

BAB 2 TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

VRP pertama kali diteliti oleh Dantzig dan Ramzer (1959) berkaitan dengan masalah distribusi BBM. Ini menjadi cikal bakal dari kasus VRP yang paling dasar yang disebut *Capacitated VRP (CVRP)*. Toth dan Vigo (2002) merupakan sebuah literatur yang lengkap tentang berbagai macam aplikasi CVRP. Semenjak itu beragam penelitian tentang VRP terus bermunculan. Kompleksitas permasalahan yang ada pada dunia logistik mendorong tren VRP yang semakin berkembang dan meluas ke berbagai varian.

Keterkaitan antara VRP dan jendela waktu melatarbelakangi lahirnya varian baru yang disebut *VRPTW*. Penelitian *VRPTW* pertama kali dilakukan melalui studi kasus oleh Pullen dan Webb (1967). Toth dan Vigo (2002) merupakan sebuah literatur yang lengkap tentang berbagai macam aplikasi *VRPTW*.

Variasi lain dari VRP dapat terjadi pada jenis pengiriman yang digunakan. *SDVRP* membagi pemenuhan *demand* dari suatu konsumen ke dalam beberapa rute pengiriman. Dror dan Trudeau (1989) adalah yang pertama kali meneliti *SDVRP*. Archetti dan Speranza (2008) merupakan sebuah literatur yang lengkap tentang berbagai macam aplikasi *SDVRP*. Varian *SDVRP* dapat dikembangkan lebih lanjut dengan memperhitungkan aspek jendela waktu sehingga membentuk varian baru yaitu *SDVRPTW*. *SDVRPTW* diperkenalkan oleh Frizzell dan Giffin (1992).

Variasi VRP yang lain terdapat pada penggunaan armada angkut dengan kapasitas yang heterogen dan jumlah yang terbatas. Varian itu lebih dikenal dengan *HFVRP*. *HFVRP* masih berkerabat dekat dengan sebuah varian VRP yang disebut *Fleet Size and Mix VRP (FSMVRP)*. Perbedaannya walaupun sama-sama menggunakan armada angkut yang heterogen, *FSMVRP* armada angkutnya tidak terbatas. Golden dkk. (1984) adalah yang pertama kali meneliti *FSMVRP*. Taillard (1999) adalah yang pertama kali meneliti *HFVRP*. Baldacci dkk. (2007) merupakan sebuah literatur yang lengkap tentang berbagai macam aplikasi *FSMVRP* dan *HFVRP*. Sama seperti *SDVRP*, baik *HFVRP* maupun *FSMVRP* dapat dikembangkan lebih lanjut dengan mempertimbangkan aspek

jendela waktu sehingga kasusnya menjadi HFVRPTW dan FSMVRPTW. FSMVRPTW diperkenalkan pertama kali oleh Liu dan Shen (1999).

Ditinjau dari segi fungsi tujuannya, kasus VRP klasik biasanya hanya memasukkan 1 macam fungsi tujuan ke dalam pertimbangan. Fungsi tujuan yang dipertimbangkan juga tidak jauh-jauh dari masalah biaya. Padahal penerapan VRP di dunia nyata bisa jauh lebih kompleks. Didorong oleh hal tersebut maka lahirlah *Multi – objective* VRP. Jozefowicz dkk. (2007) merupakan sebuah literatur yang lengkap tentang berbagai macam aplikasi *Multi – objective* VRP.

Penyelesaian permasalahan VRP dapat dilakukan dengan 3 macam metode yaitu cara heuristik sederhana, metaheuristik maupun cara eksak (Kritikos dan Ioannou, 2013). Kumar dan Panneerselvam (2012) merupakan sebuah survey yang lengkap mengenai penyelesaian VRP dengan cara-cara tersebut. Menurut Lenstra dan Rinnoy Kan (1981) semua permasalahan VRP adalah NP-Hard karena tidak dapat diselesaikan dalam waktu polinomial sehingga cara-cara heuristik lebih banyak disukai dibandingkan cara eksak.

Penelitian yang dilakukan oleh penulis dapat diklasifikasikan sebagai perpaduan antara HFVRPTW dengan SDVRPTW serta mempertimbangkan *Multi – objective* VRP. HFVRP tercermin dari disediakannya armada angkut dengan kapasitas heterogen dan jumlah terbatas oleh perusahaan ekspedisi selaku penyedia jasa pengiriman. SDVRP tercermin dari besarnya *demand* di ler sehingga *demand* harus dipecah ke beberapa rute pengiriman supaya dapat dikirimkan. *Time Window* (TW) tercermin dari turut diperhitungkannya jam operasional di ler dan depot pabrik. Sementara *Multi-objective* VRP tercermin dari turut diperhitungkannya 2 tujuan pengiriman sekaligus yaitu pemerataan pengiriman dan maksimalisasi utilitas angkut.

Selama ini penelitian VRP yang sudah ada cenderung berfokus pada satu varian saja. Sangat jarang penelitian VRP yang memadukan beragam varian VRP ke dalam 1 penelitian yang sama. Padahal dalam penerapannya di dunia nyata, bisa terjadi interaksi antar beberapa varian VRP seperti yang ada di saluran distribusi PT. SIS ke di ler. Akibat kompleksitas masalah yang dihadapi, hampir tidak ada referensi dan literatur yang mempelajari pokok bahasan yang sama dengan penelitian penulis. Penelitian yang cukup relevan adalah Belfiore dan Yoshizaki (2008). Kedua peneliti tersebut melakukan sebuah studi kasus untuk menyelesaikan permasalahan pengiriman yang dialami oleh sebuah grup ritel

besar di Brasil. Kedua peneliti tersebut memandang penelitian mereka sebagai perpaduan antara HFVRPTW dengan SDVRPTW. Meski jumlah truk yang digunakan tak terbatas, namun mereka berdua tetap menganggap kasusnya sebagai perpaduan HFVRPTW dengan SDVRPTW dan bukan perpaduan antara FSMVRPTW dengan SDVRPTW. Mereka berdua juga mengklaim sebagai yang pertama dalam meneliti perpaduan HFVRPTW dengan SDVRPTW. Tujuan yang ingin dicapai hanya 1 yaitu meminimumkan biaya total distribusi.

Belfiore dan Yoshizaki (2008) memproduksi solusi awal dengan menggunakan algoritma *Constructive Heuristic* yang didahului dengan pra proses. Solusi awal kemudian akan diperbaiki menggunakan *Meta Heuristic* yaitu *Scatter Search* (SS). Pra proses diperlukan dalam Belfiore dan Yoshizaki (2008) karena *demand* konsumen jauh lebih besar daripada kapasitas angkut kendaraan. Melalui algoritma pra proses tersebut *demand* masing-masing konsumen akan terus dikirimkan secara eksklusif menggunakan truk penuh dan baru dihentikan ketika sisa *demand* sudah lebih kecil daripada kapasitas angkut truk terbesar. Pengiriman eksklusif artinya dalam sebuah rute pengiriman hanya dikunjungi 1 konsumen saja. *Demand* milik konsumen yang lain tidak mungkin ditampung dalam rute pengiriman yang sama karena kapasitas rute pengiriman sudah terpakai seluruhnya untuk menampung *demand* yang besar.

Sisa *demand* setiap konsumen yang belum terkirimkan oleh algoritma pra proses jumlahnya sudah lebih sedikit daripada kapasitas angkut truk sehingga memungkinkan melakukan konsolidasi *demand* dari beberapa konsumen berbeda ke dalam satu rute pengiriman yang sama. Konsolidasi akan diproses menggunakan algoritma *Constructive Heuristic*. Belfiore dan Yoshizaki (2008) menggunakan 2 macam algoritma *Constructive Heuristic* berbeda untuk menghasilkan diversifikasi solusi awal, yaitu algoritma milik Dullaert dkk. (2002) dan ekstensi algoritma Dullaert dkk. (2002) melalui penambahan ide dasar Ho dan Haughland (2004). Diversifikasi solusi awal diperlukan supaya perbaikan menggunakan SS dapat dilakukan. SS adalah metode evolusioner yang cukup unik. Tidak seperti metode evolusioner lain yang berbasis pada randomisasi, SS menggunakan sekumpulan solusi yang terdiversifikasi sebagai referensi untuk membentuk solusi baru dengan cara kombinasi linear berbobot dari solusi-solusi tersebut.

Dalam algoritma *Constructive Heuristic* yang pertama, algoritma Dullaert dkk. (2002) diterapkan secara murni sehingga tidak mengizinkan adanya pembagian *demand*. *Demand* suatu konsumen harus tertampung seluruhnya dalam sebuah rute pengiriman tunggal bila akan dibebankan. Sedangkan dalam algoritma *Constructive Heuristic* yang kedua, adanya ekstensi ide dasar Ho dan Haughland (2004) pada algoritma Dullaert dkk. (2002) membuat pembagian *demand* diperkenankan. *Demand* suatu konsumen tidak harus tertampung seluruhnya dalam suatu rute pengiriman tunggal. *Demand* dapat dibebankan sebagian hingga kapasitas rute pengiriman penuh sedangkan sisa *demand* yang tidak tertampung dapat dibebankan kepada rute pengiriman yang lain.

Dalam algoritma milik Dullaert dkk. (2002) pembuatan rute dimulai untuk tipe kendaraan dengan kapasitas terkecil, sementara dalam rute yang dibuat akan diawali dari konsumen terjauh dari depot atau yang jendela waktunya paling ketat. Di antara depot dan konsumen terjauh tersebut kemudian akan disisipkan konsumen lain untuk dikunjungi berdasarkan nilai suatu kriteria. Kriteria yang digunakan dalam algoritma Dullaert dkk. (2002) merupakan ekstensi dari kriteria yang digunakan dalam algoritma *Sequential Insertion Heuristic* Solomon (1987). Kriteria yang digunakan dalam algoritma Solomon (1987) sebenarnya juga merupakan sebuah ekstensi dari kriteria yang digunakan dalam algoritma Clarke dan Wright (1964).

Algoritma Clarke dan Wright (1964) merupakan algoritma penyelesaian CVRP sehingga penugasan dan penentuan urutan kunjungan konsumen dalam rute hanya memperhitungkan penghematan jarak ketika sebuah konsumen disisipkan di antara 2 buah titik tujuan di dalam rute. Algoritma Solomon (1987) merupakan algoritma penyelesaian VRPTW sehingga untuk mengakomodasi jendela waktu kriteria tidak hanya memperhitungkan penghematan jarak saja melainkan juga memperhitungkan penghematan waktu. Algoritma Dullaert dkk. (2002) merupakan algoritma penyelesaian FSMVRPTW sehingga untuk mengakomodasi armada angkut yang heterogen kriteria tidak hanya memperhitungkan penghematan jarak dan waktu melainkan juga memperhitungkan penghematan biaya ketika suatu truk tipe tertentu ditugaskan untuk melayani rute pengiriman.

Penelitian yang dilakukan oleh penulis adalah penjadwalan dan penentuan rute pengiriman di saluran distribusi PT. SIS ke diler. Penelitian yang dilakukan oleh

penulis dapat diklasifikasikan sebagai perpaduan HFVRPTW dengan SDVRPTW, mirip seperti penelitian Belfiore dan Yoshizaki (2008). Perbedaannya armada angkut dalam penelitian penulis jumlahnya terbatas sedangkan dalam Belfiore dan Yoshizaki (2008) jumlahnya tak terbatas. Perbedaan yang lain terletak pada jumlah dan bentuk tujuan yang dipertimbangkan. Penelitian Belfiore dan Yoshizaki (2008) hanya mempertimbangkan pemenuhan sebuah tujuan yang dinyatakan dalam bentuk paling umum yaitu biaya. Sementara penelitian penulis mempertimbangkan pemenuhan 2 buah tujuan sekaligus dan tidak dinyatakan dalam bentuk biaya. Penelitian Belfiore dan Yoshizaki (2008) bertujuan untuk meminimasi biaya total distribusi sedangkan penelitian penulis bertujuan untuk meratakan pengiriman sekaligus memaksimalkan utilitas angkut. Pemerataan pengiriman akan dinilai berdasarkan besar-kecilnya standar deviasi pengiriman sedangkan maskimasi utilitas angkut akan dinilai berdasarkan besar-kecilnya persentase kapasitas angkut tidak terpakai terhadap kapasitas angkut aktif yang tersedia.

Penulis mengusulkan 3 tahap algoritma *Constructive Heuristic* sederhana untuk menyelesaikan penelitian. Algoritma tahap 1 dan tahap 2 lebih mengejar pemerataan pengiriman dengan memaksa *demand* konsumen selama sebulan dikirimkan dalam kuantitas yang sama setiap minggunya. Akibatnya akan tetap ada rute pengiriman yang utilitas angkutnya tidak termaksimalkan. Algoritma tahap 3 diusulkan untuk memberi perimbangan ke arah tujuan maksimalisasi utilitas angkut. Algoritma tahap 1 akan diterapkan terlebih dahulu sebelum algoritma tahap 2, sementara algoritma tahap 3 diterapkan setelah algoritma tahap 2.

Algoritma tahap 1 mengambil ide dasar dari algoritma pra proses milik Belfiore dan Yoshizaki (2008) karena adanya kesamaan kasus yaitu *demand* milik konsumen jauh lebih besar daripada kapasitas angkut kendaraan. Perbedaannya, jika dalam pra proses Belfiore dan Yoshizaki (2008) pengiriman eksklusif dengan truk penuh akan dihentikan ketika sisa *demand* sudah lebih kecil daripada kapasitas angkut truk terbesar maka dalam penelitian penulis pengiriman eksklusif dengan truk penuh baru akan dihentikan ketika sisa *demand* sudah lebih kecil daripada kapasitas angkut truk terkecil. Modifikasi terhadap algoritma pra proses milik Belfiore dan Yoshizaki (2008) dilakukan supaya lebih banyak *demand* terkirimkan secara eksklusif hingga hampir habis. Diler akan lebih senang jika sebagian besar *demand* dikirimkan secara eksklusif.

Sisa *demand* yang belum terkirimkan oleh algoritma tahap 1 akan dikonsolidasikan bersama sisa *demand* di ler lain ke dalam sebuah rute pengiriman yang sama. Konsolidasi dilakukan melalui algoritma tahap 2.

Algoritma tahap 2 akan mengkonsolidasikan sisa *demand* beberapa di ler berbeda ke dalam rute pengiriman yang sama berdasarkan kesamaan letak geografis dan perusahaan ekspedisi yang digunakan. Dalam Belfiore dan Yoshizaki (2008) konsolidasi tidak dibatasi oleh kesamaan letak geografis maupun kesamaan perusahaan ekspedisi yang digunakan karena seluruh armada angkut merupakan milik perusahaan. Sedangkan dalam penelitian penulis perusahaan menyewa jasa pengiriman yang disediakan beberapa perusahaan ekspedisi. Setiap perusahaan ekspedisi melayani beberapa di ler sekaligus yang umumnya terletak di lokasi geografis yang sama sehingga konsolidasi harus memperhatikan kesamaan letak geografis dan perusahaan ekspedisi yang digunakan.

Dalam algoritma *Constructive Heuristic* Belfiore dan Yoshizaki (2008) penugasan dan penentuan urutan kunjungan konsumen dalam rute pengiriman yang dikonsolidasikan menggunakan algoritma Dullaert dkk. (2002) sementara algoritma Ho dan Haughland (2004) hanya digunakan ide dasarnya saja untuk mengakomodasi pembagian *demand*. Sedangkan dalam algoritma tahap 2 penulis, algoritma Ho dan Haughland (2004) selain digunakan untuk mengakomodasi pembagian *demand* juga digunakan untuk menugaskan dan menentukan urutan kunjungan di ler dalam rute pengiriman yang dikonsolidasikan. Dalam Ho dan Haughland (2004) penjumlahan yang minimum dari waktu tempuh dan waktu menunggu dari suatu konsumen ke konsumen lain digunakan sebagai kriteria penugasan dan penentuan urutan kunjungan konsumen di dalam rute pengiriman.

Algoritma tahap 3 penulis berbasis pada realokasi *demand* dan eliminasi rute-rute yang telah dihasilkan oleh algoritma tahap 1 dan tahap 2, khususnya yang utilitas angkutnya tidak termaksimalkan. *Demand* milik suatu rute akan disumbangkan ke rute lain dengan harapan utilitas angkut dari rute lain tersebut dapat dimaksimalkan sementara rute pendonor dapat dieliminasi karena *demand*-nya habis disumbangkan ke rute yang lain. Tabel 2.1 menunjukkan perbandingan penelitian milik penulis dengan penelitian milik Belfiore dan Yoshizaki (2008).

Tabel 2.1. Perbandingan Penelitian Penulis Terhadap Penelitian Lain

Penelitian	Tempat Penelitian	Tujuan Penelitian	Tipe Armada Angkut	Constructive Heuristic			Meta Heuristic
				Algoritma Tahap 1	Algoritma Tahap 2	Algoritma Tahap 3	
Belfiore dan Yoshizaki (2008)	Grup ritel besar di Brazil	Tujuan penelitian hanya 1 dan dinyatakan dalam bentuk biaya yaitu minimasi total biaya distribusi	Heterogen tetapi jumlahnya tak terbatas	Merupakan pra proses. Pra proses akan mengirimkan <i>demand</i> konsumen secara eksklusif dengan truk penuh hingga sisa <i>demand</i> lebih kecil daripada kapasitas angkut truk terbesar.	Menggunakan 2 macam algoritma <i>Constructive Heuristic</i> untuk menghasilkan diversifikasi solusi. Algoritma pertama merupakan adaptasi algoritma Dullaert dkk. (2002). Algoritma kedua merupakan ekstensi dari algoritma Dullaert dkk. (2002) dengan penambahan ide dasar algoritma Ho dan Haughland (2004).	-	<i>Scatter Search</i>
Dewanto (2014)	Saluran distribusi PT. SIS - diler	Tujuan ada 2 dan tidak dinyatakan dalam bentuk biaya yaitu pemerataan pengiriman dan maksimalisasi utilitas angkut	Heterogen dan jumlahnya terbatas	Modifikasi dari pra proses Belfiore dan Yoshizaki (2008). Bedanya pengiriman eksklusif dengan truk penuh akan dilakukan hingga sisa <i>demand</i> lebih kecil daripada kapasitas angkut truk terkecil.	Algoritma Ho dan Haughland (2004) dipergunakan untuk menentukan urutan kunjungan konsumen dan untuk mengakomodasi pembagian <i>demand</i> .	Realokasi <i>demand</i> dan eliminasi rute.	-

2.2. Desain Jaringan Transportasi

Menurut Chopra dan Meindl (2007) terdapat 3 macam desain jaringan transportasi yaitu *direct shipment*, *warehousing* dan *tailored network*.

2.2.1. Direct Shipment (Pengiriman Langsung)

Menurut Chopra dan Meindl (2007) *Direct Shipment* adalah pengiriman secara langsung dari depot pabrik ke konsumen tanpa melalui gudang perantara. Keuntungan utama dari *Direct Shipment* ini adalah kesederhanaan dan kemudahan operasi. *Direct Shipment* sangat cocok untuk pengiriman ke konsumen yang memiliki *demand* besar, paling tidak sebesar kapasitas angkut kendaraan. Jika terpaksa melayani konsumen dengan *demand* kecil maka *Direct Shipment* dapat dikombinasikan dengan strategi *Milk Run*. Strategi *Milk Run* adalah mengkonsolidasikan *demand* yang kecil dari beberapa konsumen ke dalam 1 kendaraan angkut yang sama, sehingga kapasitas angkut tidak terpakai dapat dikurangi dan biaya transportasi per unit dapat diturunkan.

2.2.2. Warehousing (Gudang Perantara)

Menurut Chopra dan Meindl (2007) *Warehousing* diterapkan dengan cara membagi pengiriman ke dalam beberapa wilayah pengiriman kemudian pada setiap wilayah didirikan *warehouse* (gudang perantara). Pengiriman akan dilakukan dari depot pabrik ke gudang perantara, baru kemudian dari gudang perantara barang dikirim ke konsumen. Gudang perantara dapat menjadi kepanjangan tangan perusahaan untuk lebih dekat dengan konsumennya, khususnya untuk konsumen yang jaraknya cukup jauh dari depot pabrik. Jika *replenishment lot* dari konsumen yang dilayani kecil maka gudang perantara dapat berfungsi sebagai penyimpan persediaan selama jangka waktu tertentu. Sedangkan jika *replenishment lot* dari konsumen yang dilayani besar maka gudang perantara tidak perlu menyimpan persediaan dalam jangka waktu yang lama melainkan hanya berfungsi sebagai fasilitas *Cross-Docking*. Dalam *Cross-Docking* pengiriman akan datang dalam jumlah besar dari depot pabrik kemudian sesegera mungkin akan dipecah menjadi kuantitas yang lebih kecil untuk dikirimkan ke konsumen. Seperti juga *Direct Shipment*, *Warehousing* dapat dikombinasikan dengan strategi *Milk Run*.

2.2.3. Tailored Network (Jaringan Gabungan)

Menurut Chopra dan Meindl (2007) *Tailored Network* adalah suatu desain jaringan transportasi pada perusahaan yang menganut *Direct Shipment* dan *Warehousing* secara bersamaan sehingga mampu menyesuaikan dengan situasi dan kondisi yang dihadapi. Dalam *Tailored Network* konsumen dengan *demand* besar akan diprioritaskan untuk dilayani secara *Direct Shipment* sedangkan konsumen dengan *demand* kecil akan dilayani secara *Warehousing*. Dibandingkan dengan *Direct Shipment* dan *Warehousing*, *Tailored Network* memerlukan biaya yang lebih besar terkait pengadaan infrastruktur maupun upaya koordinasi yang harus dilakukan.

2.3. Capacitated Vehicle Routing Problem (CVRP)

CVRP merupakan pengembangan dari *Traveling Salesman Problem* (TSP). TSP merupakan permasalahan penentuan rute kunjungan terpendek yang biasa dialami oleh tenaga penjual. Rute kunjungan terpendek menjadi penting ketika biaya minimum ingin dicapai. Pada TSP tidak dikenal kendala kapasitas sehingga hanya diperlukan sebuah rute pengiriman untuk melayani *demand* seluruh konsumen. Sedangkan CVRP memperkenalkan kendala kapasitas sehingga mungkin dibutuhkan lebih dari sebuah rute pengiriman untuk memenuhi *demand* seluruh konsumen.

Menurut Toth dan Vigo (2002) beberapa asumsi dasar yang dibangun untuk CVRP :

1. Setiap rute berbentuk sirkuit yaitu diawali dan diakhiri di tempat yang sama (depot).
2. Setiap konsumen hanya boleh dikunjungi sebanyak 1 kali saja.
3. Total *demand* yang diangkut dalam sebuah rute tidak melebihi kapasitas angkut rute tersebut.
4. Kapasitas kendaraan yang digunakan homogen (seragam).

2.4. Vehicle Routing Problem with Time Window (VRPTW)

CVRP dapat dikembangkan dengan menambahkan kendala jendela waktu sehingga kasusnya menjadi VRPTW. Asumsi standar pada CVRP tetap dipertahankan pada VRPTW. Beberapa asumsi yang berkaitan dengan jendela waktu ditambahkan seperti memperhitungkan jam operasional setiap konsumen (Toth dan Vigo, 2002).

Pelayanan pada konsumen hanya bisa dilakukan dalam rentang jam operasionalnya saja. Kendaraan angkut diperbolehkan tiba di konsumen sebelum jam operasional, tetapi tetap harus menunggu hingga jam operasional dimulai. Pelayanan di setiap konsumen dilakukan selama jangka waktu tertentu, jangka waktu tersebut disebut sebagai waktu pelayanan.

Waktu pelayanan pada umumnya diasumsikan konstan, tidak berbeda antara suatu konsumen dengan konsumen yang lain. Mungkin pada penerapannya di lapangan hal ini terlihat kurang realistis. Asumsi tersebut dapat diterima ketika waktu pelayanannya bersifat *capacity independent*, tidak terlalu bergantung pada banyak – sedikitnya *demand* yang diantarkan ke konsumen (Salani dan Vacca, 2011).

2.5. Split Delivery Vehicle Routing Problem (SDVRP)

SDVRP merupakan relaksasi dari CVRP. SDVRP tidak dipelajari sefamiliar seperti CVRP (Jin dkk., 2006). Dibandingkan dengan CVRP, SDVRP lebih sulit untuk diselesaikan karena mempunyai *feasible space* yang lebih luas (Hubert dan Cavalier, 2012).

Semua asumsi dasar yang berlaku pada CVRP tetap valid untuk SDVRP, kecuali asumsi yang berkaitan dengan jumlah maksimal kunjungan untuk setiap konsumen. Pada SDVRP diperbolehkan melakukan kunjungan lebih dari sekali untuk konsumen yang sama (Hubert dan Cavalier, 2012). Dengan kata lain *demand* suatu konsumen dapat dibagikan ke beberapa rute yang berbeda untuk dipenuhi secara bersamaan.

Adanya pembagian *demand* pada SDVRP membuat konsumen yang mempunyai *demand* lebih besar daripada kapasitas angkut dapat terlayani (Ho dan Haughland, 2004). SDVRP cocok diterapkan jika sebagian besar konsumen mempunyai *demand* sebesar setengah dari kapasitas kendaraan angkut atau jika varian *demandnya* rendah (Archetti dan Speranza, 2008). Seperti CVRP, SDVRP juga dapat dikembangkan dengan memperhitungkan jendela waktu sehingga kasusnya menjadi SDVRPTW.

2.6. Heterogeneous Fleet Vehicle Routing Problem (HFVRP)

HFVRP merupakan pengembangan lanjut dari CVRP. CVRP menggunakan armada angkut yang homogen kapasitasnya, sementara HFVRP menggunakan

armada yang heterogen. HFVRP masih berkerabat dekat dengan FSMVRP, bedanya HFVRP menggunakan armada yang terbatas sementara FSMVRP armadanya tak terbatas. HFVRP lebih jarang dipelajari dan lebih sulit diselesaikan daripada FSMVRP (Jiang dkk., 2013). Fakta bahwa jumlah kendaraan angkut terbatas dan tetap, maka tidak ada jaminan dapat diperoleh solusi yang *feasible* (Brandão, 2010). Jika terjadi di dunia nyata maka perusahaan harus cepat-cepat mencari armada tambahan. Seperti juga SDVRP, baik HFVRP maupun FSMVRP dapat dikembangkan lebih lanjut dengan mempertimbangkan jendela waktu sehingga kasusnya menjadi HFVRPTW dan FSMVRPTW.

Secara praktis kebutuhan akan kapasitas angkut kendaraan yang berbeda sangat ditentukan oleh karakteristik konsumen yang bersangkutan. Kendaraan dengan kapasitas angkut yang besar lebih cocok digunakan untuk melayani konsumen yang memiliki *demand* besar, sementara kendaraan dengan kapasitas angkut kecil lebih cocok untuk melayani konsumen dengan *demand* kecil atau yang mempunyai kendala berupa keterbatasan akses (Brandão, 2010).

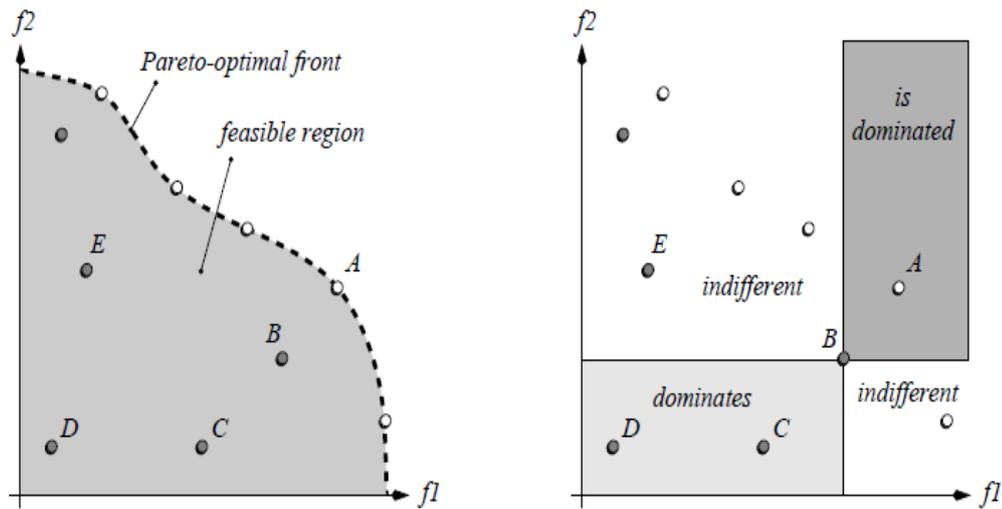
Pengurutan untuk memilih konsumen dan kendaraan dalam suatu algoritma penyelesaian HFVRP mempunyai dampak yang signifikan terhadap kualitas solusi yang dihasilkan (Jiang dkk., 2013).

2.7. Multi – objective VRP dan Multi - objective Optimization Problem (MOP)

Kasus VRP biasanya hanya mempertimbangkan pemenuhan 1 macam tujuan dan biasanya hanya berfokus pada masalah biaya. *Multi – objective* VRP lahir supaya kasus VRP dapat mempertimbangkan pemenuhan lebih dari 1 macam tujuan. Beberapa tujuan ada yang tidak berkaitan atau tidak dinyatakan secara langsung dalam bentuk biaya seperti penyeimbangan beban kerja, utilisasi sumber daya, keleluasaan akses, *makespan* dan lain-lain (Jozefowicz dkk., 2007). Penyelesaian *Multi-objective* VRP banyak melibatkan konsep keilmuan MOP.

Daripada sebuah solusi tunggal yang optimum, pada MOP akan diperoleh sekumpulan solusi dengan tingkat *trade off* yang berbeda antar fungsi tujuan yang bersangkutan (Zitzler, 1999). Kumpulan solusi tersebut disebut sebagai kumpulan solusi *pareto optimal*. *Pareto optimal* adalah suatu kondisi di mana kita tidak mungkin membuat suatu individual menjadi lebih baik tanpa membuat individual lain menjadi lebih buruk (Pareto, 1971). Konsep *pareto optimal* tersebut

menjelaskan adanya *trade off* di antara fungsi tujuan. Gambar 2.1 menunjukkan contoh MOP dengan 2 buah fungsi tujuan yang ingin dimaksimalkan.



Gambar 2.1. Contoh MOP

Pada Gambar 2.1 f_1 mewakili tujuan pertama sedangkan f_2 mewakili tujuan kedua. Terdapat titik-titik yang mewakili *decision vector* (vektor keputusan). Setiap vektor keputusan berada dalam *feasible region* (daerah layak). Titik-titik yang berwarna putih bernilai istimewa karena merupakan *pareto optimal*. Keunikan MOP dapat dicermati dengan cara menjadikan vektor keputusan B sebagai referensi pembandingan terhadap vektor keputusan yang lain. Jika dibandingkan terhadap C dan D , vektor keputusan B mampu mendominasi karena menghasilkan pemenuhan terhadap f_1 dan f_2 yang lebih baik. Sehingga vektor keputusan B dapat dikatakan lebih baik daripada C dan D . Sebaliknya vektor keputusan B didominasi oleh A , yang merupakan *pareto optimal* dan menghasilkan pemenuhan terhadap f_1 dan f_2 yang lebih baik. Sehingga vektor keputusan B dapat dikatakan lebih buruk daripada A . Jika dibandingkan terhadap E , vektor keputusan B tidak dapat dikatakan mendominasi ataupun didominasi melainkan bersifat *indifferent*. Vektor keputusan B menghasilkan pemenuhan terhadap f_1 yang lebih baik dibandingkan E , tetapi pemenuhan terhadap f_2 lebih buruk. Sehingga di antara B dan E tidak dapat dikatakan mana yang lebih baik dan mana yang lebih buruk kecuali preferensi dari f_1 dan f_2 sudah diketahui sebelumnya.

2.8. Haversine Formula

Haversine Formula digunakan untuk menghitung jarak *Orthodromic*. Jarak *Orthodromic* adalah jarak terpendek antara dua titik di permukaan bumi. Jarak *Orthodromic* bukan merupakan jarak garis lurus yang menghubungkan dua titik seperti pada jarak *Euclidean*. Jarak *Orthodromic* ikut memperhitungkan jari – jari kelengkungan bumi. Menurut Smart (1960) *Haversine Formula* untuk menghitung jarak *Orthodromic* dapat dinyatakan sebagai berikut:

$$d = 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right) \quad (2.1)$$

Dengan r sebagai jari – jari kelengkungan bumi yang bernilai 6.371 km, ϕ sebagai *latitude* dan λ sebagai *longitude*. *Latitude* adalah kordinat yang menyatakan posisi suatu tempat berada di utara atau selatan ekuator. Sementara *longitude* adalah koordinat yang menyatakan posisi suatu tempat berada di timur atau barat garis *Greenwich*. Ketika *latitude* dinyatakan dalam derajat, bila angkanya positif artinya tempat itu ada di belahan bumi utara (bujur utara) sedangkan bila angkanya negatif artinya tempat itu ada di belahan bumi selatan (bujur selatan). Ketika *longitude* dinyatakan dalam derajat, bila angkanya positif artinya tempat itu ada di belahan bumi timur (bujur timur) sedangkan bila angkanya negatif artinya tempat itu ada di belahan bumi barat (bujur barat).

2.9. Varian dan Standar Deviasi

Varian merupakan salah satu ukuran persebaran data. Varian merupakan ukuran persebaran data yang paling sering digunakan dan salah satu yang paling efektif dalam sejarah statistika (Mulholland dan Jones, 1968). Standar deviasi merupakan akar kuadrat dari varian. Standar deviasi untuk suatu populasi dapat dinyatakan sebagai berikut:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{X})^2} \quad (2.2)$$

Dengan N adalah banyaknya anggota populasi, x_i adalah nilai dari suatu anggota populasi dan \bar{X} nilai rata-rata populasi. Standar deviasi dari beberapa populasi yang tidak saling tumpang tindih dapat kita gabungkan menjadi satu. Rumus standar deviasi gabungan menurut Gupta dan Kapoor (2002) adalah:

$$\sigma' = \sqrt{\frac{1}{N_1 + N_2 + \dots + N_k} N_1(\sigma_1^2 + d_1^2) + N_2(\sigma_2^2 + d_2^2) + \dots + N_k(\sigma_k^2 + d_k^2)} \quad (2.3)$$

$$d_i = \bar{X}_i - \bar{X}' \quad (2.4)$$

$$\bar{X}' = \frac{N_1\bar{X}_1 + N_2\bar{X}_2 + \dots + N_k\bar{X}_k}{N_1 + N_2 + \dots + N_k} \quad (2.5)$$

2.10. Fungsi dan Fitur *Microsoft Excel* yang Dipakai

Dalam *Microsoft Excel* terdapat berbagai macam fungsi dan fitur yang dapat membantu perhitungan dan pengolahan data. Berikut ini adalah beberapa fungsi dan fitur yang digunakan dalam penelitian ini:

1. Fungsi *IF*

Fungsi *IF* digunakan sebagai fungsi logika untuk menyatakan suatu kondisi benar atau salah. Sintaks penulisan fungsi *IF* adalah sebagai berikut:

$$=IF(\text{logical_test}; \text{value_if_true}; \text{value_if_false})$$

Logical_test merupakan pernyataan yang akan diuji kebenarannya. *Value_if_true* merupakan nilai yang ingin ditampilkan jika pernyataan bernilai benar sementara *value_if_false* merupakan nilai yang ingin ditampilkan jika pernyataan bernilai salah.

2. Fungsi *AND*

Fungsi *AND* digunakan untuk menentukan apakah seluruh pernyataan bernilai benar atau salah. Sintaks penulisan fungsi *AND* adalah sebagai berikut:

$$=AND(\text{logical1}; [\text{logical2}]; \dots)$$

Fungsi *AND* dapat disarangkan ke dalam Fungsi *IF* pada posisi *logical_test*.

3. *INDEX*

Fungsi *INDEX* digunakan untuk mencari dan mengembalikan suatu nilai milik *cell* yang spesifik dari sekumpulan *cell* yang ada pada suatu *range*. Sintaks penulisan fungsi *INDEX* adalah sebagai berikut :

$$=INDEX(\text{array}; \text{row_num}; [\text{column_num}])$$

Array merupakan *range* tempat pencarian nilai yang diinginkan, *row_num* merupakan posisi baris dari *cell* yang ingin dicari nilainya relatif terhadap *cell* yang terletak pada bagian paling kiri atas *range* sedangkan *column_num* merupakan posisi kolom dari *cell* yang ingin dicari nilainya relatif terhadap *cell* yang terletak pada bagian paling kiri atas *range*.

4. Fungsi *MATCH*

Fungsi *MATCH* digunakan untuk mencari posisi relatif dari suatu *cell* spesifik dalam suatu *range*. Sintaks penulisan fungsi *MATCH* adalah sebagai berikut :

$$=MATCH(lookup_value;lookup_array;[match_type])$$

Lookup_value merupakan nilai referensi sebagai pembanding, *lookup_array* adalah *range* tempat pencarian dilakukan, sedangkan *match_type* adalah tipe kecocokan. Bila *match type* diisikan angka 0 maka pencarian akan dilakukan pencarian untuk menemukan posisi relatif dari *cell* yang memiliki nilai sama persis seperti *lookup_value*. Jika diisikan angka -1 maka akan dilakukan pencarian untuk menemukan posisi relatif dari *cell* yang memiliki nilai lebih besar dari *lookup_value*. Sedangkan jika diisikan angka 1 maka akan mencari posisi relatif dari *cell* yang memiliki nilai lebih kecil dari *lookup_value*. Fungsi *MATCH* dapat disarangkan ke dalam Fungsi *INDEX* sehingga akan berfungsi seperti *VLOOKUP*.

5. Fungsi *OFFSET*

Mengembalikan referensi ke suatu *range* berdasarkan posisi baris dan kolom dari suatu referensi. Berikut ini adalah sintaks penulisan fungsi *OFFSET* :

$$=OFFSET(reference;rows;cols;[height];[width])$$

Reference adalah *cell* atau *range* yang dijadikan referensi awal, *rows* adalah jarak baris relatif terhadap referensi, *cols* adalah jarak kolom relatif terhadap referensi; *height* adalah tinggi *range* yang ingin dihasilkan dan *width* adalah lebar *range* yang ingin dihasilkan. Baik *rows* maupun *cols* dapat diisi angka positif maupun negatif. Untuk *rows* jika negatif artinya dilakukan *offset* ke atas, sementara jika positif artinya *offset* dilakukan ke bawah. Untuk *cols* jika diisi angka negatif artinya dilakukan *offset* ke kiri, sedangkan jika positif artinya dilakukan *offset* ke kanan.

6. Fungsi *ROW*

Fungsi *ROW* digunakan untuk mengembalikan nilai posisi baris dari suatu referensi. Berikut ini adalah sintaks penulisan fungsi *ROW* :

$$=Row([reference])$$

Bila *reference* dikosongkan maka akan mengembalikan nilai posisi baris dari *cell* tempat fungsi dituliskan.

7. Fungsi *MOD*

Fungsi *MOD* digunakan untuk menghitung sisa hasil bagi. Berikut ini sintaks penulisan fungsi *MOD*:

$$=MOD(number;divisor)$$

Number merupakan angka pembilang sedangkan *divisor* merupakan angka penyebut.

8. Fungsi *ROUND*

Fungsi *ROUND* digunakan untuk melakukan pembulatan. Berikut ini adalah sintaks penulisan fungsi *ROUND* :

$$=ROUND(number;round_digits)$$

Number adalah angka yang akan dibulatkan sementara *round_digits* adalah jumlah angka desimal yang dikehendaki.

9. Fungsi *MIN*

Fungsi *MIN* digunakan untuk mencari nilai minimal dari sekumpulan nilai. Sintaks penulisan fungsi *MIN* adalah :

$$=MIN(number1;[number2];...)$$

10. Fungsi *MAX*

Fungsi *MAX* digunakan untuk mencari nilai maksimum dari sekumpulan nilai. Sintaks penulisan fungsi *MAX* adalah:

$$=MAX(number1;[number2];...)$$

11. Fungsi *AVERAGEIF*

Fungsi *AVERAGEIF* digunakan untuk mencari nilai rata – rata dari sekumpulan nilai yang memenuhi kriteria tertentu. Sintaks penulisan fungsi *AVERAGE* adalah:

$$=AVERAGEIF(range;criteria;[average_range];...)$$

12. Fungsi *SUM*

Fungsi *SUM* digunakan untuk melakukan operasi penjumlahan terhadap sekumpulan nilai. Sintaks penulisan fungsi *SUM* adalah :

$$=SUM(number1;[number2];...)$$

13. Fungsi *COUNTIFS*

Fungsi *COUNTIFS* digunakan untuk menghitung jumlah *cell* yang memenuhi beberapa kriteria tertentu. Sintaks penulisan fungsi *COUNTIFS* adalah :

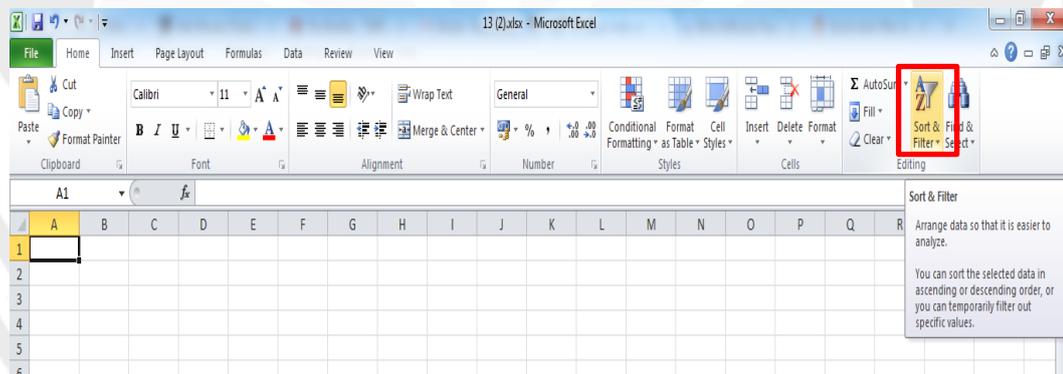
$$= COUNTIFS (criteria_range1;[criteria1];[criteria_range2];[criteria2];...)$$

14. Formula *Array*

Formula *array* merupakan formula khusus untuk melakukan suatu operasi pada *array* tertentu. *Array* adalah kumpulan dari beberapa sel yang tersebar. Dibandingkan dengan formula biasa, formula *array* ini memiliki sedikit perbedaan dalam tata cara penulisannya. Pada formula biasa kita tinggal menuliskan formula yang diinginkan pada *formula bar* kemudian kita tekan *enter*. Sedangkan untuk mengaktifkan formula *array* kita harus menekan *ctrl + shift + enter*. Pada akhir formula yang kita tuliskan akan ada perbedaan berupa munculnya tanda kurung kurawal.

15. Fitur *Sort & Filter*

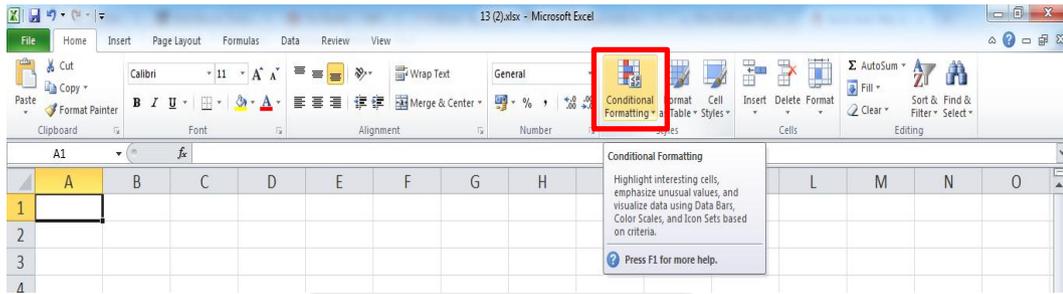
Fitur *Sort & Filter* dapat memudahkan kita untuk menemukan dan menyaring suatu nilai di antara nilai – nilai lain. *Sort & Filter* juga dapat digunakan untuk mengurutkan nilai menurut urutan tertentu. *Sort & Filter* dapat diakses melalui *Home Tab, Editing Group*, pilih *Sort & Filter*. Gambar 2.2 menggambarkan letak fitur *Sort & Filter* dalam antar-muka *Microsoft Excel* versi 2010(ditandai oleh kotak bergaris merah).



Gambar 2.2. Letak Fitur *Sort & Filter*

16. Fitur *Conditional Formatting*

Fitur *Conditional Formatting* dapat memudahkan kita untuk memformat *cell* secara otomatis bila *cell* tersebut memenuhi kriteria tertentu. *Cell* yang diformat biasanya mempunyai nilai-nilai tertentu yang khas sehingga perlu dibedakan formatnya dengan *cell* lain supaya mudah dikenali. *Conditional Formatting* dapat diakses melalui *Home Tab, Styles Group*, pilih *Conditional Formatting*. Gambar 2.3 menggambarkan letak *Conditional Formatting* dalam antar-muka *Microsoft Excel* versi 2010 (ditandai oleh kotak bergaris merah).



Gambar 2.3. Letak Fitur *Conditional Formatting*

