

PENERAPAN ALGORITMA SIMULATED ANNEALING PADA PENJADWALAN DISTRIBUSI PRODUK

Eri Wirdianto¹, Jonrinaldi², Betris Surya³

¹) Laboratorium POSI Jurusan Teknik Industri Fakultas Teknik Universitas Andalas

²) Laboratorium Sistem Produksi Jurusan Teknik Industri Fakultas Teknik Universitas Andalas

³) Studio TLFP Jurusan Teknik Industri Fakultas Teknik Universitas Andalas

Abstract

A good logistics system can be achieved by performing various logistic activities efficiently. One of the most affected logistic components is transportation problem. The reason is that the expense of transportation can range from one-third to two-third of total logistic costs. Therefore, various ways in decreasing the expenses of transportation are needed; by optimizing the resources. The optimization of these resources can be done through efficiencies the amount of vehicles allocation and determination the route so that the minimum of total vehicles allocated and travelling distance are obtained.

The distribution route at PT. PDR was unscheduled and only focused on location in the same area, so that vehicles capacities were not exploited maximally. Determination of the route represents problem of VRP (Vehicle Routing Problems). In this research, the method used to determine early solution is 2-Opt method. But solution yielded by this method is not optimal because 2-Opt method is one of heuristic method. Therefore, 2-Opt method solution is then improved by using metaheuristic method which is Simulated Annealing (SA) to get the nearly optimal route solution. SA is conducted by large amount of iterations, so that in this research java programming language is used to facilitate the achievement of the result.

This research obtains total reduction of travel distance for 65,82 km and reduction of vehicles allocation by 1 unit for six-wheel vehicle and 1 unit for four-wheel vehicle with average of the vehicles capacities used is 87,75% and mean of distribution time is 5,8 hours.

Keywords : *Transportation, Vehicle Routing Problems, Metaheuristic, Simulated Annealing*

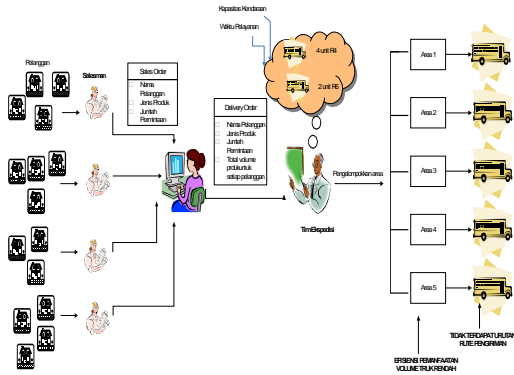
1. Pendahuluan

PT. PDR merupakan perusahaan distributor produk "W" untuk daerah pemasaran Sumatera Barat dengan depot yang terletak di kota Bukittinggi dan Padang. Daerah pemasaran untuk depot Bukittinggi meliputi daerah Bukittinggi, Payakumbuh, Padangpanjang, Batusangkar, dan sebagian daerah Pasaman. Sedangkan untuk daerah lainnya di Sumatera Barat ditangani oleh depot Padang. Perusahaan ini memasarkan produk *multi item* dengan sistem pesan terlebih dahulu (*taking order*) dan kanvas. Sistem kanvas digunakan untuk mendistribusikan barang ke *retailer* yang tersebar. Untuk daerah Padang, perusahaan mengalokasikan sebuah truk roda 4 untuk kanvas. Sedangkan untuk sistem *taking order*, *sales* akan mencatat semua pesanan pelanggan dalam *sales order* dan kemudian dilakukan input data pesanan ke komputer. Keluaran dari komputer berupa DO yang berisikan nama pelanggan, list pesanan, dan volume keseluruhan kemudian di kirim ke bagian ekspedisi untuk dikelompokkan berdasarkan area pemasarannya.

Dalam melaksanakan pekerjaannya, tim ekspedisi ini bekerja hanya berdasarkan perkiraan saja dan lebih terfokus kepada

pengalokasian kendaraan untuk pelanggan pada area yang sama sehingga sering terjadi kapasitas kendaraan berlebih karena tidak adanya standar dalam melakukan pengelompokkan. Padahal seharusnya kapasitas yang berlebih ini dapat dimanfaatkan untuk mensuplai daerah yang tidak hanya berada pada satu area saja. Selain itu, pengelompokkan area ini tidak menghasilkan solusi rute. Rute yang akan ditempuh hanya ditentukan oleh sopir secara trial dan error. Namun banyaknya jumlah titik distribusi yang tersebar menyulitkan sopir dalam menentukan urutan rute yang akan ditempuh. Salah satu akibatnya adalah jarak yang ditempuh menjadi lebih panjang. Jika hal ini terus berlanjut maka biaya yang dikeluarkan perusahaan untuk mendistribusikan barang menjadi besar. Secara umum, gambaran sistem pendistribusian perusahaan ini dapat dilihat pada Gambar 1.

Untuk itu perlu dilakukan penentuan rute dan penjadwalan pengiriman produk yang dapat mengatasi permasalahan diatas dengan mengoptimalkan seluruh sumber daya yang dimiliki sehingga diperoleh solusi rute yang dapat meminimasi total ongkos transportasi.



Gambar 1. Gambaran Sistem Pendistribusian PT. PDR untuk Wilayah Padang

ada, dan juga dengan meminimumkan biaya transportasi.

Gambar 2 di bawah menunjukkan dimana ada satu depot yang harus melaksanakan distribusi barang ke konsumen. Pada Gambar 2 terlihat ada 12 konsumen yang tersebar dengan jarak dan titik koordinat yang berbeda. Sedangkan pada Gambar 3 dapat melihat bahwa ada garis yang menghubungkan antara konsumen atau depot dengan konsumen yang disebut dengan rute kendaraan. Dari Gambar 3 dapat terlihat bagaimana dalam melakukan distribusi kendaraan membentuk rute tertentu, yaitu 12 konsumen dicapai dengan cara mengirimkan produk berdasarkan rute yang telah ditetapkan.

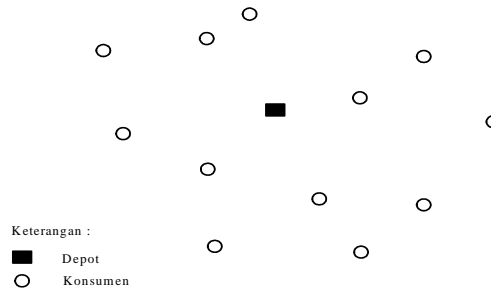
2. Tinjauan Pustaka

Menurut *Council of Logistic Management* (CLM) di dalam Ballou (1998) logistik merupakan suatu proses perancangan, implementasi dan pengendalian efisiensi, aliran biaya dan penyimpanan bahan baku, penyimpanan barang dalam proses, produk jadi dan informasi-informasi lain yang berhubungan dengan titik awal sampai titik akhir dari suatu proses produksi untuk memenuhi kebutuhan konsumen.

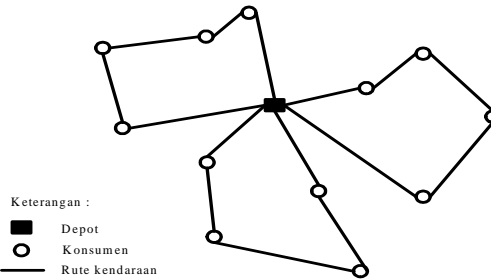
Dalam sistem logistik tingkat performansi tertentu yang ingin dicapai yaitu keseimbangan antara kualitas pelayanan yang diharapkan oleh pelanggan dengan biaya-biaya yang dikeluarkan untuk mencapai tujuan perusahaan. Dua faktor utama yang terkait dengan performansi sistem logistik adalah faktor pelayanan yang menyangkut tingkat pelayanan yang diberikan kepada pelanggan, dan faktor biaya yang menyangkut besarnya biaya yang dikeluarkan perusahaan sehubungan dengan tingkat pelayanan yang diberikan kepada pelanggan. Salah satu faktor yang paling berpengaruh dalam peningkatan pelayanan kepada pelanggan adalah ketepatan waktu pengiriman. Dalam hal ini berkaitan dengan sistem distribusi perusahaan.

Vehicle Routing Problems (VRP)

Vehicle Routing Problem (VRP) adalah suatu pencarian solusi yang meliputi penentuan sejumlah rute, masing-masing rute dilalui oleh satu kendaraan yang berawal dan berakhir di depot asalnya, agar dapat melayani semua konsumennya dengan tetap memenuhi kendala operasional yang



Gambar 2. Contoh Masukan VRP



Gambar 3. Contoh Keluaran VRP

Tujuan utama dari VRP adalah pencapaian solusi yang optimal. Untuk sebuah permasalahan VRP berskala besar dimana terdapat banyak konsumen yang harus dilayani, maka upaya pencarian solusi yang optimal akan memakan waktu yang sangat lama. Hal ini dikarenakan VRP merupakan *NP-hard problems*, dimana waktu yang dibutuhkan untuk mencari solusi permasalahan bergerak secara eksponensial seiring dengan bertambahnya konsumen. Sehingga berkembanglah berbagai macam pendekatan heuristik yang mampu menghasilkan solusi yang mendekati optimal dan dengan waktu yang relatif lebih singkat. Pendekatan heuristik ini terus berkembang

dengan dibuatnya pendekatan kedalam prosedur yang terkomputerisasi, sehingga menjadi sangat aplikatif dan mudah digunakan. Penggunaan prosedur yang terkomputerisasi ini didalam perencanaan proses pendistribusian telah mampu menghasilkan penghematan yang cukup signifikan, yaitu berkisar antara 5-20% biaya transportasi.

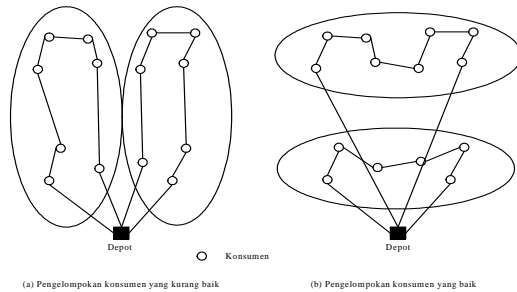
Vehicle Routing and Shedulling

Vehicle routing and scheduling merupakan perluasan dari VRP.

Prinsip-prinsip dalam pembuatan *routing and scheduling* yang baik adalah sebagai berikut [Ballou, 1998, hlm.199] :

1. Mengisi truk sebanyak volume pemberhentian yang akan didatangi, dimana titik-titik pemberhentian tersebut letaknya berdekatan satu sama lain.

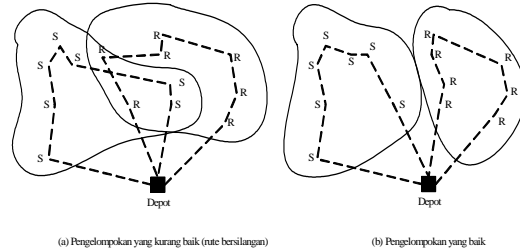
Pengelompokan titik pemberhentian yang baik dan buruk dapat dilihat pada Gambar 4.



Gambar 4. Pengelompokkan Titik Pemberhentian untuk Penugasan Volume Pemberhentian pada Kendaraan

2. Titik-titik pemberhentian yang memiliki hari pengiriman yang berbeda-beda, sebaiknya diatur untuk memperoleh pengelompokan yang lebih dekat.

Titik-titik pemberhentian yang memiliki hari pengiriman yang berbeda-beda, harus dipisahkan untuk *routing* dan *scheduling* tiap harinya. Pembuatan rute dan penjadwalan harian juga harus menghindari pengelompokkan yang tumpang tindih (*overlapping clusters*). Hal ini akan membantu meminimalkan jumlah truk yang dibutuhkan untuk melayani seluruh titik pemberhentian. Ilustrasinya dapat dilihat pada Gambar 5.



Gambar 5. Pengelompokan Titik Pemberhentian Berdasarkan Hari Pengiriman

3. Dalam pembuatan rute dimulai dari titik pemberhentian terjauh dari depot agar mendapatkan rute yang efisien.

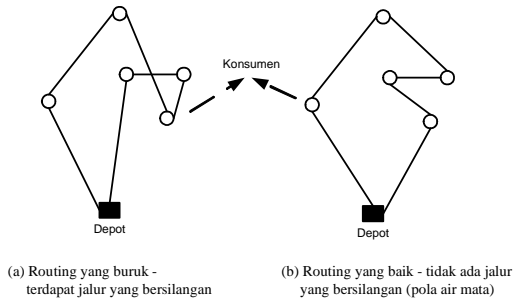
Rute yang efisien dapat dikembangkan dengan dimulai dari titik pemberhentian yang paling jauh dari depot sampai ke titik yang paling dekat. Saat titik pemberhentian terjauh dari depot teridentifikasi, kapasitas yang tersisa dari kendaraan yang telah ditugaskan sebaiknya diisi dengan memilih sekelompok titik pemberhentian yang berdekatan dengan titik pemberhentian tersebut. Setelah kendaraan ditugaskan untuk volume titik-titik pemberhentian tersebut, mulailah membuat rute dengan kendaraan lain dan identifikasi titik pemberhentian terjauh dari sisa titik-titik pemberhentian yang belum ditugaskan pada kendaraan. Terus lakukan prosedur ini sampai seluruh titik pemberhentian telah ditugaskan pada kendaraan.

4. Urutan pemberhentian pada sebuah rute sebaiknya membentuk pola air mata (*teardrop pattern*).

Hal ini ditujukan agar tidak ada jalur yang bersilangan. Batasan *time windows* dan pemberlakuan aturan pengambilan barang setelah seluruh pengiriman barang dilakukan, dapat mengakibatkan jalur rute yang bersilangan. Pembuatan urutan pemberhentian yang baik dan buruk dapat dilihat pada Gambar 6.

5. Rute yang paling efisien dibangun dengan menggunakan kendaraan dengan kapasitas terbesar.

Idealnya, penggunaan truk berkapasitas besar untuk melayani banyak titik pemberhentian dalam satu rute akan meminimalkan jarak tempuh kendaraan. Sehingga, truk dengan kapasitas terbesar harus dialokasikan terlebih dahulu.



Gambar 6. Contoh Penentuan Urutan Pemberhentian

6. Pengambilan barang (*pickup*) sebaiknya digabungkan dengan rute pengiriman barang (*delivery*), ketimbang pengambilan barang baru dilakukan setelah semua pengiriman dilakukan.

Hal ini guna meminimalkan jalur yang bersilangan, yang dapat terjadi bila pengambilan dilakukan setelah seluruh pengiriman dilakukan.

7. Titik pemberhentian yang terpisah dari pengelompokan rute adalah kandidat terbaik untuk penggunaan alat transportasi lain.

Titik pemberhentian yang terpisah dari pengelompokan, terutama titik pemberhentian dengan volume yang kecil, dilayani dengan waktu dan biaya yang relatif besar. Untuk itu digunakan kendaraan berkapasitas kecil untuk melayani titik pemberhentian tersebut sehingga lebih ekonomis.

8. Batasan *time windows* titik pemberhentian yang berdekatan harus dihindari.

Batasan *time windows* yang sangat dekat diantara pemberhentian, dapat memaksa pembentukan urutan pemberhentian jauh dari pola ideal. Oleh karena *time windows* tidak bersifat absolut, maka sebaiknya dilakukan negosiasi terhadap titik pemberhentian yang dipaksa untuk dilayani sesuai dengan pola *routing* yang diinginkan.

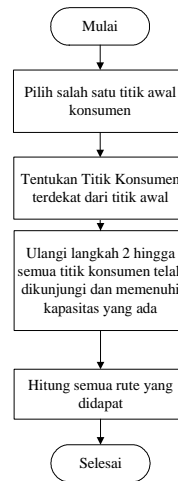
Metode Penyelesaian VRP

Metode-metode dalam penyelesaian permasalahan VRP dapat digabungkan antara suatu metode dengan metode yang lain. Dalam penelitian ini, metode yang digunakan adalah metode heuristik yaitu *nearest neighbour*, *k-opt*, dan *k-insert* dikombinasikan dengan metode

metaheuristik yaitu algoritma *simulated annealing*.

1. Metode *Nearest Neighbour*

Nearest neighbour adalah metode yang pertama digunakan untuk mendapatkan solusi *vehicle routing problem*. Metode ini sangat mudah dan cepat untuk diimplementasikan. Caranya adalah pertama kita pilih satu titik konsumen sebagai titik awal lalu bergerak ke kota berikutnya yang terdekat. Diagram alir penyelesaian permasalahan dengan menggunakan metode ini dapat dilihat pada Gambar 7.



Gambar 7. Diagram Alir Metode *Nearest Neighbour*

2. Metode *N-Insert*

N-insert heuristik adalah memindahkan konsumen dari rute yang satu ke rute yang lain. Pemilihan konsumen berdasarkan informasi jarak, waktu tempuh dan *time window* konsumen. *N-insert* dapat diterima apabila dapat memberikan solusi yang lebih baik dari solusi awal seperti pada Gambar 8.

3. Metode *N-Opt*

N-Opt yang digunakan dalam pengolahan data adalah algoritma *2-Opt*. Algoritma *2-Opt* adalah pertukaran yang menghilangkan 2 titik dan menggantinya dengan hanya pasangan yang mungkin dari dua titik yang baru yang tidak memecah rute ke dalam dua siklus yang tidak bersama-sama. Algoritma *2-Opt* dimulai dengan melakukan pertukaran dan berakhir ketika tidak ada lagi petukaran yang menghasilkan perbaikan. Ilustrasi mengenai *2-Opt* ini dapat dilihat pada Gambar 9 dan Gambar 10.

4. Metode *Saving*

Tujuan metode *saving* adalah meminimasi total jarak perjalanan semua kendaraan dan untuk meminimalkan secara tidak langsung jumlah kendaraan yang dibutuhkan untuk melayani semua titik pemberhentian. Asumsi awal yang digunakan pada metode ini yaitu setiap kendaraan yang berbeda berangkat dari depot untuk mengantar produk ke lokasi tertentu dan kembali lagi ke depot.

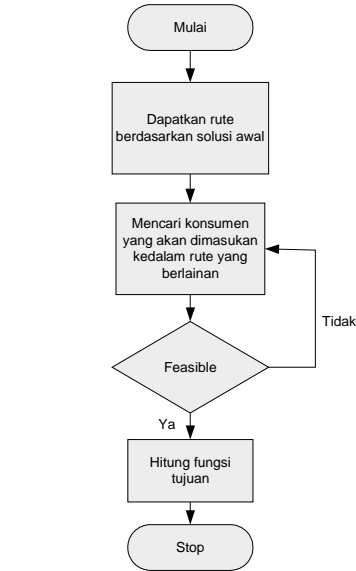
5. Metode *Sweep*

Metode *sweep* merupakan metode sederhana untuk menentukan rute kendaraan bahkan untuk masalah dalam skala besar. Pada beragam permasalahan, keakuratan metode ini menghasilkan rata-rata error kira-kira 10%. Kerugian metode ini yaitu dalam hal cara menghasilkan rute. Proses pembentukan rute pada metode ini dibagi menjadi dua tahap dengan pemberhentian ditetapkan terlebih dahulu pada kendaraan. Kemudian ditentukan urutan rute yang akan ditempuh kendaraan. Karena proses dua tahap ini, maka masalah waktu seperti total waktu serta *time windows* tidak dapat diatasi dengan baik.

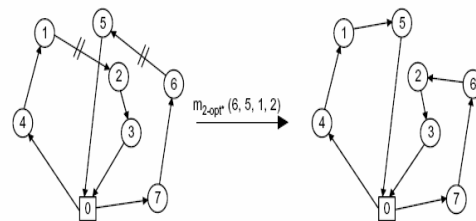
6. Metode *Tabu Search*

Tabu search adalah suatu strategi pencarian yang bekerja pada satu solusi pada waktu tertentu. *Tabu search* bekerja dengan memelihara daftar dari langkah (*move*) yang baru dilakukannya dan mencoba untuk tidak mengulangi langkah tersebut, terutama jika langkah tersebut tidak ada gunanya. Pada masing-masing langkah pencarian, dilakukan evaluasi semua langkah dari suatu daftar calon, dan biasanya menerima satu-satunya yang terbaik dari yang tersedia. Jika tidak ada langkah *improving non tabu*, algoritma dapat mengambil langkah *tabu*. Jika tidak ada lagi langkah yang bisa meningkatkan solusi, algoritma bisa mengambil langkah *unimproving*. Pada setiap langkah, algoritma mengevaluasi daftar calon *move* dan menggolongkan mereka ke dalam pilihan berikut:

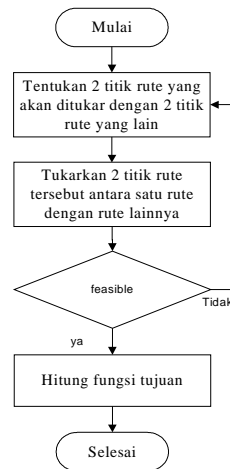
- best improving, non tabu*
- best improving, tabu*
- least-worst improving, non-tabu*
- least-worst improving, tabu*



Gambar 8. Diagram Alir Metode *N-Insert*



Gambar 9. Iustrasi Pergerakan 2-Opt



Gambar 10. Diagram Alir Penyelesaian dengan Menggunakan 2-Opt

7. Algoritma Genetika

Algoritma genetika (GA) adalah algoritma pencarian metaheuristik yang didasarkan atas mekanisme evolusi biologis. Keberagaman pada evolusi biologis adalah

variasi dari kromosom antar individu organisme. Variasi kromosom ini akan mempengaruhi laju reproduksi dan tingkat kemampuan organisme untuk tetap hidup

Pada algoritma ini, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin yang dikenal dengan istilah populasi. Individu yang terdapat dalam satu populasi disebut dengan istilah kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Populasi awal dibangun secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan istilah generasi. Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut dengan fungsi *fitness*. Nilai *fitness* dari suatu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut. Generasi berikutnya dikenal dengan istilah anak (*offspring*) terbentuk dari gabungan 2 kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyilangan (*crossover*). Selain operator penyilangan, suatu kromosom dapat juga dimodifikasi dengan menggunakan operator mutasi. Populasi generasi yang baru dibentuk dengan cara menyeleksi nilai *fitness* dari kromosom induk (*parent*) dan nilai *fitness* dari kromosom anak (*offspring*), serta menolak kromosom-kromosom yang lainnya sehingga ukuran populasi (jumlah kromosom dalam suatu populasi) konstan. Setelah melalui beberapa generasi, maka algoritma ini akan konvergen ke kromosom terbaik.

8. *Simulated Annealing* (SA)

Algoritma SA merupakan algoritma metaheuristik dengan konsep awal pada proses fisika. Suatu benda padat dipanaskan hingga mencair pada tingkat temperatur tertentu. Pada temperatur ini, setiap atom dapat bergerak dengan bebas. Dengan melakukan perpindahan, atom-atom ini akan memiliki banyak alternatif kombinasi struktur yang akan terbentuk apabila temperatur diturunkan. Penurunan temperatur ini harus dilakukan secara perlahan yang disebut dengan proses *annealing*. Hal ini bertujuan agar pada setiap tingkatan temperatur terjadi perubahan sistem hingga tercapai keseimbangan termal. Dengan proses *annealing* ini maka susunan atom yang terbentuk akan memiliki nilai energi yang rendah. Jika tidak demikian, keadaan akhir dari benda padat

tersebut akan memiliki banyak cacat karena terbentuknya struktur yang optimal secara lokal. Penurunan temperatur ini terus dilakukan hingga mencapai nilai temperatur akhir 0,01°C. Pada temperatur akhir ini semua atom sudah membentuk suatu struktur kristal dan tidak terdapat lagi perpindahan atom.

Pada penyelesaian permasalahan sistem distribusi, rute-rute yang akan disusun dianalogikan seperti atom-atom yang bergerak bebas. Dengan menganalogikan nilai temperatur sebagai tingkatan dari iterasi yang akan dilakukan, maka rute-rute ini akan memiliki alternatif kombinasi rute apabila dilakukan penurunan temperatur. Penurunan ini dilakukan secara perlahan sehingga alternatif yang diperoleh akan semakin banyak karena iterasi yang dilakukan juga banyak.

Pada setiap tingkatan temperatur, dilakukan pertukaran secara acak antara 2 titik rute. Hasil rute setelah pertukaran ini disebut dengan *neighbourhood* atau solusi tetangga. *Neighbourhood* ini akan memiliki nilai solusi yang dianalogikan sebagai nilai energi yang berbeda-beda. Kemudian dipilih satu rute yang memiliki nilai solusi terendah dan dievaluasi. Apabila rute Untuk menghasilkan solusi yang dapat menyelesaikan permasalahan distribusi dengan lebih akurat dan sesuai dengan kondisi riil, sebaiknya dirancang sebuah program yang dapat mengatasi terjadinya penambahan jumlah pelanggan, keterbatasan stock gudang, penambahan jumlah produk, dan hal-hal lainnya. Hasil pertukaran ini lebih baik dari rute awal, maka langkah (*move*) yang diambil bersifat memperbaiki dan rute inilah yang akan digunakan sebagai rute awal pada penurunan tingkatan iterasi selanjutnya.

Konsep mekanika statistik diringkaskan dan dibandingkan dengan optimasi dapat dilihat pada Tabel 1.

Tabel 1. Analogi SA dalam Sistem Termodinamika dan Pencarian Solusi di Berbagai Bidang

Sistem termodinamik	Optimasi
Kedadaan Sistem	Solusi
Energi Sistem	Biaya Solusi
Perubahan Sistem	Solusi Neighbour
Min. Energi	Min. Biaya

Langkah algoritma SA secara umum adalah [Heragu, 1997] :

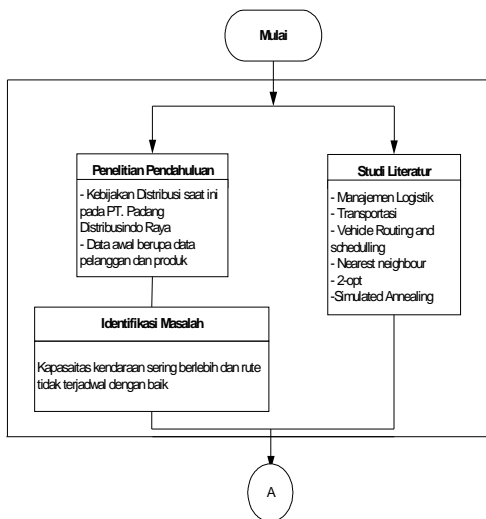
- Langkah 1 Munculkan sebuah solusi awal S
- Langkah 2 Pilih sebuah nilai temperatur awal, $T_0 > 0$
- Langkah 3 Ulangi sebanyak N kali
Munculkan sebuah S' dari S
Bandingkan S' dengan S
Cek suhu ≤ 0.01
- Langkah 4 Jika suhu ≤ 0.01 penghentian telah dicapai, stop. Bila tidak turunkan suhu dan set $F(T) = T + 1$, ulangi langkah 3

Efektifitas *Simulated Annealing* bergantung pada desain *neighbourhood* dan juga bagaimana pencarian dilakukan di dalam *neighbourhood*. Jika *neighbourhood* didesain sedemikian rupa sehingga memfasilitasi *move* ke solusi lebih baik dan keluar dari minimal lokal, maka prosedur SA akan memiliki kinerja yang baik.

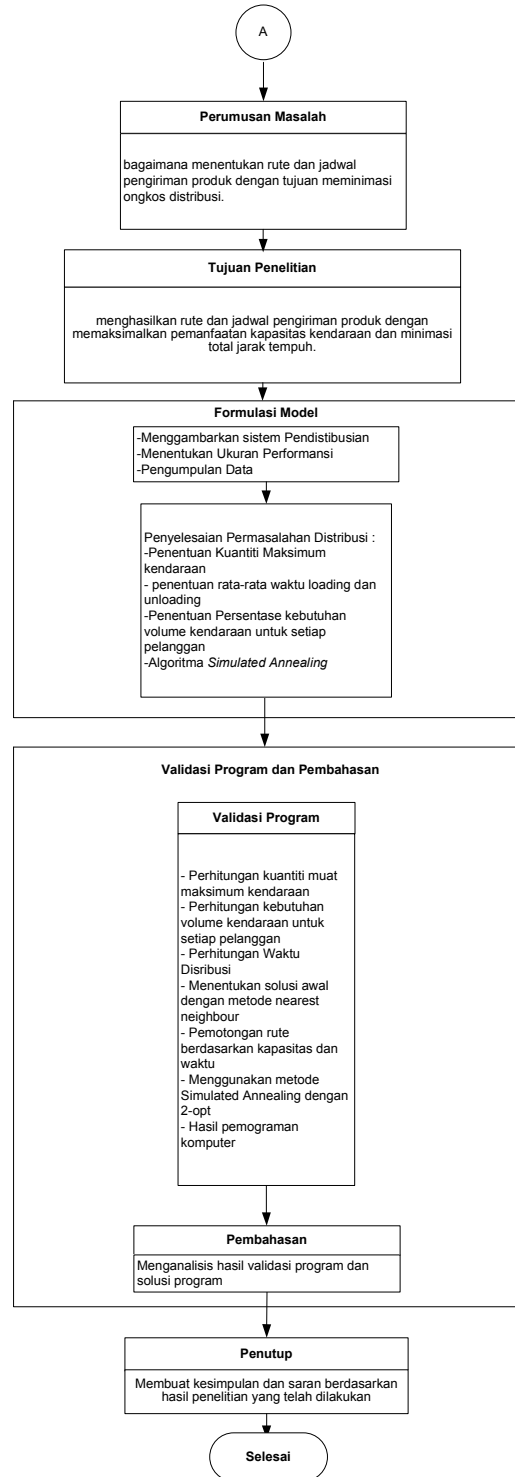
Algoritma SA ini merupakan algoritma yang cukup sederhana sehingga mudah diterapkan untuk berbagai permasalahan khususnya permasalahan distribusi. Selain itu, algoritma ini tidak memerlukan pengacakan rute secara keseluruhan untuk memperoleh alternatif kombinasi rute sehingga nilai solusi akhir dengan mudah didapatkan.

3. Metodologi Penelitian

Tahapan penyelesaian masalah dalam penelitian ini dapat dilihat pada Gambar 11.



Gambar 11. Skema Metodologi Penelitian



Gambar 11. Skema Metodologi Penelitian (lanjutan)

4. Formulasi Model

A. Ukuran Performansi

Performansi rute pengiriman produk ditunjukkan dengan menggunakan fungsi objektif yang dirancang secara tepat. Dalam penelitian ini, fungsi objektif yang digunakan sebagai ukuran performansi rute pengiriman adalah total ongkos transportasi yang didefinisikan oleh total jarak tempuh kendaraan dengan pembatas jumlah kendaraan, persentase volume kendaraan, dan total waktu distribusi. Formulasi matematis dari fungsi objektif ini adalah [Carter, 2003, him 2] :

$$\min \sum_{i=0}^P \sum_{j=0}^P x_{ij}$$

P = jumlah pelanggan

x_{ij} = jarak dari daerah i ke daerah j

B. Pengumpulan Data

Data yang dikumpulkan dalam penelitian ini adalah :

1. Jenis produk

Data jenis produk diperlukan untuk mengetahui jenis produk yang didistribusikan oleh PT.PDR. Berdasarkan data yang diperoleh, terdapat 685 jenis produk yang didistribusikan.

2. Data volume produk

Data volume produk merupakan data volume dari masing-masing produk yang diukur langsung oleh perusahaan dengan memberikan kelonggaran untuk setiap produknya. Data ini diperlukan untuk mengetahui kapasitas ruang yang terpakai dalam kendaraan.

3. Data permintaan pelanggan.

Data permintaan pelanggan diperoleh dari data pemesanan yang terdapat pada masing-masing sales. Data ini akan digunakan untuk mengetahui permintaan harian pelanggan terhadap produk.

4. Data lokasi dan data jarak.

Data lokasi pelanggan diperoleh dari supervisor. Data lokasi pelanggan yang digunakan hanya pelanggan yang berada dalam area Padang saja. Data ini akan digunakan untuk menentukan jarak lokasi pelanggan dari depot dan jarak antar pelanggan. Berdasarkan data yang diperoleh, perusahaan ini memiliki 250

pelanggan tetap. Data jarak pelanggan dari depot dan data jarak antar pelanggan diperoleh dengan melakukan pengukuran menggunakan peta kota Padang.

5. Data jumlah dan jenis kendaraan

Jumlah kendaraan yang dimiliki oleh PT.PDR untuk area Padang adalah sebanyak 4 unit kendaraan roda empat dan 2 unit kendaraan roda enam.

C. Penyelesaian Permasalahan Distribusi

Penyelesaian Permasalahan Distribusi dengan menggunakan algoritma SA dimulai dengan menentukan kuantiti muat maksimum kendaraan, perhitungan persentase kebutuhan volume kendaraan, perhitungan waktu *loading* dan *unloading*, dan tahap algoritma SA.

1. Penentuan Kuantiti Muat Maksimum Kendaraan

Penentuan kuantiti maksimum kendaraan dilakukan untuk mengetahui jumlah maksimum suatu jenis produk dapat dimuat ke dalam kendaraan. Dalam perhitungannya, yang menjadi ukuran adalah volume produk dan volume box kendaraan. Volume produk yang digunakan adalah dalam bentuk karton kemasan karena permintaan pelanggan lebih sering dalam ukuran karton, bukan dalam *pieces* produk.

Asumsi yang digunakan pada tahap ini adalah berat produk dapat diabaikan.

Sedangkan notasi yang digunakan adalah sebagai berikut:

Notasi	Keterangan	Satuan
V_p	Volume produk	$cm^3/karton$
V_k	Volume kendaraan	cm^3
K	Kuantiti muat maksimum kendaraan	karton

Kuantiti muat maksimum kendaraan didapatkan dengan membandingkan besarnya volume kendaraan dengan volume produk per karton. Secara matematis dapat dinyatakan dalam persamaan berikut ini.

$$[Kuantiti Muat Maksimum kendaraan/ karton] = \frac{V_k}{V_p}$$

2. Penentuan Rata-rata Waktu *Loading* dan *Unloading* Kendaraan

Data waktu *loading* dan *unloading* kendaraan digunakan untuk menghitung waktu distribusi. Data ini merupakan suatu konstanta yang diperoleh dari rata-rata waktu yang dibutuhkan dalam melakukan *loading* dan *unloading*.

Asumsi yang digunakan pada tahap ini adalah waktu *loading* merupakan suatu nilai yang konstan dan tidak dipengaruhi oleh kondisi pelanggan.

Sedangkan notasi yang digunakan adalah sebagai berikut:

Notasi :

L : Waktu *Loading* (jam)

U : Waktu *Unloading* (jam)

W_L : Waktu *Loading* Kendaraan (jam/% permintaan)

W_U : Waktu *UnLoading* Kendaraan (jam/% permintaan)

Waktu *loading* dan *unloading* didapatkan dengan membandingkan waktu *loading* dengan kapasitas kendaraan. Secara matematis dapat dinyatakan dalam persamaan berikut ini.

$$[\text{Waktu Loading Kendaraan}] = \frac{\text{Waktu Loading (jam)}}{\text{Kapasitas Kendaraan (\%)}}$$

$$[\text{Waktu UnLoading Kendaraan}] = \frac{\text{Waktu UnLoading (jam)}}{\text{Kapasitas Kendaraan (\%)}}$$

3. Penentuan Persentase Kebutuhan Volume Kendaraan untuk Setiap Pelanggan

Dari perhitungan ini dapat diketahui besarnya volume kendaraan yang digunakan berdasarkan besarnya jumlah permintaan pada hari tersebut. Dalam perhitungannya, depot sebagai titik awal akan diberi kode P000 dan setiap titik distribusi diberi urutan kode P001, P002, P003 dan seterusnya, sedangkan kapasitas kendaraan diberi kode R4 untuk kendaraan roda 4 dan R6 untuk kendaraan roda 6.

4. Algoritma *Simulated Annealing*

Parameter yang digunakan dalam algoritma SA adalah :

a. Temperatur Awal, *T₀*

Temperatur awal dianalogikan sebagai suatu kondisi awal untuk memulai iterasi. Semakin tinggi nilai temperatur awal yang diberikan maka alternatif solusi juga akan semakin banyak. Nilai yang terlalu tinggi akan menyebabkan alternatif solusi terlalu banyak tanpa memberikan perubahan yang cukup berarti. Sebaliknya temperatur awal yang terlalu rendah menyebabkan sistem tidak dapat keluar dari optimal lokal karena alternatif solusi terlalu sedikit. Dalam penelitian ini digunakan nilai temperatur 100°C dan 200°C. Solusi rute yang diperoleh dari kedua nilai temperatur ini kemudian dibandingkan dan ditentukan nilai yang memberikan solusi yang lebih stabil.

b. Laju pendinginan, *α*

Dalam sistem distribusi, fungsi laju pendinginan dianalogikan seberapa cepat pencapaian solusi akhir dilakukan. Laju pendinginan yang terlalu lambat menyebabkan lamanya solusi akhir diperoleh sedangkan laju pendinginan yang terlalu cepat mengakibatkan solusi yang diperoleh adalah solusi optimal lokal karena sedikitnya tingkatan iterasi yang terjadi. Dalam penelitian ini nilai *α* yang digunakan adalah 0,9 dan 0,95. Nilai ini digunakan karena umumnya literatur yang ada merekomendasikan kedua nilai ini sebagai nilai *α*.

c. Jumlah iterasi pada setiap tingkatan temperatur, *N*

Iterasi yang dilakukan pada setiap tingkatan temperatur adalah dengan cara melakukan pertukaran 2 titik secara acak sebanyak 5 kali. Dan setelah itu ditampilkan solusi yang terbaik dari kelima iterasi ini. Nilai solusi terbaik ini akan digunakan sebagai solusi awal pada penurunan temperatur selanjutnya.

d. Temperatur akhir, *T_t*

Nilai temperatur akhir merupakan suatu titik dimana iterasi dihentikan. Pada tahap ini dimunculkan nilai solusi akhir yang merupakan solusi terbaik. Nilai temperatur akhir yang digunakan dalam penelitian adalah 0,01°C. Jadi apabila penurunan suhu telah mencapai nilai 0,01 maka iterasi dihentikan.

Secara umum, aplikasi algoritma SA terdiri dari beberapa tahap, yaitu:

1. Menentukan solusi awal

Solusi awal diperoleh dengan menggunakan metode *nearest neighbour*. Lokasi awal sebagai lokasi dimulainya perjalanan adalah gudang, kemudian dicari daerah yang terdekat dengan gudang, setelah ditentukan daerah yang terdekat dengan gudang, kemudian ditentukan daerah-daerah lain yang terdekat. Hasil dari metode ini adalah berupa rute panjang dengan kriteria jarak terpendek. Rute ini selanjutnya dipotong dan dibentuk rute-rute kecil yang sesuai dengan kapasitas angkut kendaraan dan waktu distribusi kendaraan.

Formulasi matematis untuk pemotongan rute dengan mempertimbangkan persentase kebutuhan volume kendaraan adalah [Gen dan Cheng, 1997, hlm 345] :

$$\max \sum_{i=1}^P D_i$$

D = Permintaan pelanggan
P = Jumlah Pelanggan

Sedangkan formulasi matematis untuk pemotongan rute dengan mempertimbangkan waktu distribusi kendaraan adalah:

$$\text{Min } \sum (L + X/V + U)$$

Min \sum (waktu loading + waktu tempuh + waktu unloading)

$$\text{waktu tempuh} = \frac{\text{jarak}}{\text{kecepatan kendaraan}}$$

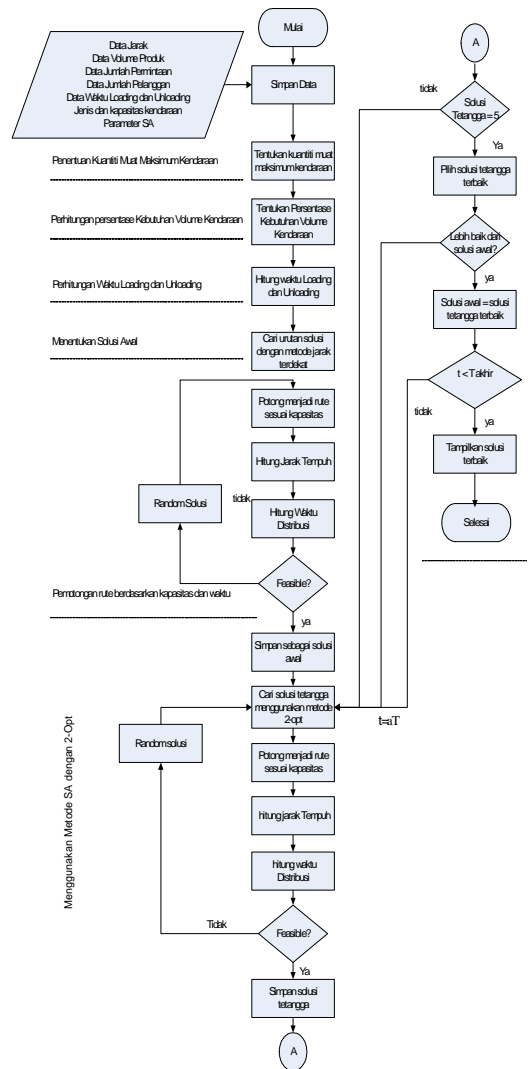
Kemudian lakukan uji fisibilitas yang berhubungan dengan waktu distribusi, persentase kebutuhan volume kendaraan, dan jumlah kendaraan yang digunakan terhadap rute yang dipotong.

2. Menentukan solusi tetangga

Tahap selanjutnya adalah mencari solusi tetangga dari solusi yang terbaik yang dihasilkan pada tahap sebelumnya. Metode yang digunakan dalam tahap ini adalah metode 2-opt. Solusi tetangga pada tahap ini juga akan diuji, apakah rute yang dihasilkan sesuai dengan jarak dan kapasitas. Selanjutnya pada tahap ini juga akan terdapat beberapa solusi, oleh karena itu harus dipilih salah satunya untuk menjadi solusi terbaik. Solusi terbaik pada tahap ini selanjutnya akan dibandingkan dengan solusi yang dihasilkan pada tahap satu.

3. Pengecekan solusi secara keseluruhan

Solusi tahap kedua masih perlu dilakukan pengecekan. Pengecekan pertama adalah apakah solusi pada kedua tahap ini lebih baik dari solusi pada tahap pertama. Jika tidak lebih baik maka pencarian tahap kedua masih terus dilanjutkan. Jika solusi pada kedua tahap ini lebih baik dilakukan pengecekan tahap akhir untuk semua solusi yang dihasilkan yaitu melihat apakah suhu yang diturunkan sudah mencapai batas pencarian solusi (T akhir). Jika pencarian solusi belum mencapai T akhir maka pencarian solusi masih terus dilanjutkan. Proses pembentukan algoritma dapat dirangkum pada Gambar 12.



Gambar 12. Diagram Alir Algoritma Pemecahan Masalah Distribusi

5. Validasi Program

Untuk menghasilkan hasil rute distribusi yang mendekati optimal, maka pada algoritma ini dibutuhkan jumlah iterasi yang sangat besar. Untuk itu dirancang sebuah program dengan menggunakan bahasa pemrograman Java untuk mempermudah proses pencarian hasil.

Program yang dibuat kemudian di validasi untuk mengetahui apakah sesuai dengan yang diharapkan dan dapat diaplikasikan pada sistem distribusi perusahaan. Validasi ini dilakukan dengan cara membandingkan antara perhitungan manual dengan output program. Apabila output perhitungan manual sama dengan output program, maka diasumsikan program yang dirancang telah sesuai dengan yang diharapkan. Akan tetapi apabila output perhitungan manual dan output program berbeda, maka program yang dirancang belum sesuai dengan yang diharapkan, dan perlu dilakukan koreksi kembali terhadap program yang dirancang.

Penentuan Kuantiti Muat Maksimum Kendaraan

Pada penelitian di PT. PDR diasumsikan kendaraan yang digunakan dapat diisi dengan muatan penuh 100% dengan nilai kelonggaran 10%. Nilai 100% ini diberikan karena pekerja sudah ahli dalam menyusun produk dalam box kendaraan sehingga semua ruang dalam box dapat dimanfaatkan. Sedangkan nilai kelonggaran yang diberikan berdasarkan kepada adanya kemungkinan ruang tersisa pada susunan karton dengan jenis yang berbeda.

Berikut ini contoh perhitungan untuk produk B024 pada kendaraan roda 4.

$$\text{Volume produk} = 23177 \text{ cm}^3/\text{karton}$$

$$\text{Volume Kendaraan} =$$

$$\text{Panjang box} \times \text{Lebar box} \times \text{Tinggi box} = 3200 \times 1800 \times 1700 = 9792000 \text{ cm}^3$$

Dengan memberikan nilai kelonggaran 10%, maka:

$$\begin{aligned} V_k &= 0,9 \times 9792000 \\ &= 8812800 \text{ cm}^3 \end{aligned}$$

$$\text{Kuantiti Muat Maksimum Kendaraan} = \frac{8812800}{23177} = 380 \text{ karton}$$

Jadi untuk produk B024, jumlah maksimal yang dapat dimuat ke dalam kendaraan adalah 380 karton.

Penentuan Kebutuhan Volume Kendaraan untuk setiap Pelanggan

Setelah ditentukan kuantiti muat maksimum kendaraan selanjutnya dilakukan perhitungan besarnya kebutuhan volume kendaraan untuk setiap pelanggan. Berikut ini contoh perhitungan kebutuhan volume kendaraan roda 4 untuk jenis produk B024 pada lokasi P001.

$$\text{Kapasitas box roda 4} = 380 \text{ karton}$$

$$\text{Permintaan produk} = 2 \text{ karton}$$

$$\text{Persentase permintaan} = \frac{2}{308} \times 100\% = 0,53\%$$

Berarti produk B024 untuk permintaan lokasi P001 hanya menempati 0,53% dari 90% ruang box kendaraan roda 4 yang tersedia.

Penentuan Rata-rata Waktu Loading dan Unloading Kendaraan

Data waktu *loading* dan *unloading* kendaraan digunakan untuk menghitung waktu distribusi. Nilai ini diambil berdasarkan rata-rata waktu yang dibutuhkan pekerja untuk kegiatan *loading* dan *unloading*. Untuk kendaraan roda empat, rata-rata waktu yang dibutuhkan untuk *loading* adalah 0,022 jam/persen permintaan dan 0,033 jam/persen permintaan untuk *unloading*. Sedangkan untuk kendaraan roda enam, waktu yang dibutuhkan untuk *loading* adalah 0,033 jam/persen permintaan dan 0,044 jam/persen permintaan untuk *unloading*.

Menentukan Solusi Awal Dengan Metode Nearest Neighbour

Solusi awal didapatkan dengan menggunakan metode *Nearest Neighbour* yaitu memilih satu titik konsumen sebagai titik awal lalu bergerak ke kota berikutnya yang terdekat. Dari data jarak maka didapatkan solusi awal dari rute terpendek yaitu :

- P0 - P201 - P190 - P129 - P130 - P125 - P158 - P157 - P249 - P250 - P178 - P183 - P184 - P186 - P179 - P180 - P073 - P072 -

P068 - P067 - P064 - P069 - P062 - P061 - P056 - P043 - P046 - P049 - P050 - P051 - P041 - P040 - P039 - P030 - P029 - P027 - P026 - P31- P033 - P035 - P024 - P023 - P036 - P037 - P038 - P216 - P0

Pemotongan Rute Berdasarkan Kapasitas dan Waktu

Berdasarkan solusi awal maka rute tersebut dipotong berdasarkan kapasitas kendaraan dan waktu yang tersedia. Kapasitas kendaraan diasumsikan 90% dan waktu distribusi tidak boleh melebihi jam kerja karyawan pukul 17:00. Maka didapatkan rute-rute sebagai berikut :

Rute 1 : P0 - P201 - P190 - P129 - P130 - P125 - P158 - P157 - P249 - P250 - P0

Jarak = 3,08+3,17+2,71+0,02+0,74+2,26 +3,44+2,53+0,05+10,46 = 28,45 km

Persentase permintaan = 46,37%

Rata-rata waktu *loading* = 0,022 jam/persen permintaan

Rata-rata waktu *unloading* = 0,033 jam/persen permintaan

Jumlah titik disribusi = 9 titik.

Kecepatan rata-rata = 40 km/jam

waktu distribusi =

$$\begin{aligned}
 & (\text{waktu loading *permintaan}) + \frac{\text{jarak}}{\text{kecepatan kendaraan}} + \\
 & \quad (\text{waktu unloading * permintaan}) \\
 & = (0,022*46,37) + \frac{28,45\text{km}}{40\text{km/ jam}} \\
 & \quad + (0,033*46,37) \\
 & = 1,02 + 0,71 + 1,53 = 3,26 \text{ jam}
 \end{aligned}$$

Perhitungan waktu ini *feasible* karena tidak melebihi jumlah jam kerja karyawan.

Menggunakan Algoritma Simulated Annealing dengan 2-opt

Pada tahap ini metode yang digunakan adalah 2-opt yaitu pertukaran 2 titik dan menggantikannya dengan pasangan yang mungkin dari 2 titik yang baru yang tidak memecah rute berdasarkan urutan pilihan bilangan random. Solusi yang dihasilkan dengan metode 2-opt ini akan dijadikan sebagai solusi tetangga. Konsep dari *Simulated Annealing* penurunan temperatur secara perlahan-lahan,

penurunan temperatur ini gunanya untuk menentukan berapa iterasi yang akan dilalui dari pencarian solusi, dimana :

- T awal = 100°C
- T akhir = 0,01 °C
- Faktor T = 0,9

Dimana temperatur = Tawal * Faktor T
 = 100 °C * 0,9 = 90 °C

Pada temperatur 90°C akan dimunculkan bilangan random, bilangan random yang muncul adalah antara 1-45 berdasarkan jumlah titik distibusi pada hari tersebut, misalkan bilangan random yang muncul adalah **1** dan **5** maka indeks urutan lokasi 5 ditukar dengan indeks urutan lokasi 1 dari solusi awal *nearest neighbour* yaitu sebagai berikut :

Rute sebelum pertukaran :

P0 - P201 - P190 - P129 - P130 - P125 - P158 - P157 - P249 - P250 - P178 - P183 - P184 - P186 - P179 - P180 - P073 - P072 - P068 - P067 - P064 - P069 - P062 - P061 - P056 - P043 - P046 - P049 - P050 - P051 - P041 - P040 - P039 - P030 - P029 - P027 - P026 - P31- P033 - P035 - P024 - P023 - P036 - P037 - P038 - P216 - P0

Indeks urutan lokasi 5 adalah titik distribusi P125 dan indeks urutan 1 adalah titik distribusi P201.

Rute setelah pertukaran :

P0 - P125 - P190 - P129 - P130 - P201 - P158 - P157 - P249 - P250 - P178 - P183 - P184 - P186 - P179 - P180 - P073 - P072 - P068 - P067 - P064 - P069 - P062 - P061 - P056 - P043 - P046 - P049 - P050 - P051 - P041 - P040 - P039 - P030 - P029 - P027 - P026 - P31- P033 - P035 - P024 - P023 - P036 - P037 - P038 - P216 - P0

Kemudian dilakukan pemotongan rute berdasarkan kapasitas dan batasan waktu, yaitu sebagai berikut :

Rute 1 : P0 - P125 - P190 - P129 - P130 - P201 - P158 - P157 - P249 - P250 - P0

Jarak = 6,81+3,55+2,71+0,02+5,99+9,89 +3,44+2,53+0,05+10,46 = 45,42 km

Persentase permintaan = 46,37%

Rata-rata waktu *loading* = 0,022 jam/persen permintaan

Rata-rata waktu *unloading* = 0,033 jam/persen permintaan

Jumlah titik distribusi = 9 titik.

Kecepatan rata-rata = 40 km/jam

Waktu distribusi =

$$\begin{aligned}
 & (\text{waktu loading} * \text{permintaan}) + \frac{\text{jarak}}{\text{kecepatan kendaraan}} + \\
 & (\text{waktu unloading} * \text{permintaan}) \\
 & = (0,022 * 46,37) + \frac{45,42 \text{ km}}{40 \text{ km/jam}} + \\
 & (0,033 * 46,37) \\
 & = 1,02 + 1,14 + 1,53 = 3,69 \text{ jam}
 \end{aligned}$$

Rute-rute ini disimpan sebagai solusi yang baru. Kemudian iterasi dilakukan kembali dengan memunculkan bilangan random yang lain. Iterasi ini terus dilakukan sampai jumlah solusi pada setiap tingkatan temperatur adalah 5 iterasi. Kemudian solusi terbaik dari 5 iterasi ini disimpan dan dilakukan kembali penurunan temperatur, dimana yang menjadi T awal adalah T akhir dari tahap sebelumnya dikali dengan faktor penurunan temperatur. Solusi awal dari temperatur ini adalah rute terbaik yang diperoleh dari temperatur sebelumnya. Namun, apabila pada temperatur sebelumnya tidak terjadi perbaikan solusi maka yang menjadi solusi awal adalah solusi awal yang diperoleh dari metode jarak terdekat.

Dimana temperatur = T awal * Faktor T

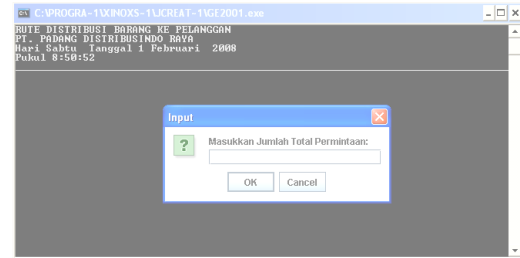
$$= 90 \text{ } ^\circ\text{C} * 0,9 = 81 \text{ } ^\circ\text{C}$$

Pada temperatur 81°C ini akan dimunculkan lagi bilangan random seperti langkah yang sebelumnya dimana bilangan random yang muncul dijadikan indeks dari urutan lokasi yang akan dipertukarkan berdasarkan dari solusi awal, sehingga didapatkan lagi rute baru. Untuk cara selanjutnya sama dengan tahap sebelumnya sampai dilakukan penurunan temperatur mencapai temperatur akhir yaitu 0,01 °C. Setelah tercapainya temperatur akhir maka ditampilkan rute terbaik dengan jumlah kendaraan terkecil dan jarak terpendek.

Pada *Simulated Annealing* ini akan dilakukan *searching* sesuai dengan penurunan temperatur sehingga terdapat sejumlah besar iterasi. Untuk mempermudah

pencarian solusi dengan menggunakan algoritma ini maka digunakanlah bahasa pemrograman Java.

Tampilan awal program untuk perintah input jumlah data permintaan dapat dilihat pada Gambar 13.



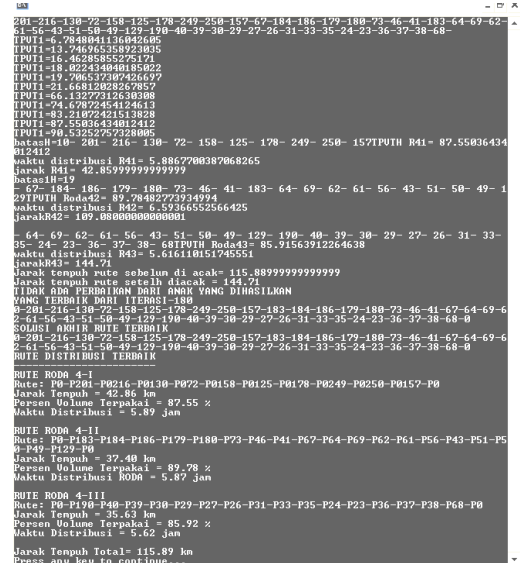
Gambar 13. Tampilan Perintah Input Jumlah Data

Tampilan program untuk perintah input data permintaan dapat dilihat pada Gambar 14.



Gambar 14. Tampilan Perintah Input Data

Tampilan hasil *running* program berupa solusi rute yang mendekati optimal dapat dilihat pada Gambar 15.



Gambar 15. Hasil *Running* Program

6. Kesimpulan dan Saran

Berdasarkan hasil penelitian penentuan rute distribusi produk dengan menggunakan algoritma *simulated annealing* dengan bantuan bahasa pemrograman JAVA dihasilkan rute kendaraan efisien yang dapat meminimasi ongkos distribusi.

Hasil rute perhitungan pada *running* program ke-5 dengan tingkat temperatur 200°C dan faktor penurunan suhu 0,95 adalah:

Rute I (Roda 4):

P0-P216-P0129-P072-P0178-P0125-P0158-P0250-P0249-P0157-P0190-P0

Rute II (Roda 4):

P0-P186-P184-P183-P179-P180-P41-P40-P68-P67-P64-P69-P62-P61-P56-P43-P51-P50-P49-P46-P0

Rute III (Roda 4):

P0-P201-P130-P39-P30-P29-P27-P26-P31-P33-P35-P24-P23-P36-P37-P38-P73-P0

Rata-rata pemanfaatan kapasitas kendaraan adalah 87,75% dan rata-rata waktu distribusi adalah 5,8 jam. Hasil ini feasible karena kapasitas maksimum kendaraan yang dapat digunakan adalah 90% dari volume keseluruhan dan jam kerja karyawan adalah 8 jam.

Perbandingan solusi yang diperoleh dengan menggunakan algoritma *simulated annealing* dan metode perusahaan adalah sebagai berikut:

- Jarak tempuh

Jarak tempuh yang dihasilkan metode perusahaan sebesar 182,59 km sedangkan jarak tempuh yang dihasilkan algoritma *simulated annealing* sebesar 116,77 km. Jadi, Algoritma *simulated annealing* memberikan hasil yang lebih baik dari metode perusahaan dengan pengurangan jarak tempuh sebesar 65,82 km.

- Waktu Distribusi

Waktu distribusi yang tersedia dapat dimanfaatkan lebih baik karena jarak tempuh yang lebih pendek.

- Waktu Pembentukan Rute

Algoritma *simulated annealing* terbukti memberikan waktu pembentukan rute yang lebih cepat dibandingkan dengan menggunakan metode perusahaan saat ini. Algoritma ini hanya membutuhkan waktu 9 menit untuk waktu komputasi. Selain itu

algoritma *simulated annealing* lebih akurat dalam membentuk rute kendaraan karena algoritma ini telah mempertimbangkan jarak tempuh sesuai dengan data yang ada.

- Jumlah alokasi kendaraan

Algoritma *simulated annealing* memberikan pengurangan jumlah alokasi kendaraan sebesar 1 unit kendaraan roda 4 dan 1 unit kendaraan roda 6. Sehingga kendaraan yang digunakan untuk mendistribusikan produk hanya 3 unit kendaraan roda 4 saja.

7. Daftar Pustaka

- Adi, A.H.B, Wirdianto, E., dan Agustina, B.Y., 2005, *Analisis dan Penjadwalan Ulang Kegiatan Delivery di CV. "Y" Padang*, Jurnal Sains dan Teknologi (ISSN 1412-5455), Vol. 5, No. 2, pp. 145-160.
- Ballou, R.H., *Business Logistics Management: Planning, Organizing and Controlling The Supply Chain*, Fourth Edition, Prentice Hall, Inc. New Jersey, 1998.
- Buseti, F., *Simulated Annealing Overview*. <http://www.geocities.com/francorbuseti/saweb.pdf>. accessed July 2007
- Carter, A., *Design and Application of Genetic Algorithms for the Multiple Traveling Salesperson Assignment Problem*. Blacksburg, Virginia, 2003.
- Cordeau, J.F., et.al., 2002, *A guide to vehicle routing heuristic*, Journal of Operational Research Society, Vol 53, pp. 512-522.
- Diaz, B.D., *What is VRP?*, 2004, <<http://neo.lcc.uma.es>> last updated Jan 2004, accessed Mar 2004)
- Gen, M., and Cheng, R., *Genetic Algorithm and Engineering Design*, John Wiley & Sons, Inc., Canada, 1997.
- Heragu, S., *Facilities Design*. PWS Publishing Company, Boston, 1997.
- Larusic, J., *A Heuristic for Solving the Bottleneck Traveling Salesman Problem*, <http://www.cs.unb.ca/undergrad/html/documents/JhonLaRusicThesis.pdf>, accessed July 2007