PENYELESAIAN *OPEN VEHICLE ROUTING PROBLEM* MENGGUNAKAN METODE HEURISTIK SARIKLIS-POWELL

INDAKA, A.[1)], SISWANDI[2)], DAN F. HANUM[2)]

[1)]Mahasiswa Program Studi Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Pertanian Bogor
Jl Meranti, Kampus IPB Darmaga, Bogor 16680, Indonesia

[2)]Departemen Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Pertanian Bogor
Jl Meranti, Kampus IPB Darmaga, Bogor 16680, Indonesia

Abstrak : Masalah rute kendaraan terbuka (*open vehicle routing problem*) sehingga kendaraan tidak diperlukan untuk kembali ke depot merupakan bagian dari *Vehicle Routing Problem* (VRP)yang mengharuskan setiap konsumen dikunjungi sekali dan hanya sekali dengan tepat satu kendaraan.Metode heuristik yang digunakan untuk menyelesaikan masalah ini merupakan suatu algoritme yang terdiri dari beberapa fase.Fase pertama ialah fase pembentukan cluster yang seimbang, sedangkan fase kedua ialah fase penentuan rute. Fase kedua dilakukan dengan penentuan *minimumspanning tree*(MST) dengan algoritme Prim, pemodifikasian MST dengan fungsi penalti, kemudian pengubahan solusi takfisibel menjadi solusi fisibel.

Kata kunci: Manajemen distribusi, metode heuristik, *open* v*ehicle routing problem*.

1. PENDAHULUAN

Masalah transportasi dan distribusi produk dapat dimodelkan sebagai *vehicle routing problem* (VRP). Model VRP akan menghasilkan rute kendaraan untuk mengunjungi setiap konsumen. Pada umumnya, setiap rute berawal dan berakhir pada tempat yang sama, yaitu depot. Selain itu, model VRP juga memastikan agar total permintaan pada suatu rute tidak melebihi kapasitas kendaraan yang beroperasi. Permasalahan dapat

terjadi ketika perusahaan tidak memiliki kendaraan atau banyaknya kendaraan tidak dapat memenuhi permintaan konsumen, sehingga perusahaan diharuskan menyewa kendaraan lain. Kendaraan sewa akan mengunjungi konsumen dan tidak kembali lagi ke depot. Untuk memecahkan masalah tersebut digunakanlah *open vehicle routing problem* (OVRP) yaitu model VRP dengan rute yang terbuka.

Model OVRP juga dapat digunakan untuk menentukan solusi masalah penentuan rute pada model VRP (yang mengharuskan kendaraan kembali ke depot), jika kendaraan tersebut diharuskan mengunjungi kembali konsumen yang telah diantarkan pesanannyadengan urutan terbalik. Karena bobot (jarak atau biaya) rute perjalanan pergi dari depot sama dengan bobot rute perjalanan pulang kembali ke depot, maka rute keseluruhan dapat diperoleh dengan hanya menentukan sala, h satu rute (pergi atau pulang) saja yang dimodelkan sebagai OVRP. Contoh masalah seperti ini misalnya dalam proses pengiriman dan pengumpulan tabung elpiji. Kendaraan mengunjungidan mengirimkan tabung elpiji yang telah dipesanoleh konsumen. Ketika kendaraan mencapai konsumen terakhir dan barang pesanan yang ada di dalam kendaraan telah kosong, maka kendaraan kembali ke depot sambil mengumpulkan tabung elpiji yang kosong dari konsumen melalui rute yang sama namun dengan urutan terbalik mulai dari konsumen terakhir.

Masalah OVRP tidak banyak dibahas sebelum tahun 2000.Solusi OVRP ditentukan dengan cara menentukan *path* Hamilton (suatu *path* yang melewati semua simpul tepat satu kali) yang merupakan masalah NP-*Complete* (Papadimitriou & Steiglitz 1982) sehingga masalah ini tidak mudah diselesaikan. Oleh karena itu dikembangkan beberapa metode heuristik untuk mengatasi masalah ini.Metode heuristik atau metaheuristik yang telah dikembangkan antara lain*tabu search* (Fu *et al.* 2005), Li *et al.* (2007), Li *et al.* (2009), Derigs & Reuter (2009),*single parameter metaheuristic* (Tarantilis *et al.*2005), *large neighbourhood search* (Pisinger & Ropke 2007),metode heuristik berbasis *integer linear programming* (Salari *et al.* 2010). Dalam penelitian ini masalah OVRP diselesaikan dengan metode seperti dalam (Sariklis & Powell 2000) dan dilengkapi dengan contoh implementasi.

2 PEMBAHASAN

2.1 Formulasi Masalah: Ciri utama permasalahan yang membedakan OVRP dengan VRP adalah kendaraan tidak diharuskan kembali ke depot. Misalkan terdapat sebuah depot dan himpunan konsumen yang memiliki permintaan terhadap barang. Pada depot terdapat sejumlah kendaraan transportasi. Setiap kendaraan memiliki kapasitas maksimum barang yang bisa dibawa dantiap kendaraan juga memiliki biaya operasional. Biaya perjalanan antara depot dan semua konsumen, seperti juga darikonsumen ke konsumen, diketahui.Permasalahannyaialah menentukan total biaya perjalanan yang minimum dan memenuhi kriteria: (i) setiap rute berawal dari depot dan berakhir pada konsumen, (ii) setiap konsumen hanya dikunjungi tepat satu kendaraan dan permintaannya terpenuhi, (iii) total permintaan konsumen yang dikunjungi di setiap rute kurang dari atau sama dengan kapasitas kendaraan yang bertugas di rute tersebut.

2.2 Metode Penyelesaian: Metode penyelesaian OVRP yang digunakan dalam penelitian ini ialah metode heuristik Sariklis-Powell. Metode ini terdiri atas dua fase yaitu fase pembentukan *cluster*yang seimbang lalu diikuti dengan fase penentuan rute. Fase kedua terdiri atas 3 tahap yaitu penentuan *minimum spanning tree* (MST) dengan algoritme Prim, pemodifikasian MST, dan pengubahan solusi takfisibel menjadi solusi fisibel. Tahap terakhir dilakukan dengan menghapus sisi, melabeli simpul, dan mengubah solusi menjadi solusi fisibel.

Fase 1: Pembentukan *Cluster* yang Seimbang

*Cluster*ialah himpunan konsumen. Setiap konsumen dalam *cluster* memiliki permintaan barang.Kapasitas *Cluster* berupa total semua permintaan konsumen dalam *cluster* tersebut. Setiap konsumen yang ada di dalam *cluster* hanya dikunjungi oleh satu kendaraan pengantar barang.*Generalised customer* ialah himpunan terurut yang anggotanya terdiri atas konsumen dan depot dengan depot sebagai anggota pertama. Setiap konsumen dalam *generalised customer*memiliki permintaan barang terhadap depot dan memiliki biaya perjalanan ke konsumen yang lainnya.

Fase pembentukan *cluster*yang seimbang memiliki dua tahap. Tahap pertama ialah membentuk *cluster* berdasarkan permintaan barang dari konsumen. Tahap kedua ialah mengubah komposisi permintaan barang konsumen di dalam *cluster* dengan tujuan meminimumkan total biaya perjalanan dari tiap *cluster*.

Prosedur pembentukan *cluster*ialah sebagai berikut:
1 Semua konsumen dalam *generalised customer* diberi nama berupa angka yang dimulai dari 2 sedangkan depot ialah 1.
2 Dipilih satu konsumen yang memiliki permintaan tidak melebihi sisa kapasitas *cluster* dan memiliki biaya perjalanan yang minimum dari konsumen tersebut ke konsumen lainnya dalam *generalised customer*.
3 Pada tahap awal, kapasitas *cluster* adalah jumlah maksimum barang yang bisa dibawa oleh kendaraan. Jika dalam pemilihan konsumen terdapat beberapa konsumen yang memiliki biaya sama, maka konsumen dengan permintaan maksimum yang dipilih.
4 Jika terdapat beberapa konsumen dengan permintaan maksimum sama, maka konsumen dipilih berdasarkan urutan angka.
5 Konsumen yang telah dipilih dimasukkan ke dalam *cluster*.
6 Prosedur ini diulangi hingga semua konsumen dalam *generalised customer* berhasil ditempatkan ke dalam *cluster*.

Penyeimbangan *cluster* bertujuan meminimumkan total biaya perjalanan tiap *cluster*. Prosedur penyeimbangan *cluster*ialah sebagai berikut:
1 Dipilih *cluster* yang memiliki total permintaan barang yang maksimum.
2 Jikadalam pemilihan*cluster* terdapat beberapa*cluster* yang memiliki total permintaan yang sama, maka dipilih *cluster* yang pertama kali terbentuk.
3 Konsumen akhir dari *cluster* akan dipilih kembali jika terdapat *cluster* lain dengan sisa kapasitas lebih besar atau sama dengan permintaan konsumen yang dipilih.
4 Jika terdapat dua atau lebih *cluster* yang memiliki sisa kapasitas untuk konsumen tersebut, maka konsumen ditempatkan ke *cluster* yang memiliki sisa kapasitas terbesar.

Prosedur ini diulangi sampai konsumen terakhir dari semua *cluster* tidak dapat ditempatkan ke dalam *cluster* lainnya. Jika konsumen akhir dari tiap-tiap *cluster* tidak bisa ditempatkan ke *cluster* lain, maka *cluster-cluster* ini dikatakan seimbang**.**

Fase 2: Penentuan rute

Rute adalah urutan pemesanan barang oleh konsumen yang dimulai dari depot. Konsumen dikunjungi satu kali dan konsumen yang memiliki biaya perjalanan minimum ke depot akan dikunjungi terlebih dahulu. Biaya suatu rute adalah total biaya perjalanan kendaraan ke setiap konsumen pada rute tersebut.

Lokasi konsumen dan depot dapat dinyatakan dalam bentuk graf berarah dengan setiap konsumen dan depot dinyatakan dengan simpul dan bobot pada sisi berarahnya menyatakan biaya perjalanan. Rute yang diinginkan berupa **rantai** yaitu suatu *walk* dengan syarat derajat masuk simpul depot adalah 0, derajat keluar simpul depot adalah 1,

dan derajat masuk dan derajat keluar semua simpul konsumen ialah 1 kecuali konsumen akhir yang memiliki derajat masuk 1 dan derajat keluarnya 0.

Fase penentuan rute memiliki tiga tahap, yaitu:

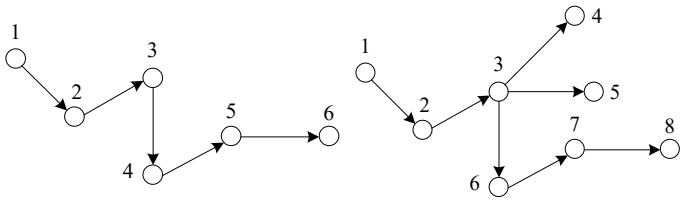
i. tahap pembentukan *minimum spanning tree* (MST),
ii. tahap pemodifikasian MST, dan
iii. tahap pengubahan solusi takfisibel menjadi solusi fisibel.

Jika hasil yang diperoleh dari tahap pertama berupa rantai, maka fase penentuan rute selesai; jika tidak, maka dua tahap selanjutnya dari fase ini harus dilakukan.Berikut ini adalah penjelasan dari setiap tahap.

(i) *Minimum spanning tree* ditentukan dengan algoritme Prim.

(ii) Pada tahap kedua digunakan prosedur penalti simpul untuk memodifikasi solusi yang berupa*minimum spanning tree.*

Misalkan $N_c$adalah banyaknya konsumen pada*clusterc*. Misalkan simpul i adalah simpul konsumen dalam suatu *cluster* sehingga $2 \leq i \leq N_c$ sedangkan simpul 1 adalah simpul depot. Misalkan$d_i$adalah derajat dari simpul i, $2 \leq i \leq N_c$dengand_1 adalah derajat simpul depot.Jika suatu simpul memiliki derajat yang tidak sama dengan syarat suatu rantai, maka simpul tersebut adalah **simpul takfisibel**.



(a) (b)

Gambar 1(a) Contoh rantai, (b) Contoh rantai bersimpul takfisibel.

Pada Gambar 1b), simpul 3 adalah simpul takfisibel karena berderajat-masuk 1 dan berderajat-keluar 3≠1. Ketakfisibelan simpul depot didefinisikan sebagai berikut:

$$\inf(1) = d_1 - 1$$

dan ketakfisibelan dari simpul konsumen adalah:

$$\inf(i) = \begin{cases} d_i - 2 & ; \text{ jika} d_i \geq 2 \\ 1 & ; \text{ jika} d_i = 1 \end{cases}$$

untuk $2 \leq i \leq N_c$. Total ketakfisibelan untuk simpul konsumen dengan $d_i \geq 2$ adalah:

$$a = \sum_{d_i \geq 2} (d_i - 2)$$

Total ketakfisibelan untuk simpul konsumen dengan $d_i = 1$ adalah:

$$b = \sum_{d_i = 1} (\inf(i) - 1)$$

Total ketakfisibelan dari suatu *cluster* adalah:

$$\inf(1) + a + b.$$

Ketakfisibelan simpul akan dihilangkan menggunakan fungsi penalti untuk simpul. Jika simpultersebut adalah depot, maka penalti didefinisikan sebagai berikut:

$$\text{pen}(1) = p(d_1 - 1) \text{dengan} p > 0.$$

Tidak ada aturan khusus dalam menentukan nilai p.

Penalti terhadap simpul konsumen diberikan oleh:

$$\text{pen}(i) = \begin{cases} p(d_i - 2) & ; \text{ jika} d_i \geq 2 \\ -p & ; \text{ jika} d_i = 1 \end{cases}$$

Langkah-langkah metode penalti:
1. Tentukan nilai awal dan nilai maksimum penalti p.
2. Tentukan kenaikan nilai p dan maksimum jumlah iterasinya.
3. Tentukan nilai fungsi penalti di depot dan di setiap simpul.
4. Nilai biaya$\overline{m}_{i,j}$ diperbarui dengan menggunakan:
$$\overline{m}_{i,j} = m_{i,j} + \text{pen}(i) + \text{pen}(j).$$
5. Prosedur akan berhenti ketika diperoleh suatu rantai atau maksimum iterasi dicapai. Jika tidak, maka langkah 4 diulangi.

Prosedur penalti diulangi untuk nilai p yang berbeda sesuai dengan kenaikan nilai p yang telah ditentukan hingga mencapai nilai maksimum p.

(iii) Tahap pengubahan solusi takfisibel menjadifisibel terdiri atas: (a) penghapusan sisi berarah, (b) pelabelan simpul, (c) pembentukan solusi fisibel.

Penghapusan sisi berarah

Ada dua carapenghapusan sisi yaitu:(i) menghapus sisi berarah yang memiliki biaya maksimum, atau (ii) menghapus sisi berarah yang akan menghasilkan solusi dengan biaya maksimum.

Prosedur penghapusan sisi berarah (yang pertama)
1 Periksa/hitung derajat keluar simpul depot.
2 Jika derajat keluar simpul depot lebih dari satu, maka sisi berarahdengan biaya minimum dipertahankan dan lainnya dihapus.
3 Simpul yang memiliki derajat-masuk dan derajat-keluar maksimum dipilih.
4 Jika simpul konsumenmemiliki derajat-masuk dan derajat-keluar sama dengan satu, maka prosedur penghapusan berhenti.
5 Jika tidak,dua sisi berarah berbiaya minimum dipertahankan dan sisanya dihapus.
6 Arah dari sisi berarah yang dipertahankan disesuaikan dengan arah sisi berarah dari setiap*subtree*.
7 Prosedur penghapusan akan berhenti sampai derajat-keluarmaksimum dan derajat-masukmaksimum simpul konsumen sama dengan 1.

Prosedurpenghapusan sisi berarah (yang kedua)
1 Sama seperti Langkah 1 s.d. 3 pada prosedur pertama. Misalkan simpul konsumen yang memiliki derajat-masuk dan derajat-keluar maksimum dipilih ialah simpul *i*.
2 Semua sisi berarah yang terhubung ke konsumen tersebut dihapus, kecuali jika ada sisi berarah yang menghubungkan simpul berderajat maksimum dengan simpul depot maka sisi ini tidak dihapus sampai akhir prosedur.
Simpul yang *adjacent*-ke dan *adjacent*-dari simpul *i*disebut **simpul basis**dan dinotasikan denganb_u dengan $1 \le u \le d_i$ .Sisi berarah (i, b_u) atau (b_u, i) dengan$2 \le i \le N_c$ adalah sisi berarah yang *incident*-ke atau *incident*-dari simpul *i*.
3 Hasil dari penghapusan sisi berarah adalah *subtree*. Jika dilakukan penelusuran rute dari simpul basis pada tiap-tiap *subtree*, maka akan berhenti pada simpul akhir yang untuk selanjutnya disebut **simpul terminal** yang disimbolkan sebagai t_u. Jika *subtree* hanya memiliki simpul tunggal, maka simpul basis sama dengan simpul terminal. Jika terdapat dua simpul akhir dalam satu *subtree*, maka dipilih simpul akhir yang memiliki biaya perjalanan minimum ke simpul sebelumnya.
4 Pilih sisi berarah dengan biaya minimum antar-*subtree*, yaitu dengan:
$$m_{g,f} = \min\{m(T \times T), m([B \times T] \backslash U)\}$$
Badalah himpunan dari simpul-simpul basis dan T adalah himpunan dari simpul-simpul terminal, sedangkan U adalah himpunan sisi berarah $(b_u, t_u), 1 \le u \le d_i$. Sisi berarah yang dipertahankan hanyalah sisi(i, b_u) dan$(i, b_v), 1 \le u \ne v \le d_i$.
5 Sisi berarah (i, b_u) yang memiliki biaya perjalanan minimum dikembalikan seperti semula dan arah dari sisi berarah yang dipertahankan disamakan dengan arah sisi berarah lainnya dari setiap *subtree*.

6. Prosedur penghapusan akan berhenti sampai derajat-keluar maksimum dan derajat-masuk maksimum simpul konsumen sama dengan 1.

Langkah penghapusan menghasilkan $d_i - 1$*subtree* dan simpul *i* memiliki derajat-masuk dan derajat-keluar yang sama dengan satu. Jika simpul *i* terhubung ke simpul depot, maka sisi berarah antara simpul *i* dengan simpul depot dipertahankan dan simpul depot tidak akan menjadi anggota himpunan *B*.

Penggunaan salah satu metode penghapusan sisi berarah akan menguraikan MST takfisibel menjadi simpul-simpul tunggal dan rantai-rantai parsial. Rantai parsial memiliki beberapa bagian yaitu:
(i) simpul depot, yaitu simpul yang hanya terhubung ke satu simpul konsumen,
(ii) simpul tengah, yaitu simpul dengan derajat masuk = derajat keluar = 1,
(iii) simpul akhir/terminal, memiliki derajat masuk 1 dan derajat keluar 0.
Simpul pertama dalam rute suatu rantai disebut akar rantai (**simpul akar**).Ketakefisienan dari simpulkonsumen didefinisikan sebagai berikut:

$$\text{ineff}(i) = 2 - d_i \quad 2 \le i \le N_c$$

Oleh karena itu, ketakefisienan dari simpul tunggal adalah 2, simpul tengah dari rantai parsial adalah 0, dan ketakefisienan simpul depot dari rantai parsial adalah 1. Dalam solusi yang fisibel hanya simpul akhir dari rantai yang mempunyai ketakefisienan 1.

Pelabelan simpul.Simpul yang tidak terhubung dengan simpul (berderajat 0) disebut simpul tunggal. Setiap simpul akan diberi tiga label dengan aturan pelabelan:

$$\text{label\_1} = \begin{cases} \text{nomor simpul}, & \text{jika simpul adalah simpul tunggal} \\ \text{akar suatu rantai}, & \text{jika simpul di dalam rantai parsial} \end{cases}$$

$$\text{label\_2} = \begin{cases} -1, & \text{jika simpulnya simpul tunggal} \\ 0, & \text{jika simpulnya simpul akar dari rantai parsial} \\ \text{nomor simpul sebelumnya}, & \text{jika simpulnya simpul tengah atau simpul terminal dari rantai parsial} \end{cases}$$

$$\text{label\_3} = \begin{cases} 0, & \text{jika simpul adalah simpul tengah dari rantai parsial atau simpul depot} \\ 1, & \text{jika simpul adalah simpul tengah atau simpul terminal dari rantai parsial} \end{cases}$$

Label_1 menunjukkan simpul akar suatu rantai, label_2 menunjukkan simpul sebelumnya, dan label_3 menunjukkan bisa tidaknya simpul tersebut dihubungkan ke simpul tunggal atau rantai parsial lainnya. Simpul depot akan memiliki label (1 0 0).

Pembentukan solusi fisibel Pada bagian ini ketakefisienan dari solusi akan dikurangi. Prosedurnya: (1) memasangkan semua simpul tunggal ke rantai parsial, (2) menghubungkan rantai-rantai parsial.Setiap kali simpul tunggal terhubungkan ke rantai parsial atau dua rantai parsial terhubungkan, ketakefisienan dikurangi 2. Oleh karena itu prosedur ini berhenti pada:

$$\frac{\sum_{2 \le i \le NC} \text{ineff}(i) - 1}{2}$$

pengulangan dan menjamin solusi yang fisibel diperoleh.Berikut ini adalah penjelasan mengenai prosedur pembentukan solusi fisibel.

(1) Pemasangan simpul tunggal ke rantai parsial
Sebuah simpul tunggal bisa dihubungkan ke salah satu simpul terdekatyaitu (a) simpul terminalataukah (b) simpul akar, jika keduanya bukan simpul depot dari rantai parsial. Pada kasus (a), setelah simpul tunggal dihubungkan dengan simpul terminal, maka label diperbarui yaitu: label_1 dari simpul tunggal diubah menjadi simpul akar dari rantai

parsial, label_2 dari simpul tunggal diubah menjadi simpul terminal dari rantai parsial sebelumnya, dan label_3 dari simpul terminal rantai parsial diubah menjadi 0.Setelah dipasangkan, simpul tunggal menjadi simpul terminal dari rantai parsial yang dihasilkan, sementara simpul akar tetap sama.Pada kasus (b) setelah dipasangkan, simpul tunggal menjadi simpul akar dari rantai parsial yang dihasilkan. Label diperbarui secara bersesuaian. Proses diulangi sampai semua simpul tunggal dipasangkan ke rantai parsial dan tidak ada simpul tunggal yang tersisa.

(2) Pemasangan rantai-rantai parsial
Dalam menghubungkan dua rantai parsial, akandibedakan tiga kasus untuk dua simpul yang akan dipasangkan, yaitu: (a) simpul akar dan simpul terminal, (b) keduanya simpul akar, (c) keduanya simpul terminal.Rantai parsial yang memuat depot hanya bisa dihubungkan dengan simpul terminalnya.

Kriteria pemasangan simpul pada kasus (a) ialah:
1 Jikasimpul terminal dari rantai parsial pertama memiliki jarak minimum dengan simpul akar dari rantai parsial keduayang bukan simpul depot, maka kedua simpul dihubungkan.
2 Label diperbarui yaitu: label_1 untuk setiap simpul pada rantai parsial yang kedua diubah menjadi simpul akar dari rantai parsial yang pertama, label_2 dari simpul akar dari rantai parsial yang kedua diubah menjadi simpul akhir dari rantai parsial yang pertama, dan label_3 simpul akar dari rantai parsial yang kedua dan simpul terminal dari rantai parsial yang pertama diubah menjadi 0.

Untuk kasus simpul terminal rantai parsial kedua memiliki jarak minimum dengan simpul akar milik rantai parsial pertama dan simpul akar rantai parsial pertama bukan depot, maka diperlakukan prosedur yang sama.Rantai parsial yang dihasilkan memiliki arah kebalikan.

Pemasangan simpul pada kasus (b):
1 Misalkan simpul akardari rantai parsial pertama memiliki jarak minimum dengan simpul akar dari rantai parsial keduadan keduanya bukan simpul depot, maka kedua simpul dihubungkan.
2 Arah dari rantai parsial yang pertama atau kedua dibalik.
3 Label diperbarui yaitu:

- Jika arah rantai parsial pertama dibalik: label_1 untuk setiap simpul diubah menjadi nomor simpul terminal dari rantai parsial pertama. Label_2 dari simpul terminal rantai parsial pertama diubah menjadi 0. Label_2 dari semua simpul kecuali simpul terminal rantai parsial pertama diubah menjadi nomor simpul sebelumnya. Label_3 dari semua simpul kecuali simpul terminal rantai parsial kedua diubah menjadi 0, sedangkan label_3 simpul terminal dari rantai parsial kedua diubah menjadi 1.
- Jika arah rantai parsial kedua dibalik: label_1 untuk setiap simpul diubah menjadi nomor simpul terminal dari rantai parsial kedua. Label_2 dari simpul terminal rantai parsial kedua diubah menjadi 0. Label_2 dari semua simpul kecuali simpul terminal rantai parsial kedua diubah menjadi nomor simpul sebelumnya. Label_3 dari semua simpul kecuali simpul terminal rantai parsial pertama diubah menjadi 0, sedangkan label_3 untuk simpul terminal dari rantai parsial pertama diubah menjadi 1.

Pemasangan simpul pada kasus (c):
1 Misalkan simpul terminal rantai parsial pertama memiliki jarak minimum dengan simpul terminal rantai parsial kedua.
2 Periksa simpul akar dari kedua rantai parsial tersebut. Jika salah satunya adalah simpul depot, maka simpul akar dari rantai parsial lainnya menjadi simpul terminal dari rantai parsial yang dihasilkan.

3 Jika dari simpul-simpul akar tidak ada yang merupakan simpul depot, maka simpul akar dipilih secara acak, kemudian kedua simpul dihubungkan.
4 Arah dari rantai parsial yang pertama atau kedua dibalik.
5 Label-label diperbarui yaitu:

- jikaarah rantai parsial pertama dibalik: label_1 untuk setiap simpul diubah menjadi nomor simpul akar dari rantai parsial kedua. Label_2 dari simpul akar rantai parsial kedua diubah menjadi 0. Label_2 dari semua simpul kecuali simpul akar rantai parsial kedua diubah menjadi nomor simpul sebelumnya. Label_3 dari semua simpul kecuali simpul akar rantai parsial pertama diubah menjadi 0, sedangkan label_3 untuk simpul akar dari rantai parsial pertama diubah menjadi 1.
- jikaarah rantai parsial kedua dibalik: label_1 untuk setiap simpul diubah menjadi nomor simpul akar dari rantai parsial pertama. Label_2 dari simpul akar rantai parsial pertama diubah menjadi 0. Label_2 dari semua simpul kecuali simpul akar rantai parsial pertama diubah menjadi nomor simpul sebelumnya. Label_3 dari semua simpul kecuali simpul akar rantai parsial kedua diubah menjadi 0, sedangkan label_3 untuk simpul akar dari rantai parsial kedua diubah menjadi 1.

3 IMPLEMENTASI

Misalkan suatu perusahaan memiliki sebuah depot$d$ dan dua kendaraan yangdigunakan untuk mengirimkan barang.Misalkanada 15 konsumen$c_1, c_2, \dots, c_{15}$dengan permintaan barang dan biaya perjalanan antara depot-konsumen dan konsumen-konsumen diberikan pada Tabel 1. Data permintaan konsumen dan biaya perjalanan diperoleh melalui data acak menggunakan *software* Matlab. Diasumsikan bahwa biaya perjalanan dari depot ke konsumen sama dengan biaya perjalanan dari konsumen ke depot. Asumsi ini juga berlaku untuk biaya perjalanan dari konsumen ke konsumen$\{c_1, c_2, c_3, c_4, c_5, \dots, c_{15}\}$. Pada tahap awal akan dibentuk cluster berdasarkan kapasitas kendaraan. Misalkan akan dibentuk beberapa cluster dengan kapasitas 150 unit barang tiap cluster.

Tabel 1 Biaya perjalanan dan permintaan konsumen

	d	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}
d	-	5	11	10	3	2	8	15	6	9	4	12	4	8	11	14
c_1	5	-	15	9	3	3	4	13	4	13	4	14	6	3	4	10
c_2	11	15	-	8	6	13	9	9	14	5	12	12	6	9	2	1
c_3	10	9	8	-	8	12	15	2	9	8	1	6	3	12	5	8
c_4	3	3	6	8	-	3	10	4	10	11	12	7	2	4	14	3
c_5	2	3	13	12	3	-	13	9	15	2	7	2	15	1	12	13
c_6	8	4	9	15	10	13	-	14	2	6	4	13	7	14	3	4
c_7	15	13	9	2	4	9	14	-	3	3	14	9	9	3	13	10
c_8	6	4	14	9	10	15	2	3	-	6	8	7	2	4	2	3
c_9	9	13	5	8	11	2	6	3	6	-	4	7	1	14	15	8
c_{10}	4	4	12	1	12	7	4	14	8	4	-	8	6	14	6	2
c_{11}	12	14	12	6	7	2	13	9	7	7	8	-	12	6	4	7
c_{12}	4	6	6	3	2	15	7	9	2	1	6	12	-	2	2	15
c_{13}	8	3	9	12	4	1	14	3	4	14	14	6	2	-	15	9
c_{14}	11	4	2	5	14	12	3	13	2	15	6	4	2	15	-	1
c_{15}	14	10	1	8	3	13	4	10	3	8	2	7	15	9	1	-
Permintaan		23	10	27	29	24	25	25	18	23	13	24	10	15	10	12

Tahap pembentukan *cluster*yang seimbang

Cluster 1

Konsumen yang memiliki permintaan tidak melebihi sisa kapasitas *cluster* 1 ialah$c_1$sampai c_{15}.Konsumen dengan biaya perjalanan minimum ke konsumen lainnya ialah $c_2, c_3, c_5, c_9, c_{10}, c_{12}, c_{13}, c_{14}, c_{15}$ yang memiliki biaya perjalanan 1.Konsumen dengan permintaan maksimum, yaituc_3 yang memiliki permintaan 27. Dengan cara yang

sama diperoleh secara berurutan $c_5, c_9, c_{13}, c_{10}, c_{15}, c_2, c_{12}, c_{14}$. Jadi *cluster* 1 berupa rangkap-9 (9-*tuple*)$(c_3, c_5, c_9, c_{13}, c_{10}, c_{15}, c_2, c_{12}, c_{14})$ dengan total permintaan adalah 144 dan sisa kapasitas *cluster*1 adalah 6.

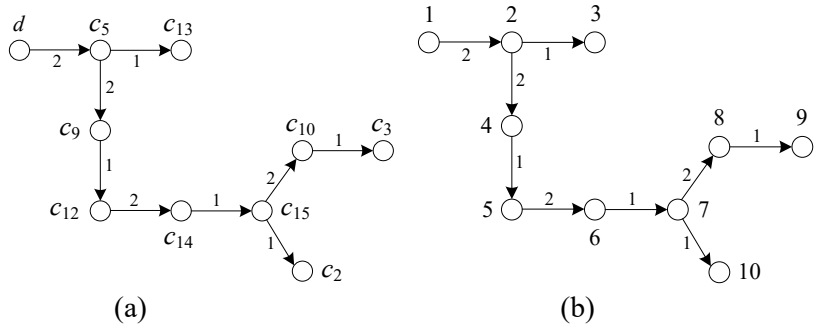
Cluster 2

Sekarang tersisa konsumen $c_1, c_4, c_6, c_7, c_8, c_{11}$.Dengan cara yang sama diperoleh *cluster*2 berupa rangkap-6 (6-*tuple*) $(c_4, c_6, c_7, c_{11}, c_1, c_8)$dengan total permintaan adalah 144 dan sisa kapasitas *cluster* adalah 6.

Karena konsumen dengan urutan terakhir dari tiap *cluster* tidak dapat ditempatkan ke *cluster* lain, maka kedua *cluster*ini sudah seimbang.

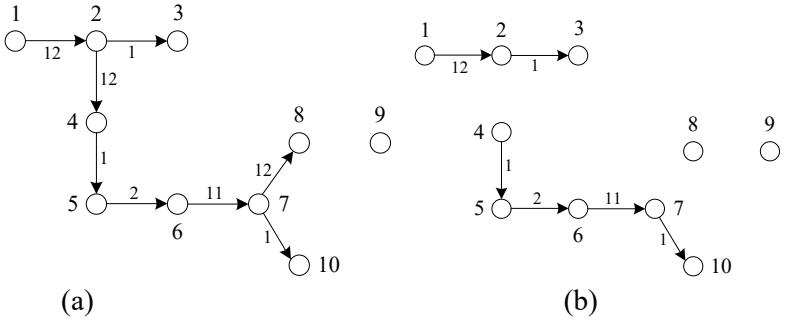
Tahap penentuan rute

Minimum spanning tree (MST) dari *cluster* 1 yang diperoleh (baik dengan verteks asli maupun dengan penamaan baru) ialah:



Gambar 2 (a) MST asli, (b) MST dengan penamaan simpul yang baru.

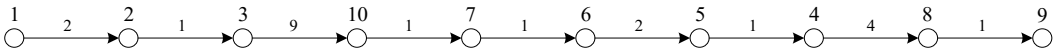
MST yang dihasilkan bukan suatu rantai, maka dengan menentukan ketakfisibelan depot dan semua konsumen diperoleh total nilai ketakfisibelan dari *cluster*1 ialah 2. Misalkan nilai awal penalti $p = 0.1$, kenaikan nilai penalti 0.1, nilai maksimum penalti adalah 1, dan maksimum iterasinya adalah 10.Dengan menggunakan prosedur penalti, maka diperoleh MST dengan bobot yang baru seperti pada Gambar 3a.



Gambar 3 (a) MST dengan bobot yang baru, (b) MST setelah penghapusan.

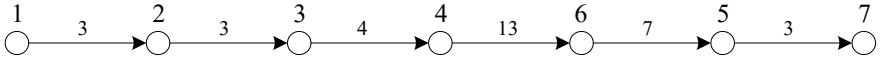
Pengubahan Solusi Takfisibel menjadi Solusi Fisibel

Diketahui $d_2 = 3$ dan $d_7 = 3$adalah derajat maksimum dalam MST.Simpul depot terhubung ke simpul 2,maka sisi berarah $(1,2)$ dipertahankan,sisi berarah $(2,4)$dan sisi berarah $(7,8)$ dihapus karena memiliki biaya maksimum, sehingga diperoleh hasil seperti pada Gambar 3b.Ketakefisienan solusi ini ialah 6. Tahap selanjutnya ialah pelabelan simpul-simpul pada *cluster* 1 berturut-turut dari simpul 1 sampai 10 adalah sebagai berikut: (1 0 0), (1 1 0), (1 2 1), (4 0 0), (4 4 0), (4 5 0), (4 6 0), (8 −1 1), (9 −1 1), (4 7 1). Dengan prosedur pemasangan simpul tunggal atau rantai parsial ke rantai parsial akan diperoleh rantai untuk *cluster* 1 seperti pada Gambar 4.



Gambar 2Solusi fisibel *cluster* 1.

Prosedur yang sama dilakukan untuk menentukan solusi fisibel dari *cluster* 2, sehingga dihasilkan solusi fisibelnya seperti pada Gambar 5.



Gambar 5 Solusi fisibel *cluster*2.

4 KESIMPULAN

Masalah rute kendaraan terbuka sehingga kendaraan tidak diperlukan untuk kembali ke depot merupakan bagian dari *Vehicle Routing Problem* (VRP)yang mengharuskan setiap konsumen dikunjungi sekali dan hanya sekali dengan tepat satu kendaraan dapat diselesaikan dengan metode heuristik.Metode heuristik yang digunakan untuk menyelesaikan masalah ini merupakan suatu algoritme yang terdiri dari beberapa fase. Salah satu algoritme yang digunakan adalah algoritme Prim yang digunakan untuk mencari *minimum spanning tree* (MST). Metode heuristikyang digunakan akan menghasilkan himpunan rute yang berupa rantai.

DAFTAR PUSTAKA

[1]. U. Derigs & K. Reuter, "A simple and efficient tabu search heuristic for solving the open vehicle routing problem", *Journal of the Operational Research Society*, vol. 60, pp. 1658 – 1669, 2009.

[2]. Z.Fu, R. Eglese, &LYO Li. "A new tabu search heuristic for the open vehiclerouting problem".*Journal of the Operational Research Society*56: 267–274.2005

[3]. X-Y Li, P. Tian& S.C.H. Leung, "An ant colony optimizationmetaheuristic hybridized with tabu search for open vehicle routing problems", *Journal of the Operational Research Society*, vol. 60, pp. 1012 –1025, 2009.

[4]. F. Li, B. Golden, & E. Wasil, "The open vehicle routing problem: Algorithms, large-scale test problems, and computational results", *Computers & Operations Research,* vol. 34,pp. 2918 – 2930, 2007.

[5]. C.H. Papadimitriou & K. Steiglitz.*Combinatorial Optimization*: *Algorithms and Complexity*. New Jersey, Prentice-Hall, 1982.

[6]. D.Pisinger&S.Ropke,"A general heuristic for vehicle routing problems",*Computer and Operations Research,* vol.34, pp. 2403 – 2435,2007.

[7]. M.Salari, P.Toth, &A.Tramontani, "An ILP improvement procedure for the Open Vehicle Routing Problem",*Computers and Operations Research*vol. 37, no.12, pp. 2106-2120,2010.doi:10.1016/j.cor.2010.02.010

[8]. D.Sariklis &S.Powell. A heuristic method for the open vehicle routing problem. *Journal of the Operational Research Society*vol. 51, pp. 564-573, 2000.

[9]. C.D. Tarantilis, G. Ioannou, C.T. Kiranoudis & G.P. Prastacos. "Solving the open vehicle routeing problem via a single parameter metaheuristic algorithm". *Journal of the Operational Research Society* vol. 56, pp. 588–596, 2005.