PENYELESAIAN MASALAH TSP PADA RUTE KUNJUNGAN ATM DENGAN PENDEKATAN HEURISTIK (TABU SEARCH)

Jhon Pontas Simbolon, Muhammad Zarlis

Magister (S2) Teknik Informatika Fasilkom-TI USU

Jl. Universitas No. 24-A, Kampus USU, Medan 20155

jpsblank@yahoo.com, m.zarlis@usu.ac.id

Abstract

Determination of optimum route is a problem that can be found in a variety of activities. Principal of the problem is how to organize the trip so the distance is the minimum distance that the optimum is best found on a map or graph. There are many algorithms available to solve them. Algorithm is divided into two parts, the exact methods and heuristic methods. Heuristic method is considered the best method because it can work quickly. Tabu search is a heuristic method that is often used in solving optimization problems. The algorithm works by improving a solution by using memory to avoid that the search process does not get stuck at a local optimum value by rejecting new solutions that may be in memory (taboo) so that the new solution will be more dispersed. The author will implement a tabu search algorithm to provide a better alternative solution to solve the problems of the effectiveness of the distribution of charging money at the ATM machine.

Keywords: heuristic, tabu search, memory, taboo, optimum.

Abstrak

Penentuan rute optimum adalah permasalahan yang dapat dijumpai dalam berbagai kegiatan. Pokok dari permasalahan adalah bagaimana mengatur rute perjalanan sehingga jarak yang ditempuh merupakan rute yang optimum yaitu jarak minimum terbaik yang terdapat pada suatu peta atau graph. Ada banyak algoritma yang tersedia untuk menyelesaikannya. Algoritma algoritma tersebut dibagi menjadi dua bagian, yaitu metode eksak dan metode heuristik. Metode heuristik dianggap metode yang paling baik karena dapat bekerja dengan cepat. Tabu search merupakan salah satu metode heuristik yang sering dipakai dalam menyelesaikan masalah optimasi. Algoritma ini bekerja dengan cara memperbaiki sebuah solusi dengan menggunakan memori untuk menghindari agar proses pencarian tidak terjebak pada nilai optimum lokal dengan cara menolak solusi baru yang terdapat didalam memori (tabu) sehingga solusi baru akan lebih terpencair. Penulis akan mengimplementasikan algoritma tabu search untuk memberikan pemecahan alternatif yang lebih baik dalam menyelesaikan permasalahan keefektifan rute distribusi pengisian uang pada mesin ATM.

Kata kunci: heuristik, tabu search, memori, tabu, optimum.

1. PENDAHULUAN

Kompleksitas kegiatan penduduk yang begitu padat di kota-kota besar sering kali menimbulkan kemacetan yang meresahkan sebagian besar warganya terutama bagi para pelaku bisnis. Dengan begitu banyaknya aktifitas dalam kegiatan usaha tentunya pengusaha harus dapat bergerak cepat didalam melayani mitra bisnis maupun konsumennya untuk dapat bersaing dengan para kompetitornya.

Traveling Salesman Problem (TSP) dikenal sebagai salah satu masalah optimasi yang banyak menarik perhatian para peneliti sejak beberapa dekade terdahulu. Pada mulanya, TSP dinyatakan sebagai permasalahan dalam mencari jarak minimal sebuah tour tertutup terhadap sejumlah n kota dimana kota-kota yang ada hanya dikunjungi sekali dengan kota awal juga merupakan kota akhir (tujuan). Meskipun TSP sepertinya mudah dinyatakan, akan tetapi sangat sulit untuk diselesaikan terutama untuk persoalan dengan jumlah kota yang banyak. Penyelesaian eksak terhadap persoalan ini akan melibatkan algoritma yang mengharuskan untuk melakukan perhitungan terhadap semua kemungkinan rute yang ada, kemudian memilih salah satu rute yang terpendek. Sebagai akibatnya, kompleksitas waktu dari eksekusi algoritma ini akan menjadi *eksponensial* terhadap ukuran dari masukan yang diberikan. Untuk itu jika terdapat n kota yang harus dikunjungi, maka diperlukan proses pencarian sebanyak $n!/2n$ rute. Dengan cara ini waktu komputasi yang diperlukan akan jauh meningkat seiring dengan bertambahnya jumlah kota yang harus dikunjungi. Permasalahan yang melibatkan algoritma demikian lebih dikenal sebagai permasalahan yang bersifat *Nondeterministic Polynomial-time Complete (NP-Complete)*. Sebagai ilustrasi, untuk 10 kota saja, diperlukan proses pencarian jalur sebanyak 181.440 rute. Penjelasan ini menunjukkan bahwa solusi eksak terhadap masalah TSP sangat sulit dilakukan untuk jumlah n yang besar [1].

Pada perkembangannya, ternyata TSP merupakan persoalan yang banyak ditemui pada berbagai persoalan dunia nyata, diantaranya pada persoalan perencanaan pembangunan, perencanaan produksi, rute pengambilan surat dari kotak pos, rute pengisian uang pada mesin ATM, rute patroli polisi, rute pesawat terbang dsb. Berdasarkan hal tersebut, banyak peneliti lebih memusatkan perhatian kepada pengembangan metode-metode pendekatan (*heuristic*) seperti *Simulated Annealing*, *Ant Algorithm*, Algoritma Genetika, algoritma *Tabu Search* (TS) dan yang lainnya.

Heuristik merupakan metode pencarian untuk penyelesaian masalah optimasi. Algoritma *Tabu Search* (TS) merupakan bagian dari heuristik. Algoritma *Tabu Search* suatu algoritma untuk penyelesaian masalah optimasi yang menggunakan *short-term memory* untuk menjaga agar proses pencarian tidak terjebak pada nilai optimum lokal [2].

Metode ini menggunakan tabu list untuk menyimpan sekumpulan solusi yang baru saja dievaluasi. Selama proses optimasi, pada setiap iterasi, solusi yang akan dievaluasi akan dicocokkan terlebih dahulu dengan isi pada tabu list untuk melihat apakah solusi tersebut sudah ada atau tidak. Bila solusi tersebut sudah ada maka solusi tersebut tidak akan dievaluasi lagi pada iterasi berikutnya. Apabila sudah

tidak ada lagi solusi yang tidak menjadi anggota tabu list, maka nilai terbaik yang baru saja diperoleh merupakan solusi yang sebenarnya[3]. Mengingat prinsip algoritma TS dalam menemukan jarak perjalanan paling pendek tersebut, maka TS merupakan salah satu metode yang tepat digunakan untuk diterapkan dalam penyelesaian masalah optimasi.

Anjungan Tunai Mandiri/*Automatic Teller Machine* (ATM) merupakan seperangkat alat elektronik yang di sediakan untuk melayani transaksi perbankan. Salah satu dari electronic delivery channel ini sudah menjadi ujung tombak/andalan perusahaan yang bergerak dalam jasa keuangan khususnya di bidang perbankan. Perangkat elektronik tersebut dapat melayani berbagai transaksi perbankan dari nasabah bank seperti pengambilan uang tunai, pengecekan saldo rekening tabungan/giro, transfer dana dan pembayaran tagihan tanpa perlu dilayani oleh petugas bank.

Kelangsungan dan kinerja mesin ATM dalam melayani transaksi nasabah harus dapat bekerja se-optimal mungkin. Keberadaan, kelancaran dan keberagaman transaksi ATM dapat memberi image yang baik kepada masyarakat yang menjadi nasabah perbankan. Untuk menjaga kinerja mesin ATM dalam melayani transaksi nasabah adalah dengan menekan downtime se-minimal/sekecil mungkin. Downtime adalah waktu dimana ATM tidak beroperasi/tidak dapat digunakan untuk bertransaksi oleh nasabah. Kriteria dari downtime ATM antara lain uang habis, gangguan komunikasi, gangguan pada perangkat keras/hardware ATM, penggantian uang dan perawatan berkala.

Seiring meningkatnya transaksi di ATM, maka akan mempengaruhi kinerja mesin ATM dalam melayani nasabah. Dapat diketahui bahwa terjadi fluktuasi frekuensi dan durasi downtime ATM yang cukup tinggi. Pencatatan frekuensi downtime berdasarkan jumlah kali terjadi downtime, dan durasi downtime dicatat berdasarkan jumlah menit selama downtime terjadi. Pada periode tertentu, ATM-ATM yang mengalami downtime ini harus dikunjungi untuk dilakukan perbaikan atau pengisian uang. Masalahnya, rute kunjungan pada sejumlah mesin ATM masih dilakukan secara acak sehingga menimbulkan beban energi dengan biaya yang tidak menentu. Rute pengisian uang pada sejumlah mesin ATM adalah masalah TSP, sehingga sangat sulit untuk mencari solusi dari masalah ini dengan pendekatan eksak. Untuk menyelesaikannya digunakanlah pendekatan secara heuristic. Salah satu metode heuristic yang dikenal adalah Algoritma *Tabu Search*.

Untuk itu, topik yang akan diangkat dalam penelitian ini adalah mengimplenentasikan konsep dan cara kerja algoritma *Tabu Search* untuk memberikan pemecahan alternatif yang lebih baik dalam menyelesaikan permasalahan keefektifan rute kunjungan pada sejumlah ATM. Data yang digunakan dalam implementasi ini adalah peta (*maps*) wilayah kota Medan yang di **capture** dari *Google Maps* dengan atribut ruas jalan yang sering dilewati kendaraan umum dan sejumlah verteks yang mewakili ATM. Digitasi pada peta dilakukan dengan menggunakan software khusus pengolahan peta **ArcView 3.3** Untuk membaca koordinat posisi ATM dengan bantuan fasilitas pengolahan grafik yang ada di *Visual Basic 6.0*.

Graph yang digunakan adalah *graph* tidak berarah dan berbobot Untuk menghitung jarak antar *vertex* dalam *graph* digunakan metode *Euclidean*. Output berupa daftar (*list*) solusi disetiap iterasi yang dilakukan, jalur *Global Minimum* dan *walk* (jalan) berupa map garis pandu rute perjalanan yang akan dilalui petugas pengisian uang di ATM.

Tujuan yang ingin dicapai dari penulisan tesis ini adalah untuk menerapkan konsep dan cara kerja algoritma *Tabu Search* untuk optimasi rute perjalanan kunjungan ATM. Adapun manfaat yang dapat diperoleh dari hasil penelitian ini adalah perangkat lunak yang dihasilkan dapat membantu perbankan khususnya divisi ATM dalam membuat rute perjalanan kunjungan ATM untuk pengisian uang secara terencana sehingga nantinya akan menekan waktu dan biaya energi yang dikeluarkan. Penelitian ini juga diharapkan dapat memberikan kontribusi bagi penyelesaian permasalahan TSP.

2. METODOLOGI PENELITIAN

2.1 Data

Data mentah (raw data) yang dipergunakan dalam peneitian ini diperoleh dengan melakukan *capture* map dengan bantuan *Google Maps* yang kemudian akan dilakukan digitasi untuk memetakan verteks-verteks yang mewakili lokasi ATM yang selanjutnya didapatkan titik koordinat setiap verteks yang akan diujicoba. Sebagai contoh dipergunakan 6 data vetex dengan koordinat lokasi seperti pada Tabel 1. dibawah ini.

Tabel 1. Koordinat Verteks

Verteks ke-	X	Y
1	10	00
2	30	00
3	15	20
4	25	20
5	10	30
6	30	30

Untuk menghitung jarak antar verteks dipergunakan rumus Euclidean yaitu :

$$d_{12} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Setelah dilakukan perhitungan dengan menggunakan rumus Euclidean diatas, maka didapatkan tabel jarak antar verteks seperti pada Tabel 2. berikut ini:

Tabel 2. Jarak Antar Verteks

0	20.00	20.62	25.00	30.00	36.06
20.00	0	25.00	20.62	36.06	30.00
20.62	25.00	0	10.00	11.18	18.03
25.00	20.62	10.00	0	18.03	11.18
30.00	36.06	11.18	18.03	0	20.00
36.06	30.00	18.03	11.18	20.00	0

2.2. Pelaksanaan Penelitian

Secara garis besar alur penelitian ini akan dilaksanakan berurutan dan sistematis. Semua hasil pengujian akan diolah dan dapat dibuat kesimpulan yang berupa jawaban dari tujuan penelitian. Tahapan-tahapan yang dilakukan dalam penelitian ini terdiri dari: mengambil data peta, proses digitasi, proses konversi koordinat nyata ke koordinat sistem, meletakkan posisi ATM diatas peta, memilih ATM kunjungan, baca koordinat ATM, bentuk graph, proses Tabu Search, tampilkan dan cetak peta rute.

2.3. Mengambil Data Peta dan Proses Digitasi

Didalam pembentukan *Graph* yang akan digunakan dalam penelitian ini dimulai dengan *capture* bidang potong dari *digital map* yang diperoleh dengan bantuan *Google Maps* tepatnya mengambil peta kota Medan. Bidang peta yang diperoleh dari Google Maps dan telah di digitasi terlihat seperti Gambar 1. dibawah ini.

Gambar 1. Bidang peta yang diperoleh dari Google Maps dan telah di digitasi

Setelah diperoleh bentuk dasar dari peta, perlu dilakukan digitasi yaitu proses pembentukan data vector berupa garis-garis pembentuk ruas jalan, sungai, nama-nama jalan dan titik-titik yang menentukan lokasi obyek tertentu. Tahapan ini dilakukan dengan bantuan aplikasi *ArcView GIS*.

2.4 Pemetaan posisi ATM di atas bidang peta

Pemetaan posisi ATM di atas bidang peta adalah memetakan posisi vertex pada bidang peta digital. Untuk ini dibutuhkan alat gambar untuk memetakan posisi ATM di bidang peta yang diwakili titik hitam. Untuk ini penulis menggunakan fasilitas yang ada di *Visual Basic object Picture Box*. Pemilihan fasilitas ini didasari oleh agar koordinat ATM yang diwakili titik hitam (obyek *shape*) dapat dibaca oleh program nantinya. Pemetaan posisi ATM di atas bidang peta terlihat pada Gambar 2.

Gambar 2. Pemetaan posisi ATM di atas bidang peta

2.5 Proses Baca Koordinat ATM dan Pembentukan Graph

Setelah dilakukan proses digitasi data vector dan pememetaan posisi vertex berupa titik hitam yang mewakili posisi koordinat ATM pada bidang peta digital, maka sistem akan membaca koordinat setiap vertex dalam bentuk sistem koordinat kartesian (x, y) yang akan digunakan untuk pembentukan graph dan untuk proses perhitungan jarak antar vertex. Kemudian jarak antar vertex ini akan digunakan didalam proses pencarian urutan jalur terpendek yang akan dilalui untuk mencapai semua titik dalam graph. Jumlah vertex pada contoh peta dibawah ada 6 buah untuk di ujikan.

Setelah koordinat setiap vertex didapatkan, dapatlah dilakukan proses pembentukan graph seperti terlihat pada Gambar 3. berikut ini.

Gambar 3. Graph yang terbentuk

2.6 Proses Tabu Search

Setelah jarak antar veretex diperoleh maka data jarak antar vertex diberikan ke algoritma tabu Search sebagai input dalam proses pencarian rute terpendek. Proses optimisasi menggunakan *tabu search* dimulai dengan mengeset variabel *MAX_ITER* dan *ITER* untuk jumlah iterasi algoritma, yang dilanjutkan dengan membuat sebuah solusi awal secara acak. Kemudian akan dipilih solusi baru yang terbaik dan *valid* berdasarkan *neighborhood* solusi sebelumnya. Proses pemilihan ini bersifat deterministik. Solusi baru *Vn* yang didapatkan merupakan solusi terbaik dari *neighborhood* solusi sebelumnya *Vc*. Kemudian solusi baru *Vn* menggantikan solusi lama *Vc*. Setelah itu solusi *Vc* kemudian diperbaharui lagi dengan mencari solusi baru *Vn* sampai jumlah counter mencapai nilai *ITER* (jumlah pembaharuan solusi). Solusi yang terbaik saat ini masih bersifat lokal.

Kemudian nilai *counter_total* dinaikkan dan solusi awal diciptakan lagi, diperbaharui dan proses tersebut diulang kembali sampai nilai *counter_total* menjadi *MAX_ITER*. Hal ini dilakukan agar solusi yang didapatkan merupakan solusi yang tersebar, sehingga bisa mendapatkan nilai optimal yang global. Setelah semua solusi terbaik yang bersifat lokal terkumpul, maka solusi yang terbaik dari semua solusi lokal adalah solusi terbaik yang bersifat global.

Berikut ini diberikan kerangka umum algoritma *tabu search*.

Langkah 0 : Tetapkan :

X = Matriks input berukuran (n x m).

MaxIter = Maksimum Iterasi.

Langkah 1 : S = bangkitkan solusi secara random.

Langkah 2 : GlobalMin = FCost(S).

Langkah 3 : Best = S.

Langkah 4 : TabuList = [].

Langkah 5 : Kerjakan dari k = 1 sampai MaxIter:

Langkah 6 : BestSoFar = Cost.

Langkah 7 : BestMove = S.

Langkah 8 : Kerjakan dari i = 1 sampai MaxIter:

Langkah 9 : Kerjakan dari j = i sampai n:

Langkah 10: L = Tukar(S[i],S[j]).

Langkah 11: Cost = FCost(L).

Langkah 12: Jika(L!=TabuList)atau(Cos<GlobalMin),
kerjakan :

Langkah 13: Jika (Cost < BestSoFar), kerjakan :

Langkah 14: BestSoFar = Cost.

Langkah 15: BestMove = L.

Langkah 16: S = BestMove.

Langkah 17: Tambahkan S ke Tabu List.

Langkah 18: Jika BestSoFar < GlobalMin, kerjakan :

Langkah 19: GlobalMin = BestSoFar.

Langkah 20: Best = BestMove.

2.6.1 Proses Inisialisasi

Inisialisasi merupakan proses pemberian nilai awal untuk tiap parameter yang akan dipakai di dalam *tabu search*. panjang memori, jumlah pencarian solusi baru, jumlah pembaharuan solusi. *Tabu search* akan menginisialisasi sebuah solusi awal dengan membuat solusi (rute) secara acak.

2.6.2 Proses Evaluasi dari Solusi

Pada proses ini dilakukan pengevaluasian dari solusi (rute). Kemudian nilai evaluasi akan dimasukkan ke dalam solusi.

Mulai

Hitung Panjang_Rute_Solusi

Masukkan nilai panjang rute solusi ke dalam solusi Akhir

Akhir

2.6.3 Perbaikan memori

Algoritma ini digunakan untuk mencari solusi terbaik dengan cara memperbaiki solusi yang ada. Cara kerja algoritma ini dilakukan dengan metode *swap*, yaitu dengan memilih 2 elemen untuk kemudian menukar posisinya. Algoritma ini tidak mencari seluruh permutasi dari sebuah solusi melainkan hanya sebagian kecil saja, karena elemen yang diubah hanyalah 2 elemen. Hal ini didapatkan berdasarkan cara kerja algoritma ini yang melakukan perhitungan sebanyak

$$C(n, 2) = \frac{N!}{2! * (n - 2)!}$$

Pseudocode dari algoritma ini dinyatakan di bawah ini:

Mulai

$Best \leftarrow Vc$

$$x = C(n, 2) = \frac{N!}{2! * (n - 2)!}$$

Selama $i=0$ sampai dengan $i \leq x$

Selama $y = \text{jumlah vertex}$ sampai dengan $y = x$ lakukan

Buat solusi baru Vn

$\text{Elemen_Solusi_Vn}(y) \leftarrow \text{Elemen_Solusi_Vc}(x)$

$\text{Elemen_Solusi_Vn}(x) \leftarrow \text{Elemen_Solusi_Vc}(y)$

Jika Vn lebih baik daripada $Best$, maka $Best \leftarrow Vn$

Akhir

3. HASIL DAN PEMBAHASAN

3.1 Hasil

Program aplikasi yang telah dikembangkan ini dijalankan secara interaktif yang terintegrasi dengan sistem operasi Windows. Adapun tampilan utama aplikasi yang dikembangkan terlihat pada Gambar 4. Bagian ini menyajikan hasil percobaan yang dilakukan terhadap data ATM yang mengalami downtime sesuai informasi yang didapatkan dari lapangan.

Program aplikasi yang dikembangkan penulis akan menampilkan daftar ATM yang ada dibawah pengawasan Bank seperti terlihat pada Gambar 4. Dengan

berpedoman pada informasi daftar ATM yang mengalami downtime, maka akan dipilih ATM-ATM yang akan dikunjungi. Tidak semu ATM yang mengalami downtime dipilih sekaligus, namun dapat dipartisi berdasarkan wilayahnya menjadi beberapa group pilihan ATM, setiap group akan menghasilkan satu map rute kunjungan. Semua petugas akan diberangkatkan secara serentak untuk mempersingkat waktu kunjungan ATM.

3.1.1. Hasil Pengujian Pertama

Dalam pengujian pertama ini penulis akan menguji data ATM yang mengalami downtime, maka aplikasi akan menampilkan daftar ATM. Dari daftar ATM yang ditampilkan akan dipilih ATM-ATM yang harus dikunjungi. Pada pengujian pertama ini jumlah ATM yang diuji ada sebanyak 6 unit seperti terlihat pada Gambar 4. Kemudian sistem akan membaca koordinat setiap ATM yang dipilih dan nilai koordinat inilah yang akan menjadi data masukan bagi Tabu Search untuk diproses. Kemudian sistem akan menampilkan graph yang terbentuk seperti terlihat pada Gambar 5.

Gambar 4. Pemilihan ATM-ATM yang akan dikunjungi segera

Gambar 5. Graph Terbentuk Pengujian Pertama

Kemudian sistem akan melakukan proses pencarian jalur rute kunjungan ATM terpendek disertai dengan menampilkan tabel informasi hasil proses pencarian jalur seperti terlihat pada Gambar 6., Gambar 7. dan Gambar 8. secara berturut-turut. Tampilan proses pencarian jalur ini akan berlanjut dari iterasi ke-1 s/d iterasi ke-6 yang tampilannya dapat dilihat dengan menggeser *vertical scroll button*.

Gambar 6. Tampilan Proses Pencarian Jalur Iterasi ke-1 (awal)

Gambar 7. Tampilan Proses Pencarian Jalur Iterasi Lanjutan

Gambar 8. Tampilan Proses Pencarian Jalur iterasi ke-6 (akhir)

Adapun Tabel informasi hasil proses yang dirangkum dari tampilan proses pencarian jalur dapat dilihat pada Tabel 4.

Tabel 4. Informasi Hasil Pengujian Pertama

Informasi	Nilai
Jumlah vertex	6 unit
Jumlah jalur tetangga setiap iterasi	15 jalur
Jumlah jalur tetangga	90 jalur
Jumlah jalur tabu	6 jalur

Urutan jalur yang diperoleh	0 1 2 3 4 5
Panjang jalur terpendek (<i>Cost</i>)	192.0732
Waktu proses	18 mili detik

Dari gambar 8. terlihat bahwa jalur terpendek yang diperoleh adalah index 0 1 2 3 4 5 secara berturut-turut adalah ATM-0, ATM-3, ATM-4, ATM-7, ATM-9 dan ATM-16. Kemudian sistem akan menampilkan map rute kunjungan ATM seperti terlihat pada Gambar 9.

Gambar 9. Map Rute Kunjungan ATM Pengujian Pertama

3.1.2. Hasil Pengujian Kedua

Dalam pengujian kedua ini jumlah ATM yang diuji ada sebanyak **30 unit**. Dari data ATM yang di proses ditampilkan graph yang terbentuk dapat dilihat pada Gambar 10.

Gambar 10. Graph Terbentuk Pengujian Kedua

Kemudian sistem akan menampilkan map rute kunjungan ATM seperti terlihat pada Gambar 11.

Gambar 11. Map Rute Kunjungan ATM Pengujian Kedua

Adapun Tabel informasi hasil proses yang dirangkum dari tampilan proses pencarian jalur dapat dilihat pada Tabel 5.

Tabel 5. Informasi Hasil Pengujian Kedua

Informasi	Nilai
Jumlah vertex	30 unit

Informasi	Nilai
Jumlah jalur tetangga setiap iterasi	435 jalur
Jumlah jalur tetangga	13050 jalur
Jumlah jalur tabu	30 jalur
Urutan jalur yang diperoleh	5 26 1 3 2 4 6 7 29 8 10 11 12 13 14 15 17 16 18 19 20 21 22 23 24 25 0 27 9 28
Panjang jalur terpendek (<i>Cost</i>)	496.6972
Waktu proses	38 detik 26 mili detik

3.2 Pembahasan

Dari pengujian yang telah dilakukan terlihat bahwa lintasan yang terbentuk selalu membentuk kurva tertutup. Ini dikarenakan tabu search bekerja mencari verteks yang harus dikunjungi pertama kali dan dilanjutkan dengan vertex-vertex berikutnya sekalian kembali ke vertex awal.

Map rute kunjungan ATM seperti pada Gambar 11. inilah yang akan di cetak (*print*) untuk diberikan ke petugas kunjungan ATM (petugas pengisian uang) dan yang menjadi pedoman rute perjalanannya dalam kunjungan ATM. Terlihat dari map rute kunjungan ATM bahwa rute perjalanan petugas dapat terpandu dengan melihat ruas jalan yang berdekatan dengan garis pada rute hasil proses. Akan selalu ada ruas jalan yang berdekatan dengan garis rute hasil proses.

Kunjungan dapat dilakukan dengan arah memutar kekiri atau memutar ke kanan disesuaikan dengan kondisi arah pengaturan jalan pada posisi awal keberangkatan.

Selalu ada jalan yang dapat dipilih yang berdekatan dengan garis pandu yang diberikan map rute hasil proses.

4. SIMPULAN

Penelitian ini menghasilkan beberapa kesimpulan sebagai berikut :

1. Rancangan program yang dibuat telah mampu menyelesaikan masalah pencarian rute terpendek dengan menggunakan algoritma Tabu Search.
2. Dari hasil pengujian yang telah dilakukan terlihat bahwa perjalanan petugas pengisian uang ke ATM dapat terpandu dengan melihat ruas jalan yang berdekatan dengan garis pandu yang diberikan hasil proses, dengan asumsi selalu ada jalan yang relevan yang bisa di pilih. Map rute kunjungan ATM dapat di cetak (*print*) untuk diberikan kepada petugas pengisian uang yang menjadi panduan dalam perjalanannya.
3. Melalui dua percobaan yang telah dilakukan menunjukkan bahwa waktu proses akan menaik seiring dengan semakin banyaknya jumlah vertex yang di proses.
4. Algoritma Tabu Search cenderung memberikan hasil yang stabil dan konsisten dalam waktu proses maupun rute yang diberikan.

Adapun saran yang dapat diberikan guna penelitian lebih lanjut tentang algoritma Tabu Search dalam penyelesaian masalah pencarian rute perjalanan terpendek adalah dapat dikembangkan dengan pemilihan ruas jalan antar vertex yang dapat ditempuh dengan memberikan parameter-parameter pada setiap ruas jalan, seperti kemacetan pada jam tertentu, keamanan lintasan dan masalah lainnya.

DAFTAR PUSTAKA

- [1]. Betrianis, & Aryawan P. T., "Penerapan Algoritma Tabu Search Dalam Penyelesaian Job Shop", Makara Teknologi, Vol. 7, No. 3, pp. 107-112, Desember 2003.
- [2]. Glover, F., "Tabu Search Fundamentals And Uses", University of Colorado, Boulder, June 1995.
- [3]. Glover, F., "Tabu Search : A Tutorial", Center for Applied Artificial Intelligence, University of Colorado, Boulder, August 1990.
- [4]. Glover, F., "Tabu search - Part I", ORSA Journal on Computing, Vol. 1, No. 3, pp. 190-206, 1989.
- [5]. Mutakhiroh, I., Saptono, F., Hasanah, N. & Wiryadinata, R., "Pemanfaatan Metode Heuristik Dalam Pencarian Jalur Terpendek Dengan Algoritma Semut dan Algoritma Genetik", Seminar Nasional Aplikasi Teknologi Informasi, ISSN: 1907-5022, Yogyakarta, 2007.
- [6]. Jayaswal S., "A Comparative Study of Tabu Search and Simulated Annealing for Traveling Salesman Problem", Project Report Applied Optimization MSCI 703, Department of Management Sciences University of Waterloo
- [7]. Panggabean, H. P., "Penjadwalan Job Shop Statik Dengan Algoritma Tabu Search", INTEGRAL, Vol. 10, No. 1, pp. 34-45, Maret 2005.
- [8]. Wardy, I. S., "Penggunaan graph dalam algoritma semut untuk melakukan optimisasi", Program studi Teknik Informatika, ITB, Bandung, 2007.
- [9]. Wilson, R. J. & Watkhins, J. J., "Graph An Introductory Approach, A First Course in Discrete Mathematics", John Willey and Sons, New York, 1990.
- [10]. Setemen, K. & Purnomo, M. H., "Kombinasi Algoritma Genetika dan Tabu Search dalam Pembuatan Tabel Jadwal Mata Kuliah", Seminar on Intelligent Technology and Its Applications, ISBN: 978-979-8897-24-5, 2008.
- [11]. Bona & Miklos, "A Walk Through Combinatorics An Introduction to Enumeration and Graph Theory", World Scientific Publishing, 2006.

- [12]. Edahiro, M., "Equispreading Tree in Manhattan Distance", Vol. 16, Issue 3, pp. 316-338. 2006.
- [13]. Li, Fajie & Klette, R., "Euclidean Shortest Paths". Springer, 2011.
- [14]. Wallis, W.D., "A Beginner's Guide to Graph Theory". Universitas Birkhauser, Boston, 2010.
- [15]. Hernawati, Anies, Thiang & Eleazar, "Rute Optimum Menggunakan Algoritma Genetika", Jurnal Teknik Elektro, Vol. 2, No. 2, pp. 78-83, September 2002.
- [16]. Joni & Luh, "Pencarian Rute Terpendek Tempat Wisata di Bali Dengan Menggunakan Algoritma Dijkstra", Procceding, SNATI, ISSN: 1907-5022, pp. 46-49, 2010.
- [17]. Budiyanto, E., "Sistem Informasi Geografis dengan ArcView GIS", Penerbit Andi, Yogyakarta, 2010.
- [18]. Berlianty, I. & Arifin, M., "Teknik-Teknik Optimasi Heuristik", Graha Ilmu, Yogyakarta, 2010.