
Perangkat Lunak Simulasi *Periodic Vehicle Routing Problem (PVRP)* dengan *Tabu Search*

Danny Manongga, Theophilus Wellem, Kasih Septi

Fakultas Teknologi Informasi
Universitas Kristen Satya Wacana
Jl. Dipenogoro 52-60, Salatiga 50711, Indonesia

Abstract

Logistics distribution system is an aspect for product and global economic improvement. Especially in the cases of production and market globalisation, which cause the global competition. For logistic distribution system, there is a problem arisen, which is finding the best routes to maximize the cost and time efficiency. The writers create software to help the companies find the best routes entrenched in the visits combination to the customers in a definite period of time, production capacity, and customers-visit pattern, using the *Tabu Search*. The result is the best routes, based on the total customers, the customers' position, and delivery periods.

Keywords: Logistic Distribution, Periodic Vehicle Routing Problem (PVRP), Tabu Search

1. Pendahuluan

Sistem distribusi logistik merupakan bagian yang sangat penting dalam rangkaian usaha pengembangan produk maupun dalam pengembangan ekonomi secara keseluruhan terutama dikaitkan dengan aspek globalisasi produksi dan globalisasi pasar yang akhirnya akan menimbulkan persaingan global. Di era globalisasi, setiap negara atau perusahaan dapat melakukan kegiatan produksi dimana saja yang paling menguntungkan bagi perusahaan, baik itu seluruh komponen ataupun sebagian.

Lokasi konsumen yang tersebar mengharuskan perusahaan mampu menentukan lokasi yang strategis agar logistik mencapai tujuan dalam keadaan yang dapat dipakai, ke lokasi dimana ia dibutuhkan, serta dalam jumlah yang tepat pada waktu yang dibutuhkan [1]. Perusahaan biasanya menggunakan cara manual menentukan rute yang ditempuh dalam penyampaian logistik ke pelanggan. Permasalahan yang sering dihadapi oleh sebuah perusahaan adalah bagaimana menemukan rute yang tepat sehingga menghemat biaya dan waktu agar semua permintaan pelanggan dapat terpenuhi. Dengan menghemat waktu dan biaya maka perusahaan tidak harus menanggung biaya transportasi yang cukup tinggi.

Salah satu permasalahan pencarian rute (*routing*) yang dapat diselesaikan dengan *Tabu Search* adalah *Periodic Vehicle Routing Problem (PVRP)*. PVRP

adalah permasalahan pencarian rute untuk seperangkat alat transportasi dengan satu atau beberapa depot yang harus melayani beberapa kota atau pelanggan. *Tabu Search* dapat menemukan rute terbaik dari beberapa kemungkinan rute yang dapat dilalui.

2. Periodic Vehicle Routing Problem (PVRP)

Periodic Vehicle Routing Problem (PVRP) dijabarkan dengan cara memperpanjang rencana periode menjadi m hari. Tujuan dari PVRP adalah untuk meminimalisasi jumlah kendaraan dan waktu rata-rata perjalanan untuk menyuplai pelanggan (*customers*). Dengan PVRP dimungkinkan setiap kendaraan pengangkut (*vehicle*) tidak kembali ke depot dalam satu hari. Selama m hari pelanggan (*customers*) harus dikunjungi minimal satu kali. Berikut perumusan PVRP [3]:

- a. Meminimalisasi semua biaya untuk semua rute. Setiap pelanggan (*customer*) harus mempunyai permintaan harian yang harus dipenuhi dalam satu kali pengiriman oleh satu kendaraan saja.
- b. Jika rencana periode $m = 1$, maka PVRP menjadi VRP.
- c. Setiap pelanggan (*customers*) dalam PVRP harus dikunjungi selama waktu k , dimana $1 \leq k \leq m$.
- d. PVRP bisa dilihat sebagai sebuah permasalahan untuk mengadakan satu group rute untuk setiap hari sehingga setiap *constraints* yang ada harus dipenuhi dan seluruh biaya bisa diminimalisasi. PVRP bisa dilihat juga sebagai sebuah permasalahan mengoptimalkan *combinatorial multi-level*.

1. Level 1 bertujuan untuk menemukan beberapa pilihan yang mungkin/kombinasi untuk setiap pelanggan (*customers*).

Contoh: Jika rencana periode $m = 3$ hari

d_1, d_2, d_3

maka kombinasi yang mungkin adalah:

0 \rightarrow 000

1 \rightarrow 001

2 \rightarrow 010

3 \rightarrow 011

4 \rightarrow 100

5 \rightarrow 101

6 \rightarrow 110

7 \rightarrow 111

Jika satu pelanggan meminta dua kali kunjungan, maka alternatif pengiriman adalah:

$\{d_1, d_2\} \{d_1, d_3\} \{d_2, d_3\}$

(pilihan 3, 5, 6 dari Tabel 1).

Tabel 1 Permintaan Pelanggan

Customers	Daily demand	Number of visited	Number of combinations	Possible Combinations
1	30	1	3	1, 2, 4
2	20	2	3	3, 5, 6
3	20	2	3	3, 5, 6
4	30	2	3	3, 5, 6
5	10	3	1	7

- Level 2 bertujuan memilih satu dari beberapa alternatif untuk setiap pelanggan (*customers*), sehingga *daily constraints* terpenuhi. Oleh sebab itu harus dipilih pelanggan (*customers*) mana yang harus dipenuhi setiap harinya.
- Level 3 bertujuan memecahkan VRP *problem* setiap hari.

3. *Tabu Search*

Tabu search adalah salah satu metode optimasi untuk mencari solusi terbaik dari permasalahan pencarian rute pengiriman logistik. Algoritma ini diperkenalkan pertama kali oleh Glover [4]. Metode *tabu search* tidak hanya mencari solusi yang paling baik dan meninggalkan solusi yang kurang baik, tetapi metode ini juga mampu menyimpan solusi yang pernah ditemukan dan mencegah penggunaan kembali solusi yang telah digunakan sehingga tidak terjadi perulangan [4]. Hal ini dibuktikan dengan pemakaian memori jangka pendek (*short-term memory*). *Short-term memory* digunakan untuk menyimpan solusi-solusi yang telah ditemukan dan untuk mencari solusi-solusi lain yang belum ditemukan sehingga tidak terjadi perulangan. Penggunaan memori secara matematis inilah yang menjadi kelebihan metode *tabu search* [4]. *Tabu search* terdiri dari beberapa elemen, yaitu

- Tabu list* (memori jangka pendek) yaitu memori untuk menyimpan jumlah solusi yang terbatas yang memungkinkan terjadinya perulangan.
- Aspiration criteria* yaitu elemen untuk mencegah proses pencarian mengalami stagnasi (terhambat) karena adanya proses pengujian yang disertai *tabu move*.
- Long term memory* (memori jangka panjang) yaitu elemen untuk menyimpan atribut solusi yang akan digunakan dalam *intensification* (untuk memprioritaskan pada atribut dari satu set solusi) dan *diversification* (untuk memperkecil atribut solusi ketika dipilih untuk memperluas pencarian solusi).

Metode pencarian lain akan menyimpan solusi terbaik sampai sejauh solusi yang telah ditemukan. Namun metode *tabu search* akan menemukan dan membandingkan semua solusi secara keseluruhan, kemudian akan menyimpan solusi terbaik dari keseluruhan solusi yang telah ditemukan [5].

Rumus Dasar Algoritma *Tabu Search* (secara umum) [5]

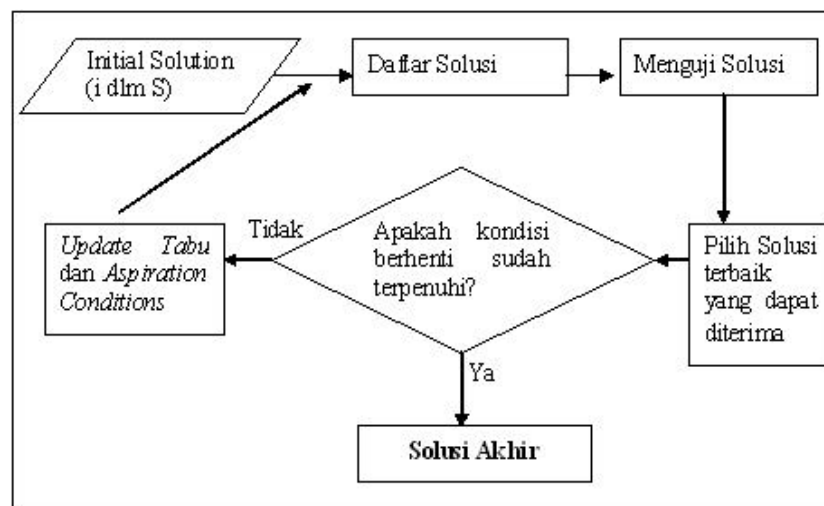
Inisialisasi :

- Langkah 1: Pilih solusi i yang mungkin dalam S . Set $i^* = i$ dan $k = 0$.

- Langkah 2: Tetapkan $k = k+1$ dan hasilkan himpunan bagian V^* dari solusi dalam himpunan solusi $N(i,k)$.
- Langkah 3: Pilih solusi terbaik j dalam himpunan bagian V^* . Tetapkan $i = j$.
- Langkah 4: Jika $f(i) \leq f(i^*)$ maka tetapkan $i^* = i$.
- Langkah 5: Jika kondisi berhenti terpenuhi maka pencarian berhenti. Jika tidak, lakukan langkah 2.

Algoritma Dasar *Tabu Search*

Gambar 1 adalah alur kerja algoritma *tabu search* dalam bentuk *flowchart*, sebagai dasar pembuatan rancangan distribusi logistik.



Gambar 1 Cara Kerja Algoritma *Tabu Search*

Algoritma dasar *Tabu Search*, dijelaskan sebagai berikut:

- Langkah 1: Pilih solusi awal i dalam himpunan S . Tetapkan $i^* = i$ dan $k = 0$ dimana i^* adalah solusi terbaik dan k adalah banyaknya perulangan yang terjadi saat dilakukan pencarian solusi terbaik i^* .
- Langkah 2: Tetapkan $k = k+1$ dan hasilkan himpunan bagian V^* dari solusi dalam himpunan solusi $N(i,k)$ sehingga *tabu conditions* tidak memenuhi dan *aspirations conditions* terpenuhi.
- Langkah 3: Pilih solusi terbaik j dalam himpunan bagian V^* . Tetapkan $i = j$.
- Langkah 4: Jika $f(i) \leq f(i^*)$ maka tetapkan $i^* = i$.
- Langkah 5: *Update tabu* dan *aspiration conditions*.
- Langkah 6: Jika kondisi berhenti (*stopping conditions*) terpenuhi, maka pencarian berhenti. Jika tidak, lakukan langkah 2.

Kondisi berhenti (*stopping conditions*) akan terpenuhi jika:

- Langkah 1: $N(i,k+1) = \emptyset$ atau jika tidak ada solusi yang mungkin di sekitar solusi i .
- Langkah 2: Nilai k lebih besar dari batas maksimum perulangan yang diijinkan.

- Langkah 3: Banyaknya perulangan yang terjadi dari mulai perbaikan solusi i adalah lebih besar dari jumlah yang ditetapkan.

dimana:

i = solusi yang ditemukan

i^* = solusi terbaik dari solusi yang ditemukan

k = perulangan

j = solusi yang ditemukan untuk perulangan berikutnya

S = himpunan solusi yang mungkin

V^* = himpunan bagian dari $N(i,k)$

$N(i,k)$ = himpunan solusi yang mungkin untuk semua perulangan

4. Metode Penelitian

Metode pengembangan sistem yang digunakan dalam aplikasi ini adalah metode *prototype*, dengan tahapan-tahapan yang dilalui yaitu *requirement gathering*, *quick design*, *building prototyping*, dan *evaluation*. Tahapan *requirement gathering* adalah tahapan analisa seluruh kebutuhan yang diperlukan dalam perancangan perangkat lunak. Tahapan ini meliputi analisa sistem, analisa kebutuhan antarmuka, dan analisa kebutuhan bahasa pemrograman. Dalam tahap *quick design* dijabarkan aspek perangkat lunak dari segi input maupun output. Tahap *building prototyping* merupakan proses pembuatan *prototype* perangkat lunak sesuai dengan hasil perancangan yang telah dilakukan. Tahap evaluasi yang merupakan tahapan terakhir, *prototype* yang dibuat akan diperiksa kembali sebagai langkah penyempurnaan.

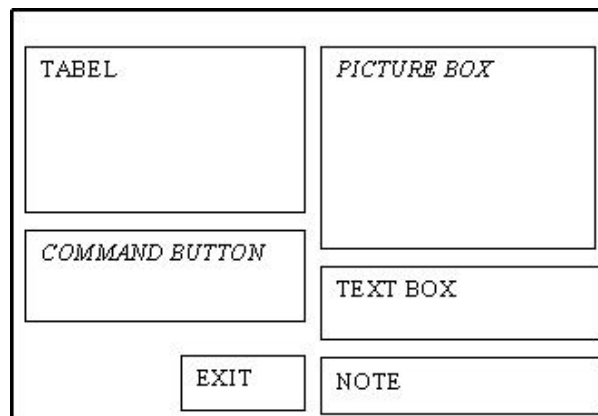
Kebutuhan antarmuka (*interface*) yang dibuat bersifat *user friendly* dengan tujuan agar program yang dibangun dapat digunakan dengan mudah oleh pengguna. Karena program yang dibangun dimaksudkan untuk *user* yang memiliki pengetahuan tentang penggunaan komputer yang beragam, maka antarmuka yang digunakan akan dilengkapi dengan jendela bantuan yang berisi petunjuk penggunaannya.

Perancangan antarmuka merupakan rancang bangun dari interaksi pengguna dengan komputer. Perancangan ini dapat berupa input data maupun menampilkan informasi kepada pengguna. Adapun bentuk rancangan *Form* Utama seperti ditunjukkan pada Gambar 2.

File	About	Help
Tool Bar		
IMAGE		

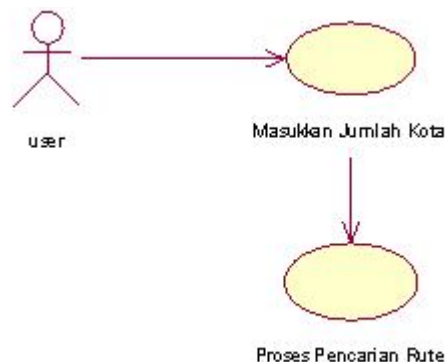
Gambar 2 Rancangan *Form* Utama

Antarmuka proses merupakan antarmuka untuk melakukan proses pencarian rute terbaik dari sejumlah data yang telah dimasukkan oleh *user*. Antarmuka ini memiliki enam komponen yaitu Tabel, *Command Button*, *Picture Box*, *Text Box*, *Note*, dan *Exit*. Tabel digunakan untuk menampilkan nama kota, posisi kota, serta permintaan kunjungan masing-masing kota. *Command button* terdiri dari enam tombol yaitu tombol Masukkan Kota, Tambah Kota, Edit Kota, Hapus Kota, Bersihkan Daftar, dan Cari Jalur.



Gambar 3 Rancangan *Form Kasus Baru*

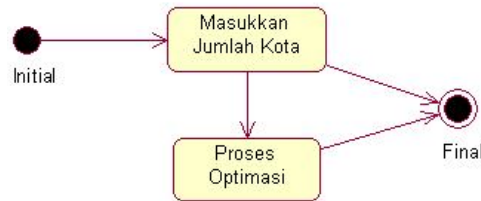
Pemodelan *Unified Modeling Language* (UML) yang digunakan adalah *use case diagram*, *statechart diagram*, dan *activity diagram*. *Use case diagram* yaitu proses-proses yang terjadi dalam suatu *software*, menggambarkan apa yang sedang dilakukan oleh seorang aktor.



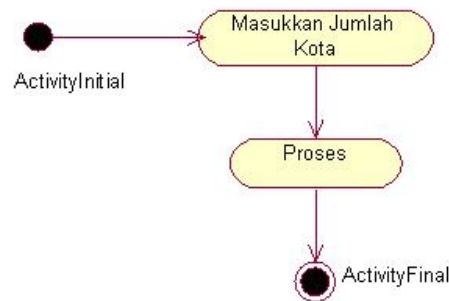
Gambar 4 *Use Case Diagram* Pencarian Rute

Sedangkan *statechart diagram* merupakan diagram yang menggambarkan perubahan *state* dari *state* yang satu ke *state* lainnya. Gambar 2 adalah bentuk diagram *state* untuk proses pencarian rute.

Activity diagram menggambarkan proses-proses yang terjadi mulai aktivitas dimulai sampai aktivitas berhenti. *Activity diagram* ini mirip dengan diagram alir.

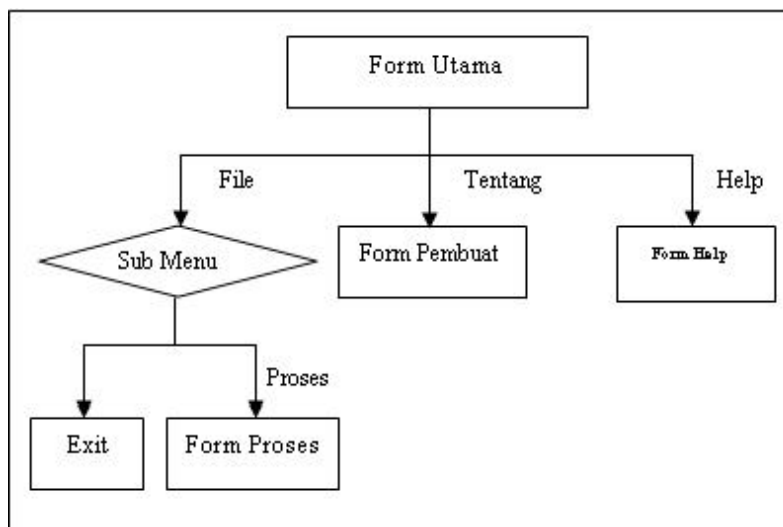


Gambar 5 State Diagram Pencarian Rute



Gambar 6 Activity Diagram Pencarian Rute

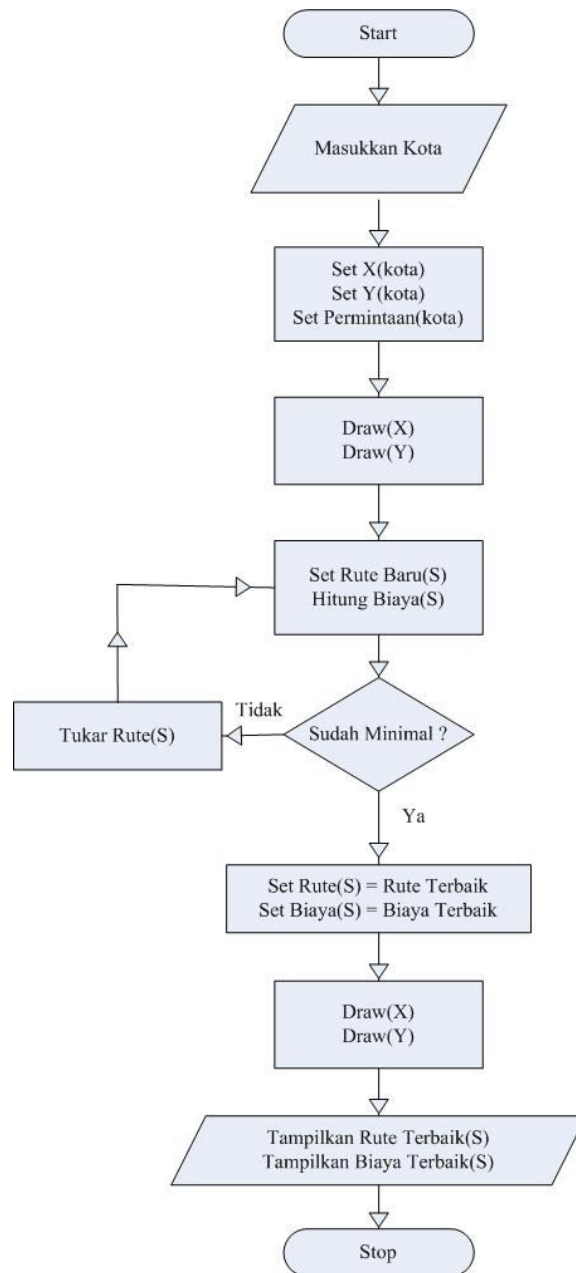
Secara keseluruhan alur kerja aplikasi pencarian rute digambarkan pada Gambar 7.



Gambar 7 Diagram Alir Aplikasi Pencarian Rute

Pada Gambar 8 dapat dijelaskan alur program adalah untuk masukan awal dibutuhkan jumlah kota. Setelah itu akan dilakukan proses pengaturan posisi X dan Y dan jumlah permintaan sejumlah kota tersebut secara random (acak). Proses selanjutnya adalah meletakkan posisi masing-masing kota ke dalam visualisasi. Setelah divisualisasikan kemudian dilakukan proses pengaturan rute baru dan menghitung biaya dari rute tersebut. Jika rute tersebut tidak memenuhi kriteria, maka akan dilakukan penukaran urutan rute. Kemudian diulangi kembali proses pengaturan

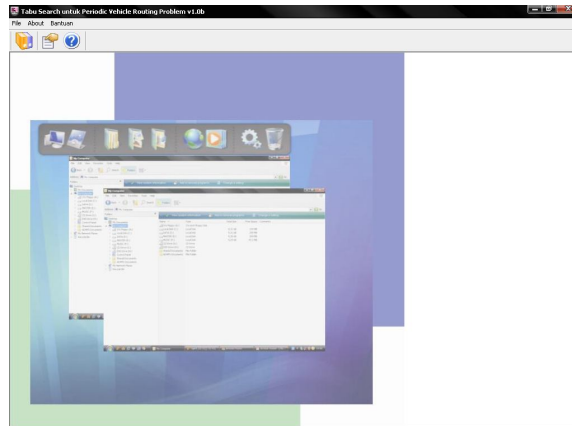
rute baru yang telah ditukar dan menghitung kembali biaya dari rute tersebut. Proses penukaran urutan rute dan proses pengaturan rute baru dilakukan terus menerus sampai ditemukan rute terbaik dan biaya terbaik. Jika telah ditemukan rute terbaik maka rute tersebut akan digambarkan secara visual dan ditampilkan pula urutan rute dan biaya rute tersebut.




Gambar 8 Flowchart Pencarian Jalur

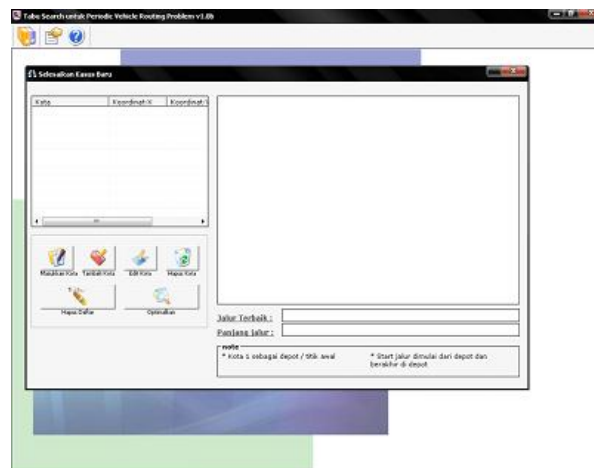
5. Hasil dan Pembahasan

Antarmuka yang pertama kali muncul pada saat kita menjalankan aplikasi Perangkat Lunak Simulasi *Periodic Vehicle Routing Problem* (PVRP) dengan *Tabu Search* adalah Form Utama. Tampilan Form Utama adalah seperti ditunjukkan oleh Gambar 9.



Gambar 9 Form Utama

Form Kasus Baru adalah *form* yang muncul setelah dipilih menu File Kasus Baru atau dengan cara menekan tombol . *Form* ini merupakan *form* yang digunakan *user* untuk memasukkan data serta melakukan proses pencarian jalur atau rute. Tampilan antarmuka *Form Kasus Baru* ditunjukkan pada Gambar 10.



Gambar 10 Form Kasus Baru

Form ini terdapat tabel *list* yaitu tabel yang digunakan untuk menampilkan jumlah kota, posisi kota pada koordinat X dan Y serta permintaan kunjungan masing-masing pelanggan. Tombol *Masukkan Kota* berfungsi untuk memasukkan jumlah kota yang akan diproses, sedangkan tombol *Tambah Kota* berfungsi untuk

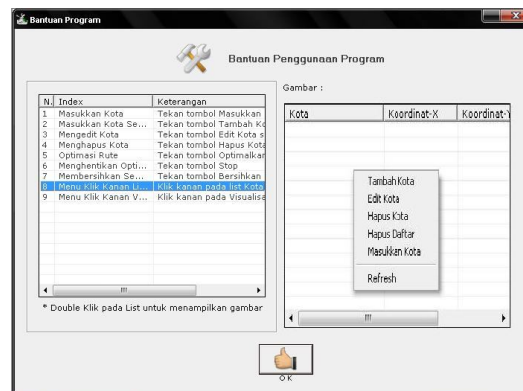
menambahkan kota setelah memasukkan kota atau memasukkan kota secara manual beserta nama, posisi dalam koordinat serta permintaan kunjungan masing-masing pelanggan. *Picture Box* berfungsi untuk menampilkan letak kota dan nama kota serta jalur yang akan dilalui. *Text Box* Jalur Terbaik akan menampilkan urutan jalur yang telah ditemukan. Sedangkan Panjang Jalur menunjukkan jarak yang harus ditempuh sesuai dengan jalur yang dilalui.

Form pembuat dapat dibuka dengan memilih menu About Program. *Form* Pembuat akan menampilkan identitas pembuat aplikasi Perangkat Lunak Simulasi *Periodic Vehicle Routing Problem* (PVRP) dengan *Tabu Search*. Gambar 11 menunjukkan tampilan *Form* About Program.



Gambar 11 *Form* About Program

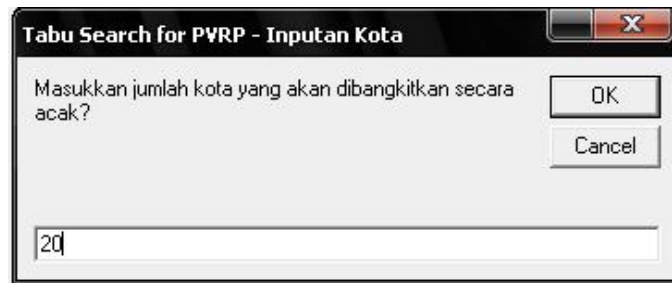
Form Help yang ditunjukkan pada Gambar 12 merupakan *form* tambahan yang berfungsi memberikan informasi kepada *user* tentang cara kerja, proses kerja, serta hasil yang akan ditampilkan. Tampilan antarmukanya seperti terlihat pada Gambar 12.



Gambar 12 *Form* Help

Langkah-langkah pencarian rute dengan cara memasukkan jumlah kota adalah sebagai berikut:

1. Masukkan jumlah kota yang akan dicari jalurnya dengan cara klik File→Kasus Baru. Kemudian klik Masukkan Kota. Masukkan 20 kota.



Gambar 11 Input Kota yang Akan Dioptimasi

2. Apabila akan dilakukan perubahan, maka dapat dilakukan perubahan dengan menekan tombol Tambah Kota untuk menambah kota, Edit Kota untuk mengubah nama, posisi serta kunjungan untuk masing-masing pelanggan serta Hapus Kota untuk menghapus kota. Edit Kota 1, Hapus Kota 20 dan Tambahkan Kota 21.

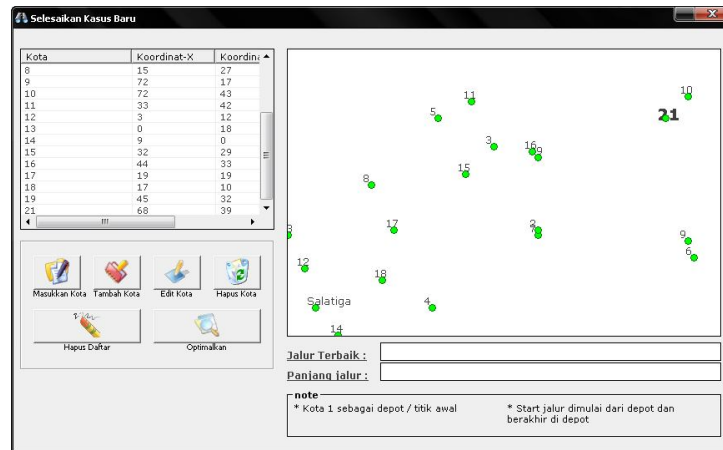


Gambar 12 Edit Kota 1

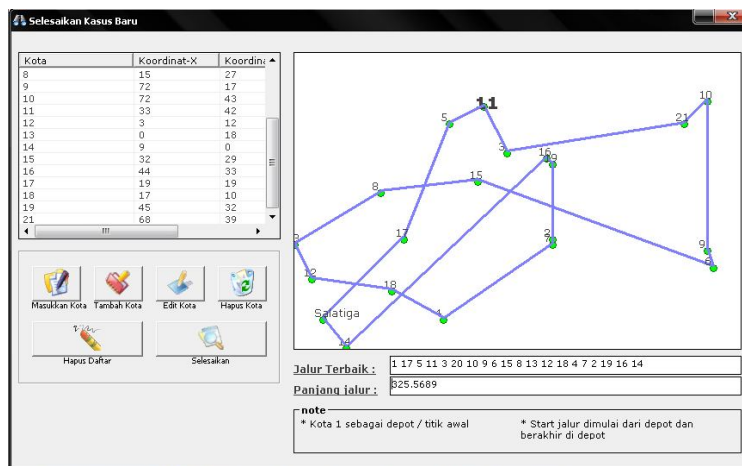


Gambar 13 Tambah Kota 21

3. Visualisasi 20 kota dan posisi kota yang telah ditentukan secara acak akan muncul pada aplikasi, seperti ditunjukkan pada Gambar 14.
4. Klik Optimalkan untuk melakukan proses pencarian jalur dan urutan rute yang akan dilalui. Proses selesai dan akan ditemukan jarak tempuh serta urutan jalur yang ditempuh. Hasil dari proses optimasi adalah urutan kota yang akan dilalui. Urutan tersebut adalah dengan panjang jalur.



Gambar 14 Visualisasi Posisi Kota



Gambar 15 Optimalisasi Kota

- Untuk menghapus proses sebelumnya dan memulai proses baru maka dapat dipilih tombol Hapus Daftar.

Dari pengujian yang telah dilakukan dapat dijelaskan bahwa output yang dihasilkan sistem adalah berupa rute urutan sejumlah kota. Dari beberapa rute yang dimungkinkan, akan dihasilkan output paling optimal berupa urutan rute yang paling pendek berdasarkan pencarian dengan *tabu search*.

Rute paling pendek ditemukan dengan cara membandingkan semua rute yang mungkin dilalui sampai ditemukan rute yang paling optimal dari semua kemungkinan rute yang ada. Jika sudah ditemukan rute yang paling optimal, maka rute tersebut akan disimpan dalam *tabu list*. Kemudian sistem akan membandingkan lagi dengan cara menukar urutan kota sampai ditemukan lagi urutan rute yang paling optimal demikian seterusnya sampai batas perulangan yang ditentukan. Batas perulangan

yang diijinkan adalah sebanyak kota yang diinputkan. Himpunan semua rute yang mungkin dilalui akan disimpan dalam *long term memory*, sedangkan rute paling optimal yang ditemukan akan disimpan dalam *short term memory* atau *tabu list*. Untuk menghindari optimum lokal, maka ditetapkan panjang *tabu list* adalah sembilan *list*.

Panjang jalur ditentukan berdasarkan urutan rute yang ditemukan. Karena sistem yang dibangun adalah sebuah simulasi maka jarak yang ditemukan tidak menggunakan parameter tertentu. Variabel jarak juga dianggap telah mewakili variabel waktu dan biaya. Diasumsikan dengan jarak yang terpendek maka akan menghemat waktu dan biaya yang dikeluarkan.

6. Simpulan

Berdasarkan analisa kebutuhan dan hasil perancangan Perangkat Lunak Simulasi *Periodic Vehicle Routing Problem* (PVRP) dengan *Tabu Search* maka dapat ditarik simpulan bahwa Perangkat Lunak Simulasi *Periodic Vehicle Routing Problem* (PVRP) dengan *Tabu Search* menghasilkan urutan rute pengiriman logistik sesuai dengan jumlah pelanggan dan posisi masing-masing pelanggan. *Tabu search* digunakan untuk mengoptimasi pencarian jalur terbaik dengan cara mencari kemungkinan-kemungkinan rute yang ada sampai ditemukan rute yang paling optimal dari semua kemungkinan yang ada. Urutan jalur yang ditemukan adalah urutan jalur yang paling optimal berdasarkan jumlah pelanggan dan posisi masing-masing kota

7. Daftar Pustaka

- [1] Bowersox, Donald J. 2000. *Manajemen Logistik*. Jakarta: Bumi Aksara.
- [2] Parthanadee, Parthana, *et al.* 2006. Periodic Product Distribution from Multi-depot under limited supplies. http://goliath.ecnext.com/coms2/summary_0199-5990831_ITM. Diakses tanggal 10 Desember 2007.
- [3] Jogiyanto H. M. 2003. *Sistem Teknologi Informasi*. Yogyakarta: Penerbit ANDI.
- [4] The VRP Web. <http://neo.lcc.uma.es/radi-aeb/WebVRP/>. Diakses tanggal 11 Desember 2007.
- [5] Glover, Fred and Manuel Laguna. 1997. *Tabu Search*. http://www.dei.unipd.it/~fisch/ricop/tabu_search_glover_laguna.pdf. Diakses tanggal 1 Desember 2007.
- [6] Hertz, Alain, *et al.*, A Tutorial On Tabu Search. <http://www.google.co.id/search?hl=id&q=%22A+Tutorial+On+Tabu+Search%22&btnG=Telusuri+dengan+Google&meta>. Diakses tanggal 1 Desember 2007.
- [7] Rice Thesis Proposal. <http://www.cs.uga.edu/~potter/dendritte/RiceThesisProposal.pdf>. Diakses tanggal 18 Desember 2007.