BAB III
METODOLOGI PENELITIAN

Bab ini mengkaji metodologi yang digunakan untuk menyelesaikan SVRP, dimulai dengan mendeskripsikan masalah kemudian dilanjutkan pembangunan model optimisasi dari SVRP hingga langkah kerja dari algoritma SA untuk menyelesaikan VRP.

3.1 Deskripsi Masalah

Penelitian ini meneliti SVRP. SVRP merupakan masalah penentuan rute pendistribusian sejumlah barang oleh kendaraan yang tersedia dari suatu depot ke sejumlah pelanggan lalu kembali ke depot dengan permintaan pelanggan yang bersifat *stochastic*. SVRP adalah perluasan dari VRP dengan faktor tambahan yaitu permintaan pelanggan yang bersifat *stochastic* atau tidak pasti. Setiap kendaraan memiliki batasan kapasitas dan melakukan pendistribusian sebanyak satu kali pengiriman, yaitu dari depot ke setiap wilayah pelayanan lalu kembali ke depot.

Penyelesaian SVRP bertujuan untuk menentukan rute pendistribusian dengan total jarak terpendek. SVRP termasuk ke dalam masalah *NP-Hard Problem*, artinya waktu komputasi yang digunakan untuk menyelesaikan permasalahan akan semakin meningkat seiring dengan meningkatnya ruang lingkup masalah. Akibatnya, jika solusi optimal dihitung dengan menggunakan metode eksak, maka diperlukan waktu komputasi yang lama. Pada penelitian ini, SVRP akan diselesaikan berbasis skenario, di mana pada setiap skenario *Vehicle Routing Problem* diselesaikan dengan menggunakan algoritma SA.

3.2 Tahapan Penelitian

Tahapan penelitian yang dilakukan adalah sebagai berikut:

1. Studi Pustaka

   Pada tahap ini dilakukan studi pustaka dengan cara mempelajari konsep serta teori-teori mengenai SVRP, model SA dan algoritmanya, yang bersumber dari berbagai literatur baik jurnal, buku, dan karya tulis lainnya.

2. Pengumpulan Data

   Data yang digunakan pada penelitian ini adalah data sekunder yang diperoleh dari jurnal berjudul "*The Vehicle Routing Problem with Stochastic Demands in an Urban Area – A Case Study*". Data tersebut terdiri dari data depot, data kendaraan, data permintaan pelanggan dalam beberapa periode, serta titik koordinat lokasi depot dan setiap pelanggan.

3. Pemodelan

   Pada tahap ini dibangun model optimisasi dari SVRP dengan terlebih dahulu mendefinisikan himpunan, parameter, dan variabel keputusan dari model optimisasi.

4. Penyelesaian Model

   Pada tahapan ini, model matematika akan diselesaikan berbasis skenario dengan terlebih dahulu memodelkan distribusi permintaan berdasarkan data permintaan sebelumnya. Selanjutnya, untuk setiap skenario akan diterapkan algoritma SA untuk menyelesaikan *Vehicle Routing Problem*.

5. Validasi

   Solusi penyelesaian SVRP akan divalidasi dengan membandingkan solusi optimal yang dihasilkan oleh program yang dibuat menggunakan sebuah *software* dengan solusi optimal hasil perhitungan manual. Validasi dilakukan untuk memeriksa apakah proses perhitungan untuk memperoleh solusi penyelesaian ada kesalahan atau tidak. Jika diperoleh solusi yang sama antara perhitungan manual dan program maka program perhitungan sudah benar sehingga akan dilanjutkan ke tahap implementasi. Jika solusi berbeda maka tahapan penyelesaian ada kesalahan sehingga perlu diperbaiki.

6. Implementasi

   Setelah model dan teknik penyelesaian valid, selanjutnya model dan teknik penyelesaian tersebut akan diimplementasikan pada penyelesaian SVRP untuk menentukan rute bagi kendaraan pengangkut sampah di sebuah kota. Selanjutnya akan dianalisis kinerja dari algoritma SA dalam *Vehicle Routing Problem*.

7. Penarikan Kesimpulan

   Penarikan kesimpulan akan dilakukan berdasarkan hasil implementasi.

3.3 Asumsi dan Model Optimasi SVRP

Model optimisasi dari SVRP dibangun dengan menggunakan asumsi-asumsi berikut:

1. Setiap pelanggan hanya dikunjungi tepat satu kali oleh satu kendaraan;
2. Setiap kendaraan mempunyai batasan kapasitas yang sama;
3. Setiap pelanggan mempunyai permintaan yang bersifat *stochastic*;
4. Jarak antara pelanggan i ke j sama dengan jarak pelanggan j ke i;
5. Kendaraan yang tersedia cukup untuk mengirim semua permintaan pelanggan.

Tahapan pertama pemodelan adalah mendefinisikan himpunan dan parameter. Berikut adalah pendefinisian himpunan dan parameter yang akan digunakan pada model optimisasi:

1. **Himpunan**
    - S: Himpunan pelanggan
    - D: Himpunan depot
    - K: Himpunan kendaraan
2. **Parameter**
    - Q: Kapasitas kendaraan
    - $\mathbb{E}(d_i)$: Ekspektasi permintaan pelanggan ke i
    - c_{ij}: Jarak i ke j di mana $i, j \in S \cup D$

Pada penelitian ini, didefinisikan variabel keputusan sebagai berikut:

$$x_{ij}^k = \begin{cases} 1, \text{jika terdapat perjalanan dari } i \text{ ke } j \text{ dengan kendaraan } k \\ 0, \text{yang lainnya} \end{cases}$$

$$y_i^k = \begin{cases} 1, \text{jika kendaraan k mengunjungi pelanggan i} \\ 0, \text{yang lainnya} \end{cases}$$

Adapun kendala-kendala dari model SVRP adalah sebagai berikut:

1. Setiap pelanggan hanya dikunjungi tepat satu kali oleh satu kendaraan.

$$\sum_{k \in K} y_i^k = 1, \forall i \in S$$

2. Jumlah kendaraan yang keluar dan masuk ke depot ada sebanyak K.

$$\sum_{j \in S} x_{0j} = \sum_{j \in S} x_{j0} = K, 0 \in D$$

3. Setiap kendaraan mengunjungi pelanggan tepat satu kali. Setelah itu kendaraan meninggalkan pelanggan tersebut dan melanjutkan perjalanan menuju pelanggan selanjutnya atau kembali menuju depot jika pelanggan tersebut adalah pelanggan terakhir dari sebuah rute. Ini berarti, kendaraan yang menuju pelanggan j akan meninggalkan pelanggan i.

$$\sum_{j\in S} x_{ij}^{k} = \sum_{j\in S} x_{jg}^{k} = y_i^k \,, \forall i, g \in S \cup D, \forall k \in K$$

4. Setiap kendaraan memiliki batasan kapasitas Q, sehingga ekspektasi jumlah permintaan pelanggan yang harus dipenuhi dalam satu rute tidak melebihi kapasitas maksimum kendaraan.

$$\sum_{i\in S} \mathbb{E}(d_i) y_i^k \le Q, \forall k \in K$$

Fungsi tujuan dari model optimisasi didefinisikan untuk mencari rute pendistribusian barang dengan meminimalkan ekspektasi total jarak dari depot ke pelanggan-pelanggan lalu kembali ke depot dengan menggunakan sejumlah k kendaraan sehingga permintaan setiap pelanggan terpenuhi. Fungsi tujuan ini dituliskan sebagai:

Meminimumkan:

$$Z = \mathbb{E}\left(\sum_{k\in K}\sum_{j\in S}\sum_{i\in S} c_{ij} x_{ij}^{k}\right)$$

Model optimisasi di atas termasuk dalam kriteria model *stochastic linear programming*. Adapun batasan variabel dari model optimisasi adalah sebagai berikut:

$$x_{ij}^{k} \in \{0,1\}, y_i^k \in \{0,1\}, \forall i \in S, j \in S, k \in K$$

Selengkapnya, model optimasi SVRP untuk menentukan rute bagi kendaraan pengangkut sampah di sebuah kota adalah sebagai berikut:

Meminimumkan:

$$Z = \mathbb{E}\left(\sum_{k\in K}\sum_{j\in S}\sum_{i\in S} c_{ij} x_{ij}^{k}\right)$$

Terhadap:

$$\sum_{k\in K} y_i^k = 1, \forall i \in S$$

$$\sum_{j\in S} x_{0j} = \sum_{j\in S} x_{j0} = K, 0 \in D$$

$$\sum_{j\in S} x_{ij}^k = \sum_{j\in S} x_{jg}^k = y_i^k, \forall i, g \in S \cup D, \forall k \in K$$

$$\sum_{i\in S} \mathbb{E}(d_i) y_i^k \leq Q, \forall k \in K$$

$$x_{ij}^k \in \{0,1\}, y_i^k \in \{0,1\}, \forall i \in S, j \in S, k \in K$$

Pada sub bab selanjutnya akan dibahas teknik penyelesaian dari model optimisasi di atas.

3.4 Teknik Penyelesaian Model menggunakan Algoritma *Simulated Annealing*

Model *stochastic linear programming* di atas akan diselesaikan berbasis skenario merujuk pada penelitian Normasari, dkk. (2019) dan Novianingsih dan Hadianti (2016). Skenario dalam penelitian ini merupakan pembangkitan permintaan setiap pelanggan berdasarkan distribusi data permintaan pada periode sebelumnya. Untuk setiap skenario, SVRP akan berubah menjadi VRP. VRP ini akan diselesaikan dengan Algoritma SA.

Misalkan Ω adalah himpunan skenario. Dimisalkan pula bahwa d_i^ω banyaknya permintaan pelanggan i untuk skenario $\omega \in \Omega$. Maka model *stochastic linear programming* pada Subbab 3.3 dapat dituliskan sebagai berikut:

Meminimumkan:

$$Z_\omega = \mathbb{E}\left(\sum_{k\in K}\sum_{j\in S}\sum_{i\in S} c_{ij} x_{ij}^k\right) = \frac{1}{|\Omega|}\sum_{\omega\in\Omega}\left(\sum_{k\in K}\sum_{j\in S}\sum_{i\in S} c_{ij} x_{ij}^k\right)_\omega$$

Terhadap:

$$\sum_{k\in K} y_i^k = 1, \forall i \in S$$

$$\sum_{j\in S} x_{0j} = \sum_{j\in S} x_{j0} = K, 0 \in D$$

$$\sum_{j\in S} x_{ij}^k = \sum_{j\in S} x_{jg}^k = y_i^k \text{ }, \forall i, g \in S \cup D, \forall k \in K$$

$$\sum_{i\in S} \mathbb{E}(d_i^{\omega}) y_i^k \leq Q, \forall k \in K, \forall \omega \in \Omega$$

$$x_{ij}^k, \in \{0,1\}, y_i^k \in \{0,1\}, \forall i \in S, j \in S, k \in K, \omega \in \Omega$$

Model di atas diselesaikan dengan langkah berikut:

1. Modelkan distribusi permintaan pelanggan berdasarkan data permintaan periode sebelumnya.
2. Bangkitkan himpunan skenario berdasarkan distribusi permintaan pelanggan.
3. Untuk setiap skenario ω, lakukan langkah berikut:
   3.1. Bangkitkan permintaan seluruh pelanggan berdasarkan distribusi yang diperoleh pada Langkah 1.
   3.2. Gunakan permintaan pada 3.1 untuk membentuk VRP.
   3.3. Selesaikan VRP dengan Algoritma SA.
4. Pilih solusi terbaik (dengan total jarak terpendek) dari sebuah skenario sebagai solusi optimal.

Dengan demikian,

$$min\, \mathbb{E}\left(\sum_{k\in K}\sum_{j\in S}\sum_{i\in S} c_{ij}x_{ij}^k\right) = \frac{1}{|\Omega|}\sum_{\omega\in\Omega}\left(\sum_{k\in K}\sum_{j\in S}\sum_{i\in S} c_{ij}x_{ij}^k\right)_{\omega} = min\{Z_{\omega}: \omega \in \Omega\}$$

Pada penelitian ini, distribusi permintaan setiap pelanggan selama periode tertentu dimodelkan menggunakan uji normalitas *Kolmogorov Smirnov* dengan cara sebagai berikut (Zaiontz, 2023):

a. Hitung rata-rata dan standar deviasi dari permintaan setiap pelanggan;
b. Urutkan data dari yang terkecil sampai terbesar;
c. Hitung frekuensi setiap data dan frekuensi kumulatifnya (F_k);
d. Hitung $\mathrm{S_n(x)} = \frac{\mathrm{F_k}}{n}$, dengan $\mathrm{S_n(x)}$ adalah probabilitas kumulatif empiris dan n banyaknya data;
e. Hitung $\mathrm{F(x)} = norm.dist(x_i; rata - rata; standar\ deviasi; kumulatif)$ dengan F(x) adalah probabilitas kumulatif normal dan x_i data permintaan pelanggan;

f. Hitung $D_{hitung} = |F(x) - S_n(x)|$ dan pilih D_{hitung} dengan nilai terbesar;

g. Lihat D_{tabel} pada tabel *Kolmogorov Smirnov* dengan tingkat kepercayaan (α) yang sudah ditentukan dan n yang diketahui;

h. Jika $D_{hitung}\ terbesar \leq D_{tabel}$, maka data berdistribusi normal. Jika $D_{hitung}\ terbesar > D_{tabel}$, maka data berdistribusi tidak normal.

i. Jika data berdistribusi tidak normal, maka akan dilakukan teknik kurva *fitting* untuk menentukan model distribusi permintaan.

Setelah distribusi permintaan dimodelkan, tahapan selanjutnya adalah membangkitkan sejumlah skenario, di mana pada setiap skenario permintaan dibangkitkan secara acak berdasarkan distribusi permintaan. Setelah permintaan dibangkitkan, maka masalah pencarian rute akan berubah menjadi VRP. Selanjutnya VRP diselesaikan dengan menggunakan Algoritma SA. Tahapan penyelesaian VRP dengan menggunakan Algoritma SA adalah sebagai berikut:

1. Inisiasi Parameter Input

Algoritma SA bekerja dengan dipengaruhi oleh beberapa parameter yang harus ditetapkan sebelum algoritma dijalankan. Parameter-parameter tersebut adalah sebagai berikut:

a. Temperatur awal (T_a)

Temperatur awal merupakan penanda awal iterasi. Suhu awal diatur cukup tinggi agar tidak melewatkan titik-titik potensial yang mungkin menjadi solusi optimal dan juga solusi yang buruk memiliki kemungkinan lebih besar untuk diterima sehingga tidak terjebak pada solusi lokal optimal. Temperatur awal yang diatur harus lebih dari 0 (Muhaddad, 2014).

b. Parameter reduksi suhu setiap iterasi (α)

Parameter reduksi suhu menyatakan presentase yang digunakan untuk menurunkan suhu di setiap iterasi dengan rentang nilainya yaitu antara 0 dan 1 (Muhaddad, 2014).

c. Temperatur akhir (T_f)

Temperatur akhir merupakan salah satu kriteria pemberhentian algoritma yang dapat digunakan yaitu dengan memperkecil nilainya, dimana $0 < T_f < T_a$ (Panggabean, 2004).

Berikut rentang nilai dari parameter input dapat dilihat pada Tabel 3.1.

Tabel 3.1 Rentang Nilai Parameter Input

Parameter Input	Rentang Nilai
Temperatur awal (T_a)	> 0
Parameter reduksi suhu (α)	$0 - 1$
Temperatur akhir (T_f)	$0 < T_f < T_a$

2. **Pembangkitan Solusi Awal**

Pada penelitian ini, solusi dibangkitkan secara *random* dengan tetap memperhatikan kapasitas dari setiap kendaraan. Representasi solusi permasalahan VRP ini adalah berupa sebuah vektor berukuran $1 \times (n + m)$, dengan *n* adalah banyaknya pelanggan dan *m* adalah banyaknya kendaraan dikurangi 1. Elemen pada vektor berisi permutasi $\{1, \dots, n\} \cup \{0, \dots 0\}$. Jika solusi yang dibangkitkan melanggar kendala, yaitu ada kendaraan yang melebihi kapasitas maksimumnya, maka solusi baru akan dibangkitkan sampai kendala dipenuhi. Misalkan sebuah perusahaan barang memiliki 6 pelanggan dengan 3 kendaraan dengan label depot 0. Dimisalkan pula representasi solusinya seperti pada Gambar 3.1. Maka, Rute Kendaraan 1 adalah 0 → 4 → 2 → 0, rute Kendaraan 2 adalah 0 → 5 → 3 → 0, dan rute Kendaraan 3 adalah 0 → 6 → 1 → 0.

4	2	0	5	3	0	6	1

Gambar 3.1 Representasi Solusi VRP

3. **Pencarian Solusi Tetangga**

Pada tahapan ini akan dicari solusi tetangga (s') dengan memodifikasi solusi awal. Terdapat beberapa cara yang digunakan untuk mencari solusi tetangga, yaitu (Ilhan, 2020):

1. *Exchange operator* yaitu menukar posisi antar dua bilangan/pelanggan dalam rute yang sama atau rute yang berbeda dengan cara membangkitkan angka *random* dua kali kemudian menukarkan posisi dari dua bilangan random tersebut;
2. *Insertion operator* yaitu menyisipkan bilangan/pelanggan yang terpilih ke posisi lain dalam rute yang sama atau rute yang berbeda. Dengan menggunakan cara yang sama seperti *exchange operator*, namun setelah

mendapatkan 2 bilangan *random*, salah satu dari bilangan tersebut di sisipkan ke posisi bilangan yang kedua;

3. *Reversion operator* yaitu membalik posisi sejumlah pelanggan terpilih dalam satu rute maupun rute yang berbeda.

Pada penelitian ini, ketiga operator akan digunakan. Tetapi pada setiap iterasi hanya ada satu operator saja yang akan dijalankan melalui pemilihan secara acak. Setelah operator tersebut dijalankan, maka akan diperoleh solusi baru yang disebut solusi tetangga (s'). Setelah solusi tetangga berhasil dibangkitkan, dilakukan pemeriksaan terhadap fungsi kendala. Jika ada kendala yang dilanggar, maka solusi tetangga ini tidak akan digunakan dan pencarian akan diulang hingga menemukan solusi tetangga yang layak. Setelah diperoleh solusi layak, maka selanjutnya dilakukan perhitungan hasil pemenuhan fungsi tujuan dari solusi tetangga tersebut untuk kemudian di evaluasi.

4. Pemeriksaan Nilai *Fitness*

Setelah solusi tetangga diperoleh, selanjutnya dilakukan pemeriksaan terhadap nilai *fitness* dari solusi tetangga. Nilai *fitness* dalam masalah VRP ini yaitu total jarak rute yang ditempuh oleh setiap kendaraan.

a. Jika nilai *fitness* solusi tetangga lebih baik dari nilai *fitness* solusi sebelumnya maka solusi tetangga tersebut diterima sebagai solusi baru. Lanjutkan proses ke Langkah 5;
b. Namun, jika nilai *fitness* solusi tetangga tidak lebih baik dari nilai *fitness* solusi sebelumnya maka dihitung peluang solusi tersebut akan diterima sebagai solusi yang baru menggunakan rumus:

$$P(\Delta f) = \exp\left(-\frac{|\Delta f|}{T_a}\right) \tag{3.1}$$

dengan $\Delta f = fitness\, s' - fitness\, s$ yang merupakan perubahan nilai *fitness* dari solusi tetangga (s') dan solusi awal (s). Selanjutnya, bangkitkan bilangan acak r, dengan $0 < r < 1$.

i. Jika $P(\Delta f) \geq r$, maka solusi tetangga diterima sebagai solusi baru. Lanjutkan proses ke Langkah 5;
ii. Jika $P(\Delta f) < r$, maka solusi awal dijadikan solusi baru dan lanjut proses ke Langkah 5.

5. **Jadwal Pendinginan**

Saat setiap iterasi selesai atau sudah didapatkan solusi baru, maka dilakukan proses pendinginan dengan cara mereduksi nilai suhu saat ini (T) dengan menggunakan parameter reduksi (α). Penurunan suhu mengikuti fungsi pendinginan berikut:

$$T = \alpha . T_a \tag{3.2}$$

Setelah suhu terbaru diperoleh, dilakukan pencarian solusi tetangga dengan menetapkan suhu terbaru menjadi T. Proses itu kemudian dilakukan berulang sampai suhu saat ini bernilai kurang dari suhu akhir (T_f).

6. **Kriteria Penghentian Algoritma**

Pada penelitian ini, kriteria penghentian algoritma yang digunakan yaitu, jika $T \leq T_f$, maka algoritma berhenti dan tampilkan s sebagai solusi akhir yang optimal dan $f(s)$ sebagai nilai fitness yang optimal. Jika $T > T_f$, kembali ke Langkah 3.

3.5 Contoh Kasus dan Penyelesaiannya

Untuk menggambarkan bagaimana tahapan penyelesaian SVRP berikut diberikan contoh kasus. Misalkan sebuah perusahaan pendistribusian barang yang memiliki sebuah depot dengan 10 pelanggan dan kendaraan untuk mendistribusikan barangnya sebanyak 3 buah dengan kapasitas masing-masing kendaraannya yaitu 55 paket. Diketahui data koordinat lokasi depot dan lokasi pelanggan pada Tabel 3.2.

Tabel 3.2 Data Koordinat dari Setiap Lokasi

Lokasi	X	Y
Depot	-6,9489	107,634
1	-6,9862	107,628
2	-6,9538	107,628
3	-6,9131	107,586
4	-6,877	107,618
5	-6,8959	107,588
6	-6,8812	107,583
7	-6,9018	107,582
8	-6,9018	107,613
9	-6,9672	107,575
10	-6,8947	107,597

Misalkan diketahui pula data permintaan dari setiap pelanggan selama 6 periode sebelumnya dengan rata-rata dan standar deviasinya pada Tabel 3.3. Kemudian, ubah titik koordinat dari masing-masing lokasi menjadi jarak antar titik lokasi. Berikut konversi data koordinat setiap lokasi pada Tabel 3.2 menjadi jarak antar titik lokasi pada Tabel 3.4.

Tabel 3.3 Data Permintaan, Rata-Rata, dan Standar Deviasi Setiap Pelanggan

Lokasi	**Permintaan**						**Mean**	**St.Dev**
	1	**2**	**3**	**4**	**5**	**6**		
1	9	8	9	8	10	9	8,833333	0,752773
2	9	8	10	10	9	9	9,166667	0,752773
3	17	15	17	19	18	16	17	1,414214
4	10	10	11	9	11	10	10,16667	0,752773
5	17	16	16	15	17	16	16,16667	0,752773
6	16	15	18	16	17	16	16,33333	1,032796
7	12	11	14	12	13	12	12,33333	1,032796
8	12	11	14	13	10	12	12	1,414214
9	20	23	21	22	19	21	21	1,414214
10	17	16	18	17	15	16	16,5	1,048809

Tabel 3.4 Data Jarak Antar Lokasi

Lokasi	**Depot**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Depot	0	4,21	0,86	6,67	8,20	7,81	9,44	7,81	5,74	6,88	7,31
1	4,21	0	3,61	9,38	12,21	10,99	12,72	10,70	9,54	6,27	10,75
2	0,86	3,61	0	6,51	8,62	7,83	9,51	7,73	6,02	6,09	7,43
3	6,67	9,38	6,51	0	5,37	1,93	3,57	1,33	3,26	6,15	2,39
4	8,20	12,21	8,62	5,37	0	3,95	3,92	4,87	2,82	11,12	3,06
5	7,81	10,99	7,83	1,93	3,95	0	1,73	0,94	2,86	8,07	1,01
6	9,44	12,72	9,51	3,57	3,92	1,73	0	2,30	4,05	9,61	2,17
7	7,81	10,70	7,73	1,33	4,87	0,94	2,30	0	3,45	7,32	1,85
8	5,74	9,54	6,02	3,26	2,82	2,86	4,05	3,45	0	8,42	1,95
9	6,88	6,27	6,09	6,15	11,12	8,07	9,61	7,32	8,42	0	8,43
10	7,31	10,75	7,43	2,39	3,06	1,01	2,17	1,85	1,95	8,43	0

Tahapan selanjutnya adalah memodelkan distribusi permintaan berdasarkan data permintaan periode sebelumnya. Dengan menggunakan uji normalitas *Kolmogorov Smirnov* dengan bantuan *software Microsoft Excel*, diperoleh bahwa data pada Tabel 3.3 berdistribusi normal. Sebagai contoh untuk permintaan pelanggan di lokasi 1 dengan rata-ratanya 8,83, standar deviasinya 0,75, dan n

sebanyak 6, data tersebut berdistribusi normal dengan proses dan hasil pada Tabel 3.5.

Tabel 3.5 Uji Normalitas Lokasi 1

Uji Normalitas Permintaan di Lokasi 1					
Data	Frekuensi	F_k	$\mathrm{S_n(x)}$	F(x)	D_{hitung}
8	2	2	0,333333	0,134143	0,19919
9	3	5	0,833333	0,587611	0,245722
10	1	6	1	0,939408	0,060592

Hasil uji normalitas diperoleh D_{hitung} maksimalnya sebesar $0{,}246$. Selanjutnya dicari D_{tabel} dengan tingkat kepercayaan sebesar 5% pada tabel *Kolmogorov Smirnov,* yaitu sebesar $0{,}519$. Karena $0{,}246 < 0{,}519$, maka data permintaan pelanggan di lokasi 1 berdistribusi normal. Dengan cara yang sama untuk memperoleh jenis distribusi data di setiap pelanggan.

Selanjutnya ditetapkan sejumlah skenario. Pada masing-masing dari skenario tersebut dibangkitkan permintaan secara acak berdasarkan distribusinya. Kemudian untuk setiap skenario, VRP diselesaikan menggunakan algoritma SA. Misalkan ditetapkan 2 skenario dengan permintaan acak yang dibangkitkan berdasarkan distribusi normal untuk setiap pelanggan seperti pada Tabel 3.6.

Tabel 3.6 Permintaan Pelanggan untuk Setiap Skenario

Pelanggan	**Skenario**	
	Skenario 1	**Skenario 2**
1	10	9
2	10	8
3	18	18
4	11	11
5	17	18
6	17	17
7	14	13
8	14	10
9	20	20
10	18	16

Kedua skenario tersebut masing-masing di proses menjadi bentuk VRP yang diselesaikan menggunakan algoritma SA. Kemudian jika kedua skenario tersebut sudah mempunyai solusi masing-masing, dipilih solusi terbaik dari sebuah skenario sebagai solusi optimal.

Misalkan ditetapkan temperatur awal (T_a) sebesar 50, parameter reduksi suhu setiap iterasi (α) sebesar 0,3, dan temperatur akhir (T_f) sebesar 5. Langkah pertama adalah membangkitkan solusi awal secara *random* dengan tetap memperhatikan kapasitas dari setiap kendaraan. Sebagai contoh untuk Skenario 1, solusi awal yang dibangkitkan dengan total jarak (*fitness*) sebesar 77,44 kilometer sebagai berikut: 1-2-3-4-0-5-6-7-0-8-9-10. Maka rute setiap kendaraan dari kendaraan dapat dilihat pada Tabel 3.7.

Tabel 3.7 Solusi Awal

Kendaraan	Rute	Total Jarak	Kapasitas	Keterangan
1	1-2-3-4-0	27,893	49	Rute dimulai dari depot (0) dilanjutkan ke pelanggan (1), lalu ke pelanggan (2), kemudian ke pelanggan (3), terakhir ke pelanggan (4), dan kembali ke depot (0).
2	5-6-7-0	19,647	48	Rute dimulai dari depot (0) dilanjutkan ke pelanggan (5), lalu ke pelanggan (6), terakhir ke pelanggan (7), dan kembali ke depot (0).
3	8-9-10	29,9	52	Rute dimulai dari depot (0) dilanjutkan ke pelanggan (8), lalu ke pelanggan (9), terakhir ke pelanggan (10), dan kembali ke depot (0).

Karena setiap kendaraan rute kendaraan di Tabel 3.7 tidak melebihi kapasitas kendaraan, maka solusi pada Tabel 3.7 digunakan sebagai solusi awal untuk skenario 1 pada Algoritma SA. Dengan cara yang sama untuk menentukan solusi awal pada skenario 2.

Tahap selanjutnya akan mulai menggunakan Algoritma SA. Tahapan pertama yaitu mencari solusi tetangga (s') dengan memodifikasi solusi awal. Lalu dipilih secara acak operator yang akan digunakan yaitu operator *exchange*, operator *insertion*, atau operator *reversion*. Pada iterasi pertama skenario 1, pencarian solusi tetangga dari solusi awal yang diketahui sebelumnya menggunakan operator *exchange* yang kemudian dibangkitkan dua bilangan acak, misalkan 5 dan 7, sehingga posisi elemen 5 dan 7 ditukar, diperoleh hasil pada Tabel 3.8.

Tabel 3.8 Solusi Tetangga pada Iterasi 1

Solusi Awal	1-2-3-4-0-5-6-7-0-8-9-10		
Solusi Tetangga	1-2-3-5-0-4-6-7-0-8-9-10		
Kendaraan	**Rute**	**Total Jarak**	**Kapasitas**
1	1-2-3-5-0	24,063	55
2	4-6-7-0	22,230	42
3	8-9-10	29,900	52

Solusi tetangga yang dihasilkan pada Tabel 3.8 tidak melanggar kendala dan menghasilkan total jarak (*fitness*) menjadi sebesar 76,193 kilometer, lebih baik dari solusi sebelumnya, sehingga solusi tetangga tersebut menjadi solusi baru. Kemudian suhu diturunkan mengikuti Persamaan (3.2):

$$T = \alpha \cdot T_a = 0{,}3 \ \cdot 50 = 15$$

Dengan tahapan yang sama seperti di atas, misalkan pada iterasi ke dua diperoleh solusi tetangga seperti pada Tabel 3.9 dengan total jarak (*fitness*) sebesar 76,849 kilometer. Solusi ini lebih buruk dari solusi pada iterasi pertama, sehingga solusi tetangga tersebut perlu dihitung peluang akan diterimanya sebagai solusi baru. Dengan menggunakan Persamaan (3.1), diperoleh peluangnya sebesar 0,96. Kemudian dibangkitkan sebuah bilangan acak *random,* misalkan 0,99. Karena 0,96 < 0,99, maka solusi tetangga ditolak sehingga solusi awal dijadikan solusi baru untuk di proses pada iterasi berikutnya.

Tabel 3.9 Solusi Tetangga pada Iterasi 2

Solusi Baru	1-2-3-5-0-4-6-7-0-8-9-10		
Solusi Tetangga	1-2-3-5-0-4-6-8-0-7-9-10		
Kendaraan	**Rute**	**Total Jarak**	**Kapasitas**
1	1-2-3-5-0	24,063	55
2	4-6-8-0	21,915	42
3	7-9-10	30,871	52

Kemudian suhu diturunkan lagi mengikuti Persamaan (3.2) untuk iterasi berikutnya:

$$T = \alpha \cdot T_a = 0{,}3 \cdot 15 = 4{,}5$$

Karena suhu saat ini, yaitu 4,5 sudah lebih kecil dari suhu akhir, yaitu 5. Maka algoritma berhenti, sehingga hasil akhir atau solusi optimal dari skenario 1, yaitu total jaraknya sebesar 76,193 kilometer dengan rute untuk seluruh kendaraan 1-2-3-5-0-4-6-7-0-8-9-10. Dengan cara yang sama untuk menentukan solusi optimal dari skenario 2

Jika setiap skenario telah selesai di proses menggunakan algoritma SA, maka hasil dari setiap skenario dibandingkan dan dipilih solusi terbaik dari sebuah skenario sebagai solusi optimal. Dalam kasus ini, dibentuk dua skenario, hasil akhir skenario pertama berupa rute pada Tabel 3.10 dengan total jarak (*fitness*) sebesar 76,193 km.

Tabel 3.10 Rute Kendaraan Skenario 1

Kendaraan	**Rute**	**Total Jarak**	**Kapasitas**
1	1-2-3-5-0	24,063 km	55
2	4-6-7-0	22,230 km	42
3	8-9-10	29,900 km	52

Sedangkan hasil akhir skenario kedua memperoleh total jarak (*fitness*) sebesar 73,117 km dengan rute setiap kendaraan pada Tabel 3.11.

Tabel 3.11 Rute Kendaraan Skenario 2

Kendaraan	**Rute**	**Total Jarak**	**Kapasitas**
1	1-2-10-4-0	26,498 km	44
2	5-6-7-0	19,647 km	48
3	8-9-3	26,972 km	48

Karena solusi terbaik yang dimaksud adalah solusi dengan total jarak terpendek, maka rute kendaraan terbaik dihasilkan oleh skenario ke dua dengan total jarak sebesar 73,117 km. Sehingga solusi tersebut yang dipilih menjadi solusi optimal dari kasus ini.