Perjanjian No. III/LPPM/2013-03/10-P

IMPLEMENTASI MODEL PERSEDIAAN YANG DIKELOLA PEMASOK (*VENDORS MANAGED INVENTORY)* DENGAN BANYAK RETAILER

Disusun Oleh:
Alfian, S.T., M.T.
Dr. Carles Sitompul

Lembaga Penelitian dan Pengabdian kepada Masyarakat
Universitas Katolik Parahyangan
2013

ABSTRAK

Penelitian ini dilakukan untuk mengimplementasikan sebuah model analitis permasalahan rantai pasok *Vendor Managed Inventory* (VMI). Adapun lingkup permasalahan VMI yang diteliti adalah kondisi sistem rantai pasok yang melibatkan 1 pemasok dan banyak *retailer.* Sebelum melakukan implementasi model, dilakukan terlebih dahulu proses validasi terhadap model analitis. Tujuannya adalah memastikan model tersebut merepresentasikan sistem VMI dengan tepat, selain itu dimungkinkan pula untuk memperbaiki fungsi-fungsi di dalamnya sehingga didapatkan model yang lebih efisien. Model yang telah melalui tahap validasi akan diterjemahkan ke dalam bahasa pemrograman Lingo. Perangkat lunak yang dihasilkan dari tahap ini akan diverifikasi untuk memastikan ketepatan penerjemahan model ke bahasa pemrograman yang digunakan. Setelah melalui tahap ini perangkat lunak akan diuji coba untuk menyelesaikan beberapa kasus hipotetis. Hasil penyelesaian kasus hipotetis ini merupakan contoh kebijakan distribusi dan pemesanan barang dalam sistem VMI yang dapat diterapkan oleh pihak-pihak yang terlibat. Hasil penyelesaian kasus pun digunakan dalam menguji kesesuaian model dalam menyelesaikan masalah yang dihadapi, salah satu caranya adalah dengan melihat logis tidaknya solusi yang dihasilkan.

DAFTAR ISI

BAB I
PENDAHULUAN

I.1. Latar Belakang Masalah

Sistem merupakan sekumpulan entitas yang saling berinteraksi untuk mencapai tujuan tertentu. Kompleksitas suatu sistem tergantung pada jumlah entitas yang terlibat di dalamnya, hubungan yang terjadi antarentitas, serta terlibat atau tidaknya faktor acak di dalamnya. Banyak jenis masalah yang mungkin terjadi dalam suatu system. Terdapat dua pendekatan untuk menyelesaikan masalah dalam suatu sistem, yaitu langsung bereksperimen dengan sistem nyata atau melakukan pemodelan terhadap sistem tersebut (Law dan Kelton 2000). Pemodelan sistem sering menjadi pilihan karena beberapa hal, yaitu biaya yang lebih rendah serta minimalnya gangguan terhadap sistem yang sekarang sedang berjalan.

Model-model dibangun untuk menirukan sifat-sifat dan interaksi antar entitas dalam sistem nyata. Menurut Law dan Kelton (2000) terdapat 2 jenis model sistem yang dapat dikembangkan, yaitu model fisik dan model matematika. Namun, hal yang paling penting dari setiap jenis model yang dikembangkan adalah kualitas dari model tersebut. Kualitas yang dimaksud adalah tingkat kemiripan model dengan sistem nyata. Semakin tinggi tingkat kemiripan ini maka model akan semakin bermanfaat bagi penyelesaian masalah dalam sistem terkait.

Oleh sebab itu, tahap validasi dan verifikasi menjadi tahap yang penting dalam rangkaian proses pemodelan suatu sistem. Model yang telah dikembangkan perlu divalidasi untuk memastikan bahwa model tersebut dapat menggambarkan sistem sebenarnya dengan tepat. Dalam Law dan Kelton (2000), proses verifikasi ditujukan apabila model yang dikembangkan hendak diterjemahkan ke dalam bahasa pemrograman tertentu. Proses verifikasi dilakukan untuk memastikan bahwa model konseptual yang telah dibuat diterjemahkan dengan benar ke dalam bahasa pemrograman. Dalam penelitian ini akan dilakukan proses validasi dan verifikasi suatu model rantai pasok yang telah dikembangkan oleh Sitompul dan Alfian (2012) sehingga pada akhirnya model tersebut benar-benar bermanfaat dalam penyelesaian masalah terkait.

I.2. Tujuan Khusus

Seperti telah disinggung sebelumnya, penelitian ini dilakukan untuk memverifikasi dan memvalidasi model rantai pasok yang telah dikembangkan sebelumnya. Adapun model yang telah dikembangkan oleh Sitompul dan Alfian (2012) adalah sebuah model *Vendor Managed Inventory.* Model ini adalah model pengelolaan rantai pasok oleh pihak *supplier.* Model rantai pasok yang telah dikembangkan ini melibatkan 1 pemasok dan banyak *retailer*. Pada penelitian sebelumnya, model sempat diterjemahkan ke dalam bahasa pemrograman AMPL, namun belum ada proses validasi dan verifikasi yang dilakukan baik untuk model konseptual maupun program yang telah dibuat. Oleh sebab itu, tidak menutup kemungkinan terdapatnya perbaikan-perbaikan model konseptual saat dilakukannya proses validasi. Perubahan-perubahan ini pun akan berdampak pada perubahan program yang telah dibuat sebelumnya.

Terdapat dua kekurangan dalam model rantai pasok yang telah ditemukan peneliti, yaitu kekeliruan dalam formulasi fungsi tujuan biaya persediaan dan ketidakefisienan penggunaan variabel dalam batasan model optimimasi.Hal ini tentunya menegaskan perlu adanya validasi terhadap model yang telah dibuat. Selain memastikan kemiripan model dengan sistem nyata, proses validasi ini pun dilakukan dengan tujuan mendapatkan model yang lebih efisien. Apabila model telah dikatakan valid, maka model ini akan diterjemahkan ke dalam bahasa pemrograman lalu diverifikasi.

Program yang telah dibuat akan digunakan untuk menyelesaikan masalah-masalah VMI yang ada. Hasil dari penyelesaian masalah ini akan dibandingkan dengan keadaan sistem yang sebenarnya. Inilah proses validasi tahap kedua. Apabila suatu model valid maka seharusnya solusi yang dihasilkan oleh model tidak berbeda secara signifikan dengan solusi/keadaan yang ada pada sistem nyata.

I.3. Keutamaan Penelitian

Permasalah rantai pasok VMI yang selama ini dikembangkan baru melibatkan 1 pemasok dan 1 *retailer.* Pada kenyataannya bisa terdapat lebih dari satu pemasok maupun *retailer* yang terlibat. Oleh sebab itu, penting sekali untuk mengembangkan model VMI dalam permasalahan yang lebih kompleks. Pada penelitian ini model rantai pasok yang dikembangkan melibatkan 1 pemasok dan banyak *retailer.* Sitompul dan Alfian (2012) telah mengembangkan model awal untuk permasalahan ini. Namun, peneliti menemukan beberapa kekurangan dalam model yang telah dikembangkan. Selain itu belum adanya proses validasi dan verifikasi terhadap model membuat model yang telah dikembangkan belum bermanfaat dengan maksimal. Oleh sebab itu, penelitian ini sangat penting untuk dilakukan sehingga model yang telah dikembangkan bisa bermanfaat dalam penyelesaian masalah VMI.

BAB II
STUDI PUSTAKA

II.1. Model *Vendors Managed Inventory*

Vendors Managed Inventory merupakan sistem pengelolaan rantai pasok yang dilakukan oleh pihak pemasok. Pada sistem tradisional, pengelolaan rantai pasok dilakukan oleh masing-masing pelaku dalam sistem, yaitu pemasok dan *retailer*. Oleh sebab itu, optimalisasi kebijakan pun dilakukan terhadap fungsi tujuan masing-masing pelaku. Pada sistem VMI, pihak pemasoklah yang merumuskan kebijakan dalam rantai pasok, baik mengenai jumlah barang yang didistribusikan ke *retailer*, frekuensi pendistribusian barang, serta waktu pemesanan barang ke pihak ketiga. Terdapat beberapa model matematis yang telah dikembangkan untuk membantu optimalisasi kebijakan sistem VMI ini. Salah satu model yang telah dikembangkan adalah model optimalisasi VMI yang melibatkan 1 pemasok dan banyak *retailer* (Sitompul dan Alfian 2012). Model yang dikembangkan ini tergolong ke dalam model deterministik yang hanya melibatkan 1 jenis barang dalam sistem rantai pasok. Model selengkapnya dapat dilihat di bawah ini.

Minimasi

$$C \sum_{i=1}^{n} \sum_{j=1}^{f_i} S_{ij} + H\left[\left\{\sum_{i=1}^{n} \sum_{j=1}^{f_i} X_{ij} T_{ij}\right\}\left\{\sum_{i=1}^{n} \sum_{j=1}^{f_i} (1 - X_{ij})(Q_{ij} - q_{ij})\right\} - \left\{\sum_{i=1}^{n} \sum_{j=1}^{f_i} T_{ij}(1 - 2X_{ij})(Q_{ij} - q_{ij})\right\}\right] + \sum_{i=1}^{n}\left(\frac{d_i}{q_i} c_i' + \frac{q_i}{2} h_i\right) \quad \text{.................................(1)}$$

subject to

$$f_i = \left\lfloor \frac{d_i}{q_i} \right\rfloor \quad ; \; i = 1, 2, \dots \dots \dots, n \quad \text{..........(2)}$$

$$T_{ij} = j\,\frac{q_i}{d_i} \quad ; \; i = 1, 2, \dots \dots \dots, n \; ; \; j = 1, 2, \dots . ., f_i \quad \text{..........(3)}$$

$$-S_{ij} + 1 \leq M_1 Y_{ij} \quad ; \; i = 1, 2, \dots \dots \dots, n \; ; \; j = 1, 2, \dots . ., f_i \quad \text{..........(4)}$$

$$Q_{ij} \leq M_1(1 - Y_{ij}) \quad ; \; i = 1, 2, \dots \dots \dots, n \; ; \; j = 1, 2, \dots . ., f_i \quad \text{..........(5)}$$

$$S_{ij} \leq Q_{ij} \quad ; \; i = 1, 2, \dots \dots \dots, n \; ; \; j = 1, 2, \dots . ., f_i \quad \text{..........(6)}$$

$$-e_{ijkl} + 1 \leq M_2 Z_{ijkl} \quad ; \; i, k = 1, 2, \dots, n \; ; \; j = 1, 2, . ., f_i \, ; l = 1, 2, . ., f_k \quad \text{........(7)}$$

$$T_{ij} - T_{kl} \leq M_2(1 - Z_{ijkl}) \quad ; \; i, k = 1, 2, \dots, n \; ; \; j = 1, 2, . ., f_i \, ; l = 1, 2, . ., f_k \quad \text{........(8)}$$

$e_{ijij} = 1 \qquad ; \ i = 1, 2, \dots, n \,; j = 1, 2, \dots, f_i$ ....(9)

$-X_{ij} + 1 \leq M_2 A_{ij} \qquad ; \ i = 1, 2, \dots, n \,; j = 1, 2, \dots, f_i$ ....(10)

$\sum_{k=1}^{n} \sum_{l=1}^{f_k} e_{ijkl} - \sum_{i=1}^{n} f_i + 1 \leq M_2 (1 - A_{ij}) \qquad ; \ i = 1, 2, \dots, n \ ; \ j = 1, 2, .., f_i$ ....(11)

$X_{ij} \leq M_3 B_{ij} \qquad ; \ i = 1, 2, \dots, n \,; j = 1, 2, \dots, f_i$ ....(12)

$-\sum_{k=1}^{n} \sum_{l=1}^{f_k} e_{ijkl} + \sum_{i=1}^{n} f_i \leq M_3 (1 - B_{ij}) \qquad ; \ i = 1, 2, \dots, n \ ; \ j = 1, 2, .., f_i$ ...(13)

$\sum_{k=1}^{n} \sum_{l=1}^{f_k} e_{ijkl} Q_{kl} \geq \sum_{k=1}^{n} \sum_{l=1}^{f_k} e_{ijkl} q_{kl} \,; \ i = 1, 2, \dots, n \,; j = 1, 2, \dots, f_i$ ....(14)

$\sum_{i=1}^{n} \sum_{j=1}^{f_i} Q_{ij} = \sum_{i=1}^{n} \sum_{j=1}^{f_i} q_{ij}$ ....(15)

$C \sum_{i=1}^{n} \sum_{j=1}^{f_i} S_{ij} + H \left[\left\{ \sum_{i=1}^{n} \sum_{j=1}^{f_i} X_{ij} T_{ij} \right\} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{f_i} \left(1 - X_{ij}\right)\left(Q_{ij} - q_{ij}\right) \right\} - \left\{ \sum_{i=1}^{n} \sum_{j=1}^{f_i} T_{ij} \left(1 - 2X_{ij}\right)\left(Q_{ij} - q_{ij}\right) \right\} \right] < \sqrt{2CDH}$ ....(16)

$\frac{d_i}{q_i} c_i^{'} + \frac{q_i}{2} h_i < \sqrt{2 c_i d_i h_i} \qquad ; \ i = 1, \dots\dots, n$ ....(17)

$q_{ij} = q_i \quad ; i = 1, 2, \dots., n \ ; j = 1, 2, \dots., f_i$ ....(18)

$M_1 = \sum_{i=1}^{n} d_i$ ....(19)

$M_2 = 1$ ....(20)

$M_3 = \sum_{i=1}^{n} f_i$ ....(21)

$S_{ij} \ , \ Y_{ij} \,, X_{ij} \,, \ A_{ij} \ , \ B_{ij} \ , \ e_{ijkl} \,, \ Z_{ijkl} \ , \ = 0 \ or \ 1 \quad ; Q_{ij} \,, \ q_i \ \geq 0$ ....(22)

Keterangan :

C : biaya pesan pemasok ke pihak ketiga

Q_{ij} : jumlah pesanan barang pemasok ke pihak ketiga untuk kebutuhan pengiriman barang kali ke j ke *retailer* i.

q_i : jumlah pengiriman barang ke *retailer* i.

T_{ij} : periode pengiriman barang kali ke-j ke *retailer* i.

f_i : frekuensi pengiriman barang ke *retailer* i dalam satu periode.

H : biaya simpan pemasok.

d_i : permintaan konsumen untuk *retailer* i dalam satu periode (contoh : setahun)

c'_i : biaya pesan/*order retailer* ike pemasok pada sistem VMI.

c_i : biaya pesan *retailer* i ke pemasok pada sistem tradisional.

h_i : biaya simpan *retailer* i.

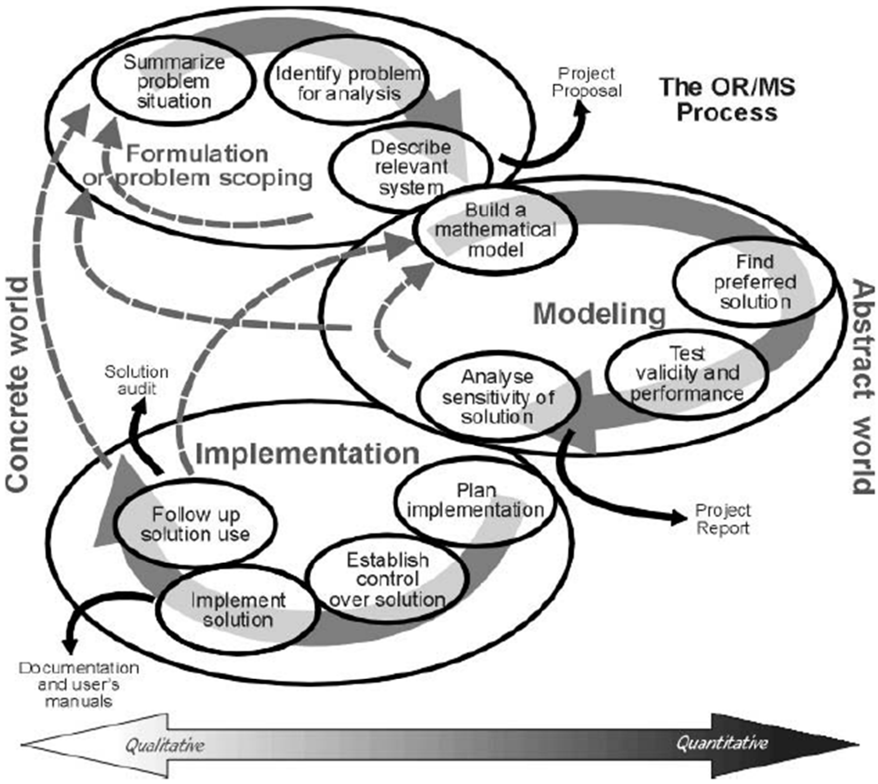
Variabel-variabel selain 9 variabel di atas merupakan variabel *dummy.*

Pada model matematis di atas dapat dilihat bahwa bentuk permasalahan adalah persoalan minimasi. Fungsi tujuan berisi biaya-biaya yang muncul dalam suatu sistem rantai pasok, yaitu biaya pemesanan dan biaya simpan. Oleh sebab itu proses optimalisasi dilakukan untuk memperoleh solusi/kebijakan penerapan VMI dengan total biaya terkecil. Biaya-biaya yang dilibatkan dalam fungsi tujuan merupakan biaya-biaya yang timbul dari pihak pemasok maupun *retailer.* Biaya-biaya dari kedua pihak ini disatukan ke dalam sebuah fungsi tujuan untuk dioptimalisasi secara bersamaan. Komponen pertama dalam fungsi tujuan adalah komponen biaya pemesanan barang yang dilakukan pemasok ke pihak ketiga. Komponen kedua adalah biaya simpan pemasok, sedangkan komponen ketiga dan keempat berturut-turut biaya pemesanan barang dan biaya simpan *retailer*-i. Biaya pesan/*order* barang yang dilakukan oleh *retailer* memang seharusnya tidak ada karena waktu pengiriman barang diatur sepenuhnya oleh pemasok. Namun, biaya yang termasuk ke dalam biaya *order* dalam sistem VMI ini adalah biaya *sharing* informasi mengenai tingkat persediaan *retailer* maupun jumlah permintaan konsumen*.* Dalam sistem VMI, informasi-informasi ini harus diinformasikan kepada pemasok secara berkala. Biaya pesan pada sistem VMI ini tentunya lebih kecil dibandingkan biaya pesan pada sistem tradisional*.*Inilah hal lain yang membedakan VMI dengan sistem rantai pasok tradisional.

Variabel-variabel keputusan dalam model rantai pasok ini adalah jumlah barang yang harus dikirimkan ke *retailer i* (q_i) dan jumlah barang yang harus dipesan pemasok untuk kebutuhan pengiriman barang kali ke-j ke *retailer* -i (Q_{ij}).Dengan mengetahui nilai variabel q_i, peneliti dapat mengetahui frekuensi dan periode pengiriman barang ke pemasok. Dalam model VMI ini pun dilibatkan konsep *game theory,* yaitu optimalisasi permasalahan VMI seharusnya menghasilkan solusi yang lebih baik dibandingkan penerapan sistem tradisional. Hal ini terlihat dari batasan permasalahan pada pertidaksamaan 16 dan 17. Penelitian-penelitian lain yang telah dilakukan memang menunjukkan bahwa sistem VMI ini lebih menguntungkan dibandingkan sistem tradisional.

II.2. Tahap-Tahap Pemodelan Sistem

Dalam proses pemodelan sistem, terdapat langkah-langkah yang harus dilakukan sehingga model yang dihasilkan dapat berfungsi dengan maksimal. Daellenbach dan McNickle (2005) memberikan panduan mengenai tahap-tahap pemodelan sistem yang secara khusus ditujukan untuk *Hard Operation Research* (*Hard OR*).Secara garis besar, terdapat 3 tahapan yang dapat dilakukan ketika seseorang dihadapkan pada permasalah *Hard OR,* yaitu formulasi permasalah dan penentuan batasan masalah, pemodelan permasalah, dan implementasi rekomendasi solusi. Setiap tahap ini dapat dijabarkan secara lebih detail seperti terlibat pada Gambar 1.



Gambar 1. Tahap-Tahap Pemodelan

(*Sumber : Daellenbach dan McNickle 2005, p. 115*)

Tahap pertama dalam pemodelan suatu sistem adalah formulasi permasalahan. Tahap ini dimulai dengan mengumpulkan informasi mengenai suatu sistem, membuat ringkasan mengenai kondisi suatu sistem, mengidentifikasi permasalahan yang terjadi dalam sistem, dan membatasi sistem yang akan diamati. Jenis hubungan antarentitas dalam sistem serta cakupan sistem yang diamati akan mempengaruhi kompleksitas model yang dikembangkan. Tahap selanjutnya adalah proses pemodelan permalahan. Proses pemodelan dapat dilakukan secara matematis seperti pada permasalahan VMI yang dihadapi. Model yang telah dikembangkan akandigunakan untuk mencari solusi dari permasalahan yang dihadapi. Namun, sebelum mengimplementasikan solusi yang didapatkan dari model, perlu dilakukan validasi terhadap model yang telah dibuat. Validasi dilakukan untuk mengetahui tingkat kesesuaian model dengan sistem nyata.

Daellenbach dan McNickle (2005) membagi proses validasi menjadi 2 fasa, yaitu validasi internal dan validasi eksternal.Validasi internal sering disebut juga verifikasi. Verifikasi merupakan proses yang dilakukan untuk memastikan setiap formula matematis dan logika yang diterapkan dalam model adalah benar. Selain itu verifikasi juga dilakukan untuk memeriksa apakah input data yang dimiliki digunakan dengan tepat pada model. Apabila suatu model diterjemahkan ke dalam bahasa pemrograman, maka verifikasi dilakukan untuk memastikan bahwa program yang telah dibuat memang sesuai dengan model matematis yang ada. Fasa kedua adalah validasi eksternal. Tahap ini dilakukan untuk memastikan bahwa model benar-benar dapat mewakili sistem nyata dan solusi yang dihasilkan dari model tersebut dapat digunakan dalam proses pengambilan keputusan untuk menyelesaikan masalah.

Setelah lolos tahap validasi, maka tahap selanjutnya adalah analisis sensitivitas. Tahap ini dilakukan untuk menguji tingkat sensitivitas model terhadap perubahan nilai-nilai parameter, atau dengan kata lain seberapa besar perubahan solusi optimal apabila terjadi perubahan pada nilai-nilai parameter. Pada tahap ini pun dapat dilakukan analisis *error* dengan tujuan mengetahui *robustness* dari suatu model. Model yang *robust* merupakan model yang tidak sensitif terhadap perubahan nilai parameter. Pergeseran solusi optimal dianggap tidak signifikan untuk interval tertentu perubahan nilai parameter.

Tahap terakhir adalah tahap implementasi solusi. Tahap ini dimulai dari perencanaan yaitu penentuan sumber daya yang akan digunakan dalam proses implementasi (tempat, waktu, manusia, dan lain lain). Langkah selanjutnya adalah merancang teknis pengendalian dan pengamatan hasil implementasi. Setelah semuanya siap, maka solusi dapat diterapkan. Hasil penerapan solusi akan dievaluasi untuk mengetahui kesesuaian setiap tahap yang telah dilakukan sebelumnya dengan kondisi sistem yang sebenarnya. Hasil evaluasi akan menjadi masukan bagi perbaikan setiap tahap dalam pemodelan sehingga masalah yang dihadapi dalam sistem benar-benar dapat diselesaikan.

BAB III
METODE PENELITIAN

Penelitian ini merupakan kelanjutan dari penelitian sebelumnya (Sitompul dan Alfian 2012) dimana tahapan yang sudah diselesaikan pada penelitian tersebut adalah identifikasi masalah dan penentuan tujuan, penentuan variabel keputusan, formulasi fungsi tujuan serta formulasi himpunan kendala.

Tahap-tahap yang akan dilakukan dalam penelitian ini dapat digambarkan pada Gambar 2 dan dijelaskan sebagai berikut.

1. Validasi Model

Tahap validasi model mencakup pemeriksaan ulang formulasi fungsi tujuan dan formulasi himpunan kendala dengan memperhatikan satuan (unit) variabel keputusan yang sudah dinyatakan. Di dalam tahapan validasi ini dimungkinkan pula adanya perbaikan atau modifikasi kendala untuk mendapatkan himpunan kendala yang lebih efisien agar dapat diselesaikan secara efisien.

2. Pengembangan perangkat lunak

Model yang telah divalidasi akan diimplementasikan ke dalam sebuah program yang disebut Lingo. Fungsi tujuan dan fungsi kendala yang nonlinear menjadikan model yang sudah dibangun tergolong ke dalam kategori *nonlinear programming.* Program Lingo dapat digunakan untuk menyelesaikan formulasi *nonlinear* selain formulasi linear yang sering dijumpai. Lingo yang akan digunakan adalah Lingo versi demo sehingga terdapat keterbatasan-keterbatasan penggunaan yang perlu diperhatikan.

3. Verifikasi perangkat lunak

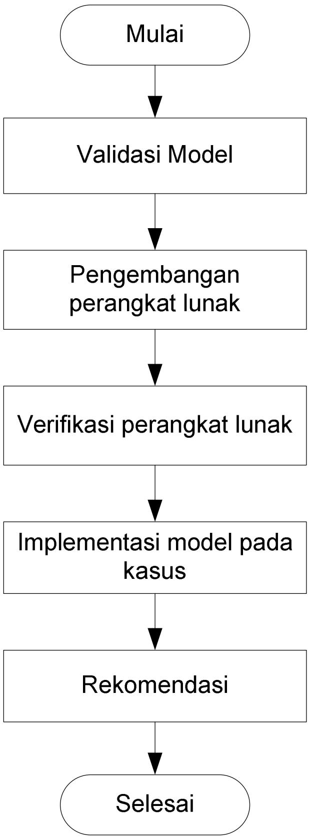
Verifikasi perangkat lunak dilakukan untuk memastikan bahwa formulasi matematis telah diterjemahkan dengan benar ke dalam perangkat lunak tersebut. Verifikasi perangkat lunak dilakukan dengan cara perintah Display Model di dalam Lingo. Dengan perintah ini, fungsi tujuan dan fungsi kendala dapat diperiksa secara manual sehingga verifikasi dapat dilakukan lebih teliti dan tepat.

4. Penerapan model pada kasus

Tahap berikutnya di dalam penelitian ini adalah menerapkan perangkat lunak yang dikembangkan untuk menyelesaikan beberapa kasus yang tersedia atau kasus yang akan dirancang mandiri. Penerapan model pada kasus harus menunjukkan bahwa keputusan yang diambil logis dan selaras dengan variabel keputusan yang dinyatakan di dalam model perangkat lunak.

5. Rekomendasi

Tahapan akhir dari penelitian ini adalah merekomendasikan penggunaan model untuk dimanfaatkan ke dalam kondisi hubungan pemasok (*supplier*) dan *retailer* yang sesuai.



Gambar 2. Diagram Alir Tahapan Penelitian

BAB IV
JADWAL PELAKSANAAN

Penelitian ini akan dilakukan di tahun anggaran 2013 dengan mengikuti jadwal pelaksanaan sebagai berikut.

Tabel 1. Jadwal Pelaksanaan Penelitian Tahun 2013

	Maret	April	Mei	Juni	Juli	Agustus	September	Oktober	November
Persiapan Penelitian									
Studi literatur									
Validasi model									
Pengembangan perangkat									
Verifikasi									
Rekomendasi									
Penyusunan laporan									

BAB V
HASIL DAN PEMBAHASAN

Pada Bab ini akan dibahas hasil pemodelan matematis yang telah dilakukan untuk permasalahan yang dihadapi. Adapun model matematis yang dikembangkan adalah perbaikan dari model yang telah dikembangkan sebelumnya oleh Sitompul dan Alfian (2012). Selain itu akan dilakukan juga verifikasi model serta implementasi model pada beberapa kasus hipotetis.

V.1. Pemodelan Masalah

Model matematis pada penelitian sebelumnya (Sitompul dan Alfian 2012) dianggap kurang efisien. Hal ini ditunjukkan dengan banyaknya variabel *dummy* yang digunakan. Salah satu penyederhanaan yang dilakukan adalah pada fungsi tujuan. Fungsi tujuan pada masalah *Vendor Managed Inventory* (VMI) adalah total biaya sistem persediaan yang harus ditanggung pemasok dan *retailer.* Fungsi tujuan pada persamaan 1 di Bab II dapat dibagi menjadi 2 bagian besar yaitu biaya pemasok dan biaya *retailer.* Biaya pemasok dapat dibagi lagi menjadi 2 komponen yaitu biaya pesan dan biaya penyimpanan. Biaya pemesanan dibebani pada pemasok setiap kali pemasok melakukan pemesanan barang kepada pihak ketiga dengan biaya pesan sebesar C, sedangkan biaya penyimpanan menjadi beban pemasok sebesar H /unit/periode akibat sejumlah barang yang dimiliki pemasok di gudang per periodenya. Adapun kedua biaya pemasok pada pemodelan terdahulu adalah sebagai berikut.

$$\text{Biaya pemasok} = C\sum_{i=1}^{n}\sum_{j=1}^{f_i} S_{ij} + H\left[\left\{\sum_{i=1}^{n}\sum_{j=1}^{f_i} X_{ij}T_{ij}\right\}\left\{\sum_{i=1}^{n}\sum_{j=1}^{f_i}\left(1-X_{ij}\right)\left(Q_{ij}-q_{ij}\right)\right\} - \left\{\sum_{i=1}^{n}\sum_{j=1}^{f_i} T_{ij}\left(1-2X_{ij}\right)\left(Q_{ij}-q_{ij}\right)\right\}\right] \quad \text{.......}(23)$$

Keterangan lengkap untuk setiap notasi pada persamaan 23 dapat dilihat pada Bab II. Penyederhanaan model matematis yang dilakukan adalah pada komponen kedua fungsi biaya pemasok, yaitu biaya penyimpanan. Adapun hasil perbaikan pada komponen kedua ini adalah sebagai berikut.

$$\text{Biaya simpan} = H\left[\sum_{i=1}^{n}\sum_{j=1}^{f_i}\frac{q_i}{d_i}\left(Q_{ij}-q_i\right)\left(f_i-j\right)\right] \quad \text{.......}(24)$$

Pada persamaan 24 dapat dilihat bahwa bentuk model matematis menjadi lebih sederhana. Pada model biaya persediaan di persamaan 23 dan 24 ini terdapat 2 hal yang diperhitungkan untuk mendapatkan biaya persediaan yaitu banyak barang yang disimpan dan lama barang tersebut disimpan. Banyak barang yang disimpan adalah sebesar $Q_{ij} – q_i$. Notasi ini terdapat pada persamaan 23 dan 24. Walaupun variabel q pada persamaan 23 memiliki dua indeks, namun arti dari variabel ini sama dengan yang terdapat pada persamaan 24. Hal ini sekali lagi menunjukkan ketidakefisienan pemodelan matematis di penelitian sebelumnya. Adapun $Q_{ij} – q_i$ menunjukkan sisa jumlah barang hasil pemesanan ke pihak ketiga untuk pengiriman barang ke *retailer-i* kali ke-j. Total lama penyimpanan barang pada persamaan 23 didapatkan dari hasil pengurangan nilai T_{ij} yang melibatkan variabel *dummy* X_{ij}, sedangkan pada persamaan 24 lama penyimpanan didapatkan dengan mengalikan q_i/d_i dengan $(f_i – j)$ tanpa harus melibatkan variabel *dummy.* Sisa barang pemesanan $(Q_{ij}-q_i)$ pada persamaan 24 tidak selalu bernilai positif. Nilai negatif dimungkinkan ketika pengiriman barang ke *retailer* i kali ke-j diambil dari barang yang telah dipesan di periode sebelumnya (pemesanan barang ke pihak ketiga untuk pengiriman barang kali ke-j ke *retailer-i* adalah sebesar nol) sehingga ini akan mengurangi lama simpan sisa barang pemesanan di periode sebelumnya. Fungsi biaya persediaan di pihak *retailer* tetap sama dengan pemodelan sebelumnya.

Perbaikan lainnya adalah pada pemodelan batasan masalah pertidaksamaan 4 sampai 6. Ketiga pertidaksamaan ini dimaksudkan sebagai tanda/status adanya pemesanan barang yang dilakukan pemasok ke pihak ketiga. Variabel *dummy* S_{ij} akan bernilai 1 apabila ada pemesanan dan bernilai nol jika tidak ada pemesanan. Nilai S_{ij} sebesar 1 akan membebankan biaya pesanan kepada pihak pemasok. Dengan tujuan yang sama, pertidaksamaan 4 sampai 6 bisa digantikan dengan hanya 1 pertidaksamaan saja yaitu

$$Q_{ij} \leq M \times S_{ij} \quad ,S_{ij} = 0 \; or \; 1, \; M = \sum_{i=1}^{n} d_i \quad \text{...(25)}$$

Perbaikan selanjutnya dilakukan pada pertidaksamaan 7 sampai 13. Pada pemodelan sebelumnya, pertidaksamaan 7 sampai 13 digunakan untuk mendukung fungsi tujuan biaya simpan pada pemasok dengan mencari periode pemesanan terpanjang. Namun, penggunaan fungsi biaya simpan yang baru menyebabkan pertidaksamaan 7 dan 13 ini tidak diperlukan lagi.

Pertidaksamaan selanjutnya yang akan dibahas adalah pertidaksamaan 14 dan 15. Kedua pertidaksamaan ini dimaksudkan untuk menjamin terpenuhinya kebutuhan pengiriman ke setiap *retailer*. Namun, persamaan 15 redundan dengan pertidaksamaan 14 apabila pertidaksamaan 14 ini mencapai nilai indeks i dan j yang terakhir. Kelemahan lainnya dari pemodelan sebelumnya adalah tidak adanya jaminan terpenuhinya kebutuhan konsumen *retailer* tertentu per periodenya. Oleh sebab itu pertidaksamaan dan persamaan 14 dan 15 akan diperbaiki dengan 2 pertidaksamaan sebagai berikut.

$\sum_{k=1}^{j} Q_{ik} \geq j \times q_i \qquad , \forall\ i = 1, \dots, n;\ \ j = 1, \dots, f_i$ ....(26)

$f_i \times q_i \geq d_i \qquad , \forall\ i = 1, \dots, n$ ....(27)

Seperti pemodelan sebelumnya, pertidaksamaan 26 digunakan untuk menjamin terpenuhinya kebutuhan pengiriman barang ke *retailer i* kali ke-j melalui pemesanan-pemesanan yang dilakukan hingga waktu pengiriman tersebut. Sebagai contoh, untuk *retailer* 1 frekuensi pengiriman dalam satu tahun adalah sebanyak 3 kali, maka pertidaksamaan yang terbentuk berdasarkan formulasi matematis pertidaksamaan 26 adalah sebagai berikut.

$Q_{11} \geq 1 \times q_1$ ....(28)

$Q_{11} + Q_{12} \geq 2 \times q_1$ ....(29)

$Q_{11} + Q_{12} + Q_{13} \geq 3 \times q_1$ ....(30)

Nilai q_1 mewakili jumlah pengiriman barang ke *retailer* 1 setiap periodenya dengan jumlah yang konstan. Nilai frekuensi pengiriman adalah berupa bilangan bulat. Penggunaan pertidaksamaan 27 dalam contoh kasus yang sama adalah sebagai berikut.

$3 \times q_1 \geq d_1$..(31)

Pertidaksamaan 31 akan membuat total jumlah pengiriman barang selama satu tahun memenuhi permintaan barang konsumen *retailer* 1 selama tahun tersebut (d_1).

Pertidaksamaan berikutnya yang dilibatkan dalam pemodelan masalah VMI ini adalah pertidaksamaan 16 dan 17 yang digunakan untuk memastikan bahwa solusi VMI yang didapatkan memiliki nilai fungsi tujuan lebih kecil dari sistem tradisional. Dalam perbaikan yang dilakukan, kedua pertidaksamaan ini tidak digunakan lagi. Sistem VMI yang telah dimodelkan justru akan digunakan untuk melihat apakah solusi yang dihasilkan memang akan lebih rendah dari total biaya sistem tradisional secara keseluruhan. Studi literatur yang dilakukan terhadap penelitian-penelitian VMI dengan melibatkan 1 pemasok dan 1 *retailer* menyatakan bahwa solusi VMI lebih baik dibandingkan dengan sistem tradisional, namun untuk kasus yang melibatkan lebih dari 1 *retailer* belum dapat diketahui. Oleh sebab itu, akan diuji apakah formulasi matematis yang telah dibangun untuk permasalahan VMI dengan banyak *retailer* mampu menghasilkan solusi yang lebih baik.

Berdasarkan perbaikan model matematis beserta pembahasan yang dipaparkan, dapat dijabarkan formulasi matematis sistem VMI dengan 1 pemasok dan banyak *retailer* sebagai berikut.

Minimize

$$Z = C\sum_{i=1}^{n}\sum_{j=1}^{f_i} S_{ij} + H\left[\sum_{i=1}^{n}\sum_{j=1}^{f_i}\frac{q_i}{d_i}\left(Q_{ij}-q_i\right)(f_i-j)\right] + \sum_{i=1}^{n}\left(c_i'\frac{d_i}{q_i}+\frac{q_i}{2}h_i\right) \quad \text{...(32)}$$

Subject to

$$f_i \times q_i \geq d_i \quad ,\forall\, i = 1,\ldots,n \quad \text{...(33)}$$

$$\sum_{k=1}^{j} Q_{ik} \geq j \times q_i \quad ,\forall\, i = 1,\ldots,n;\; j = 1,\ldots,f_i \quad \text{...(34)}$$

$$Q_{ij} \leq M \times S_{ij} \quad ,\forall\, i = 1,\ldots,n;\; j = 1,\ldots,f_i;\quad M = \sum_{i=1}^{n} d_i\,, \quad \text{...(35)}$$

$$S_{ij} = 0 \text{ or } 1,\; Q_{ij} = Q_{ik} \geq 0;\; q_i \geq 0;\; f_i \geq 1$$

Keterangan :

C : biaya pesan pemasok ke pihak ketiga

$Q_{ij} = Q_{ik}$: jumlah pesanan barang pemasok ke pihak ketiga untuk kebutuhan pengiriman barang kali ke j (k) ke *retailer* i.

q_i : jumlah pengiriman barang ke *retailer* i.

f_i : frekuensi pengiriman barang ke *retailer* i dalam satu periode.

H : biaya simpan pemasok.

d_i : permintaan konsumen untuk *retailer* i dalam satu periode (contoh : setahun)

c'_i : biaya pesan/*order retailer* ike pemasok pada sistem VMI.

c_i : biaya pesan *retailer* i ke pemasok pada sistem tradisional.

h_i : biaya simpan *retailer* i.

V.2. Pengembangan Kasus Hipotetis

Untuk menguji validitas model sekaligus mengetahui bagaimana karakteristik solusi yang dihasilkan untuk suatu permasalahan, model yang telah dikembangkan perlu diujicobakan pada kasus-kasus yang relevan. Untuk keperluan ini, akan dikembangkan beberapa kasus hipotetis VMI dengan mempertimbangkan batasan *tools* penyelesaian masalah yang akan digunakan nantinya. Terdapat 3 kasus hipotetis yang dikembangkan untuk diterapkan pada model optimalisasi VMI. Kasus-kasus ini melibatkan 2, 3, dan 4 *retailer* dengan detail seperti pada Tabel 2 sampai 4.

Tabel 2. Kasus VMI 2 *Retailer*

Retailer			
Biaya	**1**	**2**	**Satuan**
D	5000	3000	/ tahun
c'	30	15	/ pemesanan
c	45	20	/ pemesanan
h	0.6	1	/unit/thn
Supplier			
C	45		/ pemesanan
H	0.5		/unit/thn

Tabel 3. Kasus VMI 3 *Retailer*

Retailer				
Biaya	**1**	**2**	**3**	**Satuan**
D	5000	3000	5000	/ tahun
c'	30	15	20	/ pemesanan
c	50	20	30	/ pemesanan
h	1.5	0.6	0.7	/unit/thn
Supplier				
C	50			/ pemesanan
H	0.4			/unit/thn

Tabel 4. Kasus VMI 4 *Retailer*

Retailer					
Biaya	**1**	**2**	**3**	**4**	**Satuan**
D	2000	1500	5000	7500	/ tahun
c'	30	15	20	20	/ pemesanan
c	45	20	30	35	/ pemesanan
h	1.5	1	0.3	0.35	/unit/thn
Supplier					
C	50				/ pemesanan
H	0.3				/unit/thn

Tabel 2 sampai 4 berisi informasi besar setiap komponen biaya yang relevan dalam penyelesaian kasus VMI ini. Adapun keterangan setiap notasi jenis biaya pada tabel-tabel di atas adalah sebagai berikut.

D : permintaan barang per tahun

c' : biaya pesan *retailer* dengan menggunakan sistem VMI

c : biaya pesan *retailer* tanpa penerapan VMI

h : biaya simpan *retailer* per unit barang per tahun

C : biaya pesan *supplier* ke pihak ketiga

H : biaya simpan *supplier* per unit barang per tahun

V.3. Penerjemahan Model ke Perangkat Lunak Lingo

Model matematis yang telah dibangun akan diselesaikan dengan bantuan perangkat lunak matematis, untuk itu model matematis ini harus terlebih dahulu diterjemahkan ke dalam bahasa pemrograman yang digunakan dalam perangkat lunak tersebut. Perangkat lunak yang pada awalnya akan digunakan adalah AMPL. Namun, karena keterbatasan yang dihadapi pada akhirnya digunakan perangkat lunak lain yaitu Lingo. Lingo adalah perangkat lunak yang sebenarnya memiliki fungsi sama dengan AMPL untuk menyelesaikan permasalahan optimalisasi. Sayangnya, Lingo yang digunakan pada penelitian ini hanyalah versi demo. Lingo versi demo memiliki batasan besar masalah yang dapat diselesaikan meliputi jumlah variabel dan jumlah batasan permasalahan dalam model bersangkutan. Penerjemahan model ke dalam bahasa Lingo yang telah dilakukan sampai saat ini terbatas untuk setiap kasus hipotetis yang akan diselesaikan atau dengan kata lain hasil penerjemahan model ini belum bisa secara langsung menyesuaikan dengan besar masalah yang dihadapi.

Hasil penerjemahan model yang telah dilakukan sampai saat ini memang disesuaikan dengan 3 kasus hipotetis yang telah dibangun yaitu permasalahan VMI dengan 2, 3 dan 4 *retailer.* Di bawah ini dapat dilihat bahasa Lingo hasil penerjemahan model optimalisasi VMI untuk semua kasus hipotetis yang dibuat.

1. VMI 2 retailer

```
data:
 f1 =4;
 f2 =8;
enddata
```

```
sets:
 !1= retailer1, 2=retailer2;
 frek1/1..f1/;
 frek2/1..f2/;
 pesan1/1..f1/:jml1,status1;
 pesan2/1..f2/:jml2,status2;
endsets

data:
!Supplier;
 Cs = 45;
 Hs = 0.5;
!Retailer1;
 D1 = 5000;
 c1v = 30; !order cost vmi retailer 1;
 c1 = 45;
 h1 = 0.6;
!Retailer2;
 D2 = 3000;
 c2v = 15; !order cost vmi retailer 2;
 c2 = 20;
 h2 = 1;
enddata

min = Cs*(@sum(pesan1:status1)+@sum(pesan2:status2))+
      (@sum(pesan1(J):(jml1(J)-q1)*(f1-J))*(q1/D1)+
   @sum(pesan2(J):(jml2(J)-q2)*(f2-J))*(q2/D2))*Hs +
      (c1v*D1/q1+q1/2*h1)+(c2v*D2/q2+q2/2*h2);

!batasan pemenuhan demand selama 1 tahun;
f1*q1 >= D1;
f2*q2 >= D2;

!batasan pemenuhan kebutuhan pengiriman;
@for(frek1(I):@sum(pesan1(J)|J#LE#I:jml1(J))>=I*q1);
@for(frek2(I):@sum(pesan2(J)|J#LE#I:jml2(J))>=I*q2);

!if then contraint;
@for(pesan1:jml1<=(D1+D2)*status1);
@for(pesan2:jml2<=(D1+D2)*status2);

!sign constraint;
@for(pesan1:@bin(status1));
@for(pesan2:@bin(status2));
@for(pesan1:jml1 >=0);
@for(pesan2:jml2 >=0);
q1>=0; q2>=0;
```

Pada hasil penerjemahan model optimalisasi di atas, terlihat bahwa pertama-tama dibutuhkan parameter berupa frekuensi pengiriman barang ke *retailer i*. Hal ini ditunjukan oleh blok ‘data’ yang berisi parameter f1 dan f2, dimana f1 menyatakan jumlah frekuensi pengiriman barang ke *retailer* 1 sedangkan f2 adalah frekuensi pengiriman barang ke *retailer* 2. Blok selanjutnya yaitu blok ‘sets’ merupakan pendeklarasian variabel-variabel yang terlibat dalam model matematis. Deklarasi frek1 dan frek2 dilakukan untuk mendefinisikan indeks yang akan digunakan pada fungsi tujuan ataupun batasan masalah berkaitan dengan frekuensi pengiriman barang ke *retailer* 1 dan 2. Nilai indeks dimulai dari angka 1 (kali pertama pengiriman) hingga mencapai nilai fi yang telah didefinisikan sebelumnya di blok ‘data’. Deklarasi lainnya dalam blok ini adalah pesan1 dan pesan2. Pesan1 dan pesan2 digunakan untuk mendeklarasikan variabel yang terlibat di setiap persamaan dan pertidaksamaan untuk *retailer* 1 dan 2. Terdapat variabel jml1 dan status1 untuk *retailer* 1. Jml1 mewakili jumlah barang yang dipesan oleh pemasok ke pihak ketiga untuk memenuhi kebutuhan pengiriman barang ke *retailer* 1 pada kali ke i. Nilai i ini dimulai dari 1 hingga f1 yang didefinisikan dalam tanda garis miring (/../). Variabel status1 merupakan variabel dengan nilai 0 atau 1, dimana nilai 1 menandakan terjadinya pemesanan barang ke pihak ketiga pada periode pengiriman barang yang ke i ke *retailer* 1. Nilai i dimulai dari 1 hingga f1. Nilai variabel status ini digunakan untuk membebankan biaya pesan pada pemasok. Interpretasi yang sama dapat dilakukan untuk deklarasi pesan2 yang ditujukan ke *retailer* 2. Terdapat 1 variabel lagi yang terlibat dalam model, yaitu variabel q. Terdapat 2 variabel q yaitu q1 dan q2 yang mewakili pengiriman barang dengan jumlah konstan ke *retailer* 1 dan 2. Variabel ini hanya ada 2 sehingga tidak perlu didefinisikan di blok ‘sets’ dengan maksud mengotomatisasi proses *generating* variabel.

Blok selanjutnya adalah blok ‘data’. Blok ‘data’ yang kedua ini digunakan untuk mendeklarasikan nilai parameter jumlah permintaan per periode dan nilai biaya-biaya yang terlibat dalam model optimalisasi VMI. Parameter jumlah permintaan diwakili oleh D1 dan D2. Jumlah permintaan yang dimaksudkan di sini berasal dari permintaan barang konsumen *retailer* 1 dan 2 per periodenya. Parameter selanjutnya adalah biaya pesan dan biaya simpan. Biaya-biaya ini dibagi menjadi biaya untuk pemasok (*supplier*), biaya untuk *retailer* 1, dan biaya untuk *retailer* 2. Parameter Cs dan c_i mewakili biaya pesan untuk pemasok dan *retailer*. Terdapat 2 jenis biaya pesan bagi *retailer* yaitu c_i dan c_iv. Parameter c_i mewakili biaya pesan bagi *retailer* i dalam sistem tradisional sedangkan c_iv merupakan biaya pesan bagi *retailer* i yang menggunakan sistem VMI. Biaya lainnya adalah biaya simpan. Besar biaya simpan untuk pemasok dan *retailer* masing-masing didefinisikan dalam parameter Hs dan h_i.

Blok-blok program selanjutnya merupakan blok penerjemahan model matematis yang telah dibuat. Blok program dimulai dari fungsi tujuan yaitu meminimalisasi total biaya persediaan pemasok dan seluruh *retailer.* Tiga blok selanjutnya merupakan batasan permasalahan VMI. Blok pertama adalah batasan yang memastikan terpenuhinya permintaan konsumen *retailer* i per periode dengan kebijakan pengiriman yang diterapkan. Blok kedua merupakan batasan yang menjamin terpenuhinya kebutuhan pengiriman ke setiap *retailer* oleh kebijakan pemesanan barang yang diterapkan pemasok, sedangkan blok ketiga adalah batasan untuk membebankan biaya pesan pada pemasok apabila memang terjadi pemesanan pada periode tertentu. Apabila diamati, setiap blok dari 3 blok batasan ini terdiri dari 2 baris batasan masalah. Masing-masing baris merupakan batasan untuk *retailer* 1 dan 2. Oleh sebab itu, apabila terdapat tambahan 1 *retailer* lagi dalam permasalahan yang dihadapi, perlu ditambahkan 1 baris batasan di setiap blok program batasan masalah yang ditujukan kepada *retailer* 3. Tidak hanya pada blok batasan masalah, tambahan-tambahan lain yang mengacu pada *retailer* 3 pun perlu ditambahkan pada blok-blok lainnya. *Blok* program terakhir dari penerjemahan model ke dalam program Lingo adalah blok *sign constraint.* Blok program ini digunakan untuk mendefinisikan rentang nilai yang mungkin bagi setiap variabel yang terlibat. Batasan rentang nilai untuk setiap variabel adalah minimal sama dengan 0, kecuali untuk variabel status1 dan status2 yang merupakan *binary* variabel dengan nilai 0 atau 1.

2. VMI 3 retailer

```
data:
 f1 =9;
 f2 =4;
 f3 =6;
enddata
```

```
sets:
 !1= retailer1, 2=retailer2;
 frek1/1..f1/;
 frek2/1..f2/;
 frek3/1..f3/;
 pesan1/1..f1/:jml1,status1;
 pesan2/1..f2/:jml2,status2;
 pesan3/1..f3/:jml3,status3;
endsets

data:
!Supplier;
 Cs = 50;
 Hs = 0.4;
!Retailer1;
 D1 = 5000;
 c1v = 30; !order cost vmi retailer 1;
 c1 = 50;
 h1 = 1.5;
!Retailer2;
 D2 = 3000;
 c2v = 15; !order cost vmi retailer 2;
 c2 = 20;
 h2 = 0.6;
!Retailer3;
 D3 = 5000;
 c3v = 20; !order cost vmi retailer 3;
 c3 = 30;
 h3 = 0.7;

enddata

min =
Cs*(@sum(pesan1:status1)+@sum(pesan2:status2)+@sum(pesan3:statu
s3))+
      (@sum(pesan1(J):(jml1(J)-q1)*(f1-J))*(q1/D1)+
@sum(pesan2(J):(jml2(J)-q2)*(f2-
J))*(q2/D2)+@sum(pesan3(J):(jml3(J)-q3)*(f3-J))*(q3/D3))*Hs +

(c1v*D1/q1+q1/2*h1)+(c2v*D2/q2+q2/2*h2)+(c3v*D3/q3+q3/2*h3);

!batasan pemenuhan demand selama 1 tahun;
f1*q1 >= D1;
f2*q2 >= D2;
f3*q3 >= D3;

!batasan pemenuhan kebutuhan pengiriman;
@for(frek1(I):@sum(pesan1(J)|J#LE#I:jml1(J))>=I*q1);
@for(frek2(I):@sum(pesan2(J)|J#LE#I:jml2(J))>=I*q2);
@for(frek3(I):@sum(pesan3(J)|J#LE#I:jml3(J))>=I*q3);


!if then contraint;
@for(pesan1:jml1<=(D1+D2+D3)*status1);
@for(pesan2:jml2<=(D1+D2+D3)*status2);
```

```
@for(pesan3:jml3<=(D1+D2+D3)*status3);

!sign constraint;
@for(pesan1:@bin(status1));
@for(pesan2:@bin(status2));
@for(pesan3:@bin(status3));
@for(pesan1:jml1 >=0);
@for(pesan2:jml2 >=0);
@for(pesan3:jml3 >=0);
q1>=0; q2>=0; q3>=0;
```

3. VMI 4 retailer

```
data:
 f1 =6;
 f2 =6;
 f3 =5;
 f4 =7;
enddata

sets:
 !1= retailer1, 2=retailer2;
 frek1/1..f1/;
 frek2/1..f2/;
 frek3/1..f3/;
 frek4/1..f4/;
 pesan1/1..f1/:jml1,status1;
 pesan2/1..f2/:jml2,status2;
 pesan3/1..f3/:jml3,status3;
 pesan4/1..f4/:jml4,status4;
endsets

data:
!Supplier;
 Cs = 50;
 Hs = 0.3;
!Retailer1;
 D1 = 2000;
 c1v = 30; !order cost vmi retailer 1;
 c1 = 45;
 h1 = 1.5;
!Retailer2;
 D2 = 1500;
 c2v = 15; !order cost vmi retailer 2;
 c2 = 20;
 h2 = 1;
!Retailer3;
 D3 = 5000;
 c3v = 20; !order cost vmi retailer 3;
 c3 = 30;
 h3 = 0.3;

!Retailer4;
 D4 = 7500;
 c4v = 20; !order cost vmi retailer 4;
```

```
 c4 = 35;
 h4 = 0.35;
enddata

min =
Cs*(@sum(pesan1:status1)+@sum(pesan2:status2)+@sum(pesan3:statu
s3)+@sum(pesan4:status4))+
      (@sum(pesan1(J):(jml1(J)-q1)*(f1-J))*(q1/D1)+
@sum(pesan2(J):(jml2(J)-q2)*(f2-J))*(q2/D2)+
       @sum(pesan3(J):(jml3(J)-q3)*(f3-J))*(q3/D3)+
@sum(pesan4(J):(jml4(J)-q4)*(f4-J))*(q4/D4))*Hs +
(c1v*D1/q1+q1/2*h1)+(c2v*D2/q2+q2/2*h2)+(c3v*D3/q3+q3/2*h3)+(c4
v*D4/q4+q4/2*h4);

!batasan pemenuhan demand selama 1 tahun;
f1*q1 >= D1;
f2*q2 >= D2;
f3*q3 >= D3;
f4*q4 >= D4;

!batasan pemenuhan kebutuhan pengiriman;
@for(frek1(I):@sum(pesan1(J)|J#LE#I:jml1(J))>=I*q1);
@for(frek2(I):@sum(pesan2(J)|J#LE#I:jml2(J))>=I*q2);
@for(frek3(I):@sum(pesan3(J)|J#LE#I:jml3(J))>=I*q3);
@for(frek4(I):@sum(pesan4(J)|J#LE#I:jml4(J))>=I*q4);

!if then contraint;
@for(pesan1:jml1<=(D1+D2+D3+D4)*status1);
@for(pesan2:jml2<=(D1+D2+D3+D4)*status2);
@for(pesan3:jml3<=(D1+D2+D3+D4)*status3);
@for(pesan4:jml4<=(D1+D2+D3+D4)*status4);

!sign constraint;
@for(pesan1:@bin(status1));
@for(pesan2:@bin(status2));
@for(pesan3:@bin(status3));
@for(pesan4:@bin(status4));

@for(pesan1:jml1 >=0);
@for(pesan2:jml2 >=0);
@for(pesan3:jml3 >=0);
@for(pesan4:jml4 >=0);
q1>=0; q2>=0; q3>=0; q4>=0;
```

Seperti telah disinggung dalam penjelasan yang diberikan mengenai hasil penerjemahan model dengan 2 *retailer,* perbedaan antara kode program pada kasus 2 *retailer* dengan 3 atau 4 *retailer* terletak pada penambahan data, komponen fungsi tujuan, dan batasan masalah untuk *retailer* ketiga dan keempat. Namun, penambahan setiap komponen ini untuk sementara masih harus dilakukan secara manual sehingga apabila akan diterapkan untuk kasus yang cukup besar, penyesuaian yang dilakukan akan cukup banyak.

V.4. Verifikasi Program Lingo

Verifikasi program dilakukan untuk memastikan bahwa model matematis yang telah dibangun dengan tepat diterjemahkan ke dalam bahasa Lingo. Ukuran yang dijadikan bahan penilaian kesesuaian model matematis dengan hasil pemrograman adalah kesamaan fungsi tujuan dan batasan-batasan yang di-*generate* oleh Lingo berdasarkan input permasalahan. Dalam Lingo hal ini dapat dilakukan dengan menggunakan perintah **Display Model** dalam menu **Lingo ➔ Generate**. Melalui perintah ini, akan ditunjukan model matematis lengkap yang dihasilkan dari program Lingo. Berdasarkan kasus hipotetis VMI 2 *retailer* yang telah diterapkan pada program Lingo, didapatkan model matematis lengkap sebagai berikut.

```
MIN     ? Q1 + ? Q2 + ? JML2( 1) + 45 STATUS2( 1) + ? JML2( 2)
      + 45 STATUS2( 2) + ? JML2( 3) + 45 STATUS2( 3) + ? JML2( 4)
      + 45 STATUS2( 4) + ? JML2( 5) + 45 STATUS2( 5) + ? JML2( 6)
      + 45 STATUS2( 6) + ? JML2( 7) + 45 STATUS2( 7) + 45 STATUS2(
8)
      + ? JML1( 1) + 45 STATUS1( 1) + ? JML1( 2) + 45 STATUS1( 2)
      + ? JML1( 3) + 45 STATUS1( 3) + 45 STATUS1( 4)


 SUBJECT TO
 2]  4 Q1 >=   5000
 3]  8 Q2 >=   3000
 4]- Q1 + JML1( 1) >=   0
 5]- 2 Q1 + JML1( 1) + JML1( 2) >=   0
 6]- 3 Q1 + JML1( 1) + JML1( 2) + JML1( 3) >=   0
 7]- 4 Q1 + JML1( 1) + JML1( 2) + JML1( 3) + JML1( 4) >=   0
 8]- Q2 + JML2( 1) >=   0
 9]- 2 Q2 + JML2( 1) + JML2( 2) >=   0
 10]- 3 Q2 + JML2( 1) + JML2( 2) + JML2( 3) >=   0
 11]- 4 Q2 + JML2( 1) + JML2( 2) + JML2( 3) + JML2( 4) >=   0
 12]- 5 Q2 + JML2( 1) + JML2( 2) + JML2( 3) + JML2( 4) + JML2( 5)
      >=   0
 13]- 6 Q2 + JML2( 1) + JML2( 2) + JML2( 3) + JML2( 4) + JML2( 5)
      + JML2( 6) >=   0
 14]- 7 Q2 + JML2( 1) + JML2( 2) + JML2( 3) + JML2( 4) + JML2( 5)
      + JML2( 6) + JML2( 7) >=   0
 15]- 8 Q2 + JML2( 1) + JML2( 2) + JML2( 3) + JML2( 4) + JML2( 5)
      + JML2( 6) + JML2( 7) + JML2( 8) >=   0
 16]  JML1( 1) - 8000 STATUS1( 1) <=   0
 17]  JML1( 2) - 8000 STATUS1( 2) <=   0
 18]  JML1( 3) - 8000 STATUS1( 3) <=   0
 19]  JML1( 4) - 8000 STATUS1( 4) <=   0
 20]  JML2( 1) - 8000 STATUS2( 1) <=   0
 21]  JML2( 2) - 8000 STATUS2( 2) <=   0
 22]  JML2( 3) - 8000 STATUS2( 3) <=   0
 23]  JML2( 4) - 8000 STATUS2( 4) <=   0
 24]  JML2( 5) - 8000 STATUS2( 5) <=   0
 25]  JML2( 6) - 8000 STATUS2( 6) <=   0
```

```
 26]  JML2( 7) - 8000 STATUS2( 7) <=   0
 27]  JML2( 8) - 8000 STATUS2( 8) <=   0
 28]  JML1( 1) >=   0
 29]  JML1( 2) >=   0
 30]  JML1( 3) >=   0
 31]  JML1( 4) >=   0
 32]  JML2( 1) >=   0
 33]  JML2( 2) >=   0
 34]  JML2( 3) >=   0
 35]  JML2( 4) >=   0
 36]  JML2( 5) >=   0
 37]  JML2( 6) >=   0
 38]  JML2( 7) >=   0
 39]  JML2( 8) >=   0
 40]  Q1 >=   0
 41]  Q2 >=   0
 END
 INTE STATUS2( 1)
 INTE STATUS2( 2)
 INTE STATUS2( 3)
 INTE STATUS2( 4)
 INTE STATUS2( 5)
 INTE STATUS2( 6)
 INTE STATUS2( 7)
 INTE STATUS2( 8)
 INTE STATUS1( 1)
 INTE STATUS1( 2)
 INTE STATUS1( 3)
 INTE STATUS1( 4)
```

Berdasarkan model matematis lengkap di atas, dapat diketahui apakah model yang telah diterjemahkan ke dalam bahasa Lingo sesuai dengan yang diinginkan. Pada paparan model matematis di atas, baris pertama menunjukkan fungsi tujuan permasalahan dimana biaya yang dipertimbangkan dalam fungsi tujuan sudah melibatkan biaya-biaya dari kedua *retailer* dan *supplier.* Mulai baris kedua, model matematis yang ditunjukan merupakan batasan dari permasalahan VMI yang dihadapi. Adapun batasan pada baris kedua dan ketiga merupakan pertidaksamaan yang sama dengan batasan pada pertidaksamaan 33 di model optimalisasi VMI secara umum. Kedua pertidaksamaan ini pun sudah menunjukan kesesuaian model dengan data yang diberikan.

Baris keempat sampai kelima belas menunjukan batasan yang sama dengan persamaan 34 pada model di subbab V.1. Kedua belas batasan ini telah menunjukan kesesuaian dengan data yang diberikan (frekuensi pengiriman ke *retailer 1* dan 2). Selanjutnya baris 16 hingga 27 menunjukan batasan yang sesuai dengan persamaan 35 pada model matematis yang telah dirancang.

Kedua belas batasan ini berjenis *if-then constraint* yang akan berpengaruh pada fungsi tujuan biaya pemesanan *supplier* ke pihak ketiga. Sama halnya dengan batasan-batasan sebelumnya, kedua belas batasan ini pun sudah menunjukan kesesuaian dengan input data yang diberikan. Batasan-batasan lainnya yang terdapat pada baris kedua puluh delapan hingga terakhir merupakan batasan tanda (*sign constraint*) yang juga telah sesuai dengan batasan tanda pada model di subbab V.1. Berdasarkan pengamatan yang telah dilakukan pada model matematis lengkap yang ditampilkan, dapat disimpulkan bahwa penerjemahan model dengan bahasa Lingo telah sesuai. Verifikasi program Lingo hanya dilakukan pada salah satu contoh penerapan kasus karena pada kasus-kasus lainnya kode pemrograman yang digunakan tetap sama.

V.5. Implementasi Model

Model yang telah diterjemahkan ke dalam bahasa matematis Lingo akan digunakan untuk menyelesaikan tiga kasus hipotetis VMI. Seperti telah dijelaskan pada bagian V.2, kasus-kasus ini terdiri dari 2, 3, dan 4 *retailer* dengan nilai-nilai parameter biaya serta jumlah permintaan yang berbeda-beda. Model yang telah dihasilkan di dalam subbab V.1 adalah model yang sulit untuk diselesaikan secara langsung. Terdapat 3 variabel keputusan dalam model ini, yaitu Q_{ij} , q_i , dan f_i. Status variabel f_i sebagai indeks membuat model ini tidak dapat diselesaikan dengan mudah. Dari ketiga variabel keputusan yang dilibatkan, perlu penetapan satu nilai variabel terlebih dahulu sehingga nilai-nilai variabel lainnya dapat dihitung. Variabel yang dimaksud adalah variabel f_i, yaitu frekuensi pengiriman barang dari pemasok ke *retailer* i per periodenya. Penetapan nilai variabel f_i dalam menyelesaikan permasalahan mengubah status variabel f_i yang semula sebagai variabel keputusan menjadi parameter model matematis. Perubahan nilai f_i pada model matematis suatu kasus tentunya akan menghasilkan nilai Q_{ij} dan q_i yang berbeda. Hal ini tentunya menjadi kesulitan tersendiri dalam mencari solusi optimal kasus-kasus yang dihadapi. Peneliti perlu melakukan iterasi kombinasi nilai f_i yang mungkin hingga akhirnya didapatkan solusi frekuensi pengiriman, jumlah pemesanan, dan jumlah pengiriman barang dengan total biaya paling kecil. Namun hal ini tidak mungkin dilakukan mengingat kombinasi nilai parameter f_i yang sangat banyak. Oleh sebab itu untuk mempersingkat pencarian solusi terbaik dirancanglah suatu algoritma pencarian solusi seperti berikut.

1. Hitung frekuensi pemesanan barang setiap *retailer* menggunakan model EOQ. Bulatkan ke atas nilai frekuensi pemesanan yang dilakukan. Nilai frekuensi pemesanan ini menjadi solusi awal frekuensi pengiriman barang oleh pemasok ke *retailer* i (f_i). Nilai f_i ini menjadi dasar iterasi nilai-nilai f_i lainnya.
2. Hitung total biaya yang dihasilkan dengan penerapan sistem pasokan barang secara tradisional. Biaya-biaya ini meliputi biaya simpan dan biaya pesan pemasok dan setiap *retailer* menggunakan metode EOQ dengan frekuensi pemesanan hasil pembulatan pada langkah 1. Pembulatan frekuensi pemesanan pun dilakukan di pihak pemasok.
3. Lakukan iterasi dengan mengubah frekuensi pengiriman ke *retailer* i (i = 1, 2, …., n) 1 satuan lebih kecil dan 1 satuan lebih besar sedangkan frekuensi pengiriman barang ke *retailer* lainnya dibiarkan tetap. Terdapat 2 kondisi yang perlu diperhatikan selama iterasi berlangsung, yaitu :
   a. Jika pengubahan frekuensi pengiriman ke *retailer* i menjadi 1 poin lebih besar atau lebih kecil menghasilkan total biaya yang lebih kecil dari total biaya sebelumnya, lanjutkan iterasi dengan semakin memperbesar atau memperkecil frekuensi pengiriman barang dari nilai sebelumnya namun dengan membiarkan nilai frekuensi pengiriman ke *retailer* lainnya tetap.
   b. Jika pengubahan frekuensi pengiriman ke *retailer* i (lebih besar atau lebih kecil 1 poin) menyebabkan total biaya lebih besar dari total biaya sebelumnya, maka hentikan iterasi yang dilakukan saat ini pada nilai parameter terakhir yang memberikan solusi lebih baik (total biaya yang lebih kecil). Ganti nilai parameter frekuensi pengiriman yang akan diubah selanjutnya dengan parameter frekuensi pengiriman *retailer* yang memiliki indeks terkecil dari *retailer-retailer* yang tersisa. Lakukan iterasi solusi menggunakan cara yang sama dengan sebelumnya. Kombinasikan frekuensi pengiriman terbaik ke *retailer* i yang didapatkan sebelum penggantian parameter dengan nilai-nilai parameter baru hasil pengubahan.

   Lakukan langkah ketiga hingga semua parameter frekuensi pengiriman ke *retailer* i (1 hingga n) digunakan dalam proses iterasi dan tidak ada lagi parameter frekuensi pengiriman yang dapat digunakan untuk menggantikan parameter frekuensi sebelumnya berkaitan dengan kondisi b pada langkah 3. Carilah kombinasi nilai parameter frekuensi pengiriman yang menghasilkan biaya paling kecil dari semua kemungkinan yang telah dicoba. Pertimbangkan juga kombinasi solusi frekuensi pengiriman yang didapatkan dengan menggunakan sistem tradisional. Penerapan frekuensi pengiriman ini pada model optimalisasi VMI akan menghasilkan besar biaya yang berbeda karena terdapat perbedaan biaya pesan *retailer* kepada pemasok pada sistem VMI ini.

Penggunaan algoritma yang telah dikembangkan untuk menyelesaikan kasus VMI akan dicontohkan dengan menggunakan kasus hipotetis 2 *retailer.* Besar biaya dan jumlah permintaan yang dilibatkan dalam perhitungan dapat dilihat pada Tabel 1. Berikut ini adalah contoh penerapan algoritma yang dimaksud.

1. Rumus umum perhitungan EOQ adalah

   EOQ = $\sqrt{2cD/h}$……………………………………………………………..…..(36)

   Dengan menggunakan persamaan 36, bisa didapatkan EOQ untuk *retailer* 1 dan 2 berturut-turut 866.02 dan 346.41. Berdasarkan EOQ ini bisa didapatkan frekuensi pemesanan setiap *retailer* dengan membagi total permintaan *retailer* i per periode (D_i) dengan nilai EOQ yang sesuai. Cara ini akan menghasilkan frekuensi pemesanan *retailer* 1 dan 2 berturut-turut 5.77 dan 8.66 yang setelah dibulatkan ke atas menjadi 6 dan 9 kali pengiriman. Nilai-nilai ini akan menjadi nilai awal solusi pengiriman barang dari pemasok ke masing-masing *retailer,* f1 = 6 dan f2 = 9.

2. Rumus umum perhitungan total biaya yang digunakan dalam langkah awal pencarian solusi adalah sebagai berikut.

   Total biaya pemasok/*retailer* = (frekuensi pemesanan pemasok/*retailer* x biaya pesan pemasok/*retailer*) + (rata-rata jumlah persediaan pemasok/*retailer* per periode x biaya persediaan pemasok/*retailer* per unit per periode)…...…...………….(37)

   Untuk menghitung total biaya pemasok, perlu diketahui frekuensi pemesanan pemasok yang didapatkan dengan cara yang sama seperti di langkah 1, dimana total permintaan barang bagi pemasok merupakan total jumlah permintaan barang kedua *retailer.* Perhitungan yang dilakukan menghasilkan pembulatan frekuensi pemesanan pemasok sebesar 7.

Berdasarkan hasil perhitungan EOQ untuk pemasok dan *retailer,* bisa didapatkan total biaya pemasok, *retailer,* serta keseluruhan sistem rantai pasok tradisional sebagai berikut.

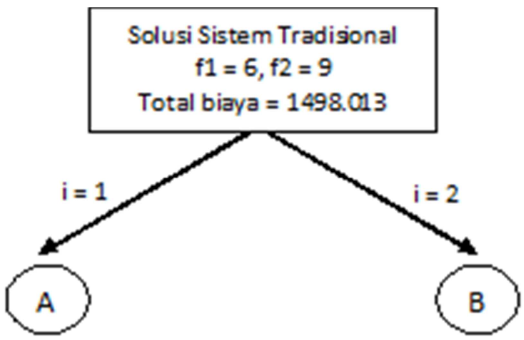
Total biaya pemasok = 7 x 45 + 1200 / 2 x 0.5 = 615

Total biaya *retailer* 1 = 6 x 45 + 866.02 / 2 x 0.6 = 529.807

Total biaya *retailer* 2 = 9 x 20 + 346.410 / 2 x 1 = 353.205

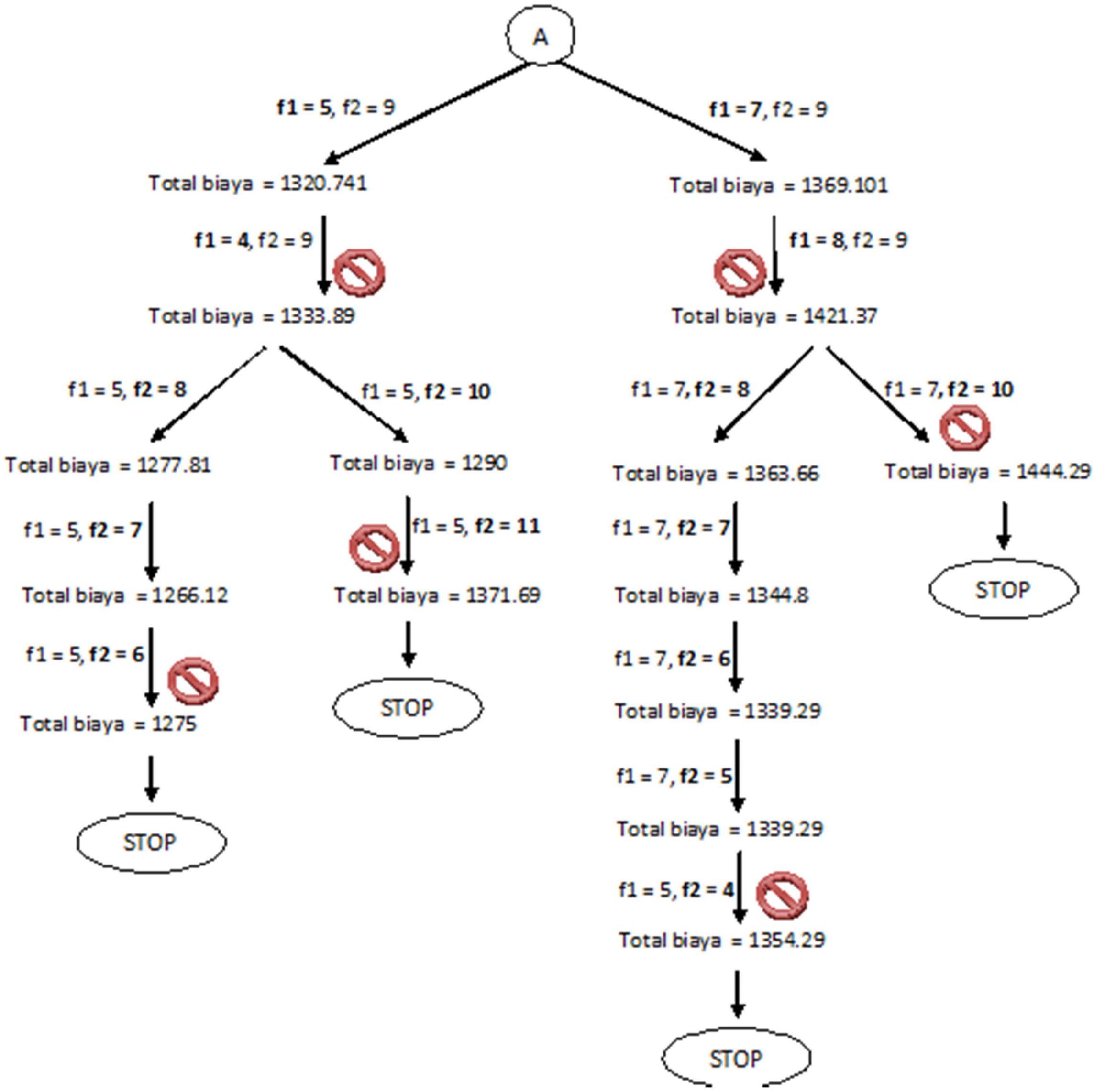
Total biaya keseluruhan = 615 + 529.807 + 353.205 = 1498.013.

3. Iterasi solusi permasalahan VMI 2 *retailer* akan dilakukan dengan dasar nilai frekuensi pengiriman dan total biaya sistem tradisional. Iterasi solusi dapat dijelaskan dengan menggunakan diagram pohon pada Gambar 2, 3, dan 4. Sesuai dengan algoritma yang dibangun untuk mencari solusi permasalahan yang dihadapi, nilai frekuensi pengiriman *retailer* i akan diubah menjadi 1 poin lebih kecil dan 1 poin lebih besar dari frekuensi yang didapatkan dengan sistem tradisional. Nilai fungsi tujuan yang menjadi dasar perbandingan awal adalah nilai fungsi tujuan (total biaya) yang telah dihitung di langkah 2.



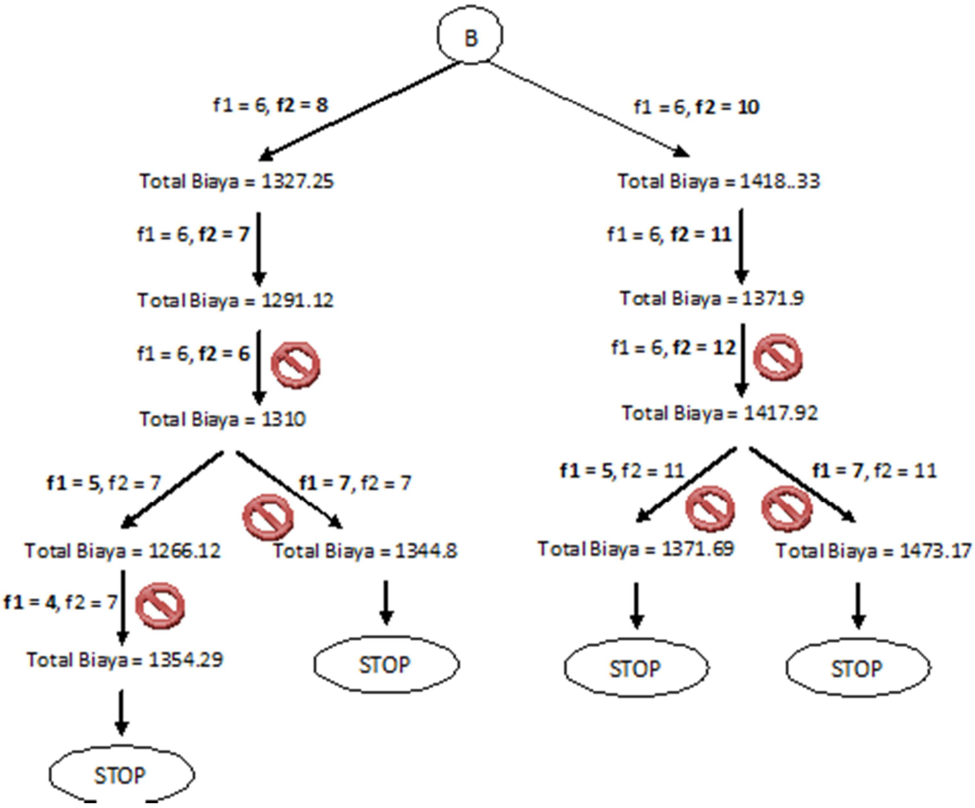
Gambar 2. Iterasi VMI 2 *Retailer*

Pada iterasi pertama, nilai f1 dan f2 akan diubah menjadi 5 dan 7 serta 8 dan 10. Masing-masing nilai f akan dikombinasikan dengan nilai f lainnya yang dianggap konstan. Dengan demikian, dihasilkan 4 kombinasi frekuensi pengiriman yang diuji cobakan pada program penyelesaian masalah VMI 2 *retailer.* Kombinasi-kombinasi tersebut adalah f1 = 5 & f2 = 9, f1 = 7 & f2 = 9, f1 = 6 & f2 = 8, serta f1 = 6 & f2 = 10. Keempat kombinasi ini menghasilkan nilai fungsi tujuan yang dapat dilihat pada Gambar 3 dan 4 di percabangan pertama diagram pohon. Seperti telah dijelaskan sebelumnya, terdapat 2 kondisi yang perlu diperhatikan ketika iterasi berlangsung. Perubahan nilai frekuensi pada i = 1 (node A) menjadi 5 dan 7 menghasilkan nilai fungsi tujuan yang lebih kecil dibandingkan nilai fungsi tujuan sebelumnya yang didapat dari sistem tradisional. Oleh sebab itu iterasi pada percabangan kedua akan berlanjut dengan semakin memperbesar dan memperkecil nilai f1 sedangkan nilai f2 dibiarkan sebesar 9.



Gambar 3. Iterasi VMI 2 *Retailer* i = 1

Penggantian parameter akibat nilai fungsi tujuan yang dihasilkan lebih buruk.



Gambar 4. Iterasi VMI 2 *Retailer* i = 2

 Penggantian parameter akibat nilai fungsi tujuan yang dihasilkan lebih buruk.

Kondisi lainnya yang mungkin dihadapi saat iterasi berlangsung adalah lebih besarnya nilai fungsi tujuan dibandingkan nilai fungsi tujuan sebelumnya. Hal ini salah satunya terjadi pada percabangan/iterasi tahap kedua pada i = 1, tepatnya saat pengubahan nilai f1 menjadi 4 dan 8. Konsekuensi dari kondisi ini adalah dihentikannya iterasi solusi yang dilakukan dengan mengubah nilai f1 dan digantikan dengan nilai f2. Nilai f1 yang dikombinasikan dengan nilai f2 adalah nilai f1 terakhir yang memberikan solusi total biaya lebih baik dari nilai total biaya sebelumnya. Nilai f2 yang dikombinasikan dengan f1 adalah nilai f2 yang akan melalui proses perubahan yang sama dengan proses pengubahan nilai f1, yaitu menjadi 1 poin lebih besar atau lebih kecil dari nilai sebelumnya. Contoh kasus ini dapat dilihat pada Gambar 3 di percabangan/iterasi kedua. Nilai f1 = 4 dan 8 akhirnya tidak digunakan untuk iterasi selanjutnya. Nilai f1 terakhir yang memberikan solusi lebih baik dari solusi sebelumnya adalah 5 dan 7. Nilai f1 sebesar 5 dan 7, masing-masing akan dikombinasikan dengan nilai f2 sebesar 8 dan 10 yang merupakan pengubahan nilai f2 sebelumnya sebesar 9. Dengan demikian dihasilkan 4 kombinasi nilai parameter f1 dan f2 yang baru, yaitu f1 = 5 & f2 = 8, f1 = 5 & f2 = 10, f1 = 7 & f2 = 8, serta f1 = 7 & f2 = 10. Total biaya untuk masing-masing kombinasi dapat dilihat pada Gambar 3.

Satu dari dua kondisi ini akan ditemui saat iterasi berlangsung. Tindakan yang diambil akan berbeda pada setiap kondisi. Tindakan-tindakan ini telah dijelaskan dalam algoritma yang dibangun dan dicontohkan pada iterasi solusi permasalahan VMI 2 *retailer.* Iterasi akan berhenti apabila tidak ada lagi parameter frekuensi pengiriman yang bisa menggantikan parameter sebelumnya ketika menghadapi kondisi no 3b dalam algoritma pencarian solusi. Dari setiap solusi yang didapatkan melalui proses iterasi, akan dipilih satu solusi terbaik. Namun, solusi terbaik ini belum tentu solusi optimal. Inilah yang masih menjadi kelemahan algoritma pencarian solusi yang dibangun. Kelemahan lain dari algoritma ini adalah kemungkinan didapatkannya alternatif solusi yang sama dari iterasi-iterasi yang dilakukan.

Algoritma pencarian solusi yang telah diterapkan untuk kasus 2 *retailer* diterapkan juga dalam penyelesaian masalah 3 dan 4 *retailer*. Solusi selengkapnya untuk penyelesaian masalah 2, 3, dan 4 *retailer* dapat dilihat pada Tabel 5 sampai 12. Iterasi nol pada tabel-tabel yang berisi alternatif solusi sebenarnya bukan iterasi sesungguhnya dengan menggunakan langkah 3 dalam algoritma pencarian solusi, melainkan hanya alternatif solusi yang berasal dari sistem tradisional yang turut dipertimbangkan.

Tabel 5. Solusi VMI 2 *Retailer*

Iterasi	**f1**	**f2**	**Total Biaya**
0	6	9	1292.78
1	6	8	1327.257
	6	10	1418.333
	5	9	1320.741
	7	9	1369.101

(lanjut)

Tabel 5. Solusi VMI 2 *Retailer* (lanjutan)

Iterasi	f1	f2	Total Biaya
2	4	9	1333.89
	8	9	1421.37
	6	11	1371.9
	6	7	1291.12
3	5	10	1290
	5	8	1277.81
	7	10	1444.29
	7	8	1363.66
	6	12	1471.92
	6	6	1310
4	5	11	1371.69
	5	7	***1266.12***
	7	7	1344.8
	7	11	1473.17
5	5	6	1275
	7	6	1339.29
	4	7	1354.29
6	7	5	1339.29
7	7	4	1354.29

Untuk iterasi solusi di kasus 3 dan 4 *retailer,* daftar alternatif solusi yang dihasilkan dari proses iterasi akan dibagi berdasarkan nilai i. Pembagian berdasar nilai i ini merupakan pembagian yang terjadi di percabangan awal pencarian solusi seperti pada Gambar 2.

Tabel 6. Solusi VMI 3 *Retailer* untuk i = 1

Iterasi	f1	f2	f3	Total Biaya
0	9	7	8	2404.218
1	10	7	8	2396.811
	8	7	8	2766.335
2	11	7	8	2390.268
	9	6	8	2382.5
	9	8	8	2397.917
3	12	7	8	2571.913
	9	5	8	2360.167
	9	9	8	2409.367

(lanjut)

Tabel 6. Solusi VMI 3 *Retailer* untuk i = 1 (lanjutan)

Iterasi	f1	f2	f3	Total Biaya
4	9	4	8	2272.33
	9	8	7	2453.373
	9	8	9	2366.142
	11	6	8	2413.833
	11	8	8	2456.625
5	11	7	7	2353.428
	11	7	9	2492.166
	9	8	10	2505.139
	9	3	8	2328.858
6	9	4	7	2273.08
	9	4	9	2315.864
	11	7	6	2401.335
7	9	4	6	***2237.037***
8	9	4	5	2321.667

Tabel 7. Solusi VMI 3 *Retailer* untuk i = 2

Iterasi	f1	f2	f3	Total Biaya
1	9	8	8	2397.917
	9	6	8	2382.5
2	9	5	8	2361.86
	10	7	8	2396.811
	8	7	8	2766.34
3	9	7	7	2413.7
	9	7	9	2750.94
	9	4	8	2272.33
4	9	3	8	2328.86
5	8	4	8	2953.75
	10	4	8	2401.25
6	9	4	7	2334.12
	9	4	9	2389.32

Tabel 8. Solusi VMI 3 *Retailer* untuk i = 3

Iterasi	f1	f2	f3	Total Biaya
1	9	7	7	2413.7
	9	7	9	2750.94
2	8	7	8	2766.34
	10	7	8	2396.81

(lanjut)

Tabel 8. Solusi VMI 3 *Retailer* untuk i = 3 (lanjutan)

Iterasi	f1	f2	f3	Z
3	9	6	8	2382.5
	9	8	8	2397.917
4	9	5	8	2361.86
5	9	4	8	2272.33
6	9	3	8	2328.86

Tabel 9. Solusi VMI 4 *Retailer* untuk i = 1

Iterasi	f1	f2	f3	f4	Z
1	5	7	5	7	2380.255
	7	7	5	7	2351.684
2	8	7	5	7	2269.63
	6	6	5	7	***2185.255***
	6	8	5	7	2355.88
3	9	7	5	7	2373.588
	6	5	5	7	2360.255
	6	9	5	7	2330.255
4	6	6	4	7	2315.255
	6	6	6	7	2375.918
	6	10	5	7	2372.755
	8	6	5	7	2357.755
	8	8	5	7	2365.88
5	8	7	4	7	2503.265
	8	7	6	7	2388.588
	6	6	5	6	2321.25
	6	6	5	8	2382.344
6	8	7	5	6	2283.75
	8	7	5	8	2359.688

Tabel 10. Solusi VMI 4 *Retailer* untuk i = 2

Iterasi	f1	f2	f3	f4	Z
1	6	6	5	7	2185.26
	6	8	5	7	2355.88
2	5	7	5	7	2380.26
	7	7	5	7	2351.68
	6	5	5	7	2360.26

(lanjut)

Tabel 10. Solusi VMI 4 *Retailer* untuk i = 2 (lanjutan)

Iterasi	f1	f2	f3	f4	Total Biaya
3	5	6	5	7	2367.76
	7	6	5	7	2359.18
	6	7	4	7	2317.755
	6	7	6	7	2370.26
4	6	7	5	6	2333.75
	6	7	5	8	2349.69
	6	6	4	7	2315.26
	6	6	6	7	2409.42
5	6	6	5	6	2321.25
	6	6	5	8	2382.34

Tabel 11. Solusi VMI 4 *Retailer* untuk i = 3

Iterasi	f1	f2	f3	f4	Total Biaya
1	6	7	4	7	2317.76
	6	7	6	7	2370.26
2	5	7	5	7	2380.26
	7	7	5	7	2351.68
3	6	6	5	7	2185.26
	6	8	5	7	2355.88
4	6	5	5	7	2360.26
	6	7	5	6	2333.75
	6	7	5	8	2349.69
5	6	6	5	6	2321.25
	6	6	5	8	2382.344

Tabel 12. Solusi VMI 4 *Retailer* untuk i = 4

Iterasi	f1	f2	f3	f4	Total Biaya
1	6	7	5	6	2333.75
	6	7	5	8	2349.69
2	5	7	5	7	2380.26
	7	7	5	7	2351.68
3	6	6	5	7	2185.26
	6	8	5	7	2355.88
4	6	5	5	7	2360.26
	6	7	4	7	2317.76
	6	7	6	7	2370.26
5	6	6	4	7	2315.26
	6	6	6	7	2409.42

Dari sekian banyak alternatif solusi hasil iterasi, diambil nilai/solusi terbaik untuk masing-masing permasalahan. Tabel 13 sampai 15 memberikan rangkuman alternatif solusi terbaik tiap kasus beserta nilai setiap variabel keputusan.

Tabel 13. Alternatif Solusi Terbaik Kasus 2 *Retailer*

Kasus 2 retailer			
f1	5	**Q21**	428.5714
f2	7	**Q22**	857.1428
q1	1000	**Q23**	0
q2	428.5714	**Q24**	857.1428
Q11	1000	**Q25**	0
Q12	1000	**Q26**	857.1428
Q13	1000	**Q27**	0
Q14	1000	**Total Biaya**	1266.122
Q15	1000		

Tabel 14. Alternatif Solusi Terbaik Kasus 3 *Retailer*

Kasus 3 retailer			
f1	9	**Q18**	1111.11
f2	4	**Q19**	0
f3	6	**Q21**	750
q1	555.55	**Q22**	750
q2	750	**Q23**	750
q3	833.33	**Q24**	750
Q11	1111.11	**Q31**	833.33
Q12	0	**Q32**	833.33
Q13	1111.11	**Q33**	833.33
Q14	0	**Q34**	1666.67
Q15	1666.66	**Q35**	0
Q16	0	**Q36**	8333.33
Q17	0	**Total Biaya**	2237.037

Tabel 15. Alternatif Solusi Terbaik Kasus 4 *Retailer*

Kasus 4 retailer									
f1	6	**Q12**	0	**Q25**	0	**Q43**	2142.857
f2	6	**Q13**	666.67	**Q26**	0	**Q44**	0
f3	5	**Q14**	0	**Q31**	1000	**Q45**	1071.429
f4	7	**Q15**	666.67	**Q32**	1000	**Q46**	2142.857
q1	333.33	**Q16**	0	**Q33**	1000	**Q47**	0
q2	250	**Q21**	750	**Q34**	1000	**Total Biaya**	2185.255
q3	1000	**Q22**	0	**Q35**	16000		
q4	1071.429	**Q23**	0	**Q41**	2142.857		
Q11	666.67	**Q24**	750	**Q42**	0		

Alternatif solusi terbaik yang didapatkan menjadi rekomendasi kebijakan distribusi dan pemesanan barang bagi pemasok dan *retailer* sesuai dengan kasus yang dihadapi. Nilai-nilai variabel keputusan dalam alternatif solusi terpilih digunakan juga untuk menguji kebenaran model yang telah dibuat. Dari seluruh nilai variabel keputusan yang ada, dapat disimpulkan bahwa nilai-nilai tersebut logis dan sesuai dengan model konseptual yang dirancang. Namun demikian, memang terdapat kejanggalan pada 1 variabel yaitu Q35 di kasus 4 *retailer,* dimana nilai variabel tersebut berjumlah 16000. Nilai variabel ini sangat besar disebabkan oleh titik waktu pemesanan yang dilakukan di akhir periode sehingga biaya simpan tidak diperhitungkan namun pemesanan tetap harus dilakukan untuk memenuhi kebutuhan pengiriman di periode tersebut.

Batasan model yang hanya mensyaratkan jumlah pemesanan sejangka waktu tertentu minimal sama dengan total pengiriman barang dalam jangka waktu tersebut merupakan hal kedua yang menyebabkan kejanggalan ini terjadi. Namun, hal ini bisa diatasi dengan mengurangkan total pengiriman barang dalam satu periode dengan total pemesanan barang yang telah dilakukan sebelum kali terakhir pemesanan barang dalam periode/*time horizon* penyelesaian masalah. Hasil pengurangan ini dapat dijadikan nilai jumlah pemesanan yang harus dilakukan pemasok di akhir periode untuk memenuhi kebutuhan pengiriman barang. Periode yang dimaksudkan di bagian ini maupun di bagian lainnya dapat dicontohkan dengan periode permintaan barang selama 1 tahun. Dalam 1 tahun pemasok dapat mengirimkan atau memesan barang ke *retailer* atau pihak ketiga selama beberapa kali berdasarkan frekuensi dan jumlah variabel pemesanan yang bernilai tidak nol. Berdasarkan argumen ini, maka model yang telah dibuat tetap dapat dikatakan sesuai untuk digunakan dalam penyelesaian masalah VMI 1 pemasok dengan banyak *retailer.*

BAB VI
KESIMPULAN DAN SARAN

Bagian ini akan berisi ringkasan hasil penelitian yang telah dilakukan beserta saran bagi pengembangan penelitian selanjutnya yang berasal dari kekurangan-kekurangan yang masih dirasakan dalam penelitian ini.

V.1. Kesimpulan

Berdasarkan proses penelitian yang telah dilakukan, didapatkan model optimalisasi VMI yang valid dan lebih efisien dibandingkan model sebelumnya. Hal ini terlihat dari nilai-nilai variabel keputusan yang dihasilkan untuk setiap kasus hipotetis yang diterapkan. Nilai-nilai ini logis dan sesuai dengan konsep awal model yang dibuat, selain itu keefisienan model yang baru ini terlihat dari berkurangnya variabel *dummy* yang digunakan dibandingkan model matematis sebelumnya. Berkurangnya variabel *dummy* membuat fungsi tujuan dan batasan-batasan permasalahan menjadi lebih sederhana.

Selama proses penelitan, terdapat beberapa kendala sumber daya yang dihadapi. Hal ini membuat beberapa tujuan khusus penelitian tidak tercapai. Verifikasi program AMPL yang telah dibuat pada penelitian sebelumnya tidak dapat dilakukan, namun sebagai penggantinya model optimalisasi diterjemahkan ke dalam perangkat lunak yang berfungsi sama dengan AMPL sehingga model tetap dapat diverifikasi dan divalidasi berdasarkan hasil penerjemahan model ke dalam program dan solusi yang dihasilkan dengan bantuan program tersebut. Tujuan khusus lain yang tidak tercapai adalah validasi model dengan menerapkan kasus nyata sistem VMI. Sebagai gantinya, dibuatlah kasus hipotetis untuk proses validasi model ini. Berdasarkan hasil penerapan kasus hipotetis, dapat disimpulkan bahwa model layak digunakan dalam menyelesaikan permasalahan VMI 1 pemasok dengan banyak *retailer.*

V.2. Saran

Beberapa saran yang dapat diberikan berkaitan dengan proses dan hasil penelitian antara lain :

1. Pada penelitian selanjutnya dapat dilakukan pada pengembangan algoritma atau metode pencarian solusi optimal dari model VMI yang telah dibangun. Metode pencarian solusi yang telah dibuat saat ini hanya mampu menghasilkan solusi yang dapat dikatakan baik namun belum tentu optimal.
2. Perlu dibuat sebuah perangkat lunak tambahan untuk proses pencarian solusi VMI apabila algoritma pencarian solusi pada peneltian ini hendak digunakan. Perangkat lunak ini harus dapat diintegrasikan dengan program optimalisasi Lingo karena program Lingo inilah yang akan mencari solusi terbaik untuk setiap kemungkinan kombinasi nilai parameter frekuensi pengiriman barang. Pengembangan perangkat lunak ini sangat diperlukan karena tingkat kesulitan dan lama pencarian solusi akan semakin tinggi seiring pertambahan jumlah *retailer* dalam model.
3. Proses validasi model perlu diperkuat dengan melibatkan kasus nyata pada area rantai pasok.

DAFTAR PUSTAKA

Daellenbach, H.G. dan McNickle, D.C. (2005) *Management science: Decision makingthrough systems thinking,* Palgrave McMillan*,* New York*.*

Law, A.M. dan Kelton W.D. (2000) *Simulation Modeling & Analysis 3^{rd} Edition*, McGraw-Hill, Singapore.

Sitompul, C. dan Alfian (2012) *Pengembangan Model Persediaan yang Dikelola Pemasok (Vendors Managed Inventory),* Lembaga Penelitian dan Pengabdian kepada Masyarakat, Universitas Katolik Parahyangan, Bandung.