SISTEM ALOKASI PENYIMPANAN BARANG PADA GUDANG

Achmad Hambali
Jurusan Teknik Informatika PENS-ITS
Kampus PENS-ITS Keputih Sukolilo Surabaya 60111
Telp (+62)31-5947280, 5946114, Fax. (+62)31-5946114
Email : lo7thdrag@ymail.co.id

Makalah Penelitian

ABSTRAK

Gudang adalah salah satu aspek penting di dalam sebuah pelabuhan. Fungsinya sebagai tempat penyimpanan memiliki peranan yang sangat vital. Oleh sebab itu diperlukan adanya pengaturan yang tepat dan cepat dalam penggunaan ruang gudang. Pada tugas akhir ini permasalahan effisiensi penataan barang dalam gudang akan diselesaikan dengan algoritma semut.

Pencarian solusi dimulai dengan melakukan pemilihan secara bertahap berdasarkan nilai fungsi pheromone trail dan informasi heuristik yang terbesar. Pheromone trail menunjukkan kualitas solusi yang telah dicapai oleh semut dari perjalanan sebelumnya, sedangkan informasi heuristik sesuai dengan input data dari suatu permasalahan. Kegiatan ini dilakukan oleh semua semut dalam satu koloni. Setelah satu koloni semut menyusun kombinasi solusi, maka dilakukan pemilihan semut terbaik yang akan dibandingkan dengan semut terbaik secara global sehingga menghasilkan solusi akhir.Hasil uji coba perangkat lunak menunjukkan bahwa algoritma semut dapat dijadikan metode alternatif untuk menyelesaikan masalah optimasi ruang.

Kata kunci : penataan barang, algoritma semut, *block stacking,* gudang

1. PENDAHULUAN

1.1 Latar Belakang

Gudang adalah salah satu aspek penting di dalam sebuah pelabuhan, fungsinya sebagai tempat penyimpanan sementara barang yang akan dimuat ke kapal maupun barang yang baru dibongkar dari kapal dan akan dikirim lagi ke daerah tujuan membuat gudang memiliki peranan yang sangat fital.

Tingginya tingkat penggunaan gudang membuat effisiensi waktu dan ruang menjadi penting. Diperlukan adanya pengaturan yang tepat dan cepat dalam penggunaan ruang gudang ini. Urutan pengalokasian barang yang masuk secara tepat dapat mengeffisiensikan ruang yang ada.

Untuk itu perlu dibuat suatu sistem yang berfungsi untuk mengalokasikan urutan barang-barang yang masuk kedalam gudang, menempatkannya sesuai dengan aturan penempatan yang tepat. Pada proyek akhir ini akan dibuat suatu sistem yang dapat mengalokasikan urutan barang yang masuk kedalam gudang secara optimal sehingga dapat menghemat ruang

1.2 Rumusan Masalah

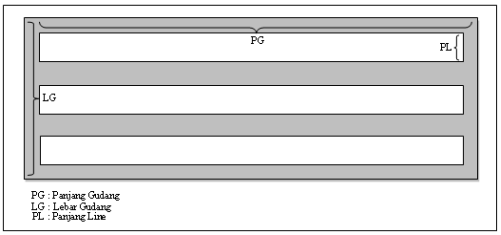
Berdasarkan uraian tersebut di atas, dalam pengerjaan proyek akhir ini timbul suatu masalah yaitu :

1. Membuat aturan penempatan barang
2. Merancang suatu algoritma untuk proses pengisian barang ke dalam gudang.
3. Membangun perangkat lunak berbasis dekstop

1.3 Batasan Masalah

Adapun batasan-batasan permasalahan pada proyek akhir ini, antara lain :

1. Penelitian dilakukan di gudang dengan :
   Panjang : 30 meter
   Lebar : 10 meter
   Tinggi : 4 meter
   Panjang Line : 3 meter



Gambar 1. *Sketsa Gudang*

2. Barang yang masuk tergolong jenis *kubikasi*, dimana berat barang tidak berpengaruh satu sama lain dalam tumpukan.
3. Barang berbentuk *rectangular box* (balok atau kubus).
4. Sistem yang dibuat, hanya untuk mencari urutan barang yang akan ditata.

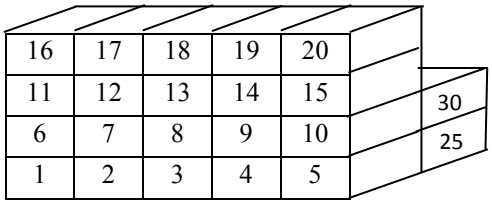
1.4 Tujuan

Tujuan dari proyek akhir ini adalah untuk membuat sebuah perangkat lunak atau software yang bisa mengalokasikan barang yang masuk kedalam gudang secara optimal, sehingga luasan *line/ block* yang dipakai lebih minimum

2. DASAR TEORI

2.1 Block Stacking

Block stacking adalah metode penataan barang dalam gudang. Dalam metode ini barang ditumpuk ke arah atas dan disimpan berjajar menjadi sebuah baris atau blok. Barang ditumpuk dengan ketinggian tertentu berdasarkan kriteria seperti berat beban, ketinggian yang diijinkan dan kapabilitas forklift gudang. Barang yang disimpan dalam metode ini dapat diambil dengan metode LIFO Metode ini tidak memakan biaya karena tidak memerlukan rak dan dapat dilakukan di berbagai tipe gudang dengan lahan yang terbuka luas. Secara umum prioritas pemilihan posisi barang dalam sistem ini sebagai berikut.



Gambar 2 *Prioritas pengisian posisi barang*

Penataan posisi barang pada blok dilakukan mulai dari ujung kiri bawah (no 1) dengan prioritas kekanan, bila barang sudah tidak mungkin maka diulangi lagi dari kiri dengan posisi tumpukan diatasnya.

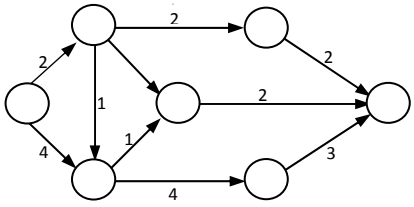
2.2 Graf

Graf adalah kumpulan simpul (*nodes*) yang dihubungkan satu sama lain melalui sisi/busur (*edges*) (Zakaria, 2006).

a. Verteks (simpul) : V = himpunan simpul yang terbatas dan tidak kosong.
b. Edge (sisi/busur) : E = himpunan busur yang menghubungkan sepasang simpul.

Simpul-simpul pada graf dapat merupakan obyek sembarang seperti kota, atom-atom suatu zat, nama anak, jenis buah, komponen alat elektronik dan sebagainya. Busur dapat menunjukkan hubungan (relasi) sembarang seperti rute penerbangan, jalan raya, sambungan telepon, ikatan kimia, dan lain-lain. Notasi graf: G(V,E) artinya graf G memiliki V simpul dan E busur

Ada beberapa macam graf. Pada sistem ini menggunakan graf berarah dan berbobot, sehingga menghasilkan matriks asimetris.



Gambar 3 *Graf Berarah dan Berbobot*

2.3 Algoritma Semut

Algoritma Semut merupakan salah satu pendekatan heuristik yang mampu memberikan hasil positif untuk menyelesaikan permasalahan optimasi dengan menemukan solusi yang baik. Perhitungan yang menyebar dapat menghindari terjebak pada keadaan lokal optimum, Algoritma Semut lebih menguntungkan daripada pendekatan penguatan tiruan (simulaten annealing) dan algoritma genetik, saat grafik mungkin berubah secara dinamis algoritma Semut dapat berjalan secara kontinyu dan menyesuaikan dengan perubahan secara waktu nyata *(real time)*.

Dalam algoritma Ant Colony, diperlukan beberapa variabel dan langkah-langkah untuk menentukan jalur terpendek

Langkah 1 :

a. Inisialisasi harga parameter-parameter algoritma

- Intensitas jejak semut antar state dan perubahannya (τ_{ij})
- Visibilitas dari suatu solusi yang akan dipilih oleh semut ($\acute{\eta}_{ij}$)
- Tetapan pengendali intensitas jejak semut (α), nilai $\alpha \geq 0$
- Tetapan pengendali visibilitas (β), nilai $\beta \geq 0$
- Tetapan siklus semut (Q)
- Banyak semut (m)
- Banyak state (n)
- Tetapan penguapan jejak semut (ρ), nilai $0 > \rho > 1$
- Jumlah siklus maksimum NC_{max}

b. Inisialisasi state pertama setiap semut

Setelah inisialisasi τ_{ij} dilakukan, kemudian m semut ditempatkan pada state secara acak

Langkah 2 :

Pengisian state pertama ke dalam tabu list. Hasil inisialisasi state pertama setiap semut dalam langkah 1 harus diisikan sebagai elemen pertama tabu list. Hasil dari langkah ini adalah terisinya elemen pertama *tabu list* setiap semut dengan indeks state tertentu, yang berarti bahwa setiap $tabu_k(1)$ bisa berisi indeks state antara *1* sampai *n* sebagaimana hasil inisialisasi pada langkah 1.

Langkah 3 :

Penyusunan rute kunjungan setiap semut ke setiap state. Koloni semut yang sudah terdistribusi ke sejumlah atau setiap state, akan mulai melakukan perjalanan dari state pertama masing-masing sebagai state asal dan salah satu state lainnya sebagai state tujuan. Kemudian dari state kedua masing-masing, koloni semut akan melanjutkan perjalanan dengan memilih salah satu dari state yang tidak terdapat pada $tabu_k$ sebagai state tujuan selanjutnya. Perjalanan koloni semut berlangsung terus menerus sampai semua state satu persatu dikunjungi atau telah menempati $tabu_k$. Jika *s* menyatakan indeks urutan kunjungan, state asal dinyatakan sebagai $tabuk(s)$ dan state-state lainnya dinyatakan sebagai $\{N\text{-}tabu_k\}$, maka untuk menentukan state tujuan digunakan persamaan probabilitas state untuk dikunjungi sebagai berikut :

$$P_{ij}^{\ k} = \begin{cases} \dfrac{[\tau_{ij}]^{\alpha}[\acute{\eta}_{ij}]^{\beta}}{\Sigma_{k\varepsilon\Omega}\{[\tau_{ik}]^{\alpha}[\acute{\eta}_{ik}]^{\beta}\}} & \text{Jika } j \varepsilon \Omega \\ 0 & \text{Selain itu} \end{cases} \quad \ldots\ldots\ldots\ldots (1)$$

Dengan i sebagai indeks state asal dan j sebagai indeks state tujuan.

Langkah 4 :

a. Perhitungan Luas minimum yang diperoleh setiap semut

Perhitungan luas minimum tertutup atau L_k setiap semut dilakukan setelah satu siklus diselesaikan oleh semua semut. Perhitungan ini dilakukan berdasarkan $tabu_k$ masing-masing dengan persamaan berikut :

$$L_k = d_{tabuk(n),\, tabuk(1)} + \sum_{s=1}^{n=1} d_{tabuk(s),\, tabuk(s+1)} \ldots\ldots (2)$$

dengan d_{ij} adalah selisih luas state i ke state j yang dihitung berdasarkan persamaan :

$$d_{ij} = Luas_i - Luas_j \ldots\ldots\ldots\ldots (3)$$

b. Pencarian volume terminimum

Setelah L_k setiap semut dihitung, akan didapat luas minimum tertutup setiap siklus atau L_{minNC} dan harga luas minimum tertutup secara keseluruhan atau L_{min}.

c. Perhitungan perubahan harga intensitas jejak kaki semut antar state

Koloni semut akan meninggalkan jejak-jejak kaki pada lintasan antar state yang dilaluinya. Adanya penguapan dan perbedaan jumlah semut yang lewat, menyebabkan kemungkinan terjadinya perubahan harga intensitas jejak kaki semut antar state. Persamaan perubahan ini adalah :

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \ldots\ldots\ldots\ldots (4)$$

dengan $\Delta\tau_{ij}^{\ k}$ adalah perubahan harga intensitas jejak kaki semut antar state setiap semut yang dihitung berdasarkan persamaan :

$$\Delta\tau_{ij}^{\ k} = \begin{cases} \dfrac{Q}{L_k} & \text{Untuk (i,j) } \varepsilon \text{ state asal dan state tujuan} \\ 0 & \text{Selain} \end{cases} \quad \ldots\ldots\ldots\ldots (5)$$

Langkah 5 :

a. Perhitungan harga intensitas jejak kaki semut antar state untuk siklus selanjutnya.

   Harga intensitas jejak kaki semut antar state pada semua lintasan antar state ada kemungkinan berubah karena adanya penguapan dan perbedaan jumlah semut yang melewati. Untuk siklus selanjutnya, semut yang akan melewati lintasan tersebut harga intensitasnya telah berubah. Harga intensitas jejak kaki semut antar state untuk siklus selanjutnya dihitung dengan persamaan :

$$\tau_{ij} = \rho * \tau_{ij} + \Delta\tau_{ij} \quad \text{..... (6)}$$

b. Atur ulang harga perubahan intensitas jejak kaki semut antar state.

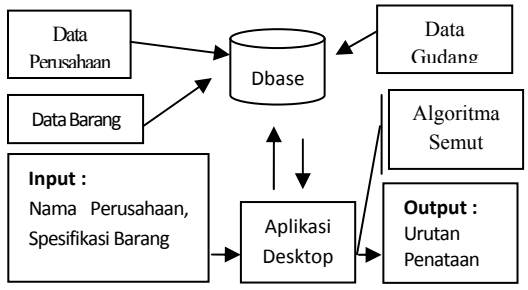
   Untuk siklus selanjutnya perubahan harga intensitas jejak semut antar state perlu diatur kembali agar memiliki nilai sama dengan nol.

Langkah 6 :

Pengosongan *tabu list*, dan ulangi langkah 2 jika diperlukan. *Tabu list* perlu dikosongkan untuk diisi lagi dengan urutan state yang baru pada siklus selanjutnya, jika jumlah siklus maksimum belum tercapai atau belum terjadi konvergensi. Algoritma diulang lagi dari langkah 2 dengan harga parameter intensitas jejak kaki semut antar state yang sudah diperbaharui.

3. DESAIN SISTEM

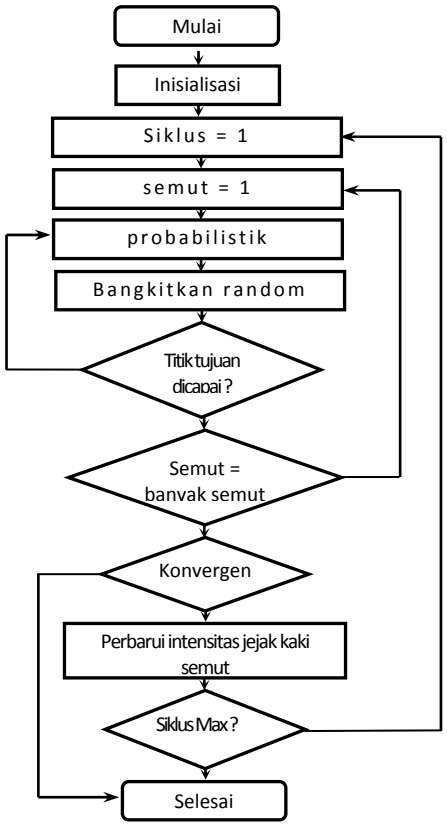
Secara umum gambaran sistem yang akan dibangun dalam proyek akhir ini seperti skema berikut :



Gambar 4 *Blok diagram proses perangkat lunak*

Dengan algoritma algoritma semut sebagai algoritma utamanya.

Berikut ini adalah flow chart dari algoritma semut.

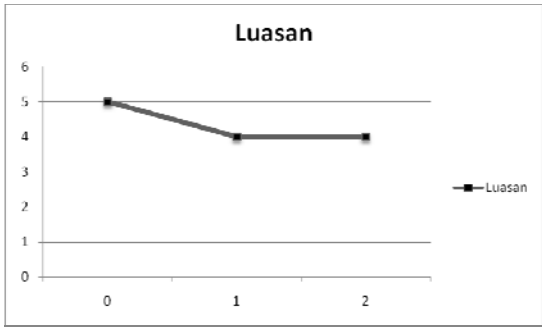


Gambar 5 *Flowchart Algoritma Koloni Semut*

Algoritma ini adalah inti dari sistem yang sedang dibangun. Dengan menggunakan algoritma semut ini nantinya bisa ditemukan urutan pengalokasian barang yang masuk ke dalam gudang. Sehingga diperolah hasil yang optimal

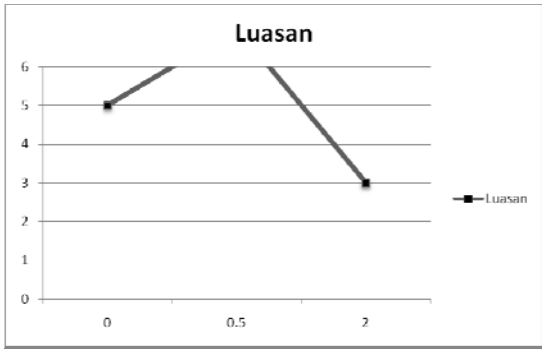
4. UJI COBA DAN ANALISA

Parameter α, β, ρ mempunyai pengaruh terhadap performa hasil yang diperoleh pada sistem. Sehingga untuk mengetahui pengaruh dari parameter tersebut, berikut diberikan hasil simulasi dengan nilai α, β, ρ masing-masing α {0, 1, 2} β{0, 0.5, 2} ρ{0.1, 0.5, 0.9}. hasil eksperimen yang diperoleh ditunjukkan pada gambar untuk nilai α, gambar untuk nilai β dan gambar untuk nilai ρ.



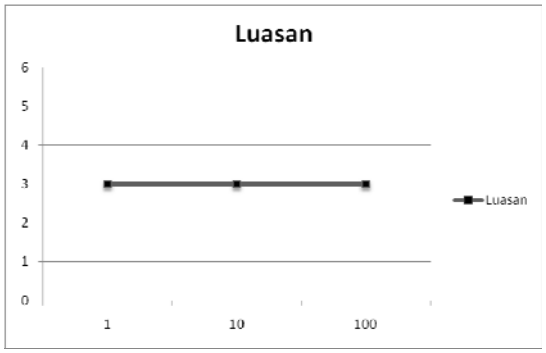
Gambar 6 *Blok diagram proses perangkat lunak*

Gambar diatas menunjukkan dengan nilai α = 1 dan 2 diperoleh performa hasil yang terbaik dan performa paling jelek diperoleh dengan nila α = 0



Gambar 7 *Blok diagram proses perangkat lunak*

Gambar diatas menunjukkan dengan nilai β = 2 diperoleh performa hasil yang terbaik dan performa paling jelek diperoleh dengan nila β = 0,5



Gambar 8 *Blok diagram proses perangkat lunak*

Gambar diatas menunjukkan tak ada pengaruh yang diciptakan oleh nilai ρ.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan analisa dari beberapa pengujian yang diterangkan pada bab sebelumnya, kesimpulan yang di dapat :

1. Optimasi ruang dengan metode algoritma semut sangat tergantung dengan kesesuaian parameter yang diinputkan. Dengan nilai parameter yang tepat solusi paling optimal bisa ditemukan.
2. Pada algoritma dengan matriks asimetris, kemampuan sistem untuk menemukan solusi tergantung dari banyaknya semut, ketika jumlah semut lebih kecil dari jumlah barang, kemungkinan sistem menemukan solusi semakin kecil, begitu juga ketika jumlah semut lebih besar dari jumlah barang.
3. Kemampuan sistem paling optimal terjadi ketika jumlah semut dan jumlah barang sama.
4. Sedangkan jumlah siklus berpengaruh terhadap pencarian solusi paling optimal.
5. Semakin besar nilai siklus maksimalnya, semakin besar pula kemungkinan menemukan solusi paling optimal.
6. Parameter lain yang juga berpengaruh adalah α (alfa), β (beta), dan ρ (rho).
7. Solusi terbaik terjadi ketika α bernilai 0, β bernilai 2 dan ρ bernilai 0,9. Sedang solusi terjelek ketika α bernilai 2, β bernilai desimal dan ρ bernilai 0,1
8. Kemampuan algoritma dalam menemukan solusi juga dipengaruhi oleh bilangan random.

5.2 Saran

Permasalahan optimasi pada tugas akhir ini hanya dibatasi pada kriteria barang bertipe kubikasi, diharapkan ada penelitian lebih lanjut, untuk barang bertipe tonase serta gabungan keduanya.

6. DAFTAR PUSTAKA

[1] Leksono, Agung. 2009. *Algoritma Ant Colony Optimization (ACO) untuk menyelesaikan Treveling Salesman Problem (TSP).* Semarang : Universitas Diponegoro Semarang

[2] Sina, Ibnu W. 2007. *Pemanfaatan Graf dalam Algoritma Semut untuk Melakukan Optimasi*. Bandung : Program Studi Teknik Informatika ITB

[3] Putro, Fidi Wincoko. 2009. *Sistem Navigasi Perjalanan Berbasis Web dengan Algoritma Koloni Semut*.Surabaya: PENS-ITS

[4] http://bernardonugroho.blogspot.com/2009/02/metode-penataan-palet-di-gudang.html. Diakses tanggal 21 Juli 2009