

BAB 2 LANDASAN TEORI

2.1. Tinjauan Pustaka

2.1.1. Penelitian Terdahulu

Penelitian yang dilakukan oleh Amri dkk (2014) mengenai penerapan *Vehicle routing problem* dengan menggunakan metode *Nearest Neighbor* untuk mengoptimalkan rute distribusi. Data yang diambil berupa data permintaan dan jarak antar lokasi. *Distance matrix* yang digunakan merupakan data jarak antar lokasi. Digunakan Metode *Nearest Neighbor* untuk merancang rute berdasarkan jarak terdekat yang selanjutnya, sedangkan efisiensi pendistribusian dievaluasi berdasarkan waktu, total jarak, serta biaya yang diperlukan.

Octora dkk (2014) melakukan penelitian terhadap masalah distribusi di PT. Panca Lestari Primamulya, yaitu terbatasnya jumlah kendaraan yang dimiliki oleh perusahaan dengan jumlah pelanggan yang banyak dan jarak yang berjauhan antar pelanggan. Digunakan metode algoritma *Clarke & Wright savings* dan algoritma *Sequential Insertion* dalam pembuatan rutenya.

Sari (2016) melakukan penelitian mengenai penyelesaian *capacitated vehicle routing problem* di Victoria RO. Pembentukan rute menggunakan *saving matriks*, *sequential insertion* dan *nearest neighbour* dan didapatkan hasil berupa rute distribusi dengan biaya bensin saat ini Rp 78.070,00, untuk metode *Saving Matriks* sebesar Rp 71.410,00, metode *Sequential Insertion* sebesar Rp 67.710,00, dan metode *Nearest Neighbour* sebesar Rp 71.484,00.

Sedangkan Palit dan Sherly (2012) melakukan penelitian mengenai *vehicle routing problem dengan time windows* pada distributor makanan. Digunakan pendekatan *VRPTW* dengan menggunakan algoritma *tabu search* dan didapatkan hasil rute yang lebih baik dengan persentase penghematan biaya transportasi mencapai 21,91%..

2.1.2. Penelitian Sekarang

Penelitian yang dilakukan saat ini menggunakan model *capacitated vehicle routing problem* (CVRP). Tujuan dari digunakan model ini adalah untuk menentukan rute kendaraan yang tiap kendaraannya melayani beberapa pelanggan, dengan batasan kapasitas angkut dan waktu pengiriman masing-

masing pelanggan. Dengan digunakannya CVRP, maka selanjutnya akan didapatkan total jarak tempuh dan jumlah armada yang minimal.

Permasalahan pada penelitian ini adalah adanya ketidakpuasan pelanggan akibat adanya pengiriman produk yang melebihi waktu yang telah ditentukan oleh konsumen. Hal ini disebabkan karena selama ini pada proses pengiriman, pihak Jogja Telor hanya menentukan pembagian area kirim dari pelanggan, sedangkan rute yang diambil ditentukan oleh sopir yang mengantarkan tanpa adanya pertimbangan lebih lanjut mengenai kapasitas angkut maupun urutan pengiriman berdasarkan batasan waktu pengirimannya.

Persamaan penelitian ini dengan penelitian yang terdahulu adalah penelitian yang digunakan sama-sama bertujuan untuk mengurangi waktu dan jarak distribusi. Perbedaan penelitian saat ini dengan penelitian terdahulu adalah lokasi penelitian, objek penelitian, serta metode yang digunakan dari penelitian yang terdahulu. Pada penelitian ini ditambahkan tahap awal menggunakan metode *split delivery vehicle routing problem* karena pada penelitian ini terdapat demand pelanggan yang lebih besar daripada kapasitas armada yang digunakan, sehingga tidak dimungkinkan melakukan pengiriman dengan 1 armada. Selain itu juga kapasitas yang digunakan pada penelitian terdahulu biasanya hanya berupa kapasitas muatan pada armada, tetapi karena tujuan dari penelitian ini adalah untuk mengurangi resiko keterlambatan pengiriman, maka perlu ditambahkan kapasitas berupa waktu pengiriman yang disesuaikan dengan shift pengantarannya. Pada penelitian yang telah dilakukan sebelumnya biasanya tidak terdapat kelompok variasi pengiriman yang menyebabkan dalam 1 kelompok area terdapat beberapa variasi pengiriman yang mungkin terjadi. Pada penelitian yang dilakukan saat ini pengiriman setiap pelanggan tidak selalu berlangsung pada hari yang sama, sehingga dengan dibantu gantt chart akan didapatkan variasi tipe pengiriman yang mungkin terjadi dalam kurun waktu 1 bulan.

Tabel 2.1. Penelitian Terdahulu

Peneliti	Tempat Penelitian	Tujuan Penelitian	Metode	Hasil Penelitian
Octora dkk (2014)	PT. Panca Lestari Primamulya	Menentukan rute distribusi pada masalah transportasi untuk menghemat waktu distribusi.	a. Algoritma <i>savings</i> b. Algoritma <i>Sequential Insertion</i>	Masing-masing algoritma menghasilkan 5 tur dengan 2 alternatif. Total jarak yang dihasilkan menggunakan Algoritma <i>Savings</i> sebesar 262,64 km dengan total waktu sebesar 33,638 jam, sedangkan total jarak yang dihasilkan menggunakan Algoritma <i>Sequential Insertion</i> sebesar 197,88 Km dengan total waktu sebesar 31,49 Jam.

Tabel 2.1. Lanjutan

Peneliti	Tempat Penelitian	Tujuan Penelitian	Metode	Hasil Penelitian
Amri dkk (2014)	MTP Nganjuk Distributor PT. Coca Cola	Mengoptimalkan rute pendistribusian, sehingga bisa mengurangi total jarak, waktu, dan beban biaya yang ditanggung oleh MTP Nganjuk	a. Metode Nearest Neighbor b. Distance Matrix	Rute distribusi memperpendek jarak tempuh sejauh 63,1 km, (13,14 %). Waktu perjalanan dipercepat selama 108,17 menit (3,81 %), sehingga supir dan kernet tidak perlu lembur. Tidak ada biaya lembur, sehingga menekan biaya distribusi senilai Rp 98.377,- (12,08 %).

Tabel 2.1. Lanjutan

Peneliti	Tempat Penelitian	Tujuan Penelitian	Metode	Hasil Penelitian
Palit dan Sherly (2012)	Distributor bahan makanan di Surabaya Timur	Menentukan rute distribusi barang yang dapat meminimumkan biaya transportasi.	a. Metaheuristik (tabu search)	Melalui pendekatan VRPTW dengan algoritma tabu search, maka didapatkan solusi penentuan rute yang lebih baik dengan hasil simulasi usulan rancangan selama 5 hari, dimana didapatkan rata-rata persentase penghematan biaya transportasi sebesar 21,91%.
Sari dkk (2016)	Victoria RO	Melakukan perbaikan rute yang lebih efektif. Keefektifitasan dilihat dari jarak tempuh dengan selisih terbesar dengan total jarak tempuh yang diterakpkn saat ini	a. <i>Saving Matrix</i> b. <i>Sequential insertion</i> c. <i>Nearest neighbour</i>	Penghematan jarak paling besar dari ketiga rute terhadap rute saat ini sebesar 13,27% dan perbedaan biaya bensin sebesar Rp 10.360,00..

2.2. Landasan Teori

2.2.1. Pengertian Distribusi

Kegiatan menyalurkan barang atau jasa dari produsen ke konsumen dapat dikatakan sebagai kegiatan pendistribusian produk.

Menurut Nitisemito (1997), sistem distribusi merupakan cara pengaturan dalam penyaluran barang dan jasa dari produsen ke konsumen. Terdapat 2 jenis sistem distribusi, yaitu :

- a. Langsung : sistem distribusi tanpa menggunakan saluran distribusi (dari produsen langsung ke konsumen akhir)
- b. Tidak langsung : sistem distribusi yang menggunakan saluran distribusi (biasanya melalui distributor)

Saluran distribusi juga dibahas dalam bukunya sebagai lembaga-lembaga distributor atau lembaga-lembaga penyalur yang mempunyai kegiatan untuk menyalurkan atau menyampaikan barang-barang atau jasa-jasa dari produsen ke konsumen.

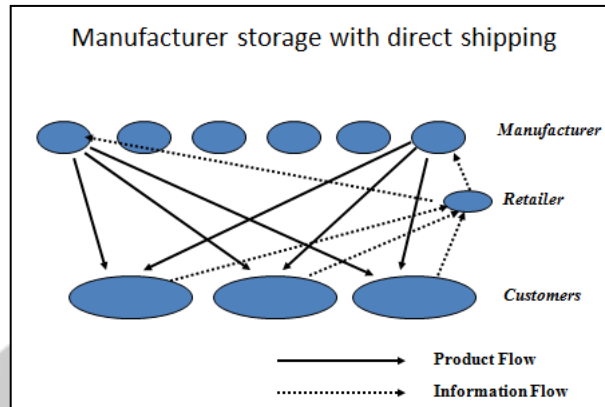
Menurut Kotler (2002), Saluran distribusi adalah sekelompok perusahaan atau perseorangan yang mempunyai hak pemilikan atas suatu produk, atau yang berperan untuk membantu memindahkan hak pemilikan produk atau jasa ketika hendak dipindahkan dari produsen ke konsumen.

2.2.2. Jenis Jaringan / Saluran Distribusi

Menurut Chopra (2007) terdapat 5 jenis jaringan supply chain dengan penjelasan sebagai berikut :

- a. Manufacturer storage with direct shipping

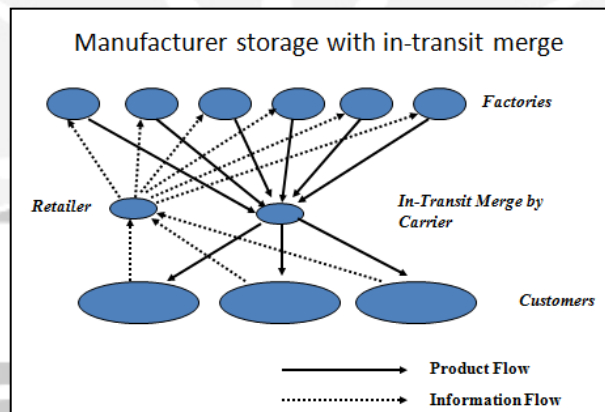
Distribusi produk yang dihasilkan langsung dikirim ke konsumen tanpa melalui retailer. Retailer hanya berfungsi sebagai penyamapi informasi dari konsumen ke pabrik. Inventori diatur oleh perusahaan, sehingga dapat mengurangi biaya inventori. Ilustrasi jenis manufacture storage with direct shipping dapat dilihat pada Gambar 2.1.



Gambar 2.1 Manufacturer storage with direct shipping

b. Manufacturer storage with in-transit merge

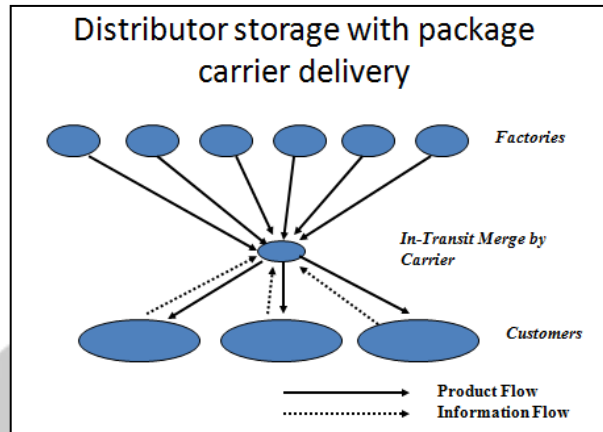
Produk dari perusahaan akan dikumpulkan dulu di suatu tempat bersamaan dengan produk-produk lain, kemudian setelah itu baru dikirimkan kepada konsumen. Ilustrasi mengenai tipe manufacture storage with in-transit merge dapat dilihat pada Gambar 2.2.



Gambar 2.2 Manufacturer storage with in-transit merge

c. Distributor storage with package carrier Delivery

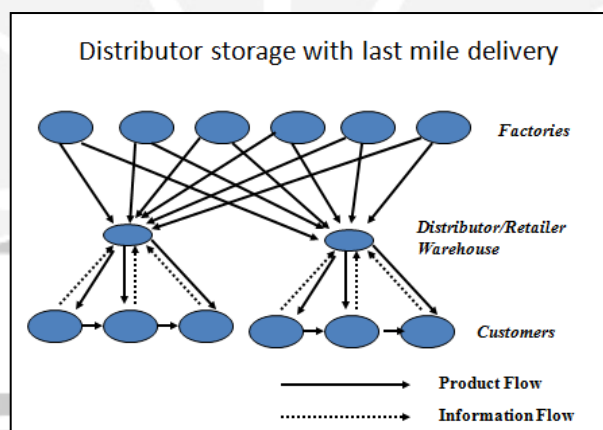
Inventori produk dipegang oleh distributor/retailer. Ilustrasi mengenai distributor storage with package carrier Delivery dapat dilihat pada Gambar 2.3.



Gambar 2.3 Distributor storage with package carrier Delivery

d. Distributor storage with last mile *Delivery*

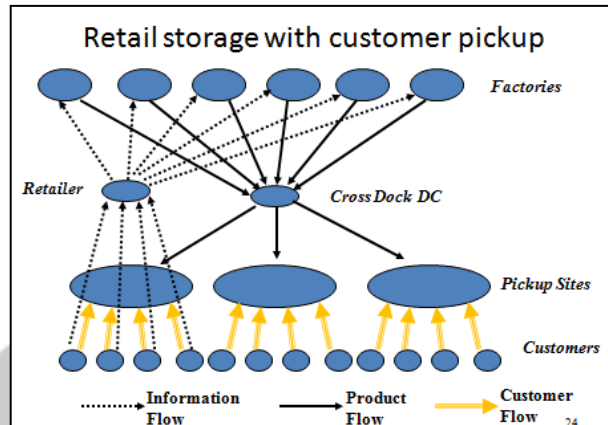
Produk dikirim ke rumah-rumah konsumen menggunakan paket, sehingga membutuhkan inventory yang dekat dengan pelanggan. Ilustrasi mengenai distributor storage with last mile *Delivery* dapat dilihat pada Gambar 2.4



Gambar 2.4. Distributor storage with last mile Delivery

e. Retail storage with customer pickup

Inventory disimpan di dalam toko. Pelanggan melakukan pemesanan/pembelian langsung ke toko. Ilustrasi mengenai retail storage with customer pickup dapat dilihat pada Gambar 2.5



Gambar 2.5. Retail storage with customer pickup

2.2.3. Faktor yang Mempengaruhi Kegiatan Distribusi

Nitisemito (1977) menjelaskan mengenai faktor-faktor yang dapat mempengaruhi proses distribusi. Terdapat 4 macam faktor yang mempengaruhi proses distribusi sebagai berikut :

a. Faktor pasar

Saluran distribusi dipengaruhi oleh pola konsumen. Yang dimaksud dengan pola konsumen adalah jumlah konsumen yang datang, letak geografis konsumen, jumlah pesanan konsumen, serta kebiasaan kosumen dalam melakukan pembelian.

b. Faktor barang

Yang dimaksud dengan faktor barang adalah pertimbangan dari barang yang akan didistribusikan, sehubungan dengan nilai unit, besar dan berat barang, mudah rusaknya barang, standar barang serta pengemasan barang.

c. Faktor perusahaan

Faktor perusahaan berkaitan dengan sumber dana yang mampu disediakan, pengalaman dan kemampuan manajemen, serta pengawasan dan pelayanan untuk kepentingan distribusi yang diberikan.

d. Faktor kebiasaan dalam pembelian

Dalam faktor kebiasaan dalam pembelian ini, perlu diperhatikan mengenai kegunaan perantara, sikap perantara terhadap kebijaksanaan produsen, volume penjualan, serta biaya distribusi.

2.2.4. Permasalahan Distribusi

a. Assignment Model (Model Penugasan)

Menurut Taha (2007), model penugasan dapat diilustrasikan sebagai beberapa pekerja yang memiliki kemampuan yang berbeda-beda untuk suatu pekerjaan. Pekerjaan yang dilakukan oleh pekerja dengan kemampuan yang sesuai membutuhkan biaya yang lebih murah daripada saat pekerjaan tersebut dilakukan oleh pekerja yang kurang berkemampuan. Tujuan dari model penugasan ini adalah untuk meminimumkan biaya penugasan pekerjaan untuk pekerja.

Salah satu metode yang digunakan untuk menyelesaikan masalah penugasan adalah metode Hungarian. Metode Hungarian merupakan metode yang memodifikasi baris dan kolom dalam matriks hingga muncul sebuah komponen nol tunggal dalam setiap baris atau kolom yang dapat dipilih sebagai alokasi penugasan.

Menurut Tamimi dkk (2017), terdapat beberapa syarat yang digunakan dalam metode Hungarian sebagai berikut:

- 1) Jumlah pekerja harus sama dengan jumlah pekerjaan.
- 2) Masing-masing pekerja ditugaskan satu pekerjaan.
- 3) Apabila jumlah pekerja tidak sama dengan jumlah pekerjaan yang ada, maka ditambahkan variabel *dummy* pekerja atau *dummy* pekerjaan dengan variabel bernilai 0.
- 4) Masalah yang akan diselesaikan berupa minimasi kerugian (biaya, waktu, jarak, dan sebagainya) atau maksimasi keuntungan.

Pada model transportasi, pekerja (diasumsikan sebagai $i = 1, 2, \dots, m$) mewakili *sources*, sedangkan pekerjaan (diasumsikan sebagai $j = 1, 2, \dots, n$) mewakili *destination*. Jumlah *supply* (*demand*) di setiap *source* (*destination*) bernilai 1. Biaya transportasi pekerja i ke pekerjaan j adalah c_{ij} . Jadi, pada model ini akan ditugaskan titik tujuan (biasanya berupa konsumen) kepada sumber (dapat berupa gudang maupun kendaraan) untuk meminimalkan biaya distribusi yang ditimbulkan. Model penugasan dapat diekspresikan sebagai berikut (Tamimi dkk, 2017)

Minimasi

$$Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (2.1.)$$

$$x_{ij} \begin{cases} 0, & \text{jika pekerja } i \text{ tidak ditugaskan ke tugas } j \\ 1, & \text{jika pekerja } i \text{ ditugaskan ke tugas } j \end{cases}$$

Batasan

$$\sum_{i=1}^m x_{1j} + x_{2j} + \dots + x_{mj} = 1, 2, \dots, m \quad (2.2.)$$

$$\sum_{j=1}^m x_{i1} + x_{i2} + \dots + x_{in} = 1, 2, \dots, n \quad (2.3.)$$

b. Traveling Salesman Problem (TSP)

Traveling Salesman Problem merupakan permasalahan seorang salesman yang harus berangkat dari depot untuk mengunjungi n pelanggan, dimana setiap pelanggan hanya boleh dikunjungi 1 kali dan salesman harus kembali ke depot awal.

Simchi-Levi dkk (2005), menjelaskan bahwa pada TSP dimisalkan $G = (V, E)$ sebagai graf yang belum terarah dengan $node$ V , $|V| = n$ dan $edges$ E . Dimisalkan d_{ij} sebagai jarak dari $edge$ (i, j) . Tujuan dari TSP adalah untuk menentukan rute yang mengunjungi setiap $node$ (pelanggan) sebanyak 1 kali dan memiliki panjang total minimum.

c. Vehicle routing problem (VRP)

Menurut Simchi-Levi dkk (2005), tipe VRP yang paling umum adalah *Capacitated Vehicle routing problem* (CVRP). CVRP dapat dideskripsikan sebagai 1 set pelanggan yang harus dilayani oleh armada kendaraan dengan kapasitas yang terbatas. Tujuan dari VRP adalah untuk menemukan satu set rute untuk kendaraan dengan jarak total minimal. Setiap rute dimulai di depot, lalu mengunjungi sebagian pelanggan dan kembali ke depot tanpa melanggar batasan kapasitas yang ada. Berikut merupakan asumsi-asumsi yang ada pada VRP:

- 1) Terdapat m -armada yang mengunjungi sejumlah pelanggan dan tiap pelanggan hanya dikunjungi oleh satu armada saja.
- 2) Setiap kendaraan mempunyai kapasitas yang sama.
- 3) Tiap rute harus berasal dari depot dan berakhir di depot yang sama.
- 4) Jumlah permintaan tiap $node$ diketahui.
- 5) Total jumlah *demand* pada suatu rute tidak boleh melebihi kapasitas armada yang digunakan untuk melewati rute tersebut.
- 6) Total jarak tempuh dan jumlah armada yang digunakan diminimalkan.

Menurut Amri dkk (2014), salah satu syarat dalam CVRP adalah bahwa semua permintaan (*demand*) dari pelanggan sudah diketahui di awal.

Berdasarkan ketiga penjelasan di atas mengenai permasalahan distribusi, dapat disimpulkan bahwa pada assignment model akan dilakukan penugasan untuk beberapa sumber ke beberapa destinasi, dimana biasanya 1 destinasi hanya dilayani oleh 1 sumber. Tujuan dari model ini adalah untuk meminimalkan biaya distribusi. Pada penelitian yang dilakukan, model ini tidak dapat dilakukan, hal ini dikarenakan pada Jogja Telor, gudang (titik pusat/sumber) yang dimiliki hanya berjumlah 1, sehingga semua destinasi akan dilayani dari sumber yang sama dan tidak ada pilihan lain untuk melihat penghematan penugasan yang dapat dilakukan.

Pada model TSP akan ditentukan rute terpendek yang harus dilalui oleh *salesman* untuk mengunjungi semua konsumennya. Kunjungan ke konsumen hanya boleh dilakukan 1 kali dan *salesman* harus kembali ke titik awalnya (sumber). Sedangkan pada model VRP, penentuan rute dilakukan untuk armada kendaraan (dengan kapasitas tertentu) yang akan mengunjungi serangkaian konsumen (dengan *demand* bervariasi). Pada model VRP ini total *demand* dalam satu rute tidak boleh melebihi kapasitas armada kendaraan. Selain itu, sama seperti pada model TSP, armada kendaraan harus kembali ke titik awal (sumber) setelah selesai mengunjungi semua konsumen yang ditugaskan. Pada penelitian yang dilakukan di Jogja Telor akan digunakan model CVRP yang dalam penentuan rutenya, selain pertimbangan mengenai jarak tempuh dan kapasitas armada juga akan mempertimbangkan waktu distribusi yang digunakan. Pada penelitian yang dilakukan pada Jogja Telor, waktu distribusi dibatasi dalam bentuk shift dan pendistribusian oleh armada sudah dikelompokkan sesuai shiftnya. Setiap pelanggan dalam kelompok shift yang sama memiliki kapasitas waktu yang sama. Pelanggan Jogja Telor selalu dapat menerima barang yang datang dalam kurun waktu shift yang telah ditentukan oleh perusahaan.

2.2.5. Persoalan Manajemen Logistik

Simchi-Levi dkk (2005) telah menjelaskan bahwa terdapat beberapa persoalan dalam manajemen logistik, berikut adalah beberapa contoh permasalahannya:

a. Manajemen Armada Kendaraan

Sebuah gudang memasok produk ke sekumpulan pengecer menggunakan armada kendaraan yang kapasitasnya terbatas. Seorang operator bertugas

menetapkan muatan ke kendaraan dan menentukan rute kendaraan. Pertama, petugas operator harus menentukan pembagian kelompok dari tiap *retailer* yang dapat dilayani dengan layak oleh kendaraan, yaitu yang sesuai beban kapasitas kendaraan. Kedua, petugas operator harus memutuskan urutan mana yang akan digunakan untuk meminimalkan biaya. Biasanya, salah satu dari dua fungsi biaya yang memungkinkan: yang pertama adalah untuk minimalkan jumlah kendaraan yang digunakan, sedangkan yang kedua fokusnya adalah untuk mengurangi total jarak tempuh. Contoh dari satu-depot *Capacitated Vehicle routing problem* (CVRP), adalah saat satu set pelanggan harus dilayani oleh armada kendaraan dengan kapasitas terbatas. Kendaraan awalnya terletak di depot (dalam hal ini, gudang) dan tujuannya adalah untuk menemukan satu set rute dengan panjang total yang minimal.

b. Rute Truk

Pertimbangkan sebuah truk yang meninggalkan gudang untuk mengirim produk ke sejumlah *retailers*. Urutan di mana retail akan dikunjungi akan menentukan berapa lama waktu pengiriman dan pada jam berapa kendaraan dapat kembali ke gudang. Oleh karena itu, penting bahwa kendaraan mengikuti rute yang efisien. Permasalahan dalam menemukan rute dengan meminimalkan waktu maupun jarak, dari gudang melalui satu set *retailers* adalah contoh dari *Traveling Salesman Problem* (TSP). Rute dari truk adalah subpermasalahan dari contoh manajemen armada yang telah dijelaskan sebelumnya.

c. Masalah Pengemasan

Dalam banyak aplikasi logistik, barang harus dikemas dalam kotak, tempat penyimpanan atau dalam kendaraan dengan ukuran terbatas. Tujuannya adalah mengemas barang sedemikian rupa sehingga jumlah tempat yang digunakan dapat sekecil mungkin. Masalah ini disebut sebagai *Bin-Packing Problem* (BPP). Contohnya, muncul sebagai kasus khusus dalam CVRP ketika tujuannya adalah untuk meminimalkan jumlah kendaraan yang digunakan untuk mengirimkan produk. Bin-packing juga muncul di banyak aplikasi lain, termasuk standar pemotongan panjang kawat atau kertas strip ke ukuran pesanan pelanggan tertentu.

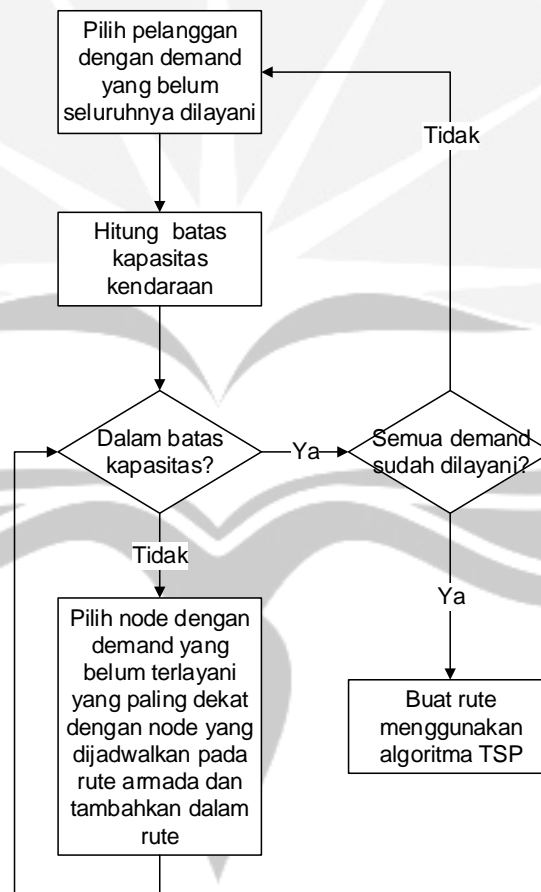
2.2.6. Split-Delivery Vehicle routing problem (SDVRP)

SDVRP merupakan VRP yang mengizinkan armada untuk mengunjungi hanya 1 pelanggan dalam rutennya. Hal ini biasanya dilakukan pada pelanggan yang

memiliki jumlah *demand* lebih besar daripada kapasitas armada yang digunakan, sehingga penghantaran tidak dapat dilakukan hanya satu kali.

Archetti dkk (2014) menyatakan bahwa pada SDVRP, sejumlah armada yang jumlahnya terbatas harus melayani sekumpulan pelanggan. *Demand* dari pelanggan bisa jadi lebih besar daripada kapasitas armada. Pelanggan dimungkinkan dikunjungi oleh beberapa armada dan demand pelanggan tersebut akan dibagi untuk beberapa armada. Jika menguntungkan, bahkan SDVRP dapat digunakan untuk kasus dimana demandnya lebih kecil dari kapasitas armada.

Wilck dan Cavalier (2012) menyatakan bahwa motivasi utama dalam membagi permintaan pelanggan dalam beberapa rute adalah untuk mengurangi jarak perjalanan dan jumlah rute kendaraan. Pada penelitian ini digunakan SDVRP dengan *construction* heuristik sebagai berikut:



Gambar 2.6. Algoritma SDVRP pada Penelitian Wilck dan Cavalier (2012)

Desaulniers (2016) menyebutkan dalam penelitiannya bahwa SDVRP diambil dari VRP dengan melonggarkan batasan pada VRP yang mengharuskan setiap

pelanggan hanya dilayani oleh satu armada. Dengan begitu, setiap pelanggan diijinkan untuk dikunjungi lebih dari 1 kali, sehingga kondisi $d_i \leq Q$ untuk setiap pelanggan $i \in N$ tidak diperlukan lagi.

2.2.7. Pendekatan *Operations Research* untuk VRP

a. *Integer Linear Programming (ILP) Formulation*

Menurut Kilby dan Shaw (2006), ILP dapat memberikan formulasi terhadap VRP dengan lebih formal. Untuk mempermudah pendekatan yang akan digunakan, maka dilakukan beberapa permisalan seperti berikut:

- i. Dimisalkan set atau sekumpulan pelanggan sebagai $C = \{1, \dots, n\}$ dan kendaraan $M = \{1, \dots, m\}$.
- ii. Gunakan 0 untuk menunjukkan depot pada awal rute, dan $n+1$ menunjukkan depot pada akhir rute.
- iii. Definisikan $N = C \cup \{0, n+1\}$.
- iv. Variabel keputusannya adalah x_{ijk} untuk $i, j \in N, k \in M$, dimana x_{ijk} bernilai 1 jika kendaraan k bergerak langsung dari pelanggan i ke pelanggan j , jika tidak maka 0.

- v. Konstanta yang digunakan:

c_{ij} = biaya perjalanan dari i ke $j, i, j \in N$

T_{ij} = waktu tempuh dari i ke $j, i, j \in N$, diikuti waktu pelayanan pelanggan i .

δ_{ij} = jarak dari i ke $j, i, j \in N$

r_i = *demand* pelanggan $i \in N$

Q_k = kapasitas dari kendaraan $k \in M$

a_i = waktu pelayanan paling awal yang dapat dilakukan pada pelanggan $i \in N$

b_i = waktu mulai pelayanan paling akhir pada pelanggan $i \in N$

K = bilangan integer besar

- vi. Untuk memudahkan, maka dimisalkan $r_0 = 0, r_{n+1} = 0, a_0 = 0, a_{n+1} = 0, b_0 = K, b_{n+1} = K$. Untuk masalah dengan batasan waktu, b_{n+1} dapat dijadikan sebagai rute waktu maksimal.

Masalah dalam VRP:

Fungsi Tujuan:

Minimasi

$$Z_{VRP} = \sum_{k \in M} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ijk} \quad (2.4.)$$

Kendala :

$$\sum_{k \in M} \sum_{j \in N} x_{ijk} = 1 \quad \forall i \in C \quad (2.5.)$$

$$\sum_{i \in C} r_i \sum_{j \in N} x_{ijk} \leq Q_k \quad \forall k \in M \quad (2.6.)$$

$$\sum_{j \in N} x_{0jk} = 1 \quad \forall k \in M \quad (2.7.)$$

$$\sum_{i \in C} x_{ihk} - \sum_{j \in N} x_{hjk} = Q \quad \forall h \in C, \forall k \in M \quad (2.8.)$$

$$\sum_{i \in N} x_{i(n+1)k} = 1 \quad \forall k \in M \quad (2.9.)$$

$$\sum_{i \in S} x_{ijk} \leq |S| - 1 \quad \forall S \subseteq C \quad (2.10.)$$

$$x_{ijk} \in \{0,1\} \quad \forall i, j \in N, \forall k \in M \quad (2.11.)$$

Tujuannya adalah untuk meminimalkan total biaya yang digunakan dalam arcs.

b. Construction Methods

Construction Methods digunakan untuk membuat solusi awal. Banyak metode yang sudah dilaporkan didasarkan pada prosedur penyisipan dimana pelanggan dipilih, lalu disisipkan dalam rute sedemikian rupa untuk meminimalkan fungsi biaya. Beberapa kendala dapat diberlakukan pada metode ini dan sangat mudah memeriksa kendala dalam rute (kendala yang hanya berurusan dengan 1 rute).

Tools dalam *Construction Methods*:

i. Pararel versus sekuensial

Prosedur berurutan yang fokus pada 1 rute. Menambahkan pelanggan sampai tidak ada lagi yang cocok. Metode ini membangun serangkaian rute secara bersamaan.

ii. Seed Customer

Merupakan metode paralel yang paling sukses. Menggunakan *seed customer* sebagai pelanggan tunggal pada sejumlah rute yang dipilih. Pelanggan ini memandu rute yang ada.

iii. Penyisipan pesanan

Urutan penyisipan pelanggan sangat penting. Beberapa metode menggunakan sistem skor berdasarkan kedekatan dan ukuran lainnya.

iv. Penyisipan posisi

Fungsi obyektif yang digunakan untuk menentukan posisi yang akan disisipkan digunakan untuk mencoba meminimalkan perjalanan tambahan yang diperlukan untuk mengunjungi pelanggan. Fungsi lainnya digunakan untuk menggunakan waktu dan kendala lain ke dalam akun, mengarahkan proses seleksi untuk memaksimalkan waktu luang atau sumber daya di rute yang dihasilkan.

v. *Saving Method*

Metode yang paling terkenal adalah *saving method* menurut Clarke dan Wright (1964). Metode ini dimulai dengan tiap pelanggan di rute yang mereka miliki. Penghematan diperoleh dengan menggabungkan dua rute $S_{ij} = c_{i0} + c_{0j} - c_{ij}$. Metode ini memilih nilai maksimal S_{ij} untuk rute gabungan yang layak dan beriterasi.

c. *Local Search Methods*

Local search melihat pada lingkungan solusi yang ada untuk mencari solusi yang lebih baik. Lingkungan didefinisikan sebagai satu atau lebih operator yang bergerak. Lingkungan solusi adalah solusi yang dapat dihasilkan dengan menerapkan operator yang bergerak. Ukuran dari lingkungan adalah faktor kunci, semakin besar, maka semakin memungkinkan untuk didapat solusi yang baik. Namun lingkungan yang besar juga membutuhkan biaya yang lebih banyak.

d. *Meta-Heuristics*

Meta-heuristics digunakan untuk menghindari minimal lokal. Hal ini dapat dilakukan dengan mengendalikan penerimaan pergerakan biaya yang terus meningkat, maupun dengan perluasan lingkungan. Beberapa variasi *meta-heuristics* di dalam VRP:

i. *Simulated Annealing*

Penerimaan pergerakan peningkatan biaya dikendalikan oleh parameter sistem yang disebut temperatur sesuai dengan analogi dalam proses *annealing* kristal. Peningkatan biaya δ diterima dengan probabilitas yang sebanding dengan $e^{-\delta/T}$. Pembaruan prosedur meningkatkan T selagi proses terus berjalan, sehingga didapatkan solusi minum lokal yang diharapkan lebih baik.

ii. *Tabu Search*

Tabu Search memperbolehkan adanya pergerakan penambahan biaya, lalu menempatkan pengembalian di "daftar tabu". Hal ini memungkinkan lingkungan dari solusi yang baru dapat dieksplorasi dengan benar.

iii. Genetik

Algoritma genetik menggunakan analogi evolusi Darwin. Dalam bentuk dasar, populasi dari solusi dibuat. Individu "disilangkan" untuk membentuk solusi baru dengan karakteristik dari kedua orang tua.

e. *Exact Methods*

Menurut Kilby dan Shaw (2006), banyak penekanan telah ditempatkan dalam solusi heuristik dan beberapa metode optimal telah dikembangkan, seperti *Column Generation* dan *Lagrangian Relaxation*.

Metode pembangkitan kolom atau *column generation* (CG) biasanya digunakan untuk menyelesaikan beberapa masalah penjadwalan. Masalah program linear yang memiliki banyak variabel akan sulit diselesaikan, variabel tersebut merupakan pola kemungkinan solusi optimal. Metode pembangkitan kolom cukup efisien untuk menyelesaikan masalah program linear (khususnya program linear bilangan bulat yang melibatkan banyak variabel).

Konsep dasar dari *Column Generation* adalah menyelesaikan masalah utama (masalah partisi himpunan), kemudian membentuk masalah utama terbatas atau *Restricted Master Problem* (RMP). Selanjutnya akan dibangkitkan variabel baru apabila solusi yang diperoleh belum optimal. Variabel yang dibangkitkan adalah variabel dengan nilai tereduksi minimum.

2.2.8. *Savings Algorithm* (Metode Konstruktif)

Savings Algorithm yang disarankan oleh Clarke dan Wright (1964) adalah yang paling penting dalam metode konstruktif. Heuristik yang merupakan dasar untuk sejumlah rute kendaraan adalah salah satu heuristik paling awal yang dirancang untuk masalah VRP dan yang paling banyak dikenal. Gagasan tentang *Savings Algorithm* sangat sederhana: pertimbangkan titik depot dan n permintaan. Dimisalkan pada awalnya ditetapkan bahwa digunakan kendaraan terpisah untuk setiap titik permintaan. Jarak total perjalanan kendaraan yang mengunjungi titik permintaan i adalah $2d_i$, di mana d_i adalah jarak dari depot ke titik permintaan i . Oleh karena itu, total jarak yang ditempuh dalam solusi ini adalah $2 \sum_{i=1}^n d_i$.

Jika kedua rute tersebut digabungkan, misalnya i dan j pada satu perjalanan (dengan kendaraan yang sama), total jarak yang ditempuh oleh kendaraan ini menjadi $d_i + d_{ij} + d_j$, dimana d_{ij} adalah jarak antara titik permintaan i dan j . Dengan demikian, penghematan (*savings*) yang diperoleh dari penggabungan titik permintaan i dan j , dilambangkan sebagai s_{ij} , adalah sebagai berikut:

$$s_{ij} = 2d_i + 2d_j - (d_i + d_{ij} + d_j) = d_i + d_j - d_{ij} \quad (2.12.)$$

Semakin besar nilai penghematan s_{ij} , semakin baik untuk menggabungkan titik permintaan i dan j . Berdasarkan ide ini, Clarke dan Wright menyarankan algoritme berikut.

The Savings Algorithm (Algoritma Penghematan)

Langkah 1: Dimulai dengan solusi bahwa setiap pelanggan dikunjungi oleh kendaraan yang berbeda.

Langkah 2: Hitung penghematan $s_{ij} = d_{0i} + d_{j0} - d_{ij} > 0$ untuk semua pelanggan i dan j .

Langkah 3: Golongkan penghematan pada order yang tidak meningkat.

Langkah 4: Temukan arc pertama (i, j) dalam daftar penghematan di mana

- 1) i dan j berada di rute yang berbeda,
- 2) i dan j adalah titik yang pertama atau terakhir dikunjungi pada masing-masing rute, dan
- 3) jumlah permintaan dari rute i dan j tidak lebih dari Q .

Tambahkan arc (i, j) pada solusi saat ini dan hapus arc $(0, i)$ dan $(j, 0)$. Hapus arc (i, j) dari daftar penghematan/*savings*.

Langkah 5: Ulangi langkah 4 hingga tidak ada lagi arc yang memenuhi kondisi.

Kendala tambahan yang mungkin terjadi dapat dimasukkan ke dalam langkah 4. Biasanya pemeriksaan sederhana dapat dilakukan untuk melihat apakah penggabungan rute yang mengandung i dan j melanggar salah satu kendala ini (Simchi-Levi dkk, 2005).

Menurut Rand (2009), nilai saving yang telah dihitung selanjutnya di urutkan sesuai peringkat dan pelanggan ditempatkan pada rute di mana mereka dapat digabungkan (jika tidak, akan dimulai rute baru) sampai batasan atau kendala tercapai. Asumsinya adalah bahwa semakin tinggi nilai *savings* maka semakin baik untuk menghubungkan kedua pelanggan tersebut pada satu rute yang sama. Nilai saving diurutkan dan keputusan dibuat pada setiap poin (apakah gabungan pelanggan dapat ditempatkan ke dalam rute). Pelanggan hanya dapat digabungkan jika tidak ada kendala yang dilanggar oleh rute yang akan dibuat oleh penggabungan pelanggan tersebut. Selanjutnya, jika sudah ada rute yang

terbentuk, maka pelanggan baru hanya dapat digabungkan di awal atau akhir rute.

Algoritma Savings

Pada penelitian yang dilakukan oleh Oktora dkk (2014), langkah pengerjaan menggunakan algoritma Clarke & Wright *Savings* adalah sebagai berikut:

Langkah 1 : Inisialisasi data jarak, jumlah permintaan, waktu pelayanan, kecepatan rata-rata kendaraan, dan kapasitas kendaraan. Lanjut ke langkah 2.

Langkah 2 : Membuat matriks jarak antar depot ke konsumen serta antar konsumen ke konsumen. Lanjut ke langkah 3.

Langkah 3 : Hitung nilai *saving* menggunakan persamaan

$$S(i, j) = d(D, i) + d(D, j) - d(i, j),$$

untuk mengetahui nilai penghematan. Lanjut ke langkah 4.

Langkah 4 : Mengurutkan pasangan pelanggan berdasarkan nilai *saving* matriks dari nilai terbesar hingga terkecil, Lanjut ke langkah 5.

Langkah 5 : Membentuk tur pertama ($t=1$). Lanjut ke langkah 6.

Langkah 6 : Menentukan pelanggan pertama yang ditugaskan pada tur dengan memilih kombinasi pelanggan dengan nilai *saving* terbesar. Lanjut ke langkah 7.

Langkah 7 : Menghitung banyaknya *demand* konsumen yang telah terpilih. Apabila jumlah permintaan masih memenuhi kapasitas kendaraan sebesar 620 kanton maka lanjut ke langkah 8, sedangkan apabila jumlah permintaan melebihi kapasitas kendaraan maka lanjut ke langkah 11.

Langkah 8 : Menghitung total jarak, waktu perjalanan, waktu pelayanan, dan total waktu berdasarkan pelanggan yang telah terpilih. Lanjut ke langkah 9.

Langkah 9 : Apabila total waktu ≤ 8 jam maka pelanggan tersebut terpilih untuk ditugaskan pada tur dan dilanjutkan dengan langkah 10. Apabila total waktu > 8 jam maka lanjut ke langkah 11.

Langkah 10 : Memilih pelanggan selanjutnya yang akan ditugaskan berdasarkan kombinasi pelanggan terakhir yang terpilih dengan nilai *saving* terbesar, kembali ke langkah 7.

Langkah 11 : Hapus pelanggan terakhir yang terpilih, lanjut ke langkah 12.

Langkah 12 : Masukkan pelanggan yang terpilih sebelumnya untuk ditugaskan kedalam tur. Tur (t) telah terbentuk. Apabila masih terdapat pelanggan yang belum terpilih maka, lanjut ke langkah 13. Apabila semua pelanggan telah

ditugaskan maka, proses pengerjaan Algoritma Clarke & Wright Savings telah selesai.

Langkah 13 : Pembentukan tur baru ($t=t+1$), lanjut ke langkah 6.

