

## Rancangan Sistem Pendukung Keputusan Distribusi Produk pada Perusahaan Skala Kecil dan Menengah

Lisnawanty<sup>[1]</sup>, Norma Yunita<sup>[2]</sup>

AMIK BSI Pontianak<sup>[1]</sup>

lisnawanty.lsy@bsi.ac.id<sup>[1]</sup> \_ norma.nyt@bsi.ac.id<sup>[2]</sup>

**ABSTRACT** — *Some small and medium scale companies engaged in the distribution of products still have a simple product distribution system. Some of these distribution companies tend to use push distribution systems and simple distribution scheduling approaches. In a push distribution system, the decision to resupply is in the hands of retailers so that distribution companies must be really proactive in their product offerings to retailers. In a simple distribution scheduling approach, distribution firms tend to group retailers into clusters based on geographic proximity with the assumption that every retailer in one cluster has the same resupply cycle time. As a result, distribution companies tend to be unable to keep service levels high enough to serve a number of retailers who are estimated to have stocked out. The problem is better known as Inventory Routing Problem (IRP). This research develops a simulator-based Decision Support System (DSS) that is needed to support the solution of IRP problems that will provide suboptimal feasible solutions that can be used as the basis for decision-making on when distribution is done? how much should be distributed? and which route? The purpose of such decision-making is to minimize average distribution costs over the planning time horizon without causing inventory to stock out retailers or keeping service levels high.*

**Keywords** : Decision Support System, Inventory Routing Problem (IRP), resupply, cluster, demand, suboptimal feasible solutions

**ABSTRAK** — Beberapa perusahaan skala kecil dan menengah yang bergerak dibidang pendistribusian produk masih memiliki sistem pendistribusian produk yang sederhana. Beberapa perusahaan distribusi tersebut cenderung menggunakan *push distribution system* dan pendekatan penjadwalan distribusi yang sederhana. Dalam *push distribution system*, keputusan untuk melakukan *resupply* (pemesanan ulang) berada di tangan retailer sehingga perusahaan distribusi harus benar-benar proaktif dalam melakukan penawaran produk mereka kepada retailer. Sedangkan dalam pendekatan penjadwalan distribusi yang sederhana, perusahaan distribusi cenderung mengelompokkan retailer ke dalam sejumlah *cluster* (kelompok) berdasarkan kedekatan geografis dengan asumsi bahwa setiap retailer dalam satu *cluster* memiliki waktu siklus *resupply* yang sama. Akibatnya, perusahaan distribusi cenderung tidak dapat menjaga *service level* tetap tinggi untuk dapat melayani sejumlah retailer yang diperkirakan telah *stock out*. Permasalahan tersebut lebih dikenal dengan istilah *Inventory Routing Problem* (IRP). Penelitian ini mengembangkan suatu Sistem Pendukung Keputusan (*Decision Support Systems/DSS*) berbasis simulasi yang dibutuhkan untuk mendukung pemecahan permasalahan IRP yang akan memberikan solusi *feasible* suboptimal yang dapat dijadikan dasar pembuatan keputusan mengenai kapan distribusi dilakukan? berapa banyak harus didistribusikan? dan melalui rute yang mana? Tujuan dari pengambilan keputusan tersebut adalah untuk meminimasi biaya distribusi rata-rata selama horizon waktu perencanaan tanpa menyebabkan persediaan pada retailer *stock out* atau menjaga *service level* tetap tinggi.

**Kata Kunci** : Sistem Pendukung Keputusan, *Inventory Routing Problem* (IRP), *resupply*, *cluster*, *demand*, solusi *feasible* suboptimal

### 1 PENDAHULUAN

Teknologi informasi memiliki peranan penting dalam aktivitas manusia, terutama sebagai fasilitator utama bagi kegiatan-kegiatan bisnis. Alasan yang paling umum terhadap penggunaan teknologi informasi adalah adanya kebutuhan untuk mempertahankan dan meningkatkan posisi kompetitif, mengurangi biaya, meningkatkan fleksibilitas, dan tanggapan. Teknologi informasi juga dapat mempengaruhi proses pengembangan strategi pemasaran karena teknologi informasi memberikan lebih banyak informasi kepada manajer melalui pemakaian sistem pengambilan keputusan

(*Decision Support Systems/DSS*) dalam suatu perusahaan.

Suatu perusahaan merupakan suatu organisasi yang memiliki kompleksitas permasalahan dalam manajemen hubungan antara distributor dengan retailer serta menjaga pendistribusian produk berjalan lancar. Perubahan struktur pasar, produk, dan teknologi produksi terus terjadi sehingga berpengaruh pada kebijaksanaan manajemen yang dijalankan. Dalam menjalankan aktivitasnya, sekalipun didukung oleh potensi ekosistem dan aksesibilitas yang serba prospektif, namun di sisi lain perusahaan kerap dihadapkan pada masalah utama dalam penentuan keputusan strategis

yang sulit direalisasikan akibat persepsi yang heterogen. Beberapa perusahaan skala kecil dan menengah yang bergerak dibidang pendistribusian produk yang masih belum memiliki armada transportasi yang memadai. Sistem pendistribusian produk yang umumnya digunakan oleh beberapa perusahaan distribusi tersebut adalah *push distribution system* dan pendekatan penjadwalan distribusi yang sederhana. Dalam *push distribution system*, keputusan untuk melakukan *resupply* (pemesanan ulang) berada di tangan retailer. Dalam hal ini, perusahaan distributor harus benar-benar proaktif dalam melakukan penawaran produk mereka kepada retailer. Sedangkan dalam pendekatan penjadwalan distribusi yang sederhana, perusahaan distribusi cenderung mengelompokkan retailer ke dalam sejumlah *cluster* (kelompok) berdasarkan kedekatan geografis dengan asumsi bahwa setiap retailer dalam satu *cluster* memiliki waktu siklus *resupply* yang sama. Pada kenyataannya, setiap retailer dalam satu *cluster* belum tentu memiliki waktu siklus *resupply* yang sama karena setiap retailer memiliki karakteristik *demand* (permintaan) yang berbeda-beda.

Sistem pendistribusian dengan asumsi yang sederhana tersebut belum cukup dijadikan sebagai dasar dalam menganalisa karakteristik *demand* pada setiap retailer. Akibatnya, perusahaan distribusi cenderung tidak dapat menjaga *service level* tetap tinggi untuk dapat melayani sejumlah retailer yang diperkirakan telah *stock out*. Dalam satu kali siklus distribusi pada suatu *cluster*, perusahaan distribusi seringkali tidak mampu melayani semua retailer yang ada pada *cluster* tersebut karena kendala waktu dan kapasitas angkut. Sehingga retailer yang belum dilayani akan dilayani pada waktu siklus berikutnya, yaitu  $\pm 2$  minggu berikutnya. Dengan penjadwalan yang sederhana tersebut, kemungkinan yang akan sering terjadi adalah retailer yang belum terlayani adalah retailer yang telah mengalami *stock out*, dan retailer yang didatangi justru belum memerlukan penambahan persediaan produk mereka.

Permasalahan pendistribusian produk di atas dikenal dengan istilah *Inventory Routing Problem* (IRP). Dalam sistem IRP, *resupply* akan dijadwalkan berdasarkan *continuous review* dengan menganalisis *reorder point* yang dialami retailer pada setiap *cluster* kemudian merencanakan penjadwalan distribusi dan menentukan rute terbaik untuk dapat ditempuh dengan waktu siklus yang lebih singkat. Maka dari itu, keputusan yang dibuat dari IRP yaitu kapan distribusi dilakukan? Berapa banyak produk yang harus didistribusikan kepada retailer? Dan rute mana yang akan dilalui?

Tujuannya adalah untuk meminimasi biaya distribusi rata-rata selama horizon waktu perencanaan tanpa menyebabkan persediaan pada retailer menjadi *stock out*.

Berdasarkan keadaan tersebut, maka perlu dikembangkannya suatu prototipe Sistem Pendukung Keputusan berbasis simulasi guna mendukung dalam perolehan pendekatan solusi berupa perencanaan dalam permasalahan IRP. Prototipe Sistem Pendukung Keputusan ini akan membantu pihak distributor dalam menentukan rute terbaik (*best route*) yang akan dilalui sesuai dengan selama horizon waktu perencanaan sesuai dengan jadwal yang telah ditetapkan. Dengan adanya prototipe Sistem Pendukung Keputusan ini diharapkan dapat memberikan keputusan *feasible* (nyata) suboptimal bagi pihak distributor dan dapat dijadikan dasar dalam pembuatan keputusan dalam menentukan kapan distribusi dilakukan? Berapa banyak produk yang harus didistribusikan kepada retailer? Dan rute mana yang akan dilalui? Dengan kata lain, *output* (keluaran) dari Sistem Pendukung Keputusan ini adalah perencanaan penjadwalan distribusi selama horizon waktu perencanaan sehingga waktu dan sumber daya yang ada dapat dimanfaatkan dengan sebaik-baiknya agar distributor dapat menjaga *service level* tetap tinggi.

## 2 LANDASAN TEORI

### 2.1 Sistem Pendukung Keputusan (*Decision Support System*)

Little (1970) mendefinisikan Sistem Pendukung Keputusan sebagai sekumpulan prosedur berbasis model untuk data pemrosesan dan penilaian guna membantu para manajer mengambil keputusan. Moore dan Chang (1980) mendefinisikan Sistem Pendukung Keputusan sebagai sistem yang dapat diperluas untuk mampu mendukung analisis data *ad hoc* dan pemodelan keputusan, berorientasi terhadap perencanaan masa depan, dan digunakan pada interval yang tidak regular dan tidak terencana. Bonczek, dkk (1980) mendefinisikan Sistem Pendukung Keputusan sebagai sistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi: sistem bahasa (mekanisme untuk memberikan komunikasi antara pengguna dan komponen Sistem Pendukung Keputusan lain), sistem pengetahuan (repositori pengetahuan domain masalah yang ada pada Sistem Pendukung Keputusan entah sebagai data atau sebagai prosedur), dan sistem pemrosesan masalah (hubungan antara dua komponen lainnya, terdiri dari satu sistem atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan) [1].

## 2.2 Pembangkitan *Random Number Generator* (RNG) Dengan Algoritma *Prime Modulus Linear Congruential Generators* (PMMLCGs)

Simulasi dari suatu sistem atau proses memerlukan suatu metode pembangkitan atau angka-angka yang dihasilkan secara random dalam horizon waktu yang ditentukan mengandung unsur deterministik. Pemunculan angka-angka secara random (*random number*) diperoleh dari algoritma yang menghasilkan suatu angka sebagai penentu bagi *random number* berikutnya, dan demikian seterusnya. Metode pembangkitan yang digunakan ini dikenal dengan istilah *Prime Modulus Multiplicative Linear Congruential Generators* (PMMLCGs). PMMLCGs merupakan suatu metode yang digunakan untuk menghasilkan urutan-urutan atau *sequence* dari angka-angka sebagai hasil dari perhitungan dengan komputer yang diketahui distribusinya sehingga angka-angka tersebut muncul secara random dan digunakan terus-menerus.

Rumusan matematis yang digunakan dalam urutan (*sequence*), PMMLCGs didefinisikan sebagai berikut.

$$Z_i = (a \cdot Z_{i-1} + c) \pmod{m} \quad (1)$$

Keterangan:

$Z_i$  = angka random number yang baru

$Z_{i-1}$  = angka random number yang lama

$m$  = angka modulus

$a$  = angka konstan yang bebas (*multiplier*)

$c$  = angka konstan yang bersyarat (*increment*)

## 2.3 Pembangkitan *Random Variates*

### 1. Pendekatan Umum Pembangkitan *Random Variates* Dengan *Invers-Transform*

Metode *inverse-transform* dapat digunakan untuk distribusi diskrit, dimana pembangkitan beberapa *random variate* diskrit ini memiliki *range* nilai yang terbatas. Metode *invers-transform* lebih sesuai digunakan pada kasus secara kontinu. Dalam probabilitas distribusi ini terdapat *density function*  $f(x)$  yang dikenal dengan *probability density function* (pdf). *Density function*  $f(x)$  dari *random variable* merupakan peluang relatif yang dapat digunakan dalam pengamatan untuk *range* yg berbeda [2].

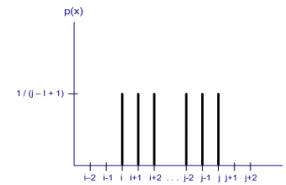
### 2. Jenis Pembangkitan *Random Variates* Kontinu

#### *Discrete Uniform*

*Discrete uniform* digunakan sebagai sebuah model pertama untuk menyatakan suatu kuantitas dalam probabilitas distribusi  $[0,1]$  yang berubah di antara bilangan *integer*  $i$  dan  $j$ . Dalam

pembangkitan *random variate* terdapat peluang untuk muncul angka yang sama (*uniform*) [2].

Gambar 1 berikut ini merupakan gambaran dari pembangkitan *random variate* pada model *discrete uniform*.



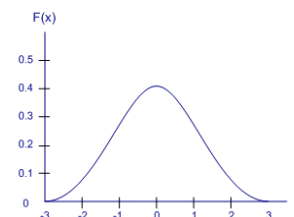
Sumber : Law, 1991

**Gambar 1 Discrete Uniform**

#### *Normal*

Distribusi  $N(0, 1)$  sering disebut sebagai *standard* atau *normal distribution*. Dalam memperkirakan *normal*, semakin kecil interval maka akan semakin baik perkiraannya. *Normal* dapat ditransformasikan menjadi distribusi *random variate* yang lain secara langsung [2].

Gambar 2 berikut ini merupakan gambaran dari pembangkitan *random variate* pada model *normal*.



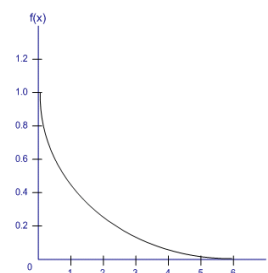
Sumber : Law, 1991

**Gambar 2 Normal**

#### *Exponential*

*Exponential distribution* merupakan model yang berada antara *Gamma* dan *Weibull* untuk ukuran parameter  $\alpha = 1$  dan skala parameter  $\beta$ . Perolehan nilai pada suatu sistem dengan model *exponential* adalah konstan [2].

Gambar 3 berikut ini merupakan gambaran dari pembangkitan *random variate* pada model *exponential*.



Sumber : Law, 1991

**Gambar 3 Exponential**

Untuk memperoleh nilai U (nilai *uniform* yang baru) ditentukan dengan rumusan matematis sebagai berikut:

$$U_i = (Z_i) / (\text{mod } m) \quad (2)$$

Keterangan :

$U_i$  = *random number uniform* yang baru

$Z_i$  = hasil *random number* PMMLCGs

$m$  = angka modulus

## 2.4 Optimasi Rute Tour Kendaraan Dengan Algoritma *Farthest Insertion*

Salah satu algoritma dari pendekatan heuristik (*insertion heuristic*) yang dapat diaplikasikan untuk pencarian rute tour kendaraan yaitu algoritma *farthest insertion*. Algoritma *farthest insertion* dapat menentukan jarak minimal suatu lokasi tour maksimal dari titik-titik (*node*) yang di *input* ke dalam peta secara geografis. Dibandingkan dengan algoritma *insertion heuristic* yang lain, algoritma *farthest insertion* sering digunakan dalam memecahkan permasalahan kombinatorial yang memerlukan waktu komputasi panjang dengan kompleksitas *procedure* pemrograman yang dapat menghasilkan peformansi yang kompetitif dan memberikan solusi dalam memilih rute tour lebih ringkas.

## 3 METODOLOGI PENELITIAN

### 3.1 Metode Pengembangan Sistem

Metode yang digunakan dalam penelitian ini adalah metode heuristik. Algoritma yang digunakan dengan pendekatan heuristik yang dapat diaplikasikan untuk pencarian rute tour kendaraan adalah algoritma *farthest insertion* dengan menentukan jarak minimal suatu lokasi tour maksimal dari titik-titik (*node*) yang diinput ke dalam peta secara geografis.

### 3.2 Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan dalam penelitian ini, antara lain:

#### A. Observasi

Pengamatan dilakukan pada Perusahaan Yakult di Kota Pontianak sebagai objek studi kasus untuk mengetahui prosedur distribusi produk yang berjalan.

#### B. Wawancara

Wawancara dilakukan dengan mengajukan pertanyaan kepada karyawan dari Perusahaan Yakult sehubungan dengan pendataan permintaan produk oleh retailer, serta proses distribusi produk dari Perusahaan Yakult ke masing-masing retailer.

#### C. Studi literatur

Studi literatur digunakan dengan mengumpulkan beberapa referensi yang bersumber dari buku dan jurnal.

## 4 HASIL DAN PEMBAHASAN

### 4.1 Perancangan Pembangkitan *Random Number* Untuk *Demand*, Transportasi dan Pelayanan

Perancangan Sistem Pendukung Keputusan ini menggunakan tiga konsep model *random variate*, yaitu *uniform*, *exponential*, dan *normal* sebagaimana yang diuraikan di bawah ini.

#### a) *Uniform*

Model *uniform* digunakan sebagai model awal dengan probabilitas distribusi [0,1] dalam proses pembangkitan *random variates*.

#### b) *Exponential*

Model *exponential* digunakan untuk memperkirakan *demand* yang terjadi pada setiap retailer dengan pembangkitan *random number* sesuai dengan parameter yang telah ditetapkan. Asumsi ini diambil mengikuti pola permintaan konsumen yang semakin berkurang dengan semakin berkurangnya produk pada rak *display* retailer.

Berikut ini merupakan nilai parameter yang ditentukan untuk pembangkitan *demand* tiap retailer.

$$Z_i = (a \cdot Z_{i-1} + c) (\text{mod } m) \quad (3.1)$$

$$U_i = Z_i / m \quad (4)$$

$$E_i = b * \ln(U_i) \quad (5)$$

Rumus PMMLCGs di atas ditentukan nilai parameter  $a = 630360016$ ,  $Z_1 = 132645$ ,  $c = 0$  dan  $m = 2^{31}-1$ . Panjang dari  $2^{31}-1$  menghasilkan 2.147.483.647 pembangkitan *random number*. Penentuan nilai tersebut mengacu pada literatur Law and Kelton, 1991.

#### c) *Normal*

Model *normal* digunakan untuk mengasumsikan kecepatan transportasi dan waktu pelayanan dari sejumlah retailer. Asumsi ini diambil dengan acuan bahwa rata-rata kecepatan transportasi atau waktu pelayanan dan masing-masing nilai simpangan tersebut menjadi parameter penentu dalam menentukan karakteristik kecepatan transportasi atau waktu pelayanan tiap retailer. Pembangkitan *random variate* untuk kecepatan transportasi dan waktu pelayanan dapat ditentukan sebagai berikut.

- Kecepatan transportasi

Kecepatan transportasi dibangkitkan dengan rumusan sebagai berikut.

$$X' = \mu_T + \sigma_T X \quad (6)$$

Keterangan:

$\mu_T$  = nilai rata-rata kecepatan transportasi yang diperoleh dari sejumlah retailer

$\sigma_T$  = simpangan rata-rata dengan nilai konstanta positif yang bebas ditentukan sendiri

- Waktu pelayanan  
Waktu pelayanan dibangkitkan dengan rumusan sebagai berikut.  
 $X' = \mu_S + \sigma_S X$  (7)  
Keterangan:  
 $\mu_S$  = nilai rata-rata waktu pelayanan yang diperoleh dari sejumlah retailer  
 $\sigma_S$  = simpangan rata-rata dengan nilai konstanta positif yang bebas ditentukan sendiri

Nilai parameter rumusan pembangkitan kecepatan transportasi dan waktu pelayanan ditetapkan berdasarkan perkiraan pengguna.

#### 4.2 Perancangan Konseptual Sistem Pendukung Keputusan

Adapun alur proses diagram alir Sistem Pendukung Keputusan diuraikan sebagai berikut.

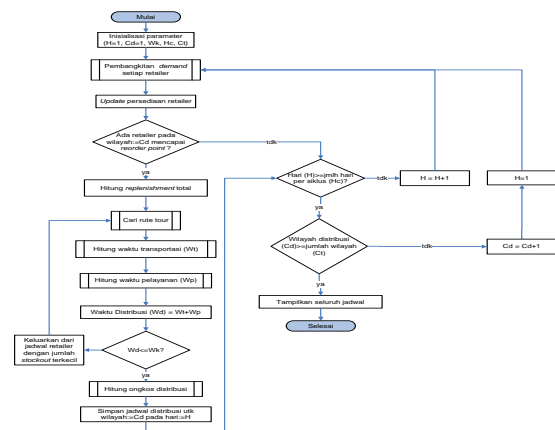
Program dimulai dengan inialisasi parameter, yaitu hari (H) pertama, wilayah distribusi (Cd) pertama, serta waktu kerja (Wk), jumlah hari per siklus (Hc), dan jumlah wilayah (Ct) yang telah ditetapkan. sesuai dengan kebutuhan perusahaan distribusi yang bersangkutan. membangkitkan *demand* untuk setiap retailer pada hari (H) pertama, wilayah distribusi (Cd) pertama, serta waktu kerja (Wk), jumlah hari per siklus (Hc), dan jumlah wilayah (Ct) yang telah ditetapkan sesuai dengan kebutuhan perencanaan depot.

Dengan pembangkitan *demand* yang menggunakan model *exponential* ini, depot memperkirakan *resupply* yang akan dilakukan salah satu atau beberapa retailer yang mengalami *stock out*. Hasil dari pembangkitan *demand* tersebut akan digunakan untuk *update* jumlah persediaan retailer pada hari itu dan dilakukan seterusnya hingga diketahui ada retailer yang mencapai *reorder point*.

Jika diketahui terdapat satu atau lebih retailer yang telah mencapai *reorder point*, maka selanjutnya akan dilakukan perhitungan *replenishment* total dari keseluruhan retailer yang mencapai *reorder point* tersebut sehingga depot dapat menentukan rute tour optimal yang akan dilalui depot. Waktu transportasi (Wt) dan waktu pelayanan (Wp) akan dibangkitkan dengan model normal. Masing-masing dari hasil pembangkitan tersebut digunakan untuk menghitung total waktu transportasi dan total waktu pelayanan sehingga diketahui jumlah waktu distribusi ( $Wd = Wt + Wp$ ) yang diperlukan. Hasil perhitungan tersebut disesuaikan dengan waktu kerja. Jika waktu distribusi (Wd) melebihi

waktu kerja (Wk), maka retailer dengan jumlah *stockout* terkecil akan dikeluarkan dari penjadwalan dan program akan kembali mencari rute tour. Namun jika waktu distribusi (Wd) kurang dari atau sama dengan waktu kerja (Wk), maka ongkos distribusi diperhitungkan sesuai dengan biaya per ruas yang telah ditetapkan depot.

Jadwal distribusi untuk wilayah Cd dan hari H disimpan, kemudian program akan mengecek jumlah hari (H) yang telah berdistribusi terhadap jumlah hari per siklus (Hc). Namun jika tidak terdapat retailer yang mencapai *reorder point*, maka program akan mengecek jumlah hari (H) yang telah berdistribusi terhadap jumlah hari per siklus (Hc). Jika hari (H) tidak melebihi atau sama dengan jumlah hari per siklus (Hc), maka distribusi dilanjutkan pada hari berikutnya ( $H = H + 1$ ) dan program akan membangkitkan *demand* kembali. Jika hari (H) telah lebih dari atau sama dengan jumlah hari per siklus (Hc) dan diketahui wilayah distribusi (Cd) tidak melebihi atau sama dengan jumlah wilayah (Ct) yang telah ditetapkan, maka penjadwalan distribusi akan dilanjutkan untuk wilayah distribusi (Cd) berikutnya ( $Cd = Cd + 1$ ), waktu distribusi dimulai hari pertama kembali ( $H = 1$ ), dan *demand* dibangkitkan kembali untuk mencari retailer yang diperkirakan mencapai *reorder point* selanjutnya. Demikian seterusnya hingga semua wilayah distribusi dijadwalkan untuk pelaksanaan distribusi. Jika hari (H) telah lebih dari atau sama dengan jumlah hari per siklus (Hc) dan diketahui wilayah distribusi (Cd) telah melebihi atau sama dengan jumlah wilayah (Ct) yang telah ditetapkan, seluruh hasil penjadwalan akan ditampilkan. Jika semua proses telah selesai dijalankan, data akhir akan disimpan. Program simulasi akan berakhir jika jumlah hari simulasi telah mencapai horizon waktu perencanaan yang telah ditetapkan.



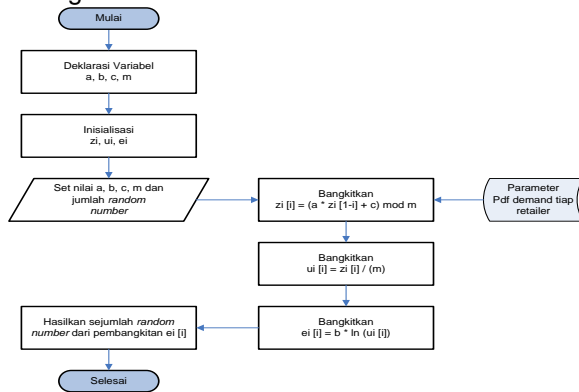
Gambar 4 Diagram Alir Sistem Pendukung Keputusan

Berdasarkan *flowchart* dan uraian dari diagram alir Sistem Pendukung Keputusan di atas, maka berikut ini diuraikan pula *subflowchart* sistem yang menggambarkan proses yang lebih spesifik dari setiap modul di atas.

**a) Subflowchart dari modul pembangkitan demand setiap retailer**

Gambar 5 berikut ini merupakan *subflowchart* dari alur proses pembangkitan demand dan dapat diuraikan sebagai berikut.

Pembangkitan dimulai dengan mendeklarasikan variabel a, b, c, dan m serta menginisialisasi variabel zi, ui, ei. Nilai dari setiap variabel yang telah dideklarasikan (a, b, c, m dan jumlah *random number*) dimasukkan sesuai dengan nilai yang telah ditetapkan. Dengan nilai tersebut, zi[i] dibangkitkan sehingga menghasilkan sejumlah *random number* sesuai dengan banyaknya *random number* yang akan dibangkitkan.



**Gambar 5 Subflowchart Modul Pembangkitan Demand Setiap Retailer**

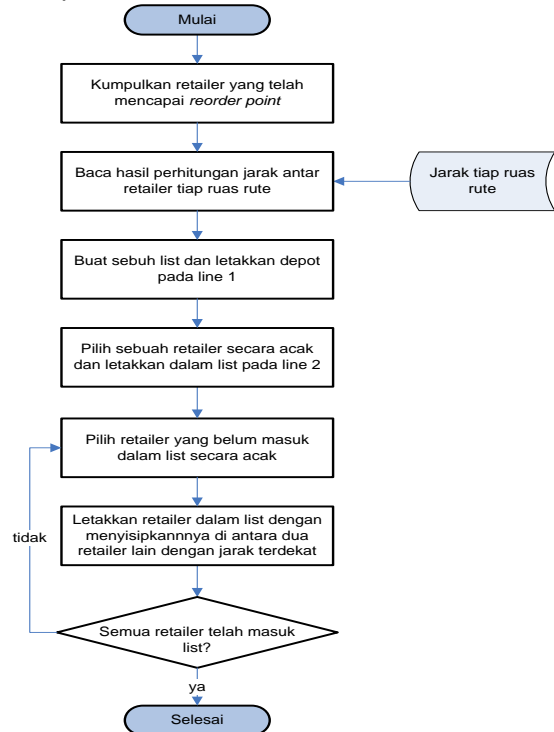
Hasil dari zi[i] akan diteruskan untuk dilakukannya pembangkitan terhadap ui[i]. Demikian seterusnya sehingga ui[i] juga digunakan untuk pembangkitan *random number* ei[i]. *Random number* yang diperoleh dari ei[i] ini digunakan sebagai pengurang terhadap stok awal retailer i. Proses pembangkitan selesai saat sejumlah *random number* ei[i] diperoleh untuk didistribusikan pada setiap retailer i dan *update* persediaan retailer i selama horizon waktu yang telah ditetapkan.

**b) Subflowchart dari modul cari rute tour**

Gambar 6 berikut ini merupakan *subflowchart* dari alur proses mencari rute tour minimal untuk tiap *cluster*.

Proses dimulai dengan mengumpulkan sejumlah retailer yang mencapai *reorder point*. Hasil perhitungan jarak antar retailer tiap ruas rute dibaca untuk digunakan dalam penentuan retailer dengan jarak terdekat. Selanjutnya, sebuah list dibuat dan meletakkan depot pada baris pertama (line 1). Sebuah retailer dipilih

secara acak dan diletakkan pada baris kedua (line 2) sebagaimana prosedur iterasi 1 dari sebuah prosedur penentuan rute dijalankan. Demikian seterusnya retailer yang belum masuk akan dipilih secara acak.



**Gambar 6 Subflowchart Modul Cari Rute Tour**

Dengan membaca hasil perhitungan jarak antar retailer tiap ruas, retailer diletakkan dalam list dengan menyisipkan retailer tersebut di antara dua retailer lain dengan jarak terdekat sehingga menghasilkan urutan rute tour. Proses akan terus berlangsung dengan memilih retailer yang belum masuk dalam *list* secara acak hingga semua retailer yang telah mencapai *reorder point* tersebut telah masuk dalam list. Proses mencari rute tour selesai jika semua retailer telah masuk dalam list.

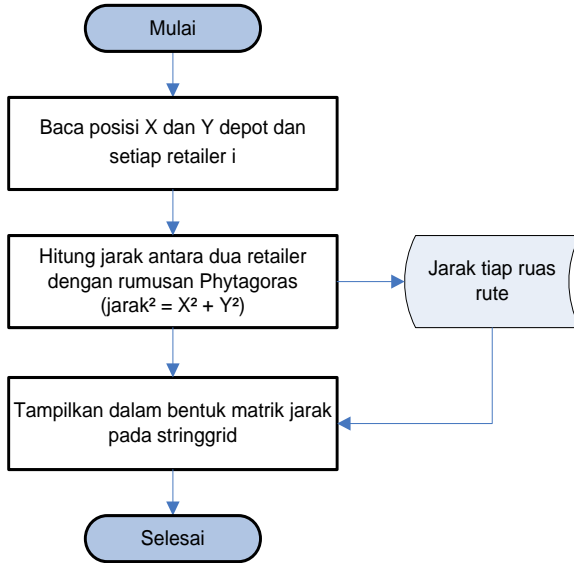
**c) Subflowchart dari modul perhitungan waktu transportasi (Wt)**

Waktu transportasi yang dibutuhkan antar retailer yang satu ke retailer berikutnya diperoleh dengan rumusan perhitungan jarak antar retailer dibagi kecepatan transportasi kendaraan. Oleh karena itu, waktu transportasi dihasilkan dari perhitungan tiga modul, yaitu modul perhitungan jarak tiap ruas rute, modul pembangkitan kecepatan transportasi, dan modul perhitungan waktu transportasi. Berikut ini merupakan *flowchart* dan uraian mengenai alur proses dari masing-masing modul tersebut.

**Modul perhitungan jarak tiap ruas rute**

Berikut ini merupakan alur proses menghitung jarak tiap ruas rute.

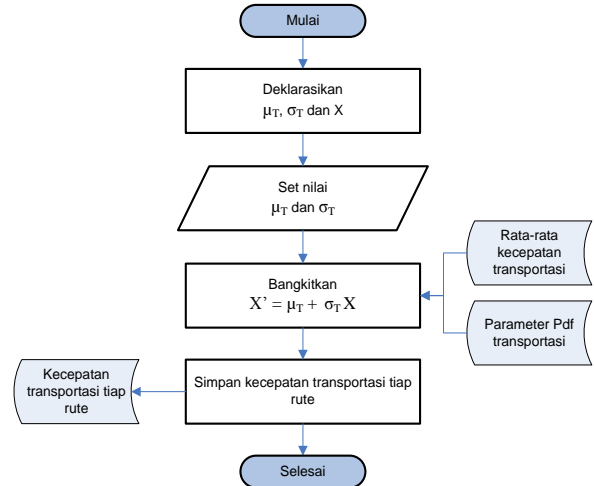
Proses dimulai dengan membaca posisi X dan Y pada depot dan setiap retailer i yang telah mencapai *reorder point*. Selanjutnya, jarak antara depot dengan retailer dan jarak antar retailer dihitung dengan rumusan Phytagoras ( $\text{jarak}^2 = X^2 + Y^2$ ). Hasil perhitungan ditampilkan dalam bentuk array matrik jarak pada stringgrid. Variabel array tersebut selanjutnya digunakan untuk perhitungan waktu transportasi. Proses dilanjutkan dengan modul pembangkitan kecepatan transportasi.



**Gambar 7 Subflowchart Modul Perhitungan Jarak Tiap Ruas Rute**

**Modul pembangkitan kecepatan transportasi**

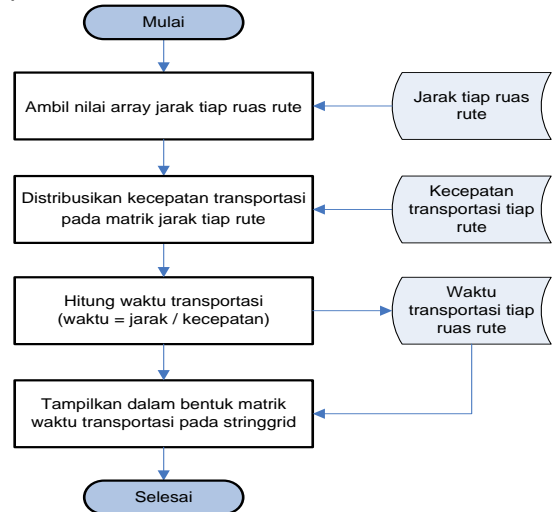
Seperti yang terlihat pada Gambar 8, pembangkitan dimulai dengan mendeklarasikan variabel  $\mu_T$ ,  $\sigma_T$  dan X. Nilai  $\mu_T$  dan  $\sigma_T$  ditentukan sebagai rata-rata kecepatan transportasi yang dibutuhkan dan besar simpangan kecepatan transportasi tersebut. Kemudian bangkitkan rumusan X' dengan parameter Pdf transportasi dan nilai X diperoleh dari *random variate*  $ui[i]$  berdistribusi normal. Selanjutnya, kecepatan transportasi tiap rute disimpan sementara. Perolehan data kecepatan transportasi tersebut digunakan untuk perhitungan waktu transportasi pada modul selanjutnya. Setelah kecepatan transportasi hasil pembangkitan diperoleh maka proses selesai.



**Gambar 8 Subflowchart Modul Pembangkitan Kecepatan Transportasi**

**Modul perhitungan waktu transportasi**

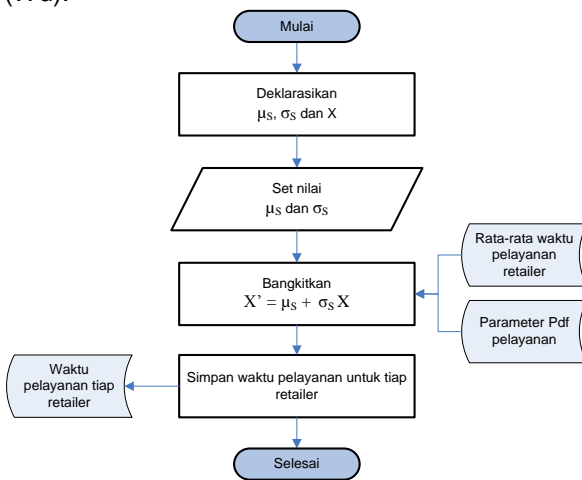
Seperti yang terlihat pada Gambar 9, proses perhitungan waktu transportasi diawali dengan mengambil nilai array jarak tiap ruas rute yang telah dihasilkan pada modul perhitungan jarak tiap ruas rute sebelumnya. Selanjutnya, sejumlah *random number* berupa kecepatan transportasi yang telah dihasilkan pada modul pembangkitan kecepatan transportasi tersebut didistribusikan pada masing-masing hasil perhitungan matrik jarak. Sehingga waktu transportasi antara depot dengan retailer dan antar retailer dapat diketahui dengan rumusan jarak tiap ruas rute dibagi dengan kecepatan transportasi. Hasil perhitungan matrik waktu transportasi tersebut ditampilkan pula pada stringgrid. Waktu transportasi tersebut digunakan sebagai salah satu variabel untuk menentukan waktu distribusi (Wd). Proses perhitungan waktu transportasi selesai jika waktu transportasi tiap ruas rute telah diperoleh.



**Gambar 9 Subflowchart Modul Perhitungan Waktu Transportasi**

**d) Subflowchart dari modul pembangkitan waktu pelayanan (Wp)**

Perhitungan waktu transportasi total diawali dengan pembangkitan waktu pelayanan. Seperti yang terlihat pada Gambar 10, pembangkitan dimulai dengan mendeklarasikan variabel  $\mu_s$ ,  $\sigma_s$  dan X. Nilai  $\mu_s$  ditentukan sebagai rata-rata waktu pelayanan retailer yang dibutuhkan, sedangkan nilai  $\sigma_s$  ditentukan sebagai besar simpangan waktu pelayanan tersebut. Kemudian bangkitkan rumusan X' dengan parameter Pdf pelayanan dan nilai X diperoleh dari *random number*  $ui[j]$  berdistribusi normal. Perolehan data waktu pelayanan tersebut disimpan dan digunakan untuk perhitungan waktu pelayanan pada prosedur selanjutnya. Hasil pembangkitan waktu pelayanan tersebut digunakan sebagai salah satu variabel untuk menentukan waktu distribusi (Wd).



**Gambar 10 Subflowchart Pembangkitan Waktu Pelayanan**

**e) Subflowchart dari modul perhitungan ongkos distribusi**

Gambar 11 merupakan *subflowchart* dari alur proses perhitungan ongkos distribusi. Adapun model perhitungan ongkos distribusi dapat ditentukan sebagai berikut.

Misalkan:

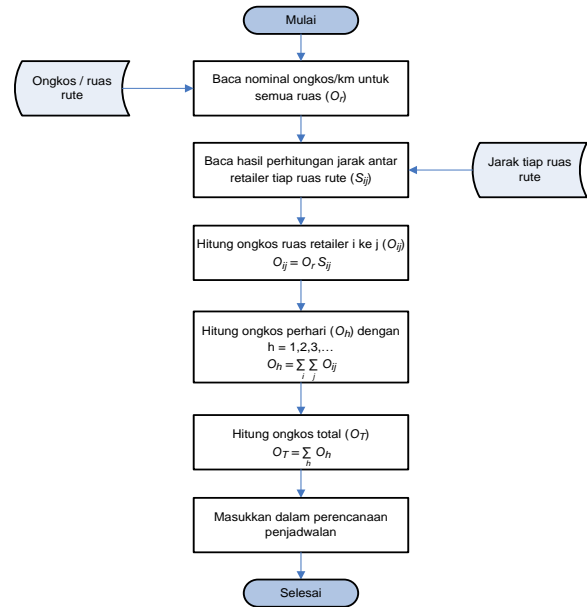
$O_{ij}$  = Ongkos untuk ruas ij (dari retailer i ke j) (dalam Rp)

$S_{ij}$  = Jarak pada ruas ij (dari retailer i ke j) dalam km atau meter

$O_r$  = Ongkos/km (Rp/km) atau (Rp/m) sama untuk semua ruas

$O_h$  = Ongkos perhari dengan  $h = 1, 2, 3, \dots$

$O_T$  = Ongkos total



**Gambar 11 Subflowchart Perhitungan Ongkos Distribusi**

Proses perhitungan ongkos distribusi diawali dengan mengetahui besar nominal ongkos/km yang digunakan untuk semua ruas ( $O_r$ ) dan hasil perhitungan jarak antar retailer tiap ruas rute ( $S_{ij}$ ). Dengan mengetahui besar ongkos/km dan jarak tiap rute yang akan ditempuh, maka depot dapat menghitung besar ongkos ruas retailer i ke j ( $O_{ij}$ ) dengan mengalikan ruas rute yang akan dilalui dengan ongkos per satuan jarak. Atau secara matematis dituliskan dengan rumusan sebagai berikut.

$$O_{ij} = O_r S_{ij} \tag{8}$$

Dari hasil perhitungan ongkos ruas retailer i ke j, ongkos perhari ( $O_h$ ) dengan  $h = 1, 2, 3, \dots$  dapat dihitung dengan rumusan sebagai berikut.

$$O_h = \sum_i \sum_j O_{ij} \tag{9}$$

Dengan demikian, ongkos total untuk pendistribusian dari satu horizon waktu yang telah ditetapkan dapat diketahui dengan rumusan perhitungan sebagai berikut.

$$O_T = \sum_h O_h \tag{10}$$

Proses selesai dengan setelah proses perhitungan ongkos selesai dengan masing-masing perolehan besar ongkos dimasukkan ke dalam perencanaan penjadwalan distribusi untuk dapat mengetahui besarnya ongkos yang akan dikeluarkan untuk proses distribusi selama horizon waktu perencanaan.



## 5 KESIMPULAN

Berikut ini merupakan kesimpulan dari hasil rancangan Sistem Pendukung Keputusan Distribusi Produk.

- a. Sistem Pendukung Keputusan Distribusi Produk menghasilkan suatu perencanaan penjadwalan distribusi untuk satu horizon waktu perencanaan dari hasil pembangkitan secara otomatis guna mengetahui waktu proses distribusi, banyaknya produk yang didistribusikan pada masing-masing retailer dan rute distribusi yang akan dilalui.
- b. Jika setiap ruas retailer tersebut dapat terlayani lebih cepat, maka dalam satu hari perusahaan distribusi dapat melayani

retailer lebih banyak. Dengan kata lain, keuntungan yang diperoleh adalah target produk untuk didistribusikan lebih banyak, kinerja karyawan dapat semakin ditingkatkan, dan keuntungan pemasukan retailer baru dapat semakin besar.

## 6 DAFTAR PUSTAKA

- [1] Turban, Efraim, Jay E. Aronson, and Ting-Peng Liang. 2005. *Decision Support Systems and Intelligent Systems*. Yogyakarta : ANDI
- [2] Law, A. and Kelton, W.D. 1991. *Simulation Modeling and Analysis*. McGraw-Hill, Inc.