Implementasi algoritma tabu *search* pada *vehicle routing problem With Double Time Windows* (VRPDTW)

Ulfa Maulida Rahma[1], Sapti Wahyuningsih[2], dan Lucky Tri Oktoviana[3]
Jurusan Matematika FMIPA Universitas Negeri Malang
E-mail: ulfamaulida159@gmail.com, sapti.wahyuningsih.fmipa@um.ac.id, lucky.tri.fmipa@um.ac.id

Abstrak: *Vehicle Routing Problem with Double Time Windows* (VRPDTW) yaitu VRP dengan terdapat dua batasan *time window*. *Time window* pertama yaitu interval waktu yang digunakan untuk persiapan dan *loading* di depot dan *time window* kedua yaitu interval waktu yang digunakan untuk perjalanan kendaraan dari depot ke *customer* sampai kembali ke depot. Pada artikel ini permasalahan VRPDTW diselesaikan dengan menggunakan algoritma *tabu search*. Pada algoritma tersebut terdapat 3 tahapan penting yaitu tahap inisialisasi, tahap pengembangan, dan tahap pemilihan solusi terbaik. Selanjutnya agar lebih mudah dalam menyelesaikan permasalahan VRPDTW dengan menggunakan algoritma *tabu search*, maka direpresentasikan dalam program komputer menggunakan *Borland Delphi* 7.0. Dalam artikel ini telah diuji coba 6, 15, 20, dan 36 titik.

Kata kunci: *Vehicle Routing Problem with Double Time Windows* (VRPDTW), Algoritma *Tabu Search*, metode *Nearest Neighbour*.

Abstract: *Vehicle Routing Problem with Double Time Windows* (VRPDTW) is VRP with two limit time window. The first time window is the interval of time that is used for the preparation and loading at the depot and the second time window that is the interval of time spent traveling vehicle from the depot to the customer to return to the depot. In this article VRPDTW problems solved using tabu search algorithm. On the algorithm, there are 3 important stages, that is the initialization phase, the development phase, and the phase of selecting the best solution. Furthermore, to make it easier to solve the problems VRPDTW using tabu search algorithm, it is represented in a computer program using Borland Delphi 7.0. In this article was tested 6, 15, 20, and 36 points.

Keywords: Vehicle Routing Problem with Double Time Windows (VRPDTW), Tabu Search Algorithm, Nearest Neighbour method.

Kegiatan distribusi merupakan kegiatan utama yang sering dilakukan oleh suatu perusahaan. Distribusi merupakan salah satu kegiatan pelayanan dalam bidang penyediaan barang atau jasa yang berasal dari perusahaan atau depot untuk dikirim kepada *customer* (pelanggan). Permasalahan yang sering muncul dalam kegiatan distribusi adalah permasalahan transportasi dalam menentukan rute. Salah satu cabang ilmu yang dapat digunakan untuk menyelesaikan permasalahan distribusi adalah teori graph. Salah satu terapan dari teori graph yang banyak digunakan untuk menyelesaikan permasalahan adalah *vehicle routing problem* (VRP), yaitu permasalahan menentukan rute kendaraan yang digunakan untuk mendistribusikan barang ke sejumlah pelanggan dari suatu depot dengan tujuan meminimumkan total biaya perjalanan yang memenuhi kendala-kendala yang diberikan (Yeun, 2008).

Tipe VRP dengan penambahan kendala waktu dikenal dengan VRPTW. Salah satu cabang dari VRPTW adalah VRPDTW (*Vehicle Routing Problem with Double Time Windows*). Batasan *time window* yang terdapat pada VRPDTW ada

1. Ulfa Maulida Rahma adalah mahasiswa jurusan Matematika FMIPA Universitas Negeri Malang
2. Sapti Wahyuningsih adalah dosen jurusan Matematika FMIPA Universitas Negeri Malang
3. Lucky Tri Oktoviana adalah dosen jurusan Matematika FMIPA Universitas Negeri Malang

dua yaitu $[e_0 , l_{ol}]$ yang merupakan *time window* pertama untuk *loading* di depot dan $[e_0 , l_{or}]$ yang merupakan *time window* kedua untuk perjalanan kendaraan dari depot ke *customer* sampai kembali ke depot (Nouaouri dkk, 2011).

Pada *International Journal of Industrial Engineering Research and Development* digunakan salah satu metode metaheuristik yang termasuk dalam algoritma *Tabu Search* untuk menyelesaikan VRPDTW. Penerapan metode *Tabu Search* memerlukan adanya solusi awal. Solusi awal tersebut dapat diperoleh dengan menggunakan metode pendekatan yaitu *Nearest Neighbour*. Solusi awal yang terbentuk berupa rute-rute sementara yang kemudian dioptimalkan dengan menggunakan algoritma *Tabu Search*. Pada penyelesaian menggunakan algoritma *Tabu search*, solusi diperoleh melalui tiga tahap yaitu tahap inisialisasi, tahap pengembangan, dan tahap pemilihan solusi terbaik.

Berdasarkan uraian diatas akan dikaji penggunaan algoritma *tabu search* pada VRPDTW. Hasil yang diharapkan yaitu (1) mengetahui pengertian dan formulasi VRPDTW. (2) mengetahui langkah-langkah algoritma tabu *search* pada VRPDTW. (3) mngimplemntasikan program untuk menyelesaikan VRPDTW dengan algoritma *Tabu Search.*

HASIL DAN PEMBAHASAN

Aldous dan Wilson (2000: 6) mengungkapkan gagasan tentang graph, yaitu himpunan tak kosong dari elemen-elemen yang disebut titik (*vertex*) dan suatu daftar pasangan terurut elemen itu yang disebut sisi (*edge*). Graph yang digunakan untuk masalah pencarian rute kendaraan adalah graph komplit yang telah diberi bobot yaitu bilangan yang berasosiasi pada setiap sisi. Graph komplit adalah graph yang setiap dua titik yang berbeda dihubungkan oleh tepat satu sisi. Graph komplit berbobot digunakan sebagai model dalam VRPDTW.

Definisi Vehicle Routing Problem with Double Time Windows (VRPDTW)

VRPDTW merupakan pengembangan dari VRPTW yaitu permasalahan tentang satu depot sebagai pusat distribusi barang, dengan sejumlah kendaraan berkapasitas tertentu melayani sejumlah pelanggan pada titik-titik lokasi terpisah. Pada VRPDTW terdapat dua batasan time window yaitu *time window* pertama untuk *loading* di depot dan *time window* kedua untuk perjalanan kendaraan dari depot ke *customer* sampai kembali ke depot. Tujuan dari permasalahan VRPDTW adalah meminimalkan jarak tempuh perjalanan untuk melayani seluruh *customer*, tanpa mengabaikan batasan kapasitas kendaraan dan *time window* depot. Desain rute dilakukan sedemikian hingga setiap pelanggan hanya dikunjungi sekali oleh satu kendaraan, dan setiap kendaraan memulai dan mengakhiri rutenya pada depot.

Formulasi dari VRPDTW dengan fungsi tujuan meminimalkan jarak tempuh perjalanan untuk melayani seluruh *customer*, tanpa mengabaikan batasan kapasitas kendaraan dan *time window* depot adalah:

$$\min \sum_{m\in M} \sum_{i\in N} \sum_{j\in N} s_{ij} x_{ij}^{m}$$

Dengan kendala:

1. Setiap titik hanya dikunjungi satu kali dengan satu kendaraan

$$\sum_{m\in M} \sum_{i\in N} x_{ij}^{m} = 1 \;\; \forall j \in N - \{0\}$$

2. Total permintaan dari setiap rute kendaraan yang tidak boleh melebihi kapasitas kendaraan
$$\sum_{i\in N} q_i \sum_{j\in N} x_{ij}^n \le Q \quad \forall m \in M$$
3. Setiap kendaraan yang meninggalkan depot pusat harus kembali lagi ke depot pusat
$$\sum_{j\in N-\{0\}} x_{j0}^m = 1 \ \forall m \in M$$
$$\sum_{i\in N-\{0\}} x_{i0}^m = 1 \ \forall m \in M$$
4. Setiap kendaraan boleh memuat lebih dari satu rute
$$\sum_{i\in N}\sum_{j\in N} x_{ij}^m = \sum_{p\in N}\sum_{q\in N} x_{pq}^m \ \forall i,j,p,q \in N \ , \forall m \in M$$
5. Waktu kendaraan m melayani rute yang dimuatnya berada pada time windows yang ditentukan
$$e_0 \le \sum_{m\in M}\sum_{i\in N}\sum_{j\in N} t_{ij} x_{ij}^m \le l_{or}$$
6. Waktu persiapan untuk rute K harus berada berada pada time windows untuk loading kendaraan
$$e_0 \le \sum_{m\in M}\sum_{i\in N}\sum_{j\in N} DD_m(K) \le l_{ol}$$
7. Batasan nilai
$$x_{ij}^m \in \{0,1\} \ \forall i \in N, j \in N, m \in M$$

Keterangan formulasi:

N	= himpunan dari customer
M	= himpunan darikendaraan yang di gunakan
s_{ij}	= jarak dari i ke j
x_{ij}^m	= status kendaraan m melewati customer i ke customer j
q_i	= jumlah permintaan yang diminta oleh customer ke i
V	= kecepatan rata-rata kendaraan
T_{max}	= waktu maksimal supir bekerja
t_{ij}	= waktu pelayanan dan waktu tempuh dari titik i ke j
Q	= kapasitas maksimal kendaraan
$q_{(K)}$	= permintaan yang dimuat pada rute K
$R(K)$	= rute ke- K
$D(K)$	= total durasi rute ke- K
M	= banyak kendaraan minimal yang dapat digunakan
φ	= himpunan dari rute-rute yang terbentuk
$DD_m(K)$	= jam mulai untuk persiapan rute $R(K)$ yang ditentukan pada kendaraan ke-m
ΔT_m	= sisa waktu yang tersedia untuk kendaraan ke-m
$[e_0 , l_{ol}]$	= *time window* untuk *loading* kendaraan di depot
$[e_0 , l_{or}]$	= *time window* untuk kendaraan melayani customer sampai kembali ke depot ($l_{ol} < l_{or}$).

Algoritma Tabu Search pada VRPDTW

Pencarian solusi dari permasalahan VRPDTW dengan menggunakan algoritma tabu search melalui tiga tahap penyelesaian yaitu tahap inisialisasi, tahap pengembangan, dan tahap pemilihan solusi terbaik. Uraian dari algoritma *tabu search* adalah sebagai berikut.

1. Tahap Inisialisasi

Langkah awal dari tahap inisialisasi adalah menghitung waktu pelayanan dan waktu tempuh setiap titik dengan menggunakan rumus $t_{ij} = (s_{ij}/V) + 2(U \text{ x } q_j)$. Waktu tersebut digunakan untuk menentukan rute sementara dengan menggunakan metode *nearest neighbour*. Dengan langkah-langkahnya adalah sebagai berikut.

Langkah 1 : Tentukan titik 0 sebagai titik awal yang dikunjungi

Langkah 2 : Cari titik selanjutnya yang belum dikunjungi dan mempunyai waktu pelayanan dan waktu tempuh minimum dari titik yang sebelumnya

Langkah 3 : Periksa kendala kapasitas dan batasan waktu. Jika memenuhi kendala kapasitas dan batasan waktu maka rute dapat dibentuk

Langkah 4 : Kembali ke langkah 2 sampai semua titik dikunjungi tepat satu kali.

i. Jika sudah tidak ada lagi titik yang memenuhi salah satu kendala kapsitas dan batasan waktu maka kembali ke langkah 1 untuk membentuk rute baru

ii. Jika semua rute telah dikunjungi tepat satu kali maka proses berhenti.

Dengan enggunakan algoritma penentuan kendraan, rute-rute sementara tersebut satu persatu ditentukan ke kendaraan. Langkah ini bertujuan untuk mengoptimumkan jam kerja sopir sehingga meminimalkan biaya yang dikeluarkan perusahaan untuk menggaji sopir. Adapun langkah-langkah dari algoritma ini adalah sebagai berikut.

Langkah 1 : Diberikan rute, bentuk himpunan φ dari rute kendaraan secara menurun berdasarkan durasi masing-masing rute

Langkah 2 : Hitung jumlah minimal kendaraan yang digunakan

$$M = \left\lceil \frac{\sum_{k=1}^{K} D(K)}{T_{max}} \right\rceil.$$

Langkah 3 : Tentukan satu persatu rute dalam φ pada kendaraan dimulai dari durasi rute paling besar kekecil

Langkah 4 : Letakkan rute pertama pada akhir *time window* $[e_0 , l_{0r}]$

Langkah 5 : Hitung untuk setiap rute $R(K)$ yang diberikan untuk kendaraan m

dengan $DD_m(K) = l_{0l} - \max(0 , (D(K) - (l_{0r} - l_{0l})))$

hitung untuk setiap kendaraan m yang diberikan

dengan $\Delta T_m = \min(T_{max} - D(K) , DD_m(K) - e_0)$

Langkah 6 : Tentukan rute pertama dari sisa $R(K) \in \varphi$ ke kendaraan m, yaitu mempunyai kemungkinan waktu terbesar ΔT_m.

Jika $\Delta T_m \geq D(K)$, tentukan $R(K)$ ke kendaraan m:

Kemudian hitung $DD_m(K) = \Delta T_m - D(K)$ dan $\Delta T_m = DD_m(K)$

Jika tidak tentukan rute $R(K)$ pada $m = m_0 + 1$ dan jadwalkan rute pada akhir *time window* $[e_0 , l_{0r}]$ dan hitung $DD_m(K)$ dan ΔT_m seperti pada langkah 5.

Langkah 7 : Ulangi langkah 6 sampai semua rute terjadwalkan pada kendaraan

Hasil dari tahap inisialisasi kemudian dimasukkan dalam daftar solusi sementara.

2. Tahap Pengembangan

   Dalam tahap ini dilakukan dua langkah yaitu tahap pertukaran posisi titik pada rute yang sudah terbentuk pada tahap inisialisasi dan tahap pemeriksaan kendala. Kendala yang harus dipenuhi adalah kendala kapasitas harus kurang dari atau sama dengan kapasitas maksimum kendaraan, total waktu rute baru kurang dari atau sama dengan waktu maksimal, dan sisa waktu yang dimiliki masing-masing kendaraan harus lebih dari atau sama dengan 0. Pertukaran dilakukan dalam dua tipe yaitu titik dalam rute pada kendaraan yang sama dan titik pada rute antar kendaraan yang bebeda. Rute hasil pertukaran posisi yang memenuhi kendala VRPDTW dengan waktu paling minimum kemudian dimasukkan dalam daftar solusi sementara.

3. Tahap Pemilihan Solusi Terbaik

   Pada tahap ini dilakukan pemilihan solusi terbaik dari daftar solusi sementara. Solusi sementara dengan total waktu minimum yang kemudian menjadi solusi terbaik.

Contoh Penerapan Algoritma *Tabu Search* pada VRPDTW

Suatu perusahaan A akan melayani permintaan beberapa customernya yang tersebar di beberapa kota. Perusahaan tersebut memiliki kendaraan pengiriman dengan kapasitas tiap kendaraan adalah 50 kardus. Maksimal sopir bekerja setiap harinya adalah selama 5 jam. Waktu loading yang ditentukan oleh perusahaan adalah pukul 08.00 sampai pukul 11.30 dan waktu untuk kendaraan melayani *customer* sampai kembali ke depot adalah pukul 08.00 sampai pukul 13.00. Kecepatan rata-rata kendaraan adalah 60 km/jam. Proses *loading*/ *unloading* pada tiap customer adalah 0,0125 jam/kardus. Berikut ini adalah data permintaan tiap customer dan data jarak antara depot ke*customer* serta antar *costomer kecustomer* dalam satuan kilometer.

Tabel 1 Tabel Permintaan Setiap *Customer*

Customer (i)	1	2	3	4	5	6
Banyak permintaan (q_i)	29	25	21	18	16	17

Tabel 2 Tabel Jarak Antar Depot-*Customer* dan Antar *Customer-Customer*

c_{ij}	**0**	**1**	**2**	**3**	**4**	**5**	**6**
0	**0**	14	25	19	24	16	20
1	14	**0**	22	26	27	19	17
2	25	22	**0**	18	26	25	30
3	19	26	18	**0**	14	11	21
4	24	27	26	14	**0**	31	22
5	16	19	25	11	31	**0**	26
6	20	17	30	21	22	26	**0**

***) keterangan:** untuk kolom pertama dan baris prtama, 0 menyatakan depot dan 1,2,...,6 mrumakan *customer*. Jarak dari 0 (depot) ke *customer* 1 adalah 14 Km, dari 0 (depot) ke *customer* 2 adalah 25 Km, dan seterusnya.

Penyelesaian dengan menggunakan algoritma *tabu search*.

1. Tahap inisialisasi
   - Dengan menggunakan rumus pada pembahsan sebelumnya didapatkan waktu pelayanan dan waktu tempuh antar titik seperti pada tabel berikut.

Tabel 3 Tabel Waktu Pelayanan Dan Waktu Tempuh Antar Titik

	0	**1**	**2**	**3**	**4**	**5**	**6**
0	**0**	0,96	1,04	0,84	0,85	0,67	0,76
1	0,23	**0**	0,99	0,96	0,90	0,72	0,71
2	0,42	1,09	**0**	0,83	0,88	0,82	0,93
3	0,32	1,16	0,93	**0**	0,68	0,58	0,78
4	0,40	1,18	1,06	0,76	**0**	0,92	0,79
5	0,27	1,04	1,04	0,71	0,97	**0**	0,86
6	0,33	1,01	1,13	0,88	0,82	0,83	**0**

***) keterangan:** untuk kolom pertama dan baris prtama, 0 menyatakan depot dan 1,2,...,6 mrumakan *customer*. Waktu pelayanan dan waktu tempuh dari 0 (depot) ke *customer* 1 adalah 0,96 jam, dari 0 (depot) ke *customer* 2 adalah 1,04 jam, dan seterusnya.

- Rute sementara

  Dengan menggunakan metode *nearest neighbour* :

  Langkah 1 : Tentukan titik 0 sebagai titik awal yang dikunjungi

  Langkah 2 : Cari titik selanjutnya yang belum dikunjungi dan mempunyai waktu pelayanan dan waktu tempuh minimum dari titik yang sebelumnya yaitu titik 0.
  Diperoleh titik 5 dengan $t_{05} = 0{,}67$

  Langkah 3 : Periksa kendala kapasitas dan batasan waktu
  Karena memenuhi kendala kapasitas dan batasan waktu maka rute terbentuk $R(1) = [0 - 5 - 0]$

  Langkah 4 : Kembali ke langkah 2 sampai semua titik dikunjungi tepat satu kali

  Dengan demikian diperoleh rute sementara yaitu:

  1. R(1)=[0-5-3-0] dengan $D(1) = 1{,}70$ dan $q_{(2)} = 37$
  2. R(2)=[0-6-4-0] dengan $D(2) = 1{,}98$ dan $q_{(2)} = 35$
  3. R(3)=[0-1-0] dengan $D(3) = 1{,}19$ dan $q_{(3)} = 29$
  4. R(4)=[0-2-0] dengan $D(4) = 1{,}46$ dan $q_{(4)} = 25$

- Penentuan rute ke kendaraan

  Langkah 1 : Diberikan rute, bentuk himpunan φ dari rute kendaraan secara menurun berdasarkan durasi masing-masing rute
  $$\varphi = \{R(2), R(1), R(4), R(3)\}$$

  Langkah 2 : Hitung jumlah minimal kendaraan yang digunakan
  $$M = \left\lceil \frac{\sum_{k=1}^{K} D(K)}{T_{max}} \right\rceil = \left\lceil \frac{D(1)+D(2)+D(3)+D(4)}{5} \right\rceil$$
  $$= \left\lceil \frac{1{,}70 + 1{,}98 + 1{,}19 + 1{,}46}{5} \right\rceil = \left\lceil \frac{6{,}33}{5} \right\rceil = \lceil 1{,}27 \rceil = 2$$

  Langkah 3 : Tentukan satu persatu rute dalam φ pada kendaraan dimulai dari durasi rute paling besar kekecil. Dimulai dari 2 rute pertama yaitu $R(2)$ dan $R(1)$

  Langkah 4 : Letakkan rute pertama yaitu $R(2)$ dengan $D(2) = 1{,}98 < \Delta T_1$ pada akhir *time window* $[e_0\ , l_{0r}]$ pada kendaraan 1 dan R(1) dengan $D(1) = 1{,}70 < \Delta T_2$ pada kendaraan 2.

Langkah 5 : Hitung untuk setiap rute $R(K)$ yang diberikan untuk kendaraan m dengan $DD_m(K) = l_{0l} - \max(0, (D(K) - (l_{0r} - l_{0l})))$ dan hitung untuk setiap kendaraan m yang diberikan dengan $\Delta T_m = \min(T_{max} - D(K), DD_m(K) - e_0)$.

$DD_1(2) = l_{0l} - \max(0, D(2) - (l_{or} - l_{0l}))$
$= 11{,}5 - \max(0, (1{,}98 - (13 - 11{,}5)))$
$= 11{,}5 - 0{,}48 = 11{,}02$ (artinya persiapan untuk $R(2)$ yang ditentukan pada kendaraan 1 dimulai pada jam 11 lebih 0,02 jam yaitu pukul 11.01)

$\Delta T_1 = \min(T_{max} - D(2), DD_1(2) - e_0)$
$= \min((5 - 1{,}98), (11{,}02 - 8))$
$= \min(3{,}02, 3{,}02) = 3{,}02$

$DD_2(1) = l_{0l} - \max(0, D(1) - (l_{or} - l_{0l}))$
$= 11{,}5 - \max(0, (1{,}70 - (13 - 11{,}5)))$
$= 11{,}5 - 0{,}20 = 11{,}30$ (pukul 11.18)

$\Delta T_2 = \min(T_{max} - D(1), DD_1(1) - e_0)$
$= \min((5 - 1{,}70), (11{,}30 - 8))$
$= \min(3{,}3, 3{,}3) = 3{,}3$

Langkah 6 : Tentukan rute pertama dari sisa $R(K) \in \varphi$ ke kendaraan m, yaitu mempunyai kemungkinan waktu terbesar ΔT_m.
Sisa rute $\varphi = \{R(4), R(3)\}$
Karena $\Delta T_2 > \Delta T_1$ dan rute pertama dari sisa φ adalah $R(4)$ dengan $D(4) = 1{,}46 < \Delta T_2$ maka tentukan $R(4)$ pada kendaraan 2.
dimana,

$DD_2(4) = DD_2(1) - D(4)$
$= 11{,}30 - 1{,}46$
$= 9{,}84$ (pukul 09.50)

$\Delta T_2 = DD_2(4) - e_0$
$= 9{,}84 - 8$
$= 1{,}84$

Langkah 7 : Ulangi langkah 6 sampai semua rute terjadwalkan pada kendaraan

Dengan demikian diperoleh rute beserta kendaraannya yaitu:

Tabel 4 Solusi Sementara Tahap Inisialisasi

Kendaraan ke-	$\boldsymbol{R(K)}$	$DD_m(K)$	ΔT_m	$\boldsymbol{q}_{(K)}$	$\boldsymbol{D(K)}$	**total**
1	$R(2) = 0-6-4-0$	11.01	1,83	35	1,98	3,17
	$R(3) = 0-1-0$	09.49		29	1,19	
2	$R(1) = 0-5-3-0$	11.18	1,84	37	1,70	3,16
	$R(4) = 0-2-0$	09.50		25	1,46	
Total				**126**		**6,33**

Dari Tabel 4 diperoleh kendaraan 1 memuat R(3)= 0-1-0 dengan waktu persiapan dimulai pada 9,83 (pukul 09.49) kemudian R(2)= 0-4-6-0 dengan waktu persiapan dimulai pada 11,03 (pukul 11.01), dengan sisa waktu yang dimiliki oleh kendaraan 1 adalah 1,83 jam. Kendaraan 2 memuat R(4)= 0-5-0 dengan waktu persiapan dimulai pada 9,88 (pukul

09.50), kemudian R(1)= 0-2-3-0 dengan waktu persiapan dimulai pada 11.30 (pukul 11.18), dengan sisa waktu yang dimiliki oleh kendaraan 2 adalah 1,84 jam. Total waktu yang ditempuh oleh kendaraan 1 dan kendaraan 2 adalah 6,33 jam.

2. Tahap pengembangan

Dari solusi yang dihasilkan pada tahap inisialisasi dilakukan pertukaran posisi antar titik. Rute baru yang diperoleh dari hasil pertukaran posisi antar titik kemudian diperiksa kendala kapasitas dan kndala waktu. Rute dengan total waktu paling minimum kemudian dimasukkan dalam daftar solusi sementara.

- **Tabel 5 Solusi Sementara Pertukaran Titik dalam Satu Kendaraan**

Kendaraan ke-	$\boldsymbol{R(K)}$	$DD_m(K)$	ΔT_m	$\boldsymbol{q_{(K)}}$	$\boldsymbol{D(K)}$	**total**
1	$R(2) = 0-4-6-0$	11.01	1,84	35	1,97	3,16
	$R(3) = 0-1-0$	09.50		29	1,19	
2	$R(1) = 0-2-3-0$	10.48	1,87	46	2,19	3,13
	$R(4) = 0-5-0$	09.52		16	0,94	
Total				**126**		**6,29**

- **Tabel 6 Solusi Sementara Pertukaran Titik dalam Kendaraan yang berbeda**

Kendaraan ke-	$\boldsymbol{R(K)}$	$DD_m(K)$	ΔT_m	$\boldsymbol{q_{(K)}}$	$\boldsymbol{D(K)}$	**total**
1	$R(2) = 0-2-4-0$	10.40	1,53	43	2,32	3,51
	$R(3) = 0-1-0$	09.31		29	1,19	
2	$R(1) = 0-5-3-0$	10.18	2,21	37	1,70	2,79
	$R(4) = 0-6-0$	10.12		17	1,09	
Total				**126**		**6,30**

3. Tahap pemilihan solusi terbaik

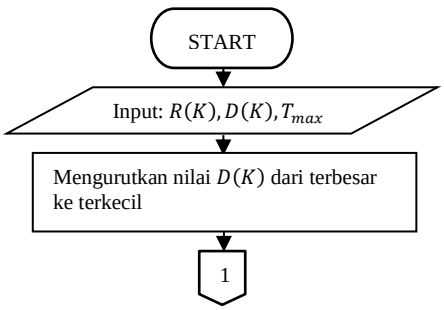
Dari solusi sementara yang dihasilkan oleh tahap inisialisasi dan tahap pengembangan, solusi dengan total waktu minimum merupakan solusi optimum.

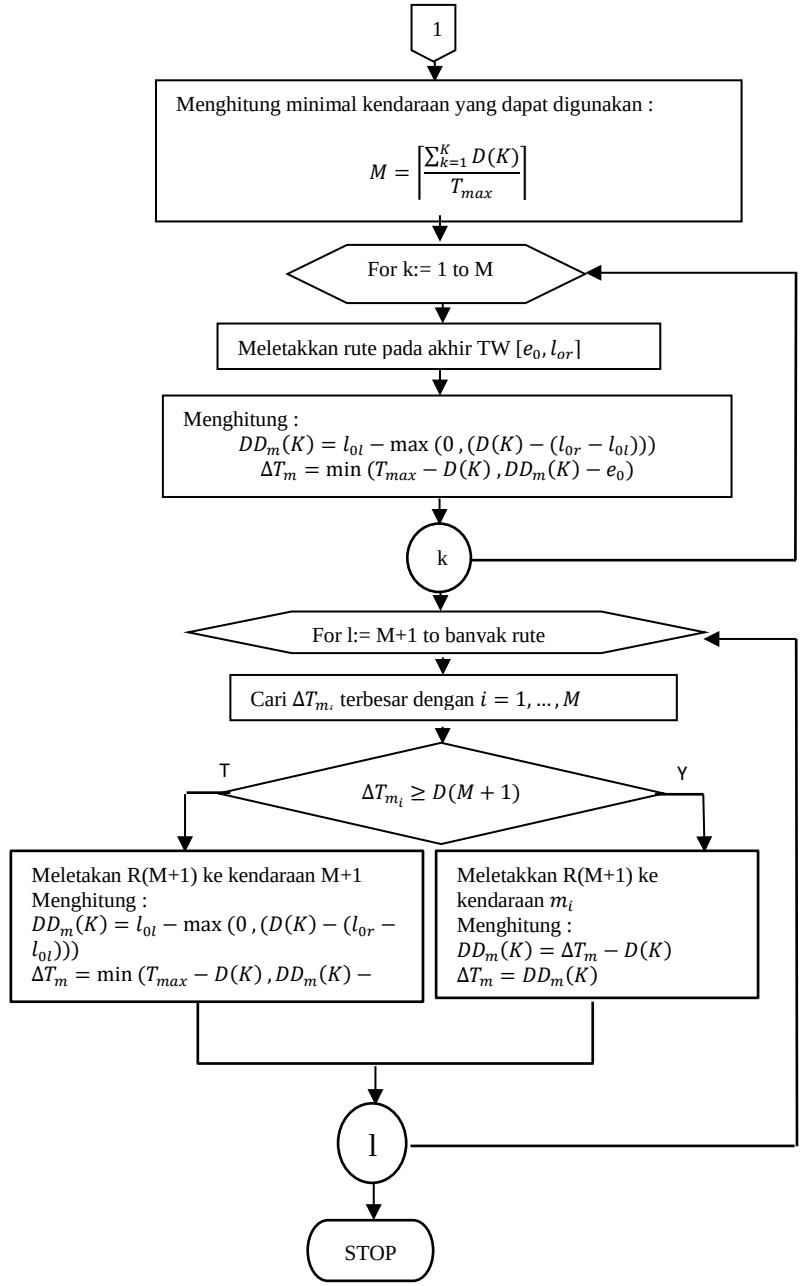
Tabel 7 Solusi Optimum

Kendaraan ke-	$\boldsymbol{R(K)}$	$DD_m(K)$	ΔT_m	$\boldsymbol{q_{(K)}}$	$\boldsymbol{D(K)}$	**total**
1	$R(2) = 0-4-6-0$	11.01	1,84	35	1,97	3,16
	$R(3) = 0-1-0$	09.50		29	1,19	
2	$R(1) = 0-2-3-0$	10.48	1,87	46	2,19	3,13
	$R(4) = 0-5-0$	09.52		16	0,94	
Total				**126**		**6,29**

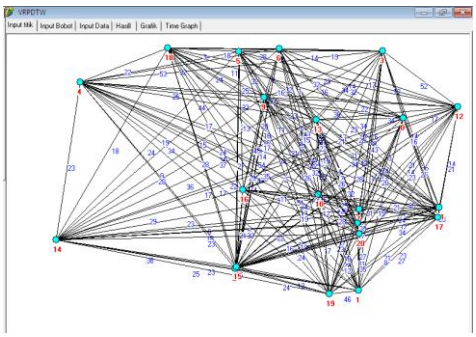
Implementasi Program VRPDTW.

Kendala yang menonjol pada VRPDTW adalah kendala dua batasan *time windows*. Pada penyelesaian VRPDTW dengan memperhatikan kendala tersebut terletak pada tahap inisialisasi yaitu mementukan rute ke kendaraan. Tahap ini bermaksud mengatur jam mulai persiapan semua rute agar berada pada interval waktu *time window* pertama. Berikut merupakan *flowchart* tahap penentuan rute ke kendaraan.

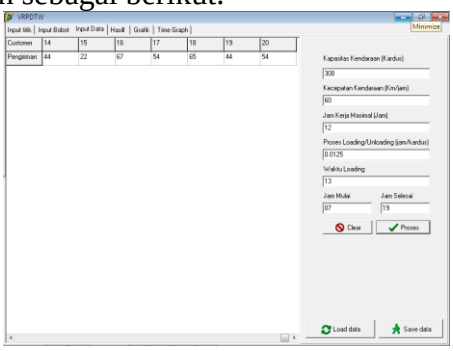




Pada program VRPDTW telah dilakukan uji coba dengan 15, 20, dan 36 titik. salah satu uji coba dengan 20 titik adalah sebagai berikut.

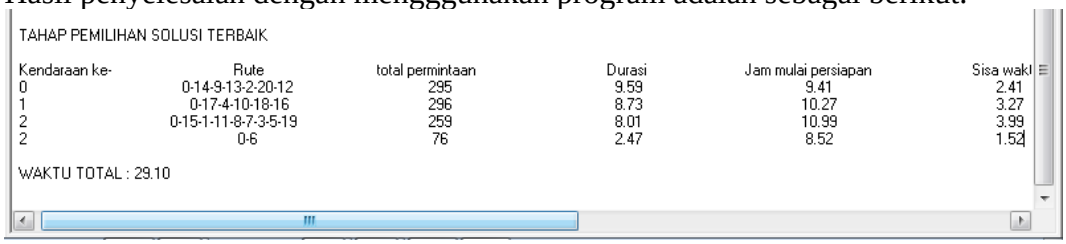


Gambar 1 Graph Contoh 20 titik



Gambar 2 Input Data Contoh 20 titik

Hasil penyelesaian dengan mengggunakan program adalah sebagai berikut.

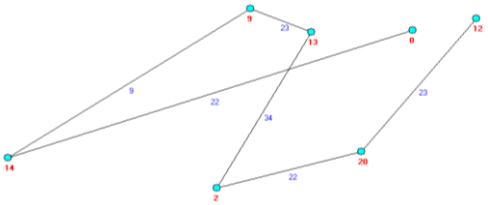


TAHAP PEMILIHAN SOLUSI TERBAIK

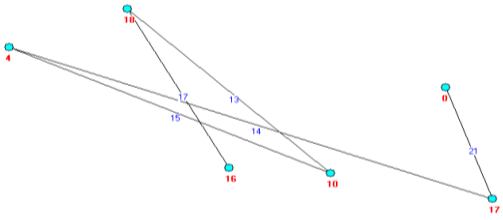
Kendaraan ke-	Rute	total permintaan	Durasi	Jam mulai persiapan	Sisa wakt
0	0-14-9-13-2-20-12	295	9.59	9.41	2.41
1	0-17-4-10-18-16	296	8.73	10.27	3.27
2	0-15-1-11-8-7-3-5-19	259	8.01	10.99	3.99
2	0-6	76	2.47	8.52	1.52

WAKTU TOTAL : 29.10

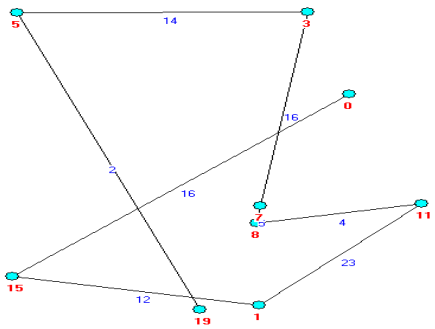
Hasil program menunjukkan bahwa contoh permasalahan dengan 20 titik jika diselesaikan dengan program VRPDTW dapat menggunakan 3 kendaraan dimana kendaraan 1 memuat rute 0-14-9-13-2-20-12-0 dengan total permintaan 295, durasi 9,59 dan memulai persiapan pada 9,41 (pukul 09.24) dengan sisa waktu yang dimiliki kendaraan 1 adalah 2,41 jam. Kendaraan 2 memuat rute 0-17-4-10-18-16-0 dengan total permintaan 296, durasi 8,73 dan memulai persiapan pada 10,27 (pukul 10.16) dengan sisa waktu yang dimiliki kendaraan 2 adalah 3,27 jam. Kendaraan 3 memuat rute 0-6-0 dengan total permintaan 76, durasi 2,47 dan memulai persiapan pada 8,52 (pukul 08.31) kemudian dilanjutkan rute 0-15-1-11-8-7-3-5-19-0 dengan total permintaan 259, durasi 8,01 dan memulai persiapan pada 10,99 (pukul 10.59) dengan sisa waktu yang dimiliki kendaraan 3 adalah 1,52 jam. Berikut adalah graph rute hasil penyelesaian.



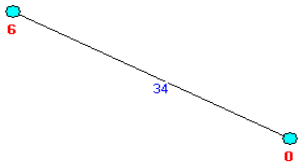
Gambar 3 Graph rute 0-14-9-13-2-20-12-0



Gambar 4 Graph rute 0-17-4-10-18-16-0



Gambar 5 Graph rute 0-15-1-11-8-7-3-5-19-0



Gambar 6 Graph rute 0-17-4-10-18-16-0

PENUTUP

Kesimpulan

Dari pembahasan sebelumnya telah dibahas mengenai pengertian dan formulasi VRPDTW serta penyelesaian VRPDTW dengan menggunakan algoritma *tabu search*. Permasalahan VRPDTW merupakan permasalahan tentang satu depot sebagai pusat distribusi barang, dengan sejumlah kendaraan berkapasitas tertentu melayani sejumlah pelanggan pada titik-titik lokasi terpisah. VRPDTW merupakan salah satu pengembangan dari VRPTW dengan dua *time windows* yaitu *time window* pertama untuk *loading* di depot dan *time window*

kedua untuk perjalanan kendaraan dari depot ke *customer* sampai kembali ke depot.

Sedangkan Implementasi algoritma *tabu Search* pada VRPDTW terbagi menjadi 3 tahap penting, yaitu tahap inisialisasi, tahap pengembangan, dan tahap pemilihan solusi terbaik. Tahap inisialisasi mencangkup proses penghitungan waktu pelayanan dan waktu tempuh, pencarian rute sementara dengan bantuan metode *nearest neighbour*, serta penentukan rute ke kendaraan. Tahap pengembangan merupakan tahap dimana hasil solusi yang terbentuk pada tahap inisialisasi dieksplorasi lebih dalam dengan cara menukarkan posisi tiap pasang titik baik dalam satu kendaraan maupun antar kendaraan. Tahap pemilihan solusi terbaik merupakan tahap akhir dari algoritma *tabu search* dimana pada tahap ini mencangkup proses pemilihan solusi terbaik dari solusi-solusi sementara dalam tabu *list*. Solusi yang dipilih harus memperhatikan kendala kapasitas dan kapasitas waktu yang telah ditetapkan.

Dari beberapa contoh permasalahan yang diberikan, diperoleh kesimpulan bahwa program ini dapat mempermudah pengimplementasian algoritma *tabu search* untuk menyelesaikan VRPDTW. Langkah awal penggunaan program ini yaitu menggambarkan titik yang mewakili bayaknya *customer*, jarak antar titik (*customer*), permintaan tiap *customer*, kapasitas maksimal kendaraan, kecepatan rata-rata kendaraan, waktu maksimal sopir bekerja, waktu proses pelayanan (*loading/unloading*), waktu mulai persiapan, batas waktu *loading*, waktu selesai pelayanan. Setelah itu klik *button* proses, maka hasil solusi optimum akan ditampilkan pada memo.

Saran

Algoritma *Tabu Search* dapat digunakan sebagai alternatif dalam menyelesaikan permasalahan VRPDTW. Dapat dimungkinkan untuk menganalisa algoritma lain sehingga dapat menjadi pembanding, dan mendapatkan solusi alternatif yang lebih optimum dan cepat. Algoritma lain yang dapat digunakan untuk menyelesaikan permasalahan VRPDTW diantaranya algoritma *Genetika*, algoritma *Clark and wright*, algoritma ACS atau algoritma lainnya.

DAFTAR PUSTAKA

Aldous, Joan and Robin J. Wilson. 2000. *Graph and Application*. Great Britain : Springer.

Nouaouri Issam, Goncalves Gilles, & Jolly Daniel. (2011). *A Hybrid Tabu Search for a Vehicle Routing Problem with Double Time Windows for The Depot and Multiple Use of Vehicle : Case of Fuel Delivery*. *International Journal of Industrial Engineering Research and Development. ISSN 0976 – 6987, Vol. 2, pp 91-105*.

Yeun, C. & Zirour, M. 2008. Vehicle Routing Problem: Models and Solution. *Journal Of Quality Measurement and Analysis 4(1): ISSN 1823-5670, pp 205-218*.

Zalynda, Putri Mety. 2013. *Memecahkan Permasalahan Vehicle Routing Problem with Time Window melalui metode Insertion Heuristic (Study Kasus : PT X Wilayah Bandung)*. Seminar Nasional. *ISSN 2337-4349.*