

SIMULASI PERGERAKAN OBYEK DALAM GRAF UNTUK OPTIMASI DISTRIBUSI BARANG ANTAR KOTA

Wiwin Suwarningsih

Pusat Penelitian Informatika LIPI,
Komplek LIPI Gd.20 Lt.3 Jl. Cisitu 21/154-d Sangkuriang Bandung
wiwin.suwarningsih@lipi.go.id

Abstrak

Dalam tulisan ini akan membahas analisa pergerakan sebuah obyek dalam sebuah graf dengan jumlah simpul 20 dengan membuat simulasi pergerakan obyek tersebut dalam sebuah contoh perangkat lunak. Metoda yang akan digunakan dalam menganalisa pergerakan obyek dalam graf ini adalah *simulated annealing*. Hasil akhir dari penelitian ini adalah sebuah model graf yang optimal untuk pergerakan obyek dalam graf. Model rute ini merupakan hasil simulasi menggunakan perangkat lunak yang menerapkan algoritma *simulated annealing* dengan jumlah energi 145.

Kata Kunci

Graf, Cross over, Simple mutator, Obyek gerak, Distribusi barang

1. PENDAHULUAN

Banyak masalah kehidupan sehari-hari yang dapat diabstraksi sebagai masalah yang berkaitan dengan himpunan benda-benda diskret dan relasi biner pada benda-benda tersebut. Salah satunya adalah distribusi barang merupakan bentuk permasalahan optimasi klasik yang berat untuk dipecahkan secara konvensional [1]. Penyelesaian eksak terhadap persoalan ini akan melibatkan algoritma yang mengharuskan untuk mencari kemungkinan semua solusi yang ada. Sebagai akibatnya, kompleksitas waktu dari eksekusi algoritma ini akan menjadi eksponensial terhadap ukuran dari masukan yang diberikan. Permasalahan yang melibatkan algoritma demikian lebih dikenal sebagai permasalahan yang bersifat *Nondeterministic Polynomial-time Complete* (NP-Complete) [2].

Distribusi barang melibatkan objek yaitu seorang kurir yang harus melakukan kunjungan ke sejumlah kota dalam pendistribusian barangnya. Rangkaian kota-kota yang dikunjungi harus membentuk suatu jalur sedemikian sehingga kota-kota tersebut hanya boleh dilewati tepat satu kali dan kemudian kembali lagi ke kota awal [3]. Penyelesaian terhadap permasalahan ini adalah untuk memperoleh jalur terpendek. Penyelesaian eksak terhadap masalah distribusi barang mengharuskan untuk melakukan perhitungan terhadap semua kemungkinan rute yang dapat diperoleh, kemudian memilih salah satu rute yang terpendek. Untuk itu jika terdapat n kota yang harus dikunjungi, maka diperlukan proses pencarian sebanyak $n!/2n$ rute [4]. Dengan cara ini waktu komputasi yang diperlukan akan jauh meningkat seiring dengan bertambahnya jumlah kota yang harus dikunjungi. Sebagai ilustrasi, untuk 10 kota saja, diperlukan proses pencarian jalur sebanyak 181.440 rute. Penjelasan ini menunjukkan bahwa solusi eksak terhadap masalah distribusi barang sangat sulit dilakukan.

Berdasarkan hal tersebut diatas, pada makalah ini akan dibuat sebuah simulasi berupa perangkat lunak yang menggunakan teori graf dan algoritma *simulated annealing* untuk memberikan pemecahan alternatif penentuan rute jalan yang lebih baik pada permasalahan distribusi barang. Adapun jumlah simpul yang digunakan pada simulasi ini adalah 20 simpul.

Secara keseluruhan, paper ini diorganisasikan kedalam bagian-bagian berikut ini : bagian 2 menyajikan penelitian terkait. Bagian 3 menjelaskan dasar teori. Bagian 4 menjelaskan hasil simulasi dan pembahasannya. Bagian 5 menjelaskan kesimpulan dari hasil simulasi.

2. PENELITIAN TERKAIT

Penelitian terdahulu yang memanfaatkan algoritma SA diantaranya adalah Benjamin dkk [1], menyatakan bahwa potensi SA dalam pemodelan harga saham sangat baik, dimana SA cukup berhasil digunakan dalam model seleksi portofolio kompleks. Algoritma SA ini mampu menangani kelas yang memiliki kendala dibandingkan dengan teknik lainnya. Kendala perdagangan yang sulit untuk ditangani karena diskontinuitas dalam ruang portofolio layak. Contoh kekuatan SA yang diungkapkan oleh Chen [4], ditambah dengan teknik pemodelan statistik mekanik baru menyatakan bahwa suku bunga dapat dipasang untuk data jauh lebih baik daripada dalam penelitian yang diterbitkan sebelumnya.

Penelitian lain yaitu menggunakan SA pada satu set masalah ekonometrik [5], termasuk fungsi biaya yang timbul dalam teori moneter penentuan nilai tukar, studi efisiensi produksi perusahaan, dan model jaringan saraf yang menghasilkan cermin ekonomi dan seri keuangan. Wah dkk [7], menunjukkan bahwa algoritma SA menemukan lebih optimal dalam bentuk *nonlinear constrained*, daripada teknik numerik lainnya seperti algoritma genetika dan algoritma quasi-Newton.

Kirkpatrick dkk [9], memperkenalkan algoritma *constrained simulated annealing* (CSA), yang merupakan perpanjangan dari simulasi SA secara konvensional untuk memecahkan nilai probabilistik dalam subruang variabel yang bermasalah. CSA menerapkan metode untuk mengontrol nilai turunan secara terpadu. Begitu pula dengan penelitian yang dilakukan oleh Wah dkk [6], mencoba melakukan konvergensi asymptotic CSA dan CPSA ke minimum global yang dibatasi dengan probabilitas optimasi diskrit. Pada penelitian ini menunjukkan bahwa CSA dan CPSA meminimalkan energi secara implisit pada setiap minimum global yang dibatasi dengan probabilitas satu. Hasilnya adalah penting karena SA menjamin konvergensi asimtotik dalam optimasi diskrit.

3. DASAR TEORI

3.1. Teori Graf

Graf G didefinisikan sebagai pasangan himpunan (V, E) , ditulis dengan notasi $G = (V, E)$, yang dalam hal ini V adalah himpunan tak kosong dari simpul-simpul (*vertices*), dan E adalah himpunan sisi (*edges* atau *arcs*) yang menghubungkan sepasang simpul. Himpunan simpul dari graf G ditulis dengan $V(G)$, sedangkan himpunan sisi dari graf G dinyatakan dengan $E(G)$ [5]. Simpul pada graf dapat dinomori dengan huruf, seperti a, b, c, d, \dots , dengan bilangan asli 1, 2, 3, Sisi yang menghubungkan simpul u dengan simpul v dinyatakan dengan pasangan (u, v) atau dengan lambang e_1, e_2, e_3, \dots . Sebuah sisi dikatakan loop jika sisi tersebut menghubungkan simpul yang sama. Dengan kata lain e adalah loop, jika $e = (v, v)$. Jika dua buah sisi atau lebih menghubungkan dua simpul yang sama, maka sisi-sisi tersebut dikatakan sisi ganda (*multiple edges* atau *parallel edges*).

Graf G disebut terhubung jika untuk setiap dua simpul yang berbeda terdapat lintasan yang menghubungkan simpul-simpul tersebut. Sebuah **lintasan geodesic** (*geodesic path*) antara titik u dan v dari graf G adalah lintasan $u-v$ dengan panjang minimum. Panjang lintasan geodesic antara simpul u dan v dinamakan jarak antara simpul u dan v . Dinotasikan $d(u, v)$.

3.2. Simulated Annealing (SA)

Simulated annealing (SA) adalah salah satu algoritma untuk optimisasi yang bersifat generik[6]. Berbasiskan probabilitas dan mekanika statistik, algoritma ini dapat digunakan untuk mencari pendekatan terhadap solusi optimum global dari suatu permasalahan. Masalah yang membutuhkan pendekatan SA adalah masalah-masalah optimisasi kombinatorial, di mana ruang pencarian solusi yang ada terlalu besar, sehingga hampir tidak mungkin ditemukan solusi eksak terhadap permasalahan itu. Publikasi tentang pendekatan ini pertama kali dilakukan oleh S. Kirkpatrick, C. D. Gelatt dan M. P. Vecchi, diaplikasikan pada desain optimal hardware komputer, dan juga pada salah satu masalah klasik ilmu komputer yaitu *Traveling Salesman Problem*.

Annealing adalah satu teknik yang dikenal dalam bidang metalurgi, digunakan dalam mempelajari proses pembentukan kristal dalam suatu materi. Agar dapat terbentuk susunan kristal yang sempurna, diperlukan pemanasan sampai suatu tingkat tertentu, kemudian dilanjutkan dengan pendinginan yang perlahan-lahan dan terkendali dari materi tersebut. Pemanasan materi di awal proses annealing, memberikan kesempatan pada atom-atom dalam materi itu untuk bergerak secara bebas, mengingat tingkat energi dalam kondisi panas ini cukup tinggi. Proses pendinginan yang perlahan-lahan memungkinkan atom-atom yang tadinya bergerak bebas itu, pada akhirnya menemukan tempat yang optimum, di mana energi internal yang dibutuhkan atom itu untuk mempertahankan posisinya adalah minimum.

Simulated Annealing berjalan berdasarkan analogi dengan proses annealing yang telah dijelaskan diatas. Pada awal proses SA, dipilih suatu solusi awal, yang merepresentasikan kondisi materi sebelum proses dimulai. Gerakan bebas dari atom-atom pada materi, direpresentasikan dalam bentuk modifikasi terhadap solusi awal/solusi sementara. Pada awal proses SA, saat parameter suhu (T) diatur tinggi, solusi sementara yang sudah ada diperbolehkan untuk mengalami modifikasi secara bebas.

Kebebasan ini secara relatif diukur berdasarkan nilai fungsi tertentu yang mengevaluasi seberapa optimal solusi sementara yang telah diperoleh. Bila nilai fungsi evaluasi hasil modifikasi ini membaik (dalam masalah optimisasi yang berusaha mencari minimum berarti nilainya lebih kecil/downhill) solusi hasil modifikasi ini akan digunakan sebagai solusi selanjutnya. Bila nilai fungsi evaluasi hasil modifikasi ini memburuk, pada saat temperatur annealing masih tinggi, solusi yang lebih buruk (uphill) ini masih mungkin diterima. Dalam tahapan selanjutnya saat temperatur sedikit demi sedikit dikurangi, maka kemungkinan untuk menerima langkah modifikasi yang tidak memperbaiki nilai fungsi evaluasi semakin berkurang. Sehingga kebebasan untuk memodifikasi solusi semakin menyempit, sampai akhirnya diharapkan diperoleh solusi yang mendekati solusi optimal.

3.3. Pemodelan dengan SA

Menurut Kirkpatrick[6] ada empat hal utama yang perlu diperhatikan dalam penggunaan SA untuk memodelkan suatu permasalahan :

1. Representasi yang akurat dari konfigurasi dalam suatu permasalahan.
2. Proses modifikasi, langkah acak atau perubahan apa yang harus dilakukan terhadap elemen-elemen konfigurasi untuk menghasilkan konfigurasi berikutnya.
3. Fungsi evaluasi atau fungsi objektif yang dapat menyatakan baik-buruknya suatu solusi terhadap permasalahan
4. Jadwal penurunan suhu dalam proses annealing, dan berapa lama proses ini harus dilakukan.

Karena didalam distribusi barang dicari rute dengan jarak terpendek, maka energi didefinisikan dengan total jarak yang ditempuh.

Energi didefinisikan:

$$E = \sum_{i=1}^n d_i \dots\dots\dots (1)$$

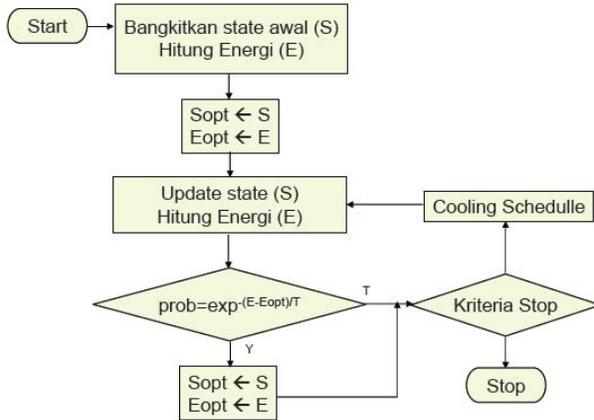
Dimana :

d_i = jarak kota ke $s(i)$ dan $s(i+1)$

Bila posisi dinyatakan sebagai koordinat-koordinat dimensi (x,y) maka d_i dapat dihitung dengan :

$$d_i = \sqrt{(s_x(i) - s_x(i+1))^2 + (s_y(i) - s_y(i+1))^2} \dots\dots(2)$$

Berdasarkan rumus diatas maka pemodelan SA dapat dibuat dalam bentuk graf.



Gambar 1. Bagan Alir SA[6]

4. METODOLOGI

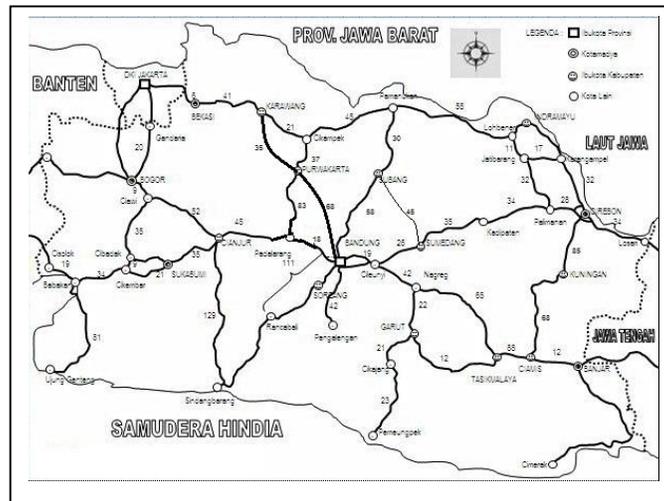
Metodologi yang akan digunakan dalam tulisan ini adalah algoritma *simulated annealing* yang terdiri dari membangkitkan state awal, update state, membangkitkan data kota, menghitung energi dan update state. Studi kasus yang akan digunakan sebagai bahan analisa adalah 20 simpul

dimana simpul-simpul ini merupakan nama kota yang ada di Jawa Barat.

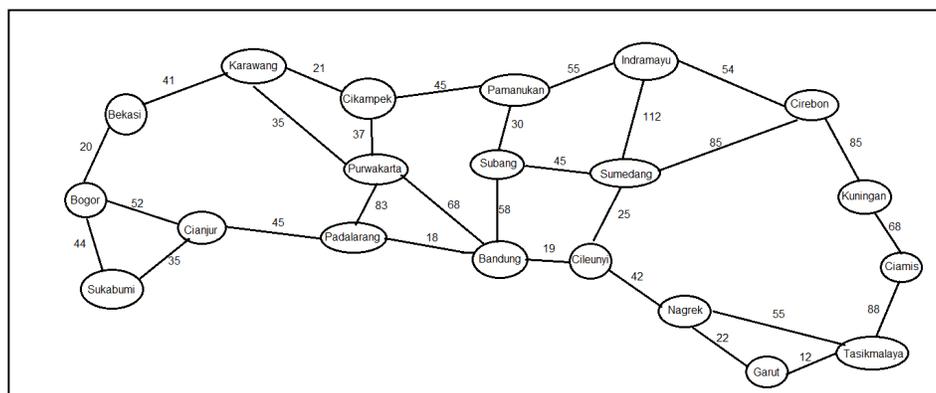
5. HASIL DAN PEMBAHASAN

5.1. Pembangkitan State Awal

Dalam sebuah graf berarah, state awal harus ditentukan dengan tujuan sebagai awal pergerakan obyek. Dalam makalah ini karena menggunakan studi kasus statanya menggunakan nama kota yang ada di Jawa Barat yaitu kota-kota yang akan dikunjungi oleh obyek yang bertugas melakukan distribusi pengiriman barang dari satu kota ke kota lain. State awal dalam kasus distribusi barang ini adalah kota Bandung, karena kota Bandung merupakan kantor pusat dimana perusahaan distribusi barang itu berada. Energi yang dibutuhkan untuk mencapai kota tujuan pertama kali ditentukan dengan cara jarak yang terdekat dan jumlah alamat tujuan di kota kedua dengan melihat juga beban barang yang harus didistribusikan. Pada gambar 1 dapat dilihat peta kota Jawa Barat dengan jarak antar kota yang akan dijadikan sebagai studi kasus. Berdasarkan gambar 2 diatas kami mengambil 20 kota sebagai simpul yang akan digunakan pada simulasi untuk pergerakan obyek dalam graf, dengan jarak antar kota sebagai busur antar simpul. (lihat gambar.3)



Gambar.2. Peta Jarak Antar Kota di Provinsi Jawa Barat[8]



Gambar 3. Graf dengan simpul 20 kota

Pada gambar 3 dapat dilihat bahwa untuk distribusi antar kota banyak alternatif jalan yang dapat dipilih dengan cara menghitung jarak terdekat antar kota. Untuk memudahkan perhitungan dan penyusunan rute maka nama kota yang

merupakan simpul pada gambar 3 akan diubah dengan menggunakan angka 1 sampai dengan 20 (lihat tabel 1).

Tabel 1. Daftar Nama Kota

No Simpul	Nama Kota	No Simpul	Nama Kota	No Simpul	Nama Kota	No Simpul	Nama Kota
1	Bandung	6	Ciamis	11	Pamanukan	16	Karawang
2	Cileunyi	7	Kuningan	12	Subang	17	Bekasi
3	Nangrek	8	Cirebon	13	Cikampek	18	Bogor
4	Garut	9	Sumedang	14	Purwakarta	19	Sukabumi
5	Tasikmalaya	10	Indramayu	15	Padalarang	20	Cianjur

Penyusunan rute secara manual dapat dibentuk dengan menggunakan penalaran berdasarkan jarak terdekat dari kota asal ke kota tujuan sehingga perhitungan total jarak dari state awal sampai dengan state akhir menghasilkan nilai yang belum tentu optimal. Hal ini dikarenakan percobaan pengaturan rute untuk menghasilkan jarak yang minimum dan semua state tersinggahi semakin rumit. Akan tetapi bila kita menggunakan algoritma simulation annealing perhitungan energi dari state awal menuju state akhir diharapkan menghasilkan nilai energi yang optimal.

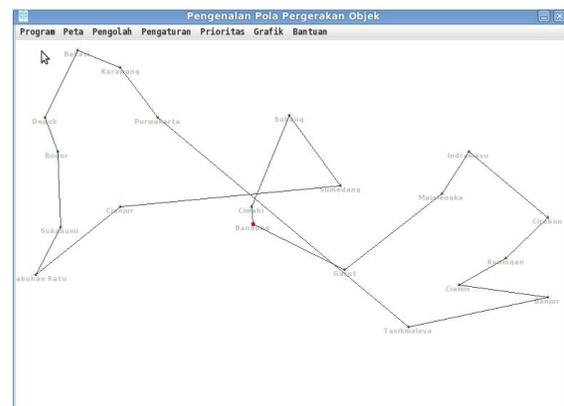
Penentuan state berikutnya dari state awal, diperlukan proses update state. Proses update state ini merupakan proses inventarisir kota tujuan berikutnya dalam sebuah percabangan dalam memilih kota mana yang akan disinggahi terlebih dahulu. Proses inventarisir berupa membangkitkan data kota yang terdiri dari item-item berupa jarak, kepadatan lalu lintas, kondisi jalan rata atau berkontur, dan jumlah alamat tujuan yang harus disinggahi pada kota tersebut. Proses pembangkitan data kota berdasarkan rumus 1 dan rumus 2, sehingga diperoleh tabel 2 daftar kota tujuan berikutnya yang merupakan hasil proses update state dan membangkitkan data kota berdasarkan item-item yang dimiliki oleh masing-masing kota tujuan.

5.2. Perhitungan Energi dan Update state

Tabel 2. Perhitungan Energi dan update State dengan SA

Id-Rute Manual	Susunan Rute Manual	Jumlah Energi	Id-Rute update	Update State Rute dengan SA	Jumlah Energi
M-01	1-2-3-4-5-6-7-8-10-9-12-11-13-16-17-18-19-20-15-1	150	U-01	1-2-10-9-12-11-13-16-17-18-19-20-15-3-4-5-6-7-8-1	149
M-02	1-15-20-19-18-17-16-14-13-11-12-9-10-8-7-6-5-4-3-2-1	201	U-02	1-2-15-20-19-18-17-16-14-13-11-12-9-10-8-7-6-5-4-3-1	145
M-03	1-12-11-13-14-16-17-18-19-20-15-1-2-9-10-8-7-6-5-4-3-2-1	220	U-03	1-2-3-4-5-6-7-8-10-9-12-11-13-16-17-18-19-20-15-1	156
M-04	1-14-13-16-17-18-19-20-15-1-2-3-4-5-6-7-8-10-11-12-9-2-1	230	U-04	1-2-19-18-17-16-14-13-11-12-14-16-17-18-19-20-15-1	167

Berdasarkan pada tabel 2, jumlah energi yang dibutuhkan dengan menggunakan SA lebih kecil dibandingkan dengan menggunakan rute manual. Id-rute update ini digunakan untuk melakukan visualisasi pergerakan obyek dengan aplikasi yang telah dibangun. Untuk id-Rute U-01 visualisasinya pada gambar 4.a, id-Rute U-02 pada gambar 4.b, id-Rute U-03 pada gambar 4.c dan id-Rute U-04 pada gambar 4.b. Sedangkan pemilihan rute optimal ditentukan berdasarkan nilai jumlah energi yang terkecil yaitu id-rute 01 dan id-rute 02 dimana masing-masing memiliki jumlah energi 149 dan 145.



(a)

