka diberi tanda $C'_{DE} = \infty$ dan nilai *penalty* ada pada sudut atas kanan yang terdapat pada entri 0 yang ada pada sel.

Tabel 10. Hasilpenghitungan*penalty*

Dari kota \ Ke kota	B	E
B	∞	0 ∞
D	0 ∞	∞

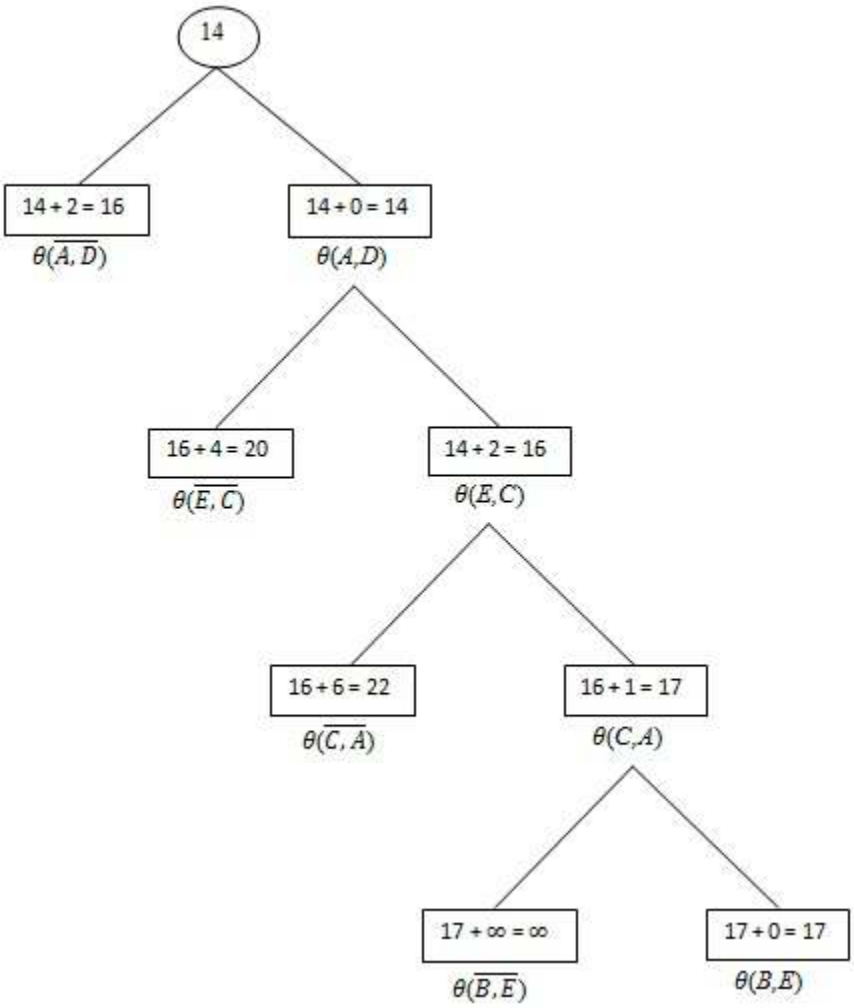
Langkah 12:
Pilih sel (B,E) sehingga diperoleh : $\theta(\overline{B,E}) = 17 + \infty = \infty$.

Langkah 13:
Hapus baris B dan kolom E dan diperoleh tabel 11.

Tabel 11. Penghapusanbaris B dankolom E

Dari kota \ Ke kota	B	E
B		
D	0	

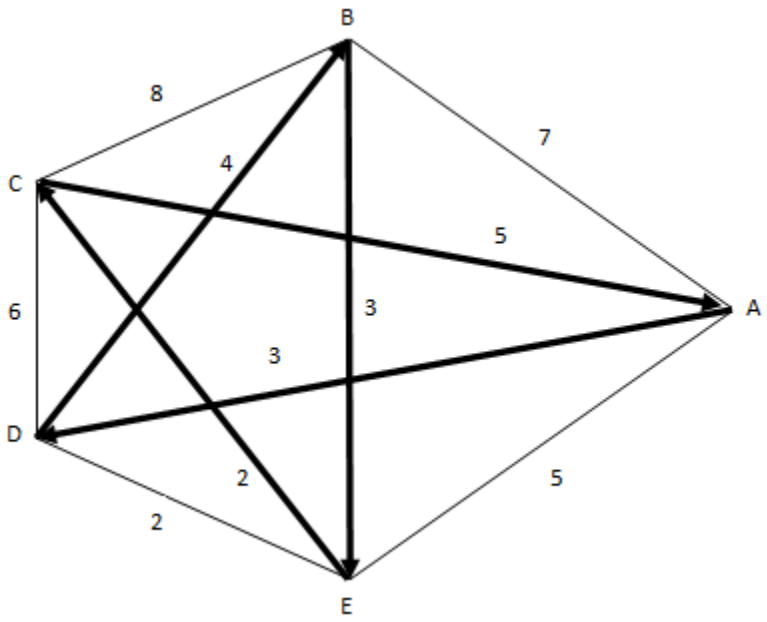
ISSN :2685-2373



Gambar 5. Hasildarilangkah 1 sampaidenganlangkah 13

Dengan demikian diperoleh rute perjalanan A → D → B → E → C → A. Dari rute perjalanan yang dilakukan pedagang keliling tersebut dapat digambarkan dalam bentuk graf berarah (*directed graph*) yang dapat dilihat pada Gambar 6.

ISSN :2685-2373



Gambar6. Graf yang mempresentasikanbiayaperjalanan yang dilalui*Salesman*

Total biaya perjalanan = (A,D) + (D,B) +(B,E) + (E,C) + (C,A)
= 3 + 4 + 3 + 2 + 5
= 17

SimpulandanSaran

Simpulan

Dengan menerapkan algoritma *Branch and Bound* pada contoh soal persoalan pedagang keliling (*Travelling Salesman Problem*) dalam penelitian ini maka diperoleh rute perjalanan terpendek yaitu A → D → B → E → C → A. Dengan total biaya perjalanan sebesar 17 rupee.Dengan menggunakan algoritma *Branch and Bound* pada Persoalan Pedagang Keliling (*Travelling Salesman Problem*) diperoleh sebanyak 5 percabangan (*Branch*) yang menghasilkan biaya minimum.

Saran

Berdasarkan hasil yang telah diperoleh dari pembahasan terdahulu maka disarankan kepada peneliti berikutnya agar mencoba menerapkan algoritma yang lain seperti algoritma genetika, *brute force*, dan algoritma semut dalam menyelesaikan Persoalan Pedagang Keliling (*Travelling Salesman Problem*).

DaftarPustaka

Firdaus, Yusrah N, dkk. (2019). Implementasi Algoritma *Branch and Bound* dalam Penentuan Jumlah Produksi Untuk Memaksimalkan Keuntungan. *Jurnal String*, 4(1), (pp. 65-70) . Jakarta: Universitas Indraprasta PGRI Jakarta.

Gillet, Billy. E. (1976). Introduction to Operations Research: A Computer-Oriented Algorithmic Approach. United States of America: McGraw-Hill.

Gupta, Prem Kumar dan D. S. Hira. (2007). *Operations Research*. Ram Nagar-New Delhi: Rajendra Ravindra.

Guritnowati, dkk. (2014). PenerapanAlgoritmaBranch and BounduntukMenentukanRuteObjekWisata di Kota Semarang. *JurnalMatematika*, 3(1), (pp.50-55). Semarang: UniversitasNegeri Semarang

Mulyono, Sri. (2004). RisetOperasi. Jakarta: FakultasEkonomiUniversitas Indonesia.

Munir, Rinaldi. (2010). MatematikaDiskrit. Bandung: Informatika.

Liu, C. L. (1995). Dasar-dasarMatematikaDiskritEdisiKedua. Jakarta: GramediaPustakaUtama.

Riyanto, Agus. (2014). UsulanPerbaikanRutePengirimandenganMenggunakanMetode*Nearest Neighbourdan Branch and Bound* di *home industry*donatenak Bandung. Jurnal OnlineInstitutTeknologi Nasional, 2(2), (pp. 278-287). Bandung: ITN.

Siagian, P. (2006). PenelitianOperasional :TeoridanPraktek. Jakarta :Universitas Indonesia.

Supatimah, Sri S, dkk. (2019). OptimasiKeuntungandenganMetode*Branch and Bound*. *JurnalAksioma*, 10(1), (pp. 13-23). Lampung : UIN RadenIntan Lampung.

Taha, Hamdy. A. (2007). *Operations Research an Introductionm Eight Edition*. Fayetteville: University of Arkansas.

Utomo, dkk. (2004). MinimasiBiayaDistribusi Tempe DenganMenggunakanMetode*Travelling Salesman Problem* (TSP)(StudiAnalisa Usaha Kecil HikmaSanan Malang). JurnalTeknikPertanian, 5(2), (pp. 87-94). Malang: UniversitasBrawijaya Malang.