

MATHEMATICAL MODEL OF VEHICLE ROUTING PROBLEM WITH COMPARTMENT, SPLIT DELIVERY, MULTI PRODUCT, AND TIME WINDOWS

Nur Mayke Eka Normasari^{*}, Anjas Fikhri Warangga

Departemen Teknik Mesin dan Industri Fakultas Teknik, Universitas Gadjah Mada

Jl. Grafika No. 2 Yogyakarta 55281 Indonesia

^{*}mayke@ugm.ac.id

Abstract

This research proposes a mathematical model of the Vehicle Routing Problem with Compartment (VRPC) by considering split delivery, multi product, and time windows. VRPC is variant of VRP which is an extension of the Capacitated Vehicle Routing Problem (CVRP) by considering the used of compartment in the vehicle. Compartment is used to separate different product to be distributed. The application of the VRPC concept in a real system can be found in the distribution system of fuel, oil, recycled waste, or food distribution. The mathematical model is used to determine the route that minimize the distance. AMPL software and CPLEX solver is used to create the mathematical programming of the model and solve it. The proposed mathematical model is Mixed Integer Nonlinear Programming (MINLP). Numerical experiments is conducted to illustrate the use of the model. The experimental results show that the model passed the verification and validation tests.

Keyword: VRPC, split delivery, multi product, time windows, CPLEX, MINLP

Abstrak

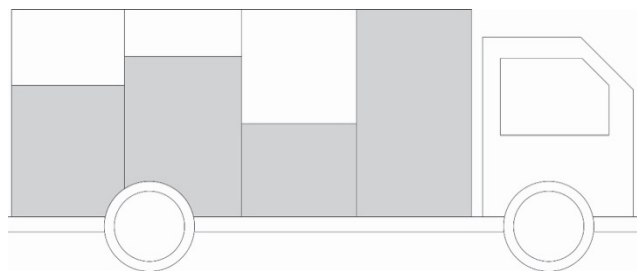
Penelitian ini mengusulkan model matematika dari *Vehicle Routing Problem with Compartment (VRPC)* dengan karakteristik *split delivery*, *multi product*, dan *time windows*. *VRPC* adalah varian *VRP* yang merupakan pengembangan dari *Capacitated Vehicle Routing Problem (CVRP)* dengan kendaraan yang digunakan memiliki kompartemen untuk misahkan beberapa jenis produk yang akan didistribusikan. Penerapan konsep *VRPC* dalam sistem nyata, dapat ditemukan pada sistem pendistribusian bahan bakar, minyak, limbah daur ulang, maupun pendistribusian makanan. Model matematika yang dibangun bertujuan untuk menentukan rute optimal dengan meminimasi jarak. Perangkat lunak *AMPL* dengan *CPLEX solver* digunakan untuk menyelesaikan model matematika yang dibangun. Model matematika yang dikembangkan berbentuk *Mixed Integer Nonlinear Programming (MINLP)*. Eksperimen numeris digunakan untuk mengilustrasikan penggunaan model yang dibangun. Hasil eksperimen menunjukkan bahwa model yang dibangun lulus uji verifikasi dan validasi.

Kata Kunci: VRPC, split delivery, multi product, time windows, CPLEX, MINLP

1. Pengantar

Vehicle Routing problem (VRP) pertama kali dikembangkan secara umum dari permasalahan *Traveling Salesment Problem (TSP)* oleh Dantzig dan Ramser [1] dengan judul “The Truck Dispatching Problem”. Mereka mencoba menemukan rute terpendek untuk mendistribusikan bahan bakar dengan menggunakan truk, yang secara umum juga dikenal sebagai *Capacitated Vehicle Routing Problem (CVRP)*. CVRP klasik mempertimbangkan sekelompok konsumen dengan permintaan yang telah diketahui dan sekelompok kendaraan dari satu depot yang sama untuk memenuhi permintaan tersebut. Rute yang terbentuk terdiri dari urutan konsumen yang ditugaskan pada kendaraan tertentu untuk dipenuhi permintaannya dengan tujuan meminimalkan total jarak yang ditempuh. Berbagai varian VRP dikembangkan sebagai penyesuaian dengan permasalahan dalam sistem nyata [2]. Varian VRP tersebut antara lain *Multi Depot Vehicle Routing Problem (MDVRP)*, *Vehicle Routing Problem with Backhauls (VRPB)*, *Vehicle Routing Problem with Time Windows (VRPTW)*, *Multiple Trip Vehicle Routing Problem (MTVRP)*, dan *Mixed Fleet Vehicle Routing Problem (MFVRP)*. Penelitian VRP terus dikembangkan untuk membuat model VRP yang lebih dekat dengan sistem nyata, seperti Frizzel dan Griffin [3] dengan mempertimbangkan karakteristik *split delivery* dan *time windows* pada VRP. Setelah itu, Ray dkk [4] mengembangkan *multi depot split delivery VRP* dan Yan dkk [5] mengembangkan *multiple trip split delivery VRPTW*.

Pada sistem pendistribusian nyata, ada kendaraan yang menggunakan kompartemen untuk mendistribusikan produknya, namun dari berbagai varian VRP tersebut, belum banyak penelitian yang mempertimbangkan penggunaan kompartemen. Contoh kendaraan dengan kompartemen dapat dilihat pada Gambar 1. Kompartemen digunakan untuk memisahkan produk yang akan dikirim oleh kendaraan karena sifat produk yang tidak dapat bercampur satu sama lain. Pengembangan VRP dengan mempertimbangkan penggunaan kompartemen dikenal dengan nama *Vehicle Routing Problem with Compartment (VRPC)*. Aplikasi konsep VRPC pada sistem nyata berupa pendistribusian bahan bakar, minyak [6], susu [7], maupun limbah daur ulang. Lahyani dkk [6] meneliti tentang VRPC dengan karakteristik *multi product*, *multi period*, dan *multi compartment* yang diangkat dari sistem nyata yaitu sistem pengumpulan *olive oil*. Selanjutnya, Sethanan dan Pitakaso [7] membuat model matematis untuk VRPC dengan karakteristik *multi trip* dan *split delivery* untuk *single product* yang diangkat dari proses pendistribusian susu mentah.



Gambar 1. Truk dengan *multi compartment*

Penelitian ini mengusulkan model matematika untuk VRPC dengan karakteristik *split delivery*, *multi product*, dan *time windows* sebagai *general model* yang diangkat dari proses distribusi minyak bumi.

2. Metodologi Penelitian

Tahap awal pada penelitian ini adalah mengamati sistem pendistribusian minyak bumi. Pada tahap ini, pengamatan dilakukan pada sistem nyata untuk mengevaluasi karakter

sistem sehingga didapatkan cara kerja, parameter dan variabel keputusan dari sistem. Pada tahap awal ini juga dilakukan studi literatur untuk mengetahui perkembangan penelitian VRP dan permasalahan yang diselesaikan.

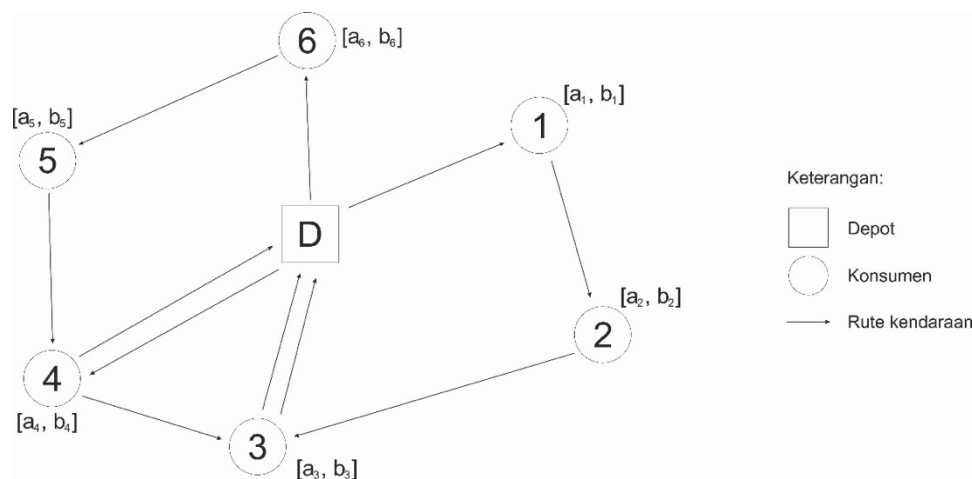
Tahap selanjutnya adalah pembuatan model matematis dengan karakteristik sistem yang didapatkan dari tahap sebelumnya. Model matematis sistem pendistribusian minyak bumi yang dibangun adalah model *VPRC* dengan karakteristik *split delivery*, *multi product*, dan *time windows*. Model matematis dibuat dengan tujuan meminimalkan biaya transportasi. Model matematis yang telah dibuat kemudian dituliskan ke dalam bentuk program komputer dengan menggunakan *software AMPL* dan diselesaikan dengan menggunakan *CPLEX solver*. Program komputer dari model tersebut akan digunakan untuk melakukan percobaan numerik dalam proses verifikasi dan validasi model.

Proses selanjutnya adalah tahap verifikasi dan validasi. Proses verifikasi dilakukan dengan menguji apakah model matematis dalam bentuk program komputer dapat berjalan sesuai dengan logika yang diharapkan. Proses verifikasi dilakukan dengan melakukan pengecekan pada hasil yang didapatkan, jika hasil yang didapatkan tidak menyalahi semua batasan yang ada, maka model dapat dikatakan sudah terverifikasi. Jika model sudah terverifikasi, selanjutnya dapat dilakukan proses validasi. Validasi yang dilakukan dalam penelitian ini adalah validasi *output*, dimana proses validasi dilakukan dengan membandingkan hasil perhitungan menggunakan model yang dibuat dengan hasil perhitungan manual dengan metode enumerasi lengkap. Persoalan sederhana dibuat sebagai media untuk melakukan proses validasi. Model dikatakan tervalidasi jika hasil kedua perhitungan yang dilakukan identik.

3. Hasil dan Pembahasan

3.1. Pembangunan Model Matematis

Dalam penelitian ini, dibangun model matematis *VRPC* dengan karakteristik *split delivery*, *multi-product*, dan *time-windows*. Di dalam model yang dibangun, setiap kendaraan yang bermula dari depot, mengirimkan beberapa jenis produk ke tiap titik konsumen dan kemudian kembali lagi ke depot. Setiap kendaraan hanya dapat mengunjungi konsumen tertentu sebanyak satu kali. Tiap kendaraan diperbolehkan untuk mengunjungi lebih dari satu konsumen, lalu konsumen tersebut juga diperbolehkan untuk dilayani oleh lebih dari satu kendaraan. Tiap konsumen memiliki rentang waktu tertentu untuk dapat dilayani $[a_j, b_j]$. Model yang dibangun memiliki tujuan untuk meminimalkan total biaya transportasi, dimana biaya tersebut juga dapat diinterpretasikan sebagai total jarak tempuh kendaraan. Gambar 2. menunjukkan ilustrasi sistem yang akan dimodelkan.



Gambar 2. Ilustrasi sistem

Model yang dibuat memiliki beberapa asumsi sebagai berikut:

1. Setiap kendaraan hanya dapat membentuk satu rute
2. Depot dianggap memiliki kapasitas yang tidak terbatas, atau dengan kata lain depot selalu dapat memenuhi kebutuhan seluruh konsumen
3. Tiap kompartemen hanya dapat diisi oleh satu jenis produk.
4. *Service time* tiap kendaraan pada tiap konsumen dianggap sama.
5. Produk dapat menempati lebih dari satu kompartemen.
6. Seluruh kendaraan yang digunakan bersifat homogen.

Model matematis yang dibuat memiliki beberapa set, indeks, parameter, dan variabel. Tabel 1. menunjukkan set dan indeks yang digunakan dalam model. Tabel 2. menunjukkan parameter yang terdapat dalam model. Tabel 3. menunjukkan variabel keputusan yang digunakan dalam model.

Tabel 1. *Set* dan indeks model matematis

Indeks	Definisi
V	Set titik depot dan konsumen ($h, i, j = 0, \dots, V$)
K	Set kendaraan ($k= 1, \dots, K$)
R	Set produk ($r= 1, \dots, R$)
M	Set kompartemen ($m= 1, \dots, M$)

Tabel 2. Parameter model matematis

Parameter	Definisi
C_{rmk}	Kapasitas produk r yang dapat tertampung di kompartemen m pada kendaraan k
c_{ijk}	Biaya perjalanan dari titik i ke titik j oleh kendaraan k
t_{ij}	Waktu tempuh dari titik i ke titik j
f_j	<i>Service time</i> di titik j
d_{jr}	Jumlah <i>demand</i> produk r di titik j
a_j	Waktu awal pelayanan titik j
b_j	Waktu akhir pelayanan titik j

Tabel 3. Variabel model matematis

Parameter	Definisi
x_{ijk}	$x_{ijk} = 1$: jika kendaraan k bergerak dari titik i ke titik j ; $x_{ijk} = 0$: lainnya
y_{rmk}	$y_{rmk} = 1$: jika terisi produk r di kompartemen m pada kendaraan k ; $y_{rmk} = 0$: lainnya
z_{jrk}	Jumlah produk r yang dialokasikan ke titik j oleh kendaraan k
sd_{jk}	Waktu keberangkatan kendaraan k dari titik j
sa_{jk}	Waktu kedatangan kendaraan k di titik j

Model matematis dibuat dalam bentuk *Mixed Integer Non-linear Programming* (MINLP) dengan notasi-notasi yang telah dijelaskan sebelumnya. Fungsi objektif dan batasan model matematis yang telah dibuat yaitu sebagai berikut:

Objective function:

$$\text{minimize } \sum_i^V \sum_j^V \sum_k^K c_{ij} x_{ijk} \quad (1)$$

Subject to:

$$\sum_j^V x_{0jk} \leq 1 ; \forall k \quad (2)$$

$$\sum_i^V x_{ihk} - \sum_j^V x_{hjk} = 0 ; \forall k, h \quad (3)$$

$$x_{ijk} + x_{jik} \leq 1 ; \forall i, j, k \quad i, j \notin \{0\} \quad (4)$$

$$\sum_i^V \sum_k^K x_{ijk} \geq 1 ; \forall j \quad (5)$$

$$\sum_k^K x_{iik} = 0 ; \forall i \quad (6)$$

$$\sum_r^R \sum_m^M y_{rmk} \leq |M| ; \forall k \quad (7)$$

$$\sum_r^R y_{rmk} \leq 1 ; \forall m, k \quad (8)$$

$$\sum_j^V z_{jrk} \leq \sum_m^M C_{rmk} y_{rmk} ; \forall r, k \quad (9)$$

$$\sum_k^K z_{jrk} = d_{jr} ; \forall j, r \quad (10)$$

$$z_{jrk} \leq d_{jr} \sum_i^V x_{ijk} ; \forall j, r, k \quad (11)$$

$$\sum_i^V (sd_{ik} + t_{ij}) x_{ijk} = sa_{jk} ; \forall j, k \quad (12)$$

$$sa_{jk} \geq a_j \sum_i^V x_{ijk} ; \forall j, k \quad j \notin \{0\} \quad (13)$$

$$sa_{jk} + f_j \sum_i^V x_{ijk} = sd_{jk} ; \forall j, k \quad j \notin \{0\} \quad (14)$$

$$sd_{jk} \leq b_j ; \forall j, k \quad (15)$$

$$x_{ijk}, y_{rmk} = \{0,1\} \quad (16)$$

$$z_{jrk}, sd_{jk}, sa_{jk} \geq 0 \quad (17)$$

Fungsi tujuan (1) dalam penelitian ini adalah meminimalkan total biaya transportasi dari seluruh kendaraan. Batasan (2) memastikan kendaraan hanya boleh keluar dari depot sekali atau tidak sama sekali. Batasan (3) memastikan keberlanjutan proses *routing* kendaraan dan memastikan kendaraan yang keluar dari depot akan kembali lagi ke depot. Batasan (4) memastikan tidak ada kendaraan yang kembali lagi ke titik sebelumnya yang sudah dikunjungi. Batasan (5) menyatakan bahwa tiap titik atau konsumen dapat dikunjungi oleh beberapa kendaraan (*split delivery*). Batasan (6) memastikan tidak ada proses *loop* di tiap titik. Batasan (7) menyatakan bahwa jumlah kompartemen yang terisi tidak dapat melebihi jumlah kompartemen kendaraan. Batasan (8) memastikan bahwa satu kompartemen hanya dapat berisi satu jenis produk. Batasan (9) menyatakan bahwa jumlah produk yang dialokasikan tiap kendaraan tidak melebihi kapasitas kendaraan itu sendiri. Batasan (10) menyatakan bahwa jumlah produk yang dialokasikan harus sama dengan jumlah *demand* pada tiap titik. Batasan (11) menyatakan bahwa produk hanya dapat dikirim saat konsumen dikunjungi oleh kendaraan. Batasan (12) menjelaskan bahwa waktu kedatangan kendaraan di suatu titik sama dengan waktu keberangkatan dari titik sebelumnya ditambah waktu tempuhnya. Batasan (13) menyatakan bahwa waktu kedatangan kendaraan tidak boleh mendahului waktu buka di titik tujuan. Batasan (14) menjelaskan bahwa waktu keberangkatan kendaraan di suatu titik sama dengan waktu kedatangan kendaraan di titik

tersebut ditambah *service time* di titik tersebut. Batasan (15) menyatakan bahwa waktu keberangkatan kendaraan tidak boleh melewati waktu tutup di titik keberangkatan. Batasan (16) mendefinisikan variabel biner. Variabel non-negatif diberikan pada batasan (17)

3.2. Verifikasi dan Validasi Model Matematis

Verifikasi model matematis dilakukan untuk memastikan bahwa model yang dibuat sudah berjalan sesuai dengan logika yang diharapkan, yaitu sesuai dengan *behaviour* dari sistem. Model dapat dikatakan terverifikasi jika model tersebut sudah dapat berjalan dalam program komputer dan menunjukkan hasil yang sesuai dengan batasan-batasan dan fungsi objektif yang ditetapkan.

Proses verifikasi dilakukan dengan menggunakan *software AMPL* dengan *CPLEX solver*. Model matematis yang telah dibuat diubah menjadi program komputer pada *software AMPL* dengan melakukan menambahkan proses linearisasi agar dapat diselesaikan dengan menggunakan *CPLEX solver*. Proses linearisasi dilakukan dengan metode yang dijelaskan oleh FICOTM Express Optimization Suite [8]. Selanjutnya, untuk melakukan proses verifikasi, dibuat kasus sederhana untuk diselesaikan oleh model yang telah dibuat pada *software AMPL* menggunakan *CPLEX solver*. Kasus tersebut yaitu kasus *VRPC* yang terdiri dari 3 titik konsumen, 2 jenis produk, 2 unit kendaraan, dan 2 kompartemen. Data dari setiap parameter seperti biaya perjalanan, *demand*, waktu tempuh, dan waktu pelayanan tiap konsumen, dibuat dengan melakukan penyesuaian terhadap batasan-batasan tertentu, seperti total *demand* dan *time windows*. Hal tersebut dilakukan agar data yang dibangkitkan merupakan data yang *feasible* untuk dilakukan penyelesaian. Data dari tiap parameter yang dibangkitkan dapat dilihat pada Tabel 4, 5, 6, 7, dan 8.

Tabel 4. Data matriks biaya perjalanan (parameter c_{ijk})

Titik	Depot	Konsumen 1	Konsumen 2	Konsumen 3
Depot	0	7	11	6
Konsumen 1	7	0	12	10
Konsumen 2	11	12	0	14
Konsumen 3	6	10	14	0

Tabel 5. Data matriks waktu tempuh (parameter t_{ij})

Titik	Depot	Konsumen 1	Konsumen 2	Konsumen 3
Depot	0	35	55	30
Konsumen 1	35	0	60	50
Konsumen 2	55	60	0	70
Konsumen 3	30	50	70	0

Tabel 6. Data matriks *demand* (parameter d_{jr})

	Depot	Konsumen 1	Konsumen 2	Konsumen 3
Produk 1	0	2000	4000	2000
Produk 2	0	1000	1000	2000

Tabel 7. Data matriks *time windows* (parameter a_j dan b_j)

	Depot	Konsumen 1	Konsumen 2	Konsumen 3
waktu awal	0	40	80	120
waktu akhir	1000	210	250	290

Tabel 8. Data kapasitas dan *service time* (parameter C_{rmk} dan f_j)

Kapasitas kompartemen	4000
<i>Service time</i>	30

Kasus yang dibuat kemudian diaplikasikan pada model matematika dalam program komputer *AMPL* dan diselesaikan dengan menggunakan *CPLEX solver*. Hasil dari perhitungan dapat dilihat pada Tabel 9.

Tabel 9. Hasil perhitungan kasus sederhana

Total biaya = 45	
Hasil variabel keputusan*	
x_{ijk}	$x_{121} = 1, x_{241} = 1, x_{411} = 1$ $x_{132} = 1, x_{312} = 1$
y_{rmk}	$y_{121} = 1, y_{122} = 1, y_{211} = 1, y_{212} = 1$
z_{jrk}	$z_{211} = 2000, z_{221} = 1000, z_{312} = 4000, z_{322} = 1000, z_{411} = 2000, z_{421} = 2000$
sd_{jk}	$sd_{11} = 5, sd_{12} = 25, sd_{21} = 70, sd_{32} = 110, sd_{41} = 150$
sa_{jk}	$sa_{11} = 180, sa_{12} = 165, sa_{21} = 40, sa_{32} = 80, sa_{41} = 120$

*depot ditunjukkan dengan indeks 1

Hasil perhitungan pada kasus sederhana dengan menggunakan *CPLEX solver* memberikan hasil optimum dengan keputusan pengambilan rute 1-3-1 untuk kendaraan pertama dan rute 1-2-4-1 untuk kendaraan kedua dengan total biaya sebesar 45. Model matematis yang diubah menjadi program komputer pada *software AMPL* sudah dapat berjalan dengan tidak adanya *error* dan menghasilkan nilai-nilai variabel keputusan yang tidak menyalahi batasan-batasan, sehingga dapat dikatakan model matematis tersebut telah lulus uji verifikasi.

Karena model telah lulus uji verifikasi, maka selanjutnya pada model matematis yang telah dibuat akan dilakukan proses validasi. Model matematis yang dibuat merupakan model *general*, sehingga proses uji validasi yang dilakukan ditujukan untuk memastikan bahwa model yang dibuat dapat menghasilkan hasil yang benar dan akurat. Proses validasi dilakukan dengan cara membandingkan hasil perhitungan yang dilakukan menggunakan *CPLEX solver* dengan hasil perhitungan manual dengan metode enumerasi lengkap. Berdasar Walukiewicz [9], jika suatu *feasible solution* pada suatu *integer programming P* dibatasi, maka $F(P)$ mengandung jumlah *feasible point* yang *finite*, dan *problem* dapat diselesaikan dengan metode *explicit enumeration/* enumerasi lengkap. Pada kasus *binary problem* metode enumerasi lengkap dilakukan dengan memeriksa tiap kombinasi dari variable biner tersebut apakah *feasible*, dan jika *feasible* maka hasil dari seluruh kombinasi yang *feasible* tersebut akan dibandingkan dan dipilih *feasible solution* optimal. Selain itu, menurut Walukiewicz [9], jika terdapat terlalu banyak kombinasi nilai variabel, maka perhitungan dengan metode enumerasi lengkap tidak terlalu membantu. Oleh karena itu perhitungan manual dengan enumerasi lengkap dilakukan dengan kasus soal yang sangat sederhana yaitu soal yang sama pada proses verifikasi. Proses perhitungan manual dengan metode enumerasi lengkap pada penelitian ini dilakukan dengan membangkitkan seluruh nilai variabel keputusan variabel x_{ijk} pada fungsi objektif yang mungkin, kemudian mengeliminasi kemungkinan nilai x_{ijk} tersebut berdasarkan batasan *routing*, dan mengeliminasi kemungkinan nilai x_{ijk} berdasarkan batasan

pemenuhan *demand* dan batasan *time windows*. Proses terakhir yaitu memilih nilai yang paling optimal di antara seluruh *feasible solution* yang dihasilkan.

Berdasarkan fungsi objektif model matematis dan kasus yang digunakan, nilai-nilai variabel x_{ijk} yang mungkin berjumlah kombinasi dari seluruh indeks yang ada pada variabel x_{ijk} yaitu indeks titik awal i , titik tujuan j , dan kendaraan k dengan kasus yang digunakan yaitu terdapat 4 titik (1 depot dan 3 konsumen), dan 2 unit kendaraan. Nilai-nilai variabel x_{ijk} yang mungkin dapat dilihat pada Tabel 10.

Tabel 10. Kemungkinan nilai variabel x_{ijk}

x_{ij1}	$x_{111}, x_{121}, x_{131}, x_{141},$ $x_{221}, x_{231}, x_{241},$ $x_{331}, x_{341},$ $x_{441} = \{0,1\}$	x_{ij2}	$x_{112}, x_{122}, x_{132}, x_{142},$ $x_{222}, x_{232}, x_{242},$ $x_{332}, x_{342},$ $x_{442} = \{0,1\}$
-----------	---	-----------	---

Kombinasi nilai variabel x_{ijk} yang dapat terjadi pada fungsi objektif dapat dihitung yaitu sejumlah $2^{4*4*2} = 2^{32} = 4.294.967.296$ kemungkinan yang terjadi. Dari kemungkinan awal tersebut, kemudian dilakukan eliminasi berdasarkan batasan-batasan yang ada pada model matematis. Dari seluruh batasan dalam model matematis, kemudian dibagi menjadi 3 jenis batasan yaitu batasan proses *routing* kendaraan yang terdiri dari batasan (2)-(6), batasan kapasitas dan *demand* yang terdiri dari batasan (7)-(11), dan batasan *time windows* yang terdiri dari batasan (12)-(15). Proses eliminasi kemungkinan nilai x_{ijk} yang dapat terjadi dengan batasan *routing* dilakukan dengan membangkitkan seluruh kemungkinan rangkaian rute distribusi yang dapat terjadi berdasarkan batasan *routing* tersebut. Sesuai batasan *routing*, kemungkinan rute tersebut memenuhi kriteria seperti berikut:

1. Tiap rute kendaraan mulai dari depot (titik 1) dan berakhir pada depot.
2. Tiap titik konsumen harus dikunjungi minimal sekali.
3. Tiap titik yang telah dikunjungi, tidak dapat dikunjungi oleh kendaraan yang sama.
4. Tiap titik dapat dikunjungi oleh lebih dari satu kendaraan.

Setelah dilakukan eliminasi berdasarkan batasan *routing* didapat sejumlah 198 kemungkinan rute yang dapat terjadi atau sejumlah 99 kemungkinan rute tanpa mempertimbangkan kombinasi kendaraan (indeks k). Dari seluruh kemungkinan rute tersebut selanjutnya akan dilakukan eliminasi berdasarkan batasan kapasitas dan *demand* dan batasan *time windows*. Setelah dilakukan eliminasi berdasarkan seluruh batasan yang ada, didapatkan sejumlah 33 rute yang dapat terjadi tanpa mempertimbangkan kombinasi kendaraan, seperti pada Tabel 11.

Tabel 11. Total biaya tiap kemungkinan rute yang terjadi

No.	Jumlah kendaraan terpakai	Rute 1	Rute 2	Total biaya
1	2	1-3-1	1-2-4-1	45
2	2	1-3-1	1-2-3-4-1	61
3	2	1-3-1	1-2-4-3-1	64
4	2	1-3-1	1-3-2-4-1	61
5	2	1-2-3-1	1-2-4-1	53

6	2	1-2-3-1	1-3-4-1	61
7	2	1-2-3-1	1-4-3-1	61
8	2	1-2-3-1	1-2-3-4-1	69
9	2	1-2-3-1	1-2-4-3-1	72
10	2	1-2-3-1	1-3-2-4-1	69
11	2	1-2-4-1	1-3-2-1	53
12	2	1-2-4-1	1-3-4-1	54
13	2	1-2-4-1	1-4-3-1	54
14	2	1-2-4-1	1-2-3-4-1	62
15	2	1-2-4-1	1-2-4-3-1	65
16	2	1-2-4-1	1-3-2-4-1	62
17	2	1-3-2-1	1-3-4-1	61
18	2	1-3-2-1	1-4-3-1	61
19	2	1-3-2-1	1-2-3-4-1	69
20	2	1-3-2-1	1-2-4-3-1	72
21	2	1-3-2-1	1-3-2-4-1	69
22	2	1-3-4-1	1-2-3-4-1	70
23	2	1-3-4-1	1-2-4-3-1	73
24	2	1-3-4-1	1-3-2-4-1	70
25	2	1-4-3-1	1-2-3-4-1	70
26	2	1-4-3-1	1-2-4-3-1	72
27	2	1-4-3-1	1-3-2-4-1	70
28	2	1-2-3-4-1	1-2-3-4-1	78
29	2	1-2-3-4-1	1-2-4-3-1	81
30	2	1-2-3-4-1	1-3-2-4-1	78
31	2	1-2-4-3-1	1-2-4-3-1	84
32	2	1-2-4-3-1	1-3-2-4-1	81
33	2	1-3-2-4-1	1-3-2-4-1	78

Selanjutnya dilakukan perhitungan total biaya rute dari seluruh kemungkinan rute tersebut dan dipilih rute dengan total biaya minimum. Hasil optimal dari kasus yang diselesaikan menggunakan perhitungan manual dengan metode enumerasi lengkap yaitu dengan menggunakan rute 1-3-1 untuk kendaraan pertama dan menggunakan rute 1-2-4-1 untuk kendaraan kedua dengan total biaya *routing* sebesar 45. Hasil tersebut merupakan hasil yang identik dengan hasil perhitungan yang dilakukan dengan menggunakan *software AMPL* pada proses verifikasi sebelumnya. Dengan hasil yang identik tersebut, maka dapat dikatakan model yang telah dibuat telah lulus rangkaian uji validasi hasil yang dilakukan.

4. Kesimpulan

Sebuah model matematis untuk permasalahan *Vehicle Routing Problem with Compartment* dengan karakteristik *split delivery*, *multi product*, dan *time windows* dengan tujuan untuk meminimalkan total biaya transportasi telah berhasil dikembangkan. Model matematis yang dibuat telah lulus uji verifikasi dan validasi. Model tersebut dapat digunakan untuk menyelesaikan permasalahan yang bersifat deterministik. Dalam penelitian ini, tingkat kepadatan lalu lintas yang bersifat tidak pasti masih belum dipertimbangkan. Penelitian selanjutnya dapat dilakukan dengan mempertimbangkan pengaruh faktor-faktor yang bersifat tidak pasti seperti permintaan dan tingkat kepadatan lalu lintas.

5. Ucapan Terimakasih

Penulis ingin menyampaikan terimakasih dan penghargaan yang sebesar-besarnya kepada pihak Departemen Teknik Mesin dan Industri (DTMI), Fakultas Teknik (FT), Universitas Gadjah Mada (UGM) atas bantuan dana untuk publikasi penelitian ini dalam skema Hibah DTMI 2018 dengan SK No. 838/H1.17/TMI/LK/2018.

Daftar Pustaka

- [1] Dantzig, G. and Ramser, J., 1959, The Truck Dispatching Problem, *Journal of Management Science*, Vol. 6, No. 1, pp. 80–91.
- [2] Wahyuningsih, S., Satyananda, D., Hasanah, D., Implementations of TSP-VRP Variants for Distribution Problem, *Global Journal of Pure and Applied Mathematics*, Vol. 12, No. 1, pp. 723-732.
- [3] Frizzel, P.W., Giffin, J.W., 1995, The split delivery vehicle scheduling problem with time windows and grid network distances, *Journal of Computers Ops Res.*, Vol. 22, No. 6, pp. 655-667.
- [4] Ray, S., Soeanu, A., Berger, J., Debbabi, M., 2014, The multi-depot split-delivery vehicle routing problem: Model and solution algorithm, *Journal of Knowledge Based Systems*, Vol. 71, pp. 238-265.
- [5] Yan, S., Chu, J.C., Hsiao, F., Huang, H., 2015, A planning model and solution algorithm for multi-trip split-delivery vehicle routing and scheduling problems with time windows, *Journal of Computers & Industrial Engineering*, Vol. 87, pp. 383-393.
- [6] Lahyani, R., Coelho, L.C., Khemakhem, M., Laporte, G., Semet, F., 2015, A multi-compartment vehicle routing problem arising in the collection of olive oil in Tunisia, *Omega*, Vol. 51, pp. 1-10.
- [7] Sethanan, K., Pitakaso, R., 2016, Differential evolution algorithms for scheduling raw milk transportation, *Computers and Electronics in Agriculture*, Vol. 121, pp. 245-259.
- [8] FICOTM Express Optimization Suite, 2009, *MIP Formulations and linearizations*, Fair Isaac Corporation, California.
- [9] Walukiewicz, S., 1991, *Integer Programming*, Kluwer Academic Publishers, Warszawa.