

BAB II

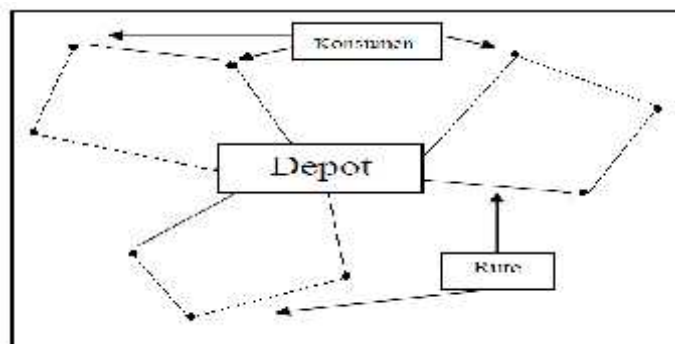
LANDASAN TEORI

Bab II dalam penelitian ini terdiri atas *vehicle routing problem*, teori lintasan dan sirkuit, metode *saving* matriks, matriks jarak, matriks penghematan, dan penentuan urutan konsumen.

2.1 *Vehicle Routing Problem (VRP)*

VRP pertama kali diutarakan oleh Dantzig dan Ramser. VRP adalah permasalahan dari penentuan rute yang akan dibentuk dari sejumlah konsumen didasarkan atas satu atau beberapa depot. Setiap konsumen akan dilayani oleh satu kendaraan dengan batasan-batasan tertentu dan rute tersebut diawali dan diakhiri pada depot. Beberapa contoh batasan-batasan yang diberikan adalah kapasitas kendaraan, keterbatasan aksesibilitas konsumen, permintaan *pick-ups delivery* dan *time windows* atau kendala waktu. VRP berhubungan dengan pengiriman dan pengambilan barang.

Permasalahan VRP dapat dibedakan menjadi dua, yaitu permasalahan statis dan dinamis. Permasalahan statis adalah permintaan konsumen telah diketahui sebelumnya. Sedangkan permasalahan dinamis adalah permintaan konsumen yang selalu berubah-ubah. Hasil dari penyelesaian masalah VRP adalah rute seperti yang digambarkan pada Gambar 2.1.



Gambar 2.1 Contoh Hasil Pemecahan VRP

Salah satu permasalahan yang menarik didalam pembahasan VRP adalah masalah mengenai *Capacitated Vehicle Routing Problem (CVRP)*. CVRP merupakan kasus penentuan dari jumlah kendaraan yang bertujuan meminimalkan total jarak yang ditempuh semua rute, yang akan memenuhi kapasitas kendaraan dan melayani setiap konsumen.

2.2 Teori Lintasan dan Sirkuit

Lintasan adalah barisan berselang-seling dari simpul dan sisi yang berawal dengan simpul v_o dan berakhir dengan simpul v_n . Lintasan terbagi atas 2 macam, yaitu lintasan terbuka dan lintasan tertutup. Lintasan terbuka adalah lintasan yang tidak berawal dan berakhir pada simpul yang sama, sedangkan lintasan tertutup adalah lintasan yang berawal dan berakhir pada simpul yang sama atau disebut juga sirkuit. Sirkuit terdiri dari 2 macam, yaitu :

2.2.1 Sirkuit Euler

Sirkuit euler adalah lintasan yang melalui masing-masing sisi di dalam graf tepat satu kali yang berawal dan berakhir pada simpul yang sama membentuk lintasan tertutup.

2.2.2 Sirkuit Hamilton

Sirkuit hamilton adalah lintasan yang melalui tiap simpul di dalam graf tepat satu kali yang berawal dan berakhir pada simpul yang sama membentuk lintasan tertutup.

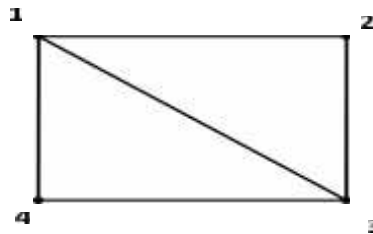
Perbedaan lintasan terbuka dan lintasan tertutup (sirkuit euler, sirkuit hamilton) dapat dilihat pada Contoh 2.1 berikut ini :

Contoh 2.1 :

Diberikan graf sederhana pada Gambar 2.2, tentukanlah lintasan dan perbedaan antara lintasan terbuka serta lintasan tertutup.

Penyelesaian :

Lintasan yang terbentuk dan perbedaan lintasan terbuka serta lintasan tertutup dapat dilihat pada Tabel 2.1.



Gambar 2.2 Graf Sederhana

Tabel 2.1 Perbedaan Lintasan Terbuka dan Tertutup (Euler dan Hamilton)

Lintasan	Lintasan Terbuka		Lintasan Tertutup (Sirkuit)	
	Euler	Hamilton	Euler	Hamilton
(1,4,3,2)	Tidak	Ya	Tidak	Tidak
(1,2,3,4,1)	Tidak	Tidak	Tidak	Ya
(1,4,3,1,2,3)	Ya	Tidak	Tidak	Tidak

2.3 Metode *Saving* Matriks

Metode *saving* matriks adalah metode yang digunakan untuk menentukan rute terbaik dengan mempertimbangkan jarak yang dilalui, jumlah kendaraan yang akan digunakan dan jumlah produk yang dapat dimuat kendaraan dalam pengiriman produk ke konsumen agar proses distribusi optimal. Langkah-langkah metode *saving* matriks adalah sebagai berikut :

- a. Menentukan matriks jarak.
- b. Menentukan matriks penghematan.
- c. Mengklasifikasikan konsumen ke sebuah rute.
- d. Menentukan urutan konsumen atau urutan pengiriman.

2.4 Matriks Jarak

Matriks jarak menyatakan jarak antara tiap pasang lokasi yang dikunjungi. Jarak antara lokasi A yang terletak pada koordinat X_a, Y_a dan lokasi B yang terletak pada koordinat X_b, Y_b dicari dengan menggunakan rumus:

$$Dist(A, B) = \sqrt{(X_a - X_b)^2 + (Y_a - Y_b)^2}$$

Contoh 2.2 :

Sebuah perusahaan akan mengirimkan produk, posisi perusahaan di koordinat (0,0) ke 8 lokasi yang koordinatnya sudah diketahui pada Tabel 2.2, ukuran order juga sudah tercantum, perusahaan ingin menentukan berapa kendaraan yang dibutuhkan jika kapasitas setiap kendaraan adalah 700 unit dan perusahaan hanya mampu menyewa maksimum 3 buah kendaraan.

Tabel 2.2 Lokasi Tujuan dan Ukuran Order

Konsumen(C)	Koordinat X	Koordinat Y	Ukuran order
1	10	8	320
2	-3	10	85
3	16	-8	300
4	10	2	150
5	9	1	200
6	4	5	120
7	10	12	180
8	2	6	230

Penyelesaian :

Lokasi depot terletak pada koordinat $DC = 0,0$

$$Dist(D, C1) = \sqrt{0 - 10^2 + 0 - 8^2}$$

$$Dist(DC, C1) = 12.8$$

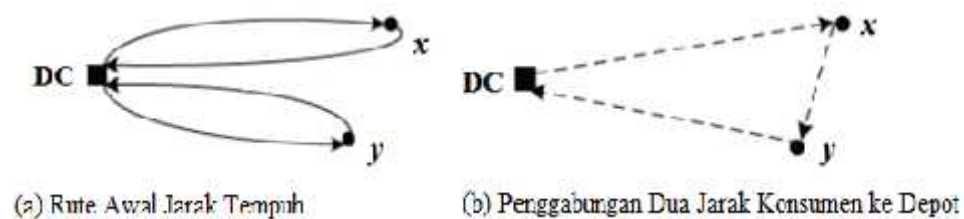
Matriks jarak secara keseluruhan dapat di lihat pada Tabel 2.3.

Tabel 2.3 Matriks Jarak dari Depot ke Konsumen

	Depot	C1	C2	C3	C4	C5	C6	C7	C8
C1	12.8	0.0							
C2	10.4	13.2	0.0						
C3	17.9	17.1	26.2	0.0					
C4	10.2	6.0	15.3	11.7	0.0				
C5	9.1	7.1	15.0	11.4	1.4	0.0			
C6	6.2	6.7	8.6	17.7	6.7	6.4	0.0		
C7	15.8	4.0	13.2	20.9	10.0	11.0	9.2	0.0	
C8	6.3	8.2	6.4	19.8	8.9	8.6	2.2	10.0	0.0

2.5 Matriks Penghematan (*Saving Matrix*)

Saving matriks merupakan penggabungan jarak yang ditempuh kendaraan dalam melakukan perjalanan dari depot ke konsumen x kemudian kembali lagi ke depot dan perjalanan dari depot ke konsumen y kemudian kembali lagi ke depot, menjadi perjalanan dari depot ke konsumen x kemudian ke konsumen y dan akhirnya kembali lagi ke depot. Secara umum dapat digambarkan sebagai berikut :



Gambar 2.3 Ilustrasi dari *Saving* Matriks

Nilai dari saving matriks dapat dihitung menggunakan rumus sebagai berikut :

$$S_{x,y} = \text{Dist}(DC,x) + \text{Dist}(DC,y) - \text{Dist}(x,y)$$

Keterangan :

$S(x,y)$ = nilai *saving* matriks atau jarak yang dihemat.

$\text{Dist}(DC,x)$ = perjalanan dari depot ke konsumen x .

$\text{Dist}(DC,y)$ = perjalanan dari depot ke konsumen y .

$\text{Dist}(x,y)$ = perjalanan dari konsumen x ke konsumen y .

Contoh 2.3 :

Tentukan matriks penghematan dari matriks jarak yang berada pada Tabel 2.3.

Penyelesaian :

$$S(x,y) = \text{Dist}(DC,x) + \text{Dist}(DC,y) - \text{Dist}(x,y)$$

$$S_{C1,C2} = \text{Dist}(DC,C1) + \text{Dist}(DC,C2) - \text{Dist}(C1,C2)$$

$$S_{C1,C2} = 12.8 + 10.4 - 13.2$$

$$S_{C1,C2} = 10.0$$

Matriks penghematan secara keseluruhan dapat dilihat pada Tabel 2.4.

Tabel 2.4 Matriks Penghematan Jarak

	C1	C2	C3	C4	C5	C6	C7	C8
C1	0.0							
C2	10.0	0.0						
C3	13.6	2.2	0.0					
C4	17.0	5.3	16.4	0.0				
C5	14.8	4.5	15.6	17.8	0.0			
C6	12.5	8.2	6.6	9.9	9.1	0.0		
C7	24.4	12.9	12.6	15.8	13.7	12.8	0.0	
C8	10.9	10.3	4.4	7.6	6.8	10.5	11.9	0.0

2.6 Penentuan Urutan Konsumen

Pengurutan konsumen menggunakan metode *farthest insert*, *cheapest insert*, *nearest neighbour*, dan *nearest insert*. Hasil yang didapatkan dari keempat metode tersebut dibandingkan dan dipilih yang menghasilkan jarak yang minimum.

2.6.1 *Farthest Insert*

Memasukkan konsumen yang memberikan perjalanan paling jauh. Untuk setiap konsumen yang belum termasuk dalam satu rute, evaluasi kenaikan jarak tempuh yang memiliki kenaikan jarak tempuh terbesar menggunakan rumus sebagai berikut :

$$\Delta f = c_{ik} + c_{kj} - c_{ij}$$

Keterangan :

Δf = kenaikan jarak tempuh.

c_{ik} = jarak tempuh antara konsumen i dan konsumen k .

c_{kj} = jarak tempuh antara konsumen k dan konsumen j .

c_{ij} = jarak tempuh antara konsumen i dan konsumen j .

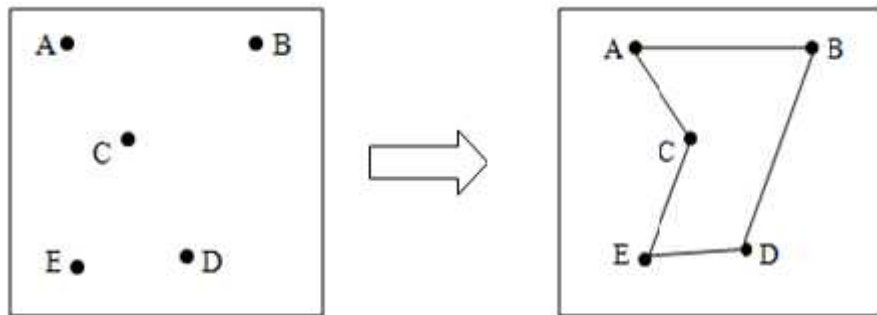
Contoh 2.3 :

Diketahui jarak antara simpul-simpul seperti diperlihatkan oleh Tabel 2.5. Bagaimana rute urutan simpul yang terbentuk dan berapa total jarak yang ditempuh dari metode *farthest insert* ?

Tabel 2.5 Jarak antara Simpul-simpul

Simpul	A	B	C	D	E
A	0				
B	8	0			
C	4	6	0		
D	9	7	5	0	
E	9	10	6	4	0

Penyelesaian :



Gambar 2.4 Rute Pengurutan Simpul Metode *Farthest Insert*

Jadi, rute yang terbentuk menggunakan metode *farthest insert* yaitu, A-B-D-E-C-A dengan total jarak 29.

2.6.2 *Cheapest Insert*

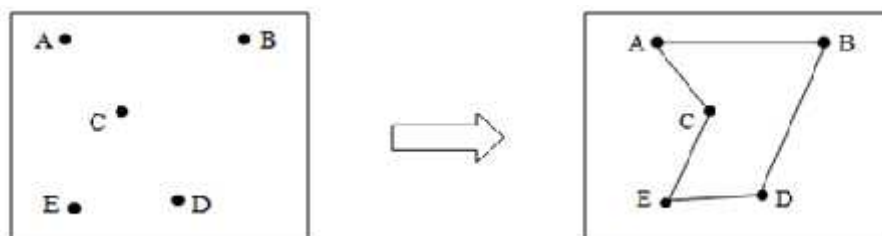
Mulai dari depot, prosedur ini memilih konsumen yang paling dekat dengan depot. Setelah itu, membentuk rute terpendek dengan memilih Δf yang paling minimal. Pada tiap langkah, rute dibangun dengan penyisipan menggunakan rumus, yaitu :

$$\Delta f = c_{ik} + c_{kj} - c_{ij}.$$

Contoh 2.6 :

Diketahui jarak antara simpul-simpul seperti diperlihatkan oleh Tabel 2.5. Bagaimana rute urutan simpul yang terbentuk dan berapa total jarak yang ditempuh dari metode *nearestneighbour* ?

Penyelesaian :



Gambar 2.7 Rute Pengurutan Simpul Metode *Cheapest Insert*

Jadi, rute yang terbentuk menggunakan metode *cheapest insert* yaitu, A-B-C-D-E-A dengan total jarak 29.

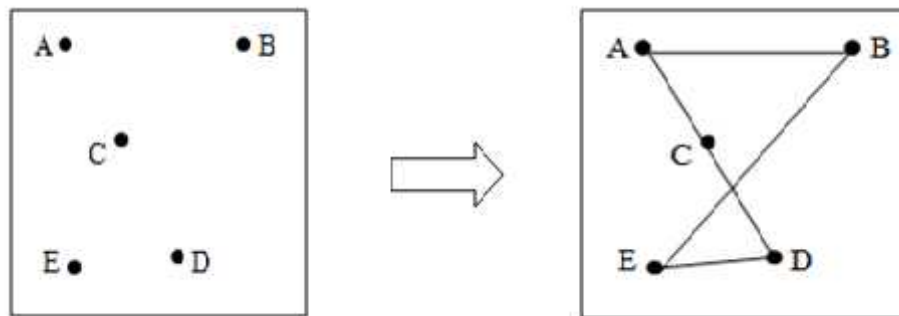
2.6.3 Nearest Neighbour

Mulai dari depot, prosedur ini menambah konsumen yang terdekat untuk melengkapi rute. Pada tiap langkah, rute dibangun dengan menambahkan konsumen yang terdekat dari titik terakhir yang dikunjungi oleh kendaraan sampai semua konsumen terkunjungi.

Contoh 2.5 :

Diketahui jarak antara simpul-simpul seperti diperlihatkan oleh Tabel 2.5. Bagaimana rute urutan simpul yang terbentuk dan berapal total jarak yang ditempuh dari metode *nearest neighbour* ?

Penyelesaian :



Gambar 2.6 Rute Pengurutan Simpul Metode *Nearest Neighbour*

Jadi, rute yang terbentuk menggunakan metode *nearest neighbour* yaitu, A-C-D-E-B-A dengan total jarak 31.

2.6.4 Nearest Insert

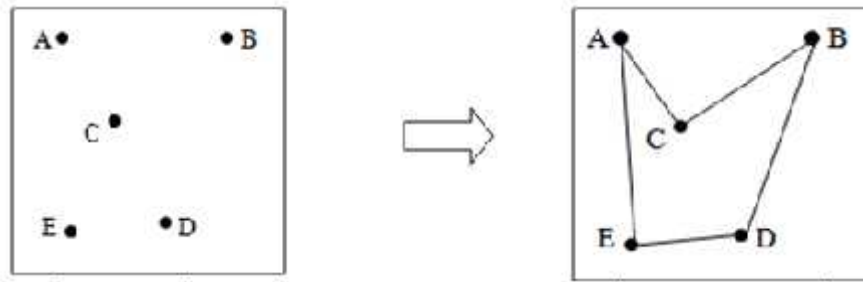
Memasukkan konsumen yang memberikan perjalanan terpendek. Setiap konsumen yang belum termasuk dalam satu rute, evaluasi kenaikan jarak tempuh yang memiliki kenaikan jarak tempuh terkecil menggunakan rumus, yaitu :

$$\Delta f = c_{lk} + c_{kj} - c_{lj}$$

Contoh 2.4 :

Diketahui jarak antara simpul-simpul seperti diperlihatkan oleh Tabel 2.5. Bagaimana rute urutan simpul yang terbentuk dan berapal total jarak yang ditempuh dari metode *nearest insert* ?

Penyelesaian :



Gambar 2.5 Rute Pengurutan Sempul Metode *Nearest Insert*

Jadi, rute yang terbentuk menggunakan metode *nearest insert* yaitu, A-C-B-D-E-A dengan total jarak 30.