

**RANCANG BANGUN *DECISION SUPPORT SYSTEM* DISTRIBUSI PRODUK PADA
INVENTORY ROUTING PROBLEM DENGAN PENDEKATAN HEURISTIK
(Studi Kasus: Perusahaan Yakult)**

Lisnawanty

Program Studi Komputerasi Akuntansi, AMIK BSI Pontianak
Jalan Abdurahman Saleh No. 18 A, Pontianak
lisnawanty.lsy@bsi.ac.id

Abstract

The company is an organization that has a complexity of problems in managing the relationship between retailers and distributors to maintain product distribution process runs smoothly. Changes in market structure, products, and technologies of production continues to occur so that the effect is determined at the discretion of management. In this regard, the case of Inventory Routing Problem (IRP) often occurs in uncertainty in coordinating some activity distribution of products to a number of retailers. Product distribution system that is run without the basic result in reduced services to retailers. In other words, the distribution companies are not able to maintain service levels remain high. When the distribution system planned without basis in estimating demand so there are a number of retailers that are not provided in time, the planning time horizon for the retailer must wait for resupply performed again at the next cycle. Determination of suboptimal solutions to some complex decision problems can face a bottleneck in terms of product quantity, time and cost can use a heuristic approach in building Decision Support System. Product Distribution DSS is a tool used to generate and produce suboptimal solutions in generating support for scheduling distribution. The decision resulting from the DSS Product Distribution is determining the time distribution will be made, determine many products that will be distributed, and determine the route to be traversed.

Keywords : *Inventory Routing Problem, heuristic, Decision Support System*

1. PENDAHULUAN

Dalam sistem pendistribusian pada beberapa perusahaan distribusi, kasus *Inventory Routing Problem* (IRP) seringkali terjadi dengan bersumber pada ketidakpastian dalam mengkoordinasi beberapa aktivitas pendistribusian produk kepada sejumlah retailer seperti halnya mengkombinasikan permasalahan pendistribusian produk dengan manajemen persediaan pada sejumlah *retailer*.

Berkenaan dengan *Inventory Routing Problem* (IRP) tersebut, Perusahaan Yakult sesuai untuk

dijadikan sebagai sampel kasus. Pemilihan Perusahaan Yakult sebagai sample kasus dikarenakan Perusahaan Yakult memiliki sistem pendistribusian yang menerapkan sistem *Inventory Management*. Manajemen perusahaan tersebut menitikberatkan pada pengontrolan penambahan persediaan pada setiap *retailer* dan mempertimbangkan waktu pengiriman, banyaknya produk yang akan didistribusikan, dan alur distribusi dari satu rute ke rute berikutnya. Sehingga permasalahan Perusahaan Yakult terjadi disebabkan variasi *demand* yang terjadi pada sejumlah retailer dikaitkan dengan

prioritas penjadwalan yang akan didistribusikan dengan fasilitas transportasi yang ada.

Berdasarkan uraian kasus di atas, maka permasalahan utama yang kerap kali terjadi pada perusahaan distribusi adalah bersumber pada ketidakpastian dalam mengkoordinir beberapa aktivitas pendistribusian produk kepada sejumlah retailer, terutama lemah dalam menganalisis perkiraan *demand* (permintaan) masing-masing *retailer*.

Asumsi bahwa horizon waktu perencanaan distribusi selama 1 bulan (4 minggu) dan terdapat 4 *cluster* (masing-masing *cluster* diwakili 1 fasilitas transportasi) yang membagi semua retailer yang ada, maka kemungkinan yang terjadi adalah waktu perencanaan distribusi produk pada satu *cluster* adalah 1 minggu. Jika ternyata perusahaan distribusi tidak melakukan perencanaan distribusi dengan baik, dalam artian retailer yang membutuhkan penambahan produk tidak terlayani tepat pada waktunya karena kendala waktu dan kapasitas angkut, maka *retailer* tersebut harus menunggu waktu siklus berikutnya untuk mendapatkan *resupply* kembali yaitu ± 1 minggu berikutnya. Namun jika diasumsikan sesuai dengan DSS yang akan dibangun, perusahaan distribusi menetapkan horizon waktu perencanaan distribusi selama 1 bulan (4 minggu) dan terjadi retailer yang tidak terlayani untuk horizon waktu distribusi saat ini harus menunggu waktu siklus berikutnya untuk mendapatkan *resupply* kembali yaitu ± 2 minggu berikutnya.

Untuk itu, perusahaan distribusi membutuhkan adanya suatu DSS untuk mendukung perusahaan distribusi dalam melakukan penjadwalan distribusi. Keputusan yang akan dibuat adalah sebagai berikut:

- Menentukan waktu distribusi akan dilakukan.
- Menentukan banyak produk yang akan didistribusikan.
- Menentukan rute yang akan dilalui.

DSS Distribusi Produk merupakan alat yang digunakan untuk membangkitkan dan menghasilkan dukungan solusi suboptimal. DSS berinteraksi dengan seorang pengambil keputusan (manajer) yang memberikan perkiraan *eksponensial* dalam memprediksi permintaan produk dengan menetapkan stok untuk *resupply* dan kapasitas efektif sehingga dapat melakukan perencanaan kapasitas dan penjadwalan distribusi.

2. TINJAUAN PUSTAKA

2.1 *Decision Support System* (DSS)

Turban, et. al. (2005:5) menguraikan beberapa definisi mengenai Sistem Pendukung Keputusan (*Decision Support System*) berdasarkan beberapa ahli sebagai berikut.

- a. Menurut Little (1970), DSS sebagai sekumpulan prosedur berbasis model untuk data pemrosesan dan penilaian guna membantu para manajer mengambil keputusan. Little menyatakan bahwa untuk sukses, sistem tersebut haruslah sederhana, cepat, mudah dikontrol, adaptif, lengkap dengan isu-isu penting, dan mudah berkomunikasi.
- b. Menurut Moore dan Chang (1980), DSS 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 reguler dan tidak terencana.

- c. Menurut Keen (1980), DSS sebagai suatu produk dari proses pengembangan dimana pengguna DSS, pembangun DSS, dan DSS itu sendiri mampu mempengaruhi satu dengan yang lain

Dan menurut Alter (2002) dalam Kadir (2003:117), DSS adalah sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data yang digunakan untuk membantu pengambilan keputusan pada situasi semi terstruktur dan situasi yang tidak terstruktur.

2.2 Pendekatan Heuristik sebagai Metode Pemecahan Masalah

Menurut Turban (2005:136), pendekatan heuristik merupakan pendekatan menggunakan heuristik untuk sampai pada solusi yang cukup baik (biasanya dalam rentang 90-90,99 persen nilai objektif dari suatu solusi optimal) dan dapat dikerjakan dengan mudah. Heuristik dapat menjadi kuantitatif, dan karena itu dapat memainkan peran penting pada basis model DSS, dimana heuristik digunakan dalam memecahkan sebuah masalah pemrograman *integer* yang kompleks. Heuristik dapat juga kualitatif dan dapat memainkan peran penting untuk memberikan pengetahuan bagi sistem pakar.

2.3 Pembangkitan Random Number (*Random Number Generator/RNG*) dengan Algoritma *Prime Modulus Multiplicative 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*) yang

digunakan pada proses pembangkitan 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*)

2.4 Pembangkitan *Random Variates*

2.4.1 Pendekatan Umum Pembangkitan *Random Variate* dengan Metode *Invers Transform*

Pendekatan yang umum digunakan untuk membangkitkan *random variate* adalah *Inverse-Transform*. Pembangkitan *random variate* X dan fungsi distribusi (*distribution function*) F yang bersifat kontinu adalah ketika $0 < F(x) < 1$. Hal ini berarti jika $x_1 > x_2$ dan $0 < F(x_1) \leq F(x_2) < 1$, maka $F(x_1) < F(x_2)$. Berikut adalah algoritma untuk membangkitkan *random variate* X yang memiliki fungsi distribusi F .

Generate $U \sim U(0,1)$

Return $X = F^{-1}(U)$

Keterangan:

\sim dibaca didistribusikan sebagai

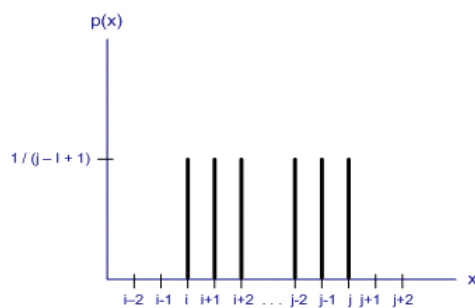
$F^{-1}(U)$ selalu didefinisikan untuk $0 \leq U \leq 1$ dan *range* dari F adalah $[0,1]$. Untuk menentukan *random number* U dengan metode *invers-transform* maka angka atau peluang harus bernilai sama (*uniform*) dengan interval $[0,1]$ pada $F(x)$.

2.4.2 Pembangkitan *Random Variate* Kontinu

a. *Discrete Uniform*

Discrete uniform digunakan sebagai sebuah model pertama untuk menyatakan suatu kuantitas dalam probabilitas distribusi yang berubah di antara bilangan integer i dan j . Dalam pembangkitan *random variate* terdapat peluang untuk muncul angka yang sama (*uniform*).

Gambar 1 berikut ini merupakan gambaran dari *Discrete Uniform*.



Sumber: Law (1991)

Gambar 1. *Discrete Uniform*

Algoritma untuk membangkitkan X pada *discrete uniform* dengan menggunakan metode *inverse-transform* adalah sebagai berikut:

Generate $U \sim U(0,1)$.

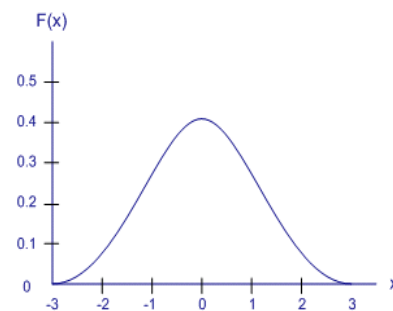
Return $X = i + [(j - i + 1)U]$.

Dalam DSS yang dirancang, metode *Discrete Uniform* digunakan untuk menghasilkan angka pembangkitan *random number*.

b. *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.

Gambar 2 berikut ini merupakan gambaran dari *Normal*.



Sumber: Law (1991)

Gambar 2. *Normal*

Algoritma untuk membangkitkan X pada *normal distribution* adalah sebagai berikut:

- 1) Generate U_1 and U_2 as IID $U(0,1)$, let $V_i = 2 U_i - 1$ for $i = 1,2$, and let $W = V_1^2 + V_2^2$.
- 2) If $W > 1$, go back to step 1. Otherwise, let $Y = (-2 \ln W) / W$, $X_1 = V_1 Y$, $X_2 = V_2 Y$. Then X_1 and X_2 are IID $N(0,1)$ *random variate*.

Model *normal* digunakan untuk mengasumsikan kecepatan transportasi dan waktu pelayanan dari sejumlah retailer. Asumsi ini diambil dengan mengacu 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 dbangkitkan dengan rumusan sebagai berikut:

$$X' = \mu_T + \sigma_T X$$

Keterangan:

μ_T = nilai rata-rata kecepatan transportasi

σ_T = simpangan rata-rata dengan nilai konstanta (bebas ditentukan sendiri)

- Waktu pelayanan
Kecepatan transportasi dbangkitkan dengan rumusan sebagai berikut:

$$X' = \mu_S + \sigma_S X$$

Keterangan:

μ_S = nilai rata-rata waktu pelayanan

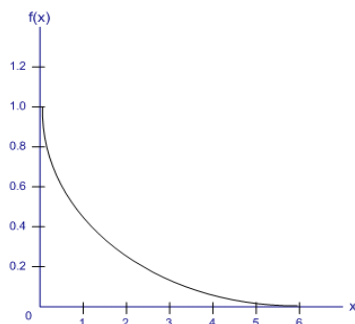
σ_S = simpangan rata-rata dengan nilai konstanta positif (bebas ditentukan sendiri)

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

c. Exponential

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

Gambar 3 berikut ini merupakan gambaran dari *Exponential*.



Sumber: Law (1991)

Gambar 3. Exponential Distribution

Algoritma untuk membangkitkan X pada *exponential* dengan menggunakan metode *inverse-transform* adalah sebagai berikut:

Generate $U \sim U(0,1)$.

Return $X = -\beta \ln U$.

Dalam DSS yang dirancang, model *exponential* digunakan untuk memperkirakan *demand* yang terjadi pada setiap retailer dengan pembangkitan *random number* yang dihasilkan dari model *discrete uniform* 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 merupakan rumusan matematis untuk pembangkitan *demand* tiap retailer.

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

$$U_i = Z_i / m$$

$$E_i = b \cdot \ln(U_i)$$

Rumus PMMLCGs di atas ditentukan nilai parameter $a = 630360016$, $Z_i = 132645$, $c = 0$, dan $m = 2^{31}-1$. Penentuan nilai tersebut mengacu pada literatur Law and Kelton, 1991.

2.4.3 Algoritma Farthest Insertion

Salah satu algoritma dari pendekatan heuristik (*insertion heuristic*) yang dapat diaplikasikan untuk pencarian rute tour kendaraan yaitu algoritma *farthest insertion* untuk dapat menentukan jarak minimal suatu lokasi tour maksimal dari titik-titik (*node*) yang di *input* ke dalam peta secara geografis.

3. METODE PENELITIAN

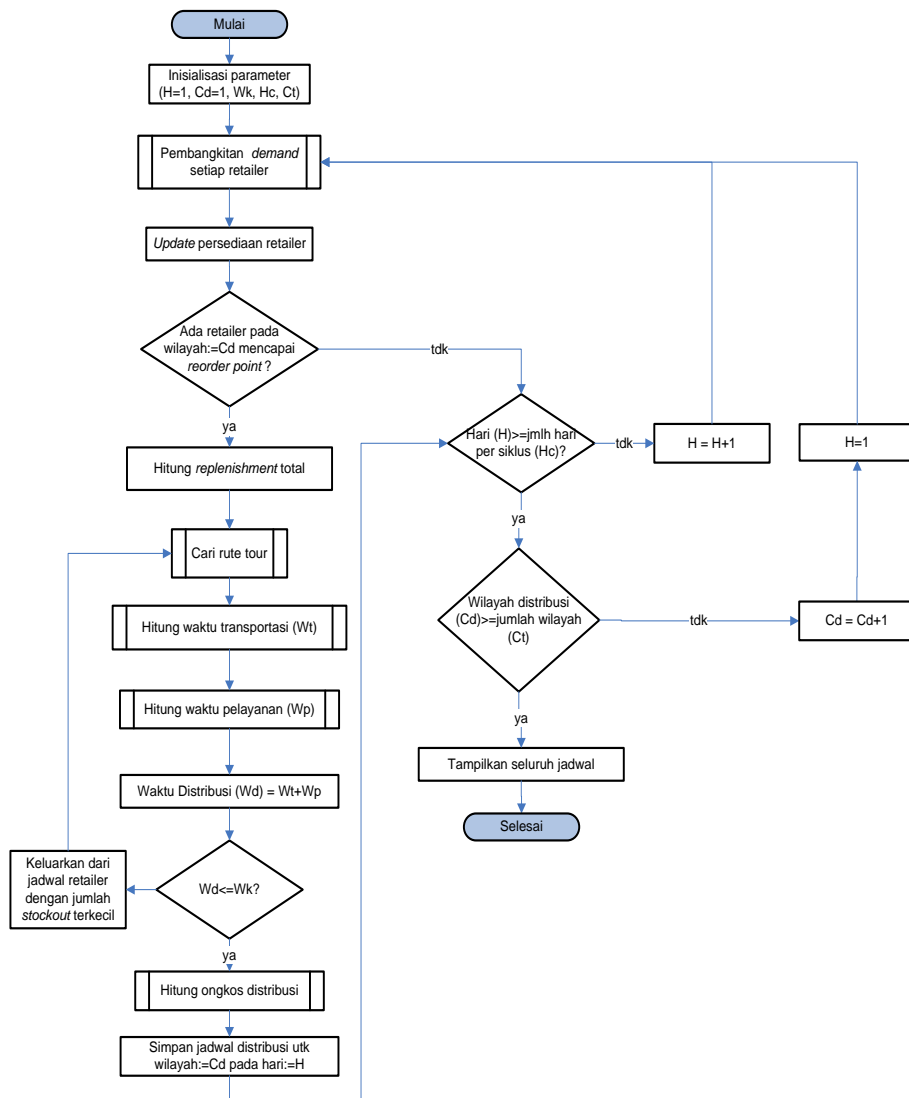
Adapun metode penelitian dan pengembangan perangkat lunak adalah menggunakan metode SDLC (*Software Development Life Cycle*) dengan model Waterfall. Metode pengujian perangkat lunak yang dirancang menggunakan metode *blackbox* dan *whitebox*.

4. PEMBAHASAN

a. Konsep Perancangan

Sistem Pendukung Keputusan dirancang untuk digunakan bagi seorang manajer atau pengelola yang berstatus sebagai *administrator*. Perancangan konseptual DSS digambarkan dalam sebuah diagram alir berupa *flowchart* (Gambar 4) yang

menggambarkan alur penyelesaian masalah dari algoritma yang digunakan dalam program dengan menggunakan ketiga model (*uniform*, *exponential*, dan *normal*) sebagaimana telah diuraikan pada bagian Tinjauan Pustaka. Berikut gambaran cara kerja DSS yang dibangun.



Gambar 4. Perancangan Konseptual DSS

Adapun alur proses diagram alir DSS pada Gambar 4 di atas diuraikan sebagai berikut.

Program dimulai dengan inisialisasi 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.

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 *route 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 *route 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.

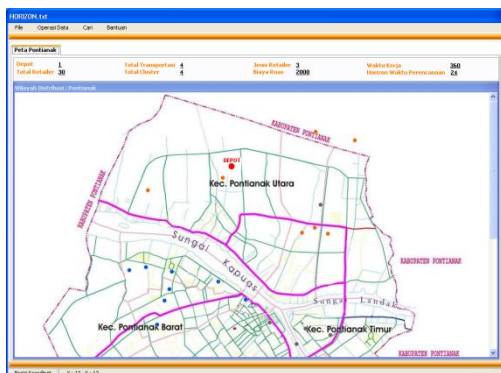
b. Hasil Perancangan

Sebelum *user* masuk ke dalam program, *user* harus login terlebih dahulu sebagaimana terlihat pada Gambar 5 berikut.



Gambar 5. Form Login

Setelah *user* berhasil login, maka *user* akan memasuki program inti (Gambar 6) untuk dapat dilakukannya proses pengolahan data dan dijalkannya proses pembangkitan sehingga perencanaan penjadwalan dapat dihasilkan guna membantu depot dalam memberikan acuan dasar terhadap proses distribusi yang akan dijalankan dalam suatu horizon waktu perencanaan tertentu.



Gambar 6. Bagian Inti Program

Ketika *user* berhasil memasuki bagian inti program sebagaimana Gambar 6, maka secara langsung pembangkitan untuk memperoleh hasil optimasi tersebut diproses. Pembangkitan dapat diproses jika data yang dibutuhkan telah dipenuhi atau diisi.

Adapun rancangan menu utama pada program ini, antara lain:

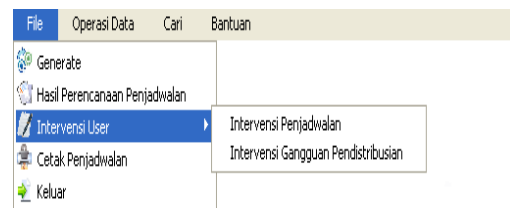
a. File

Submenu dalam menu File terlihat pada Gambar 7 sebagai berikut.



Gambar 7. Organisasi Menu File

Submenu Hasil Perencanaan Penjadwalan, Intervensi User, dan Cetak Penjadwalan dapat diaktifkan jika Generate telah diproses, sebagaimana terlihat pada Gambar 8 sebagai berikut.



Gambar 8. Menu File setelah Proses Pembangkitan

Generate merupakan fasilitas yang disediakan untuk menjalankan proses pembangkitan (Gambar 9) dan memperoleh penjadwalan.



Gambar 9. Form Aktivasi Proses Pembangkitan (*Generate*)

Keutamaan yang dibutuhkan pada proses Generate ini adalah kelengkapan data. Untuk menunjang proses kelengkapan data, maka menu Operasi Data disediakan untuk dapat dilakukannya proses penambahan, perubahan, atau penghapusan data. Output pada proses Generate ini adalah hasil perencanaan penjadwalan, yang mencakup tanggal mulai distribusi, *replenishment* total, serta rute retailer, masing-masing *replenishment*, dan biaya pada masing-masing wilayah, cluster, dan hari pelaksanaan distribusi,

sebagaimana dapat dilihat pada submenu Hasil Perencanaan Penjadwalan (Gambar 8).

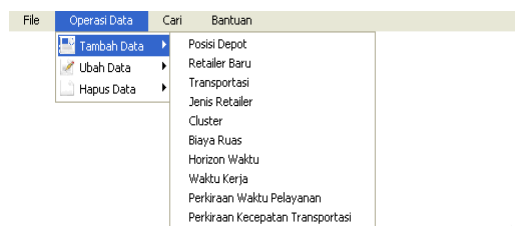
Sementara cara kerja dari Intervensi User, yaitu:

Ketika terjadi suatu kondisi kebijakan depot untuk dilakukannya perubahan perencanaan penjadwalan kembali (*reschedule*), maka proses tersebut dijalankan pada fasilitas Intervensi *User*. Terdapat dua fasilitas yang dapat diproses pada intervensi *user*, yaitu intervensi penjadwalan dan intervensi gangguan pendistribusian.

Intervensi penjadwalan diproses dengan memilih retailer yang akan dilakukan perubahan waktu distribusi (Hari) dengan pertimbangan waktu yang singkat dan jarak terdekat. Perubahan perencanaan penjadwalan dapat dilakukan. Setelah perubahan waktu distribusi (Hari) dilakukan, maka hari selanjutnya (Hari+1) akan dijadwalkan kembali (*reschedule*).

b. Operasi Data

Submenu dalam menu Operasi Data terlihat pada Gambar 10 sebagai berikut.



Gambar 10. Organisasi Menu Operasi Data

Berdasarkan Gambar 10 tersebut di atas, tidak semua data dapat dihapus. Dengan kata lain, hanya beberapa data yang dapat dihapus, diantaranya:

- Data Retailer,
- Data Transportasi,
- Data Jenis Retailer, dan
- Data Cluster.

c. Cari

Program memungkinkan *user* untuk melakukan pencarian data berdasarkan Nama Retailer dan Cluster Retailer, sebagaimana terlihat pada Gambar 11.



Gambar 11. Organisasi Menu Cari

d. Bantuan

Menu Bantuan pada program menampilkan informasi bagi *user* untuk dapat lebih memahami program yang sedang dijalankan.

2. PENUTUP

Berdasarkan hasil rancangan dan pengujian, maka dapat disimpulkan beberapa hal sebagai berikut.

1. Pengujian sistem menggunakan asumsi parameter pembangkitan waktu pelayanan model *Normal* dengan rata-rata waktu pelayanan (μ) = 30 dan simpangan waktu pelayanan (σ) = 20 tersebut menghasilkan rentang *random number* mulai dari 10 sampai dengan 50 dengan μ = 30 sebagai nilai tengah.
2. Pada pembangkitan kecepatan transportasi, dengan rata-rata kecepatan transportasi (μ) = 50 dan simpangan kecepatan transportasi (σ) = 20 tersebut pembangkitan kecepatan transportasi menghasilkan rentang *random number* mulai dari 30 sampai dengan 70 dengan μ = 50 sebagai nilai tengah.
3. Berdasarkan hasil pengujian dengan parameter waktu pelayanan (μ = 30 dan σ = 20), kecepatan transportasi (μ = 50 dan σ = 20), serta asumsi waktu kerja selama 6 jam, perencanaan penjadwalan distribusi yang telah

dihasilkan menunjukkan bahwa dalam satu hari depot dapat melayani retailer sebanyak ± 6 retailer.

4. Hasil pengujian juga menunjukkan bahwa dari 30 retailer dibutuhkan ± 3 hari untuk menyelesaikan semua retailer dalam perencanaan penjadwalan.

DAFTAR PUSTAKA

- Kadir, Abdul. 2003. Pengenalan Sistem Informasi. Yogyakarta: ANDI.
- Law, A. and Kelton, W.D. 1991. *Simulation Modeling and Analysis*. McGraw-Hill, Inc.
- Syamsi, Ibnu. 2000. Pengambilan Keputusan dan Sistem Informasi. Jakarta: Bumi Aksara.
- Turban, Efraim, Jay E. Aronson, and Ting-Peng Liang. 2005. *Decision Support Systems and Intelligent Systems*. Yogyakarta: ANDI.

