

Implementasi algoritma *tabu search* pada *split load vehicle routing problem with simultaneous delivery and pickup* (SLVRPSDP)

Irma Kurniawati¹, Sapti Wahyuningsih², dan Darmawan Satyananda³
Universitas Negeri Malang

**E-mail: irmakurnia_16@yahoo.com, sapti.wahyuningsih.fmipa@um.ac.id,
darmawan.satyananda.fmipa@um.ac.id**

ABSTRAK: *Vehicle Routing Problem with Simultaneous Delivery and Pickup* (VRPSDP) merupakan salah satu varian dari VRP di mana *customer* membutuhkan pengiriman dan pelayanan pengambilan barang secara simultan atau bersamaan dan permintaan *customer* tidak boleh melebihi kapasitas kendaraan. Pada kehidupan nyata terdapat kasus dimana permintaan *customer* boleh melebihi kapasitas kendaraan. *Split Load Vehicle Routing Problem with Simultaneous Delivery and Pickup* (SLVRPSDP) digunakan untuk menentukan sejumlah rute dengan jarak yang minimum yang berawal dan berakhir di satu depot dimana permintaan *customer* boleh melebihi kapasitas kendaraan. SLVRPSDP merupakan multi atribut dari VRP dengan dua kendala yang digabungkan yaitu pengiriman dan pengambilan secara bersamaan dan pemisahan beban. Pada artikel ini, permasalahan SLVRPSDP diselesaikan menggunakan algoritma *Tabu Search*. Algoritma *Tabu Search* memiliki empat tahap yaitu tahap *split load*, tahap inisialisasi, tahap pengembangan dan tahap pemilihan solusi yang optimum. Selanjutnya, agar mudah dalam menyelesaikan permasalahan SLVRPSDP dengan menggunakan Algoritma *Tabu Search*, maka direpresentasikan dalam program komputer menggunakan *Borland Delphi 7.0*. Pada program telah disimulasi untuk data dengan 9 titik, 26 titik, 32 titik dan 51 titik.

Kata kunci: Varian VRP, SLVRPSDP, Algoritma *Tabu Search*

ABSTRACT: *Vehicle Routing Problem with Simultaneous Delivery and Pickup* (VRPSDP) is one of the variants of the VRP in which the customer requires the delivery and pickup services simultaneously or concurrently and customer demand can not exceed the capacity of the vehicle. In the real life there are cases where customer demand can exceed the capacity of the vehicle. *Split Load Vehicle Routing Problem with Simultaneous Delivery and Pickup* (SLVRPSDP) is used to determine the number of routes with minimum distance that begins and ends at the depot where customer demand may exceed the capacity of the vehicle. SLVRPSDP is a multi-attribute of VRP with two constraints, namely the delivery and retrieval are combined simultaneously and the separation load. In this article, SLVRPSDP problems solved using the *Tabu Search* algorithm. *Tabu Search* algorithm has four phases: a *split load*, initialization phase, the selection phase and the development phase of the optimum solution. Furthermore, to make it easier to

-
1. Irma Kurniawati adalah mahasiswa Jurusan Matematika FMIPA Universitas Negeri Malang
 2. Sapti Wahyuningsih adalah dosen Jurusan Matematika FMIPA Universitas Negeri Malang
 3. Darmawan Satyananda adalah dosen Jurusan Matematika FMIPA Universitas Negeri Malang

solve the problems SLVRPSDP using Tabu Search Algorithm, it is represented in a computer program using Borland Delphi 7.0. This program has been simulated for data with 9 point, 26, point, 32 point and 51 point.

Key Word: VRP variants, SLVRPSDP, *Tabu Search* Algorithm

Dalam suatu perusahaan yang bergerak dalam bidang penyediaan barang dan jasa sistem atau saluran distribusi merupakan salah satu aspek yang sangat penting. Salah satu cabang ilmu yang dapat digunakan untuk menyelesaikan permasalahan distribusi adalah Teori Graph. Penggunaan teori graph yang tepat akan membuat permasalahan semakin mudah untuk dianalisa sehingga diperoleh penyelesaian yang tepat. Karena permasalahan tersebut akan dibuat lebih sederhana dengan mengambil aspek yang penting dan mengabaikan aspek lain yang tidak diperlukan. Salah satu terapan dari teori graph yang banyak digunakan untuk menyelesaikan permasalahan adalah *vehicle routing problem* (VRP), yaitu pencarian rute kendaraan dengan jarak tempuh yang minimum untuk melayani sejumlah *customer* (Yeun dan Zirour, 2008).

Salah satu varian dari VRP adalah *Vehicle Routing Problem with Simultaneous Delivery and Pickup* (VRPSDP), di mana *customer* membutuhkan pengiriman dan pelayanan pengambilan barang secara simultan atau bersamaan (kendaraan membawa pengiriman sekaligus mengambil barang *customer*) dengan permintaan customer tidak boleh melebihi kapasitas kendaraan. *Split Load Vehicle Routing Problem with Simultaneous Delivery and Pickup* (SLVRPSDP) merupakan multi atribut dari VRP dengan dua kendala yang digabungkan yaitu pengiriman dan pengambilan secara bersamaan dan pemisahan beban (Yin, dkk: 2013). SLVRPSDP digunakan untuk menentukan sejumlah rute yang berawal dan berakhir di satu depot. Rute tersebut dilalui sejumlah kendaraan untuk melayani *customer* dengan permintaan pengiriman dan pengambilan barang secara bersamaan dimana permintaan boleh melebihi kapasitas kendaraan. Salah satu algoritma yang dapat digunakan untuk menyelesaikan permasalahan ini adalah algoritma *Tabu Search*.

Berdasarkan uraian diatas akan dikaji penguunaan algoritma Tabu Search pada *Split Load Vehicle Routing Problem with Simultaneous Delivery and Pickup*.

HASIL YANG DIHARAPKAN

1. Mengetahui langkah- langkah dan implementasi algoritma *Tabu Search* dalam menyelesaikan permasalahan SLVRPSDP.
2. Mengetahui implementasi algoritma *Tabu Search* dalam menyelesaikan permasalahan SLVRPSDP dengan menggunakan bahasa pemrograman *Borland Delphi 7.0* dan contoh aplikasinya.

HASIL DAN PEMBAHASAN

- ***Split Load Vehicle Routing Problem with Simultaneous Delivery and Pickup***
Permasalahan SLVRPSDP didasarkan pada suatu graph $G = (V, E)$ dengan V adalah himpunan titik- titik yang mewakili depot dan beberapa *customer* dan $E \subseteq V \times V$ adalah himpunan sisi. Sehingga SLVRPSDP dapat didefinisikan sebagai suatu

himpunan n customer yang dilayani oleh kendaraan k , yang masing-masing mempunyai kapasitas Q . Setiap customer memiliki permintaan pengiriman barang sejumlah d_i , dan permintaan pengambilan barang sejumlah p_i dengan $i \in V \setminus \{0\}$, dimana nilai d_i atau p_i bisa lebih besar dari Q . Setiap kendaraan berangkat dari depot dan kembali lagi ke depot. Sehingga solusi yang dihasilkan dari permasalahan SLVRPSDP adalah himpunan rute dimana *bulk-customer* boleh dikunjungi lebih dari satu kali dengan kendaraan yang berbeda sedangkan customer yang permintaan kurang dari kapasitas hanya dikunjungi satu kali, dan pada setiap pelayanan oleh satu kendaraan, total permintaan customer tidak melebihi kapasitas kendaraan serta total jarak pada rute yang ditempuh kendaraan minimum.

Pemodelan SLVRPSDP dengan mengembangkan pemodelan matematika VRPSDP (Cao & Lai) dengan penambahan kendala yang bersesuaian dengan SLVRPSDP adalah sebagai berikut:

- a. $\min F = \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} c_{ij} x_{ijk}$
Kendala ini untuk meminimalisasi total jarak dari rute yang terbentuk
- b. $\sum_{i \in V} \sum_{j \in V \setminus \{0\}} x_{ijk} = 1, i, j \notin C, k \in K$
Batasan tersebut menjamin bahwa Setiap customer hanya dikunjungi satu kali dan hanya oleh satu kendaraan, kecuali *bulk-customer*.
- c. $\sum_{i \in V} \sum_{j \in V \setminus \{0\}} x_{ijk} > 1, i, j \in C, k \in K$
Batasan ini memastikan bahwa Untuk *bulk-customer* boleh dikunjungi lebih dari satu kali dengan kendaraan yang berbeda.
- d. $\sum_{i \in V} x_{ihk} = \sum_{j \in V} x_{hjk}, h \in V \setminus \{0\}, k \in K$
Batasan ini memastikan bahwa setiap kendaraan tiba di lokasi pengiriman tertentu akan berangkat dari tempat yang sama.
- e. $\sum_{i \in V \setminus \{0\}} p_i \sum_{j \in V} x_{ij}^k \leq Q, \forall k \in K$
 $\sum_{i \in V \setminus \{0\}} d_i \sum_{j \in V} x_{ij}^k \leq Q, \forall k \in K$
Kendala ini menjamin bahwa total pengambilan dan pengiriman barang pada tiap customer dalam satu rute tidak boleh melebihi kapasitas kendaraan
- f. $\gamma_k \leq Q, k \in K$
Kendala ini memastikan berat beban kendaraan awal tidak dapat melebihi kapasitas maksimum beban kendaraan.
- g. $\sum_{i \in V} \sum_{j \in V \setminus \{0\}} c_{ijk} x_{ijk} \leq L, k \in K$
Kendala ini menjamin bahwa jarak perjalanan setiap kendaraan tidak dapat melebihi jarak perjalanan maksimum.
- h. $\max c_{0jk} \leq \frac{L}{2}, k \in K$
Batasan ini memastikan bahwa jarak setiap customer terhadap depot tidak dapat melebihi setengah dari jarak maksimum yang dapat ditempuh setiap kendaraan.
- i. Batasan nilai: Dalam model ini, simbol x_{ijk} mewakili 0-1 variabel tergantung pada situasi dari tempat distribusi dikunjungi oleh kendaraan k . Nilai $x_{ijk} = 1$ menunjukkan bahwa situs distribusi j dikunjungi oleh kendaraan k setelah tempat distribusi i , dan $x_{ijk} = 0$ sebaliknya.

Keterangan simbol pada pemodelan:

- K : himpunan kendaraan pada depot dengan $K = \{1,2,3, \dots, k\}$
 V : himpunan titik
 L : jarak maksimal yang dapat ditempuh kendaraan
 c_{ij} : jarak antara depot dan customer dan antar customer
 d_i : permintaan pengiriman *customer i*
 p_i : permintaan pengambilan *customer i*
 x_{ij} : konstanta, yang bernilai 1 jika kendaraan k mengunjungi *customer j* setelah *customer i*, dan akan bernilai 0 jika sebaliknya.

• ALGORITMA *TABU SEARCH* PADA SLVRPSDP

Berdasarkan karakteristik utama permasalahan SLVRPSDP dengan pengiriman dan pengambilan secara bersamaan, maka algoritma tabu search digunakan untuk memperoleh solusi yang masuk akal. Karena itu keoptimalan solusi bergantung pada kualitas solusi awal (Yin, dkk: 2013).

Langkah- langkah algoritma Tabu Search secara umum adalah sebagai berikut (Singer: 2008):

Menurut Singer (2008) secara umum prosedur dari *tabu search* adalah sebagai berikut:

- Langkah 1: Memilih sebuah solusi inisial i dalam S melalui tahap inialisasi.
- Langkah 2: Membuat daftar kandidat solusi sementara yang telah diletakkan dalam tabu.
- Langkah 3: Evaluasi setiap kandidat solusi tersebut pada tahap pengembangan.
- Langkah 4: Periksa status kandidat hasil evaluasi dengan isi *tabu*.
- Langkah 5: Mengupdate *tabu* dengan kandidat terbaik.
- Langkah 6: Jika diperoleh solusi paling optimal maka berhenti. Jika tidak ulangi langkah 2.

Pada artikel ini sebelum penyelesaian dengan menggunakan algoritma tabu search, terdapat tahap pemisahan beban yang dikenal dengan istilah tahap *Split Load*. Sehingga berdasarkan hal tersebut terdapat empat tahap untuk penyelesaian permasalahan ini.

1. Tahap *Split Load*

Tahap *Split Load* bertujuan untuk membagi beban kendaraan yang melayani *customer* dengan permintaan yang jumlahnya melebihi kapasitas kendaraan. Permintaan *customer* dianggap sebagai beban kendaraan. Adapun rumus untuk pembagian tersebut adalah

$$\frac{d_i}{Q} \text{ dan } \frac{p_i}{Q}$$

Pada tahap ini solusi yang dihasilkan adalah dua daftar baru dari permintaan *customer*.

2. Tahap Inialisasi

Tahap inialisasi digunakan untuk menentukan solusi awal dari permasalahan SLVRPSDP yaitu berupa rute sementara. Metode yang digunakan untuk tahap inialisasi adalah Metode *Insertion Heuristic* yang

diambil dari jurnal yang berjudul *Vehicle Routing Problem with Simultaneous Delivery and Pickup* (Lai,dkk). Adapun langkah- langkah metode *Insertion Heuristic* adalah sebagai berikut:

Tahap 1: Pembentukan rute

Langkah 1: Bentuk rute parsial

Langkah 2: Hitung selisih antara permintaan pengiriman dan permintaan pengambilan.

Langkah 3: Perluasan rute

Langkah 4: Menghilangkan *customer* yang telah terpilih pada perluasan rute

Tahap 2 : Penyisipan Pickup

Rute yang dihasilkan dimasukkan ke dalam daftar tabu.

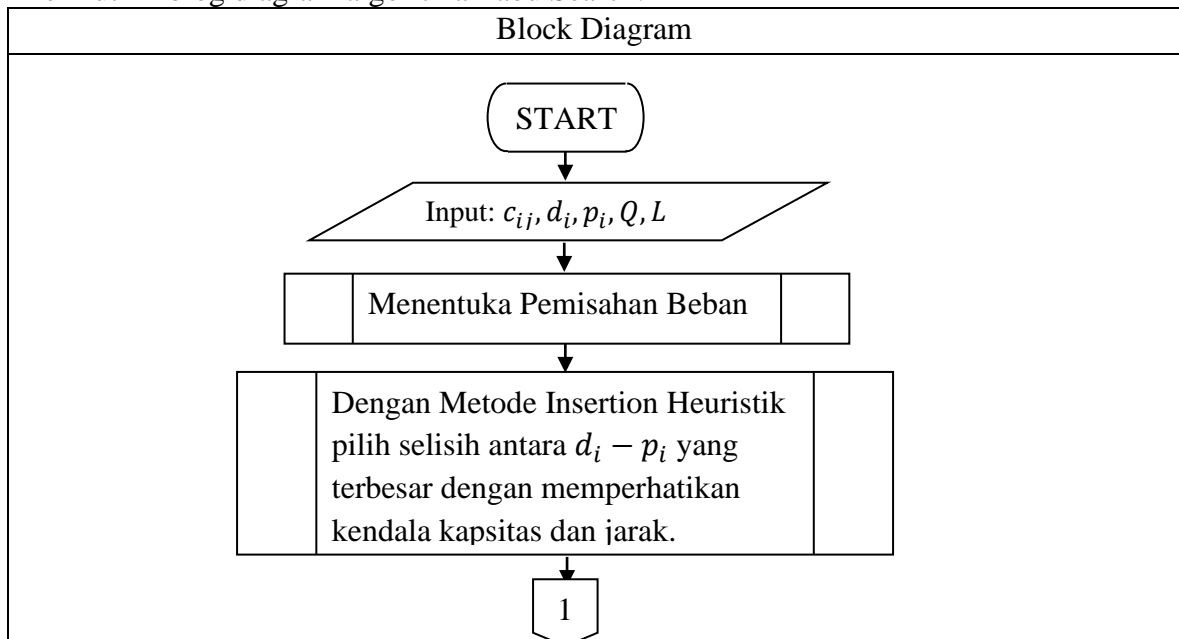
3. Tahap Pengembangan

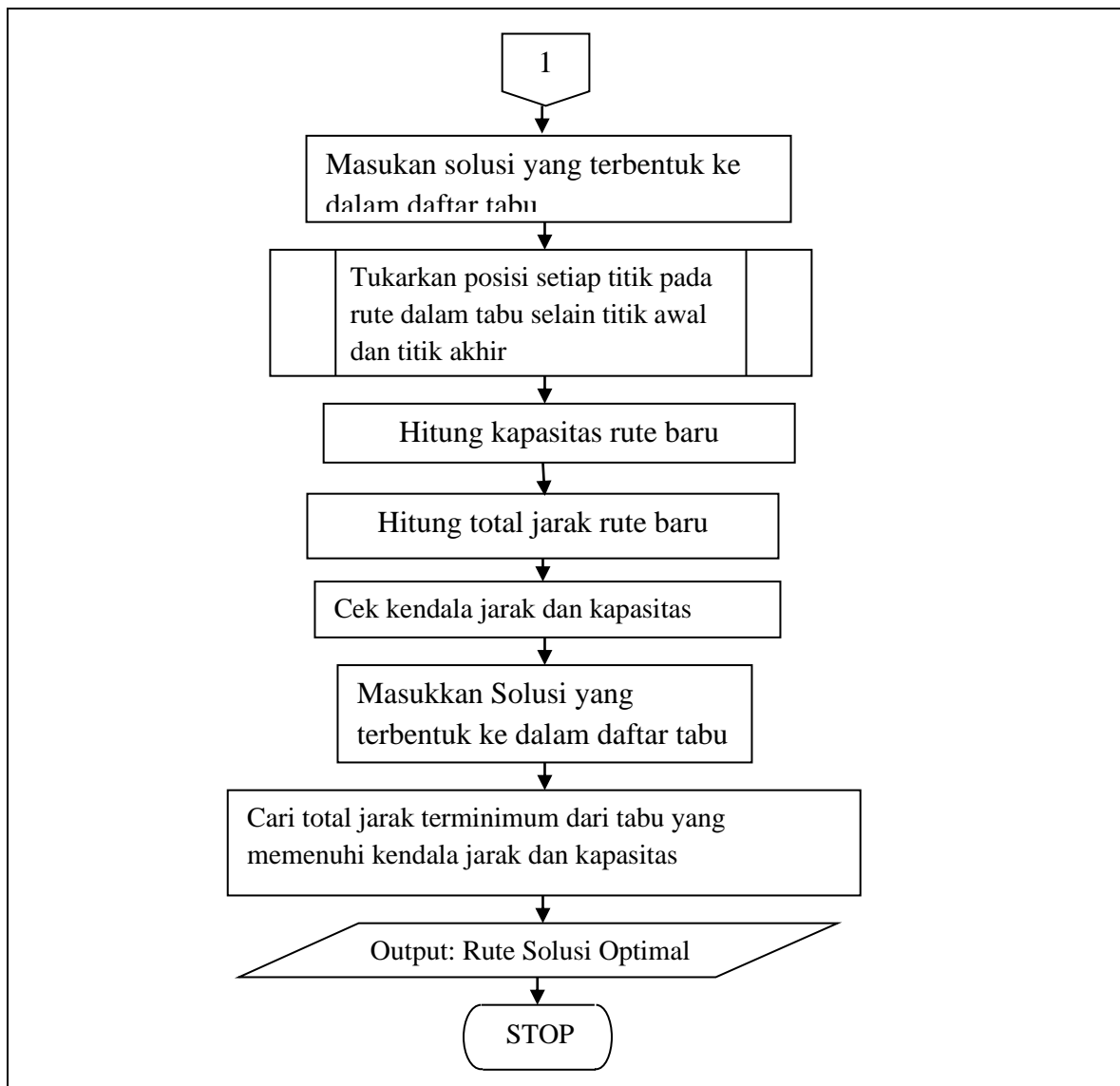
Tahap ini diperlukan untuk menghasilkan solusi yang optima, rute yang dihasilkan akan dibandingkan dengan tahap inisialisasi sebelumnya. Dalam tahap ini hasil inisialisasi dimasukkan sebagai inputan awal tahap .Tahap pengembangan ini terdiri dari dua langkah yaitu tahap pertukaran titik dan tahap pemeriksaan kendala. Semua rute yang layak dimasukkan dalam daftar tabu..

4. Tahap Pemilihan Solusi Terbaik.

Pada daftar tabu dipilih total jarak yang paling minimum dari semua rute yang terbentuk. Apabila total jarak yang paling minimum diperoleh dari tahap inisialisasi maka solusi rute yang dipilih adalah rute yang terbentuk pada tahap inisialisasi, tetapi apabila rute yang paling minimum diperoleh dari tahap pengembangan maka solusi rute yang dipilih adalah rute dari tahap pengembangan.

Berikut ini blog diagram algoritma Tabu Search.





Gambar 1. Blog Diagram Algoritma *Tabu Search*

Setelah membuat rancangan *flowchart* dalam bentuk blog diagram, maka selanjutnya merancang program yang akan dibuat. Program dimulai dengan input data. Input yang pertama adalah input titik yang mewakili depot dan *customer* dengan menggunakan mouse (Satyananda: 2012). Sisi pada pada graph diwakili oleh matriks bobot. Data jarak, permintaan *customer* dan yang lainnya diinputkan menggunakan *keyboard*. Kemudian data diproses dengan menggunakan algoritma *Tabu Search* dalam bahasa Pemrograman *Borland Delphi 7.0* dan output yang dihasilkan berupa rute yang optimum. Rute tersebut ditampilkan dalam memo dan graph hasil yang menunjukkan rute yang terbentuk. Pada program juga dilengkapi dengan button Buka, Simpan, Proses, Keluar dan Baru.

- **Contoh Penerapan *Split Load Vehicle Routing Problem with Simultaneous Delivery and Pickup* dengan Algoritma *Tabu Search***

Contoh ini diambil dari jurnal berjudul “*Mathematic Model for Split Load Vehicle Routing Problem with Simultaneous Delivery and Pickup*” (Yin,dkk: 2013) dengan mengambil delapan titik sebagai *customer* yang akan dilayani. Perusahaan A akan melakukan pengisian barang serta pengambilan kemasan isi ulang kepada 8 customer dengan kapasitas kendaraan 5(ton). Jarak maksimal yang dapat ditempuh setiap kendaraan adalah 60km. Adapun daftar permintaan tiap customer seperti pada tabel berikut:

Tabel 3.1 Tabel Permintaan *Customer* Contoh 1 (ton)

customer	1	2	3	4	5	6	7	8
Pengiriman	2,0	0,2	1,0	0,8	1,0	7,5	0	11
Pengambilan	1,0	0,3	0,7	2,4	0,8	10,6	5,3	0

Keterangan tabel: *customer* 1 memiliki permintaan pengiriman sebanyak 2 dan permintaan pengambilan sebanyak 1

Tabel jarak antara depot dan *customer*, jarak antar *customer* dapat dilihat pada tabel berikut:

Tabel 3.2 Tabel Jarak antara Depot dengan *Customer* dan antar *Customer* Contoh 1 (km)

	0	1	2	3	4	5	6	7	8
0	0								
1	12,1	0							
2	15,1	5	0						
3	18	10	13,9	0					
4	17,7	7,28	10,2	4,24	0				
5	20,1	17,5	22,2	9,43	13,6	0			
6	8,54	3,61	7,07	12,2	10,3	18	0		
7	18,4	14,8	19,4	6,71	10,8	2,83	15,5	0	
8	17,1	5,1	4,12	10,8	6,72	19,8	8,54	17	0

Keterangan: *customer* 1 memiliki jarak dengan depot sejauh 12,1 km, dengan *customer* 2 sejauh 5 km, dengan *customer* 3 sejauh 10 km.

- Penyelesaian menggunakan Algoritma *Tabu Search*.

1. Tahap *Split Load*

Pada tahap pembagian beban diperoleh hasil dua himpunan sebagai berikut:

Himpunan *customer* tambahan= {6,7,8}

Himpunan *customer* biasa = {1,2,3,4,5,6,7,8, }

Dengan tabel permintaan

Tabel 3.3 Tabel Permintaan Customer Tambahan Hasil Split Load Contoh 1

<i>customer</i>	6a	6b	7a	8a	8b
pengiriman	5	0	0	5	5
pengambilan	5	5	5	0	0

Keterangan: setelah hasil *split load customer* 6 akan menerima pendistribusian barang yang banyaknya sama dengan kapasitas kendaraan sebanyak 2 kali yaitu permintaan pengiriman 5 dan 5 dan permintaan pengambilan 5 dan 0.

Tabel 3.4 Tabel Permintaan Customer Biasa Hasil Split Load Contoh 1

<i>customer</i>	1	2	3	4	5	6	7	8
Pengiriman	2,0	0,2	1,0	0,8	1,0	2,5	0	1
Pengambilan	1,0	0,3	0,7	2,4	0,8	0,6	0,3	0

Keterangan: setelah hasil *split load customer* 6 memiliki sisa permintaan pengiriman 2,5 dan sisa permintaan pengambilan 0,6.

Setelah memperoleh dua himpunan maka selesaikan kedua himpunan tersebut melalui 3 tahap selanjutnya.

Penyelesaian customer tambahan.

Dari tahap inisialisasi diperoleh tiga rute untuk customer tambahan adalah sebagai berikut:

Rute 1: $(0,8a, 6b, 0)$ dengan pengiriman 5 pengambilan 5 dan jarak 34,14

Rute 2: $(0,8b, 7a, 0)$ dengan pengiriman 5 pengambilan 5 dan jarak 52,1

Rute 3: $(0,6a, 0)$ dengan pengiriman 5 pengambilan 5 dan jarak 17,08

Total jarak dari ketiga rute tersebut adalah 103,32

Setelah tahap inisialisasi, selanjutnya tahap pengembangan dengan melakukan pertukaran beberapa titik antar rute dan sekaligus dilakukan pemeriksaan kendala. Dari tahap ini tidak terdapat perubahan rute, rute yang diperoleh sama dengan rute pada tahap inisialisasi yaitu dua rute dengan total jarak yang sama 103,32. Selanjutnya tahap pemilihan solusi optimum, karena semua total jarak yang diperoleh sama maka pilih solusi rute dari tahap inisialisasi.

Penyelesaian customer biasa.

Dari tahap inisialisasi diperoleh tiga rute untuk customer tambahan adalah sebagai berikut:

Rute 1: $(0,1,6,0)$ dengan pengiriman 4,5 pengambilan 1,6 dan jarak 24,25

Rute 2: $(0,5,7,3,4,8,2,0)$ dengan pengiriman 4 pengambilan 4,5 dan jarak 59,82

Total jarak dari kedua rute tersebut adalah 84,07

Setelah tahap inisialisasi, selanjutnya tahap pengembangan dengan melakukan pertukaran beberapa titik antar rute dan sekaligus dilakukan pemeriksaan kendala. Dari tahap ini tidak ada rute baru yang terbentuk karena tidak memenuhi kendala sehingga rute yang ada hanya rute dari tahap inisialisasi saja. Selanjutnya tahap

pemilihan solusi optimum, karena hanya ada satu rute yang terbentuk yaitu rute dari tahap inisialisasi maka rute tersebut yang dipilih sebagai rute solusi untuk customer biasa.

Jadi dari kedua himpunan *customer* tersebut diperoleh solusi rute yang optimum adalah sebagai berikut:

Rute 1: $(0,8a, 6b, 0)$ dengan pengiriman 5 pengambilan 5 dan jarak 34,14

Rute 2: $(0,8b, 7a, 0)$ dengan pengiriman 5 pengambilan 5 dan jarak 52,1

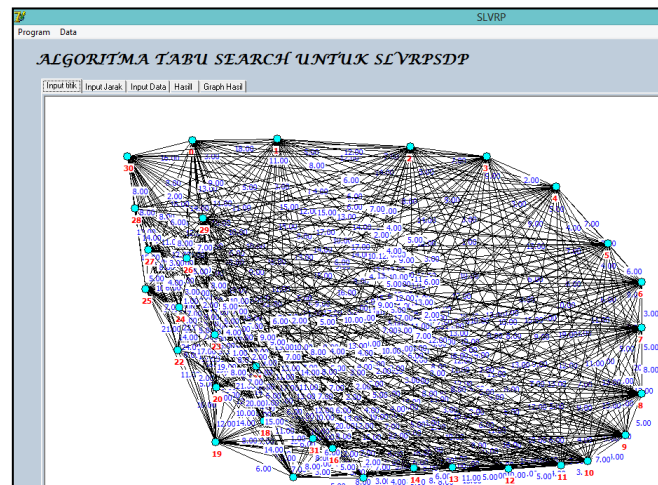
Rute 3: $(0,6a, 0)$ dengan pengiriman 5 pengambilan 5 dan jarak 17,08

Rute 1: $(0,1,6,0)$ dengan pengiriman 4,5 pengambilan 1,6 dan jarak 24,25

Rute 2: $(0,5,7,3,4,8,2,0)$ dengan pengiriman 4 pengambilan 4,5 dan jarak 59,82

- Penyelesaian menggunakan program

Data yang akan ditampilkan pada simulasi program ini adalah data dengan 30 titik. Hasil yang diperoleh adalah sebagai berikut.



Gambar 2. Graph awal

```

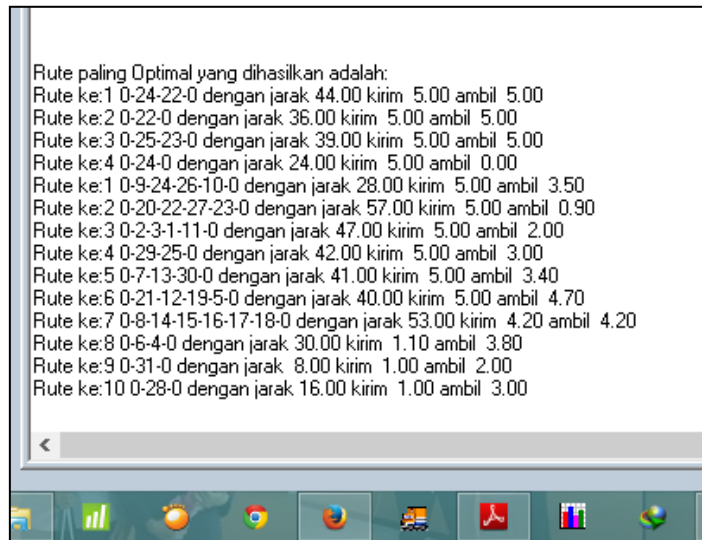
===== [ DATA PERMINTAAN AWAL ] =====
Data Customer:
COSTUMER : 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
KIRIM : 2 0.2 1 0.8 1 0.3 0.7 0.8 0 0 1.8 0.5 1.3 0.7 0.2 1 0.5 1 1.9 0.5 1.6 7.5 0 11
AMBIL : 1 0.3 0.7 2.4 0.8 1.4 0.9 1 2 0.5 0 1 0.5 0.6 0.3 0.8 0.5 1 1.5 0 1.4 10.6 5.3 0

===== [ HASIL SPLIT LOAD ] =====
Data Customer Biasa
COSTUMER : 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
KIRIM : 2.00 0.20 1.00 0.80 1.00 0.30 0.70 0.80 0.00 0.00 1.80 0.50 1.30 0.70 0.20 1.00 0.50 1.00 1.90 0.50 1.50 2.50 0.00 1.00
AMBIL : 1.00 0.30 0.70 2.40 0.80 1.40 0.90 1.00 2.00 0.50 0.00 1.00 0.50 0.60 0.30 0.80 0.50 1.00 1.50 0.00 1.40 0.60 0.30 0.00
SELISIH : 1.00 -0.10 0.30 -1.60 0.20 -1.10 -0.20 -0.20 -2.00 -0.50 1.80 -0.50 0.80 0.10 -0.10 0.20 0.00 0.00 0.40 0.50 0.20 1.90 -0.30 1.00

Data Customer Tambahan
COSTUMER : 22-1 22-2 23-1 24-1 24-2 25-1
KIRIM : 5.00 0.00 0.00 5.00 5.00 5.00
AMBIL : 5.00 5.00 5.00 0.00 0.00 0.00
SELISIH : 0.00 5.00 -5.00 5.00 5.00 5.00

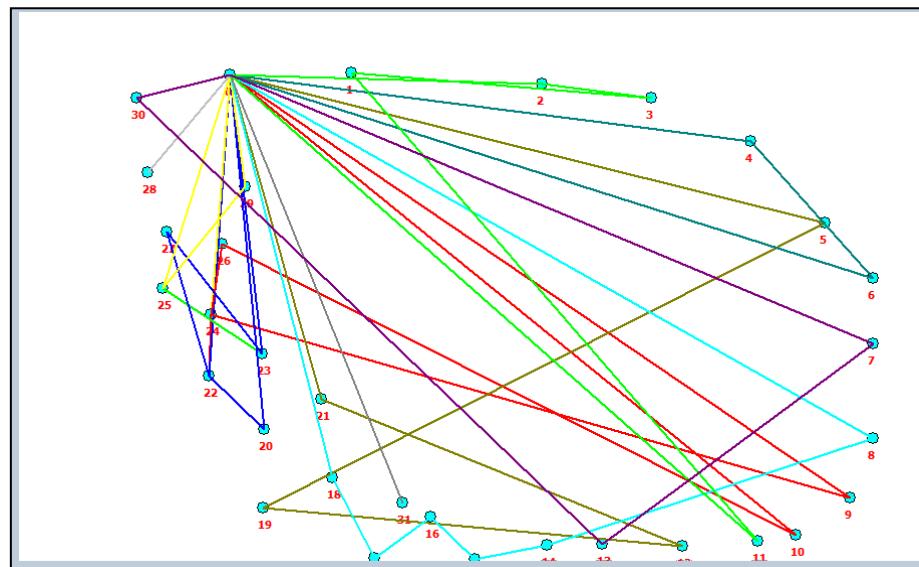
```

Gambar 3. Hasil Split Load



Gambar 4. Solusi optimum

Dari gambar 4 diatas diperoleh hasil rute [0,24,22,0] dengan jarak 44 , rute [0,22,0] dengan jarak 36 , rute [0,25,23,0] dengan jarak 39 , rute [0,24,0] dengan jarak 24, rute [0,9,24,26,10,0] dengan jarak 28, rute [0,20,22,27,23,0] dengan jarak 57 , rute [0,2,3,1,11,0] dengan jarak 47 , rute [0,29,25,0] dengan jarak 42 , rute [0,7,13,30,0] dengan jarak 41, rute [0,21,12,19,5,0] dengan jarak 40, rute [0,8,14,15,16,17,18,0] dengan jarak 53 , rute [0,6,4,0] dengan jarak 30 , rute [0,31,0] dengan jarak 8 dan rute [0,28,0] dengan jarak 16. Graph hasil yang merepresentasikan hasil tersebut adalah sebagai berikut.



Gambar 5. Rute Hasil

PENUTUP

Pada artikel ini telah dibahas permasalahan SLVRPSDP, dimana dalam menyelesaikan permasalahan tersebut menggunakan Algoritma *Tabu Search* dan program yang dibuat dengan menggunakan bahasa pemrograman *Borland Delphi 7.0*. Berdasarkan hasil dan pembahasan serta contoh penerapan yang telah dikerjakan dapat disimpulkan bahwa:

1. Penyelesaian permasalahan SLVRPSDP dengan menggunakan algoritma *Tabu Search* melalui empat tahap yaitu tahap Split Load, tahap Inisialisasi, tahap Pengembangan dan tahap Pemilihan Solusi Optimum.
2. Pencarian solusi untuk permasalahan (SLVRPSDP) menggunakan Algoritma *Tabu Search* dapat menggunakan pemrograman *Borland Delphi 7.0*. Inputan pada program berupa menggambar titik depot dan titik *customer*, mengisi kapasitas kendaraan, jarak maksimal yang dapat ditempuh kendaraan, jarak antara depot dan *customer*, jarak antar customer serta permintaan setiap *customer*. Pada program ini telah disimulasi untuk data dengan 8 titik, 25 titik, 30 titik dan 50 titik.

Pembahasan yang dilakukan pada artikel ini hanya pada graph komplit saja, untuk itu disarankan untuk pengembangan dengan menggunakan graph yang tak komplit. Selain itu permasalahan ini dapat diimplementasikan pada contoh nyata seperti pada saat PKL dengan memperhatikan batasan- batasan yang ada.

DAFTAR RUJUKAN

- Cao, E. & Lai, M.. An Improved Genetic Algorithm for the Vehicle Routing Problem with Simultaneous Delivery and Pickup Service. *The Sixth Wuhan International Conference on e-business- Innovation Management Track*, pp 2100-2106.
- Lai, C., Chen, C. & Ma, Y. Vehicle Routing Problem with Simultaneously Delivery And Pickup. *Journal Of Far East Technology*(25), pp 475-484.
- Satyananda, D. 2012. *Panduan Praktikum Struktur Data*. Malang: Fakultas Matematika dan Ilmu Alam Universitas Negeri Malang.
- Singer, B. 2008. *The Multiple Trip Vehicle Routing Problem*, BMI Thesis. Vrije Universiteit Amsterdam.
- Wilson, R.J & Aldous, J.M. 1990. *Graph and Application an Introductory Approach*. London: Springer.
- 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.
- Yin,C., Bu, L & Gong, H. 2013. Mathematical Model And Algorithm of Split load Vehicle Routing Problem with Simultaneous Delivery and Pickup. *International Journal of Innovative Computing, Information and Control* 9(11): ISSN 1349-4198, pp 4497-4508.