

Masalah Desain Tata Letak Fasilitas melalui Pemrosesan String Alfanumerik

Oleh:

Yaya S. Kusumah

Jurusan Pendidikan Matematika
Universitas Pendidikan Indonesia
Jl. Dr. Setiabudhi 229, Bandung 40154
Telp./Faks.(022) 2004508
e-mail: yskusumah@upi.edu

Abstrak

Dalam Masalah Desain Tata Letak Fasilitas (MDTLF) yang menjadi pokok persoalan adalah bagaimana menentukan susunan dan konfigurasi fasilitas yang paling efektif yang membantu pelaksanaan produksi atau memberikan keuntungan lebih tinggi dalam jasa layanan. Biasanya tujuan yang ingin dicapai adalah meminimalkan kuantitas seperti biaya produksi, ruangan yang harus digunakan, perpindahan barang, atau jarak tempuh. Jika tujuannya adalah memaksimalkan, maka beberapa contohnya adalah kuantitas seperti aliran lalu lintas barang, atau keuntungan produksi. Masalah ini biasanya menjadi lebih rumit jika semakin banyak persyaratan yang harus dipenuhi, misalnya ketersediaan bahan atau sumber, faktor keamanan, urutan pemrosesan pekerjaan, waktu, dan bentuk-bentuk barang atau gedung. Pada makalah ini akan sedikit disinggung beberapa model matematis dan metode penyelesaian yang telah dikembangkan untuk menyelesaikan masalah desain tata letak fasilitas. Pendekatan-pendekatan eksak ataupun heuristik akan didiskusikan secara sekilas. Khusus untuk algoritma pemrosesan string akan dibicarakan secara lebih rinci.

Kata Kunci: Masalah Desain Tata Letak Fasilitas, Pemrosesan String.

1. Pendahuluan

Masalah desain tata letak fasilitas (MDTLF) berkaitan dengan penentuan susunan dan konfigurasi fasilitas secara fisik yang paling efektif yang membantu produksi suatu hasil atau keuntungan jasa layanan. Beberapa contoh obyek fisik tersebut di antaranya adalah, mesin, bangunan, panel instrumentasi, markas polisi, kompleks perumahan, pusat olah raga, pusat layanan kesehatan. Biasanya tujuan yang ingin dicapai adalah meminimalkan kuantitas seperti biaya produksi, ruangan yang harus digunakan, perpindahan barang, atau jarak tempuh. Jika tujuannya adalah memaksimalkan, maka beberapa contohnya adalah kuantitas seperti arus lalu lintas barang, atau keuntungan produksi. Masalah ini biasanya menjadi lebih rumit manakala semakin banyak kendala atau persyaratan yang harus dipenuhi, misalnya ketersediaan

bahan atau sumber, faktor keamanan, urutan pemrosesan pekerjaan, waktu, dan bentuk-bentuk barang atau gedung.

2. Model Eksak untuk Masalah Desain Tata Letak Fasilitas (MDTLF)

Kusiak and Heragu (1987) menjelaskan bahwa masalah desain tata letak fasilitas dapat dimodelkan menjadi masalah penugasan kuadratis; masalah peliputan kuadratis, masalah pemrograman bilangan bulat linear, masalah pemrograman bilangan bulat campuran dan masalah teori graf. Pada bagian ini, model masalah penugasan kuadratis akan sedikit didiskusikan.

Koopmans dan Beckman (1957) memodelkan MDTLF sebagai Masalah Penugasan Kuadratis (MPK). Dalam model ini yang menjadi fungsi tujuannya adalah fungsi kuadratis dan kendalanya berupa fungsi linear. Rumusan modelnya adalah:

Misalkan n : jumlah total fasilitas,

a_{ij} : biaya tetap penempatan fasilitas i pada lokasi j ,

f_{ik} : aliran barang antara lokasi i and lokasi k ,

c_{jl} : biaya per satuan barang yang dipindahkan dari lokasi j ke lokasi l ,

$x_{ij} = 1$ jika fasilitas i berada pada lokasi j , dan 0 pada situasi lainnya.

Maka kita dapat memformulasikan MDTLF sebagai masalah penugasan kuadratis:

$$\text{Min } \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_{ij} + \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n f_{ik} c_{jl} x_{ij} x_{kl} \quad (1)$$

subject to

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n, \quad (2)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, 2, \dots, n, \quad (3)$$

$$x_{ij} \in \{0,1\}, \quad i, j = 1, 2, \dots, n. \quad (4)$$

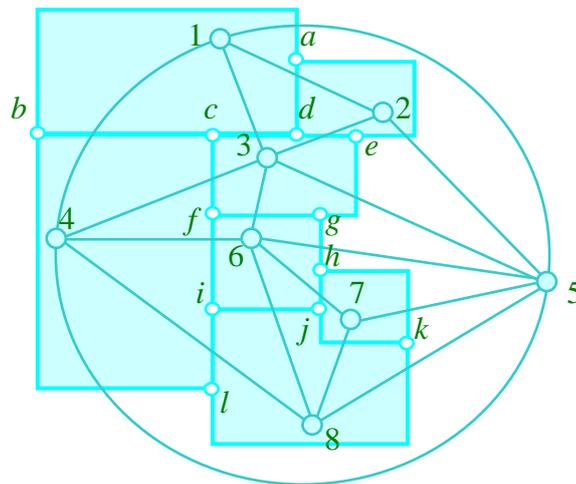
Fungsi obyektif (1) termasuk kuadratis dalam variabel x_{ij} 's. Kendala (2) memastikan bahwa tiap-tiap fasilitas yang dipasangkan dengan dengan sebuah lokasi sedangkan kendala (3) memastikan bahwa tiap-tiap lokasi dipasangkan dengan sebuah fasilitas.

Dalam menyelesaikan MDTLF terdapat beberapa algoritma yang diusulkan, misalnya yang dikemukakan oleh Gilmore (1962), Lawler (1963), Seppanen dan Moore

(1970), Foulds dan Robinson (1976), Kaku dan Thompson (1986). Algoritma eksak, yang dapat memberikan penyelesaian terbaik dari masalah yang diberikan, dapat dibagi menjadi dua bagian: Algoritma Pencabangan dan Pembatasan (*Branch and Bound algorithm*) dan Algoritma Bidang Pemotong (*Cutting Plane algorithm*).

3. Model Heuristik untuk Masalah Desain Tata Letak Fasilitas

Penyelesaian yang dihasilkan oleh keempat model tadi memberikan hasil eksak. Terdapat pula model yang memberikan hasil tak eksak (suboptimal), misalnya model teori graf. Dengan menggunakan pendekatan teori graf, kita dapat memodelkan desain tata letak fasilitas sebagai graf dengan sisi berbobot; simpul menyatakan fasilitas, sedangkan sisinya menyatakan kaitan atau kedekatan. Dengan diketahuinya sebuah graf berbobot G , maka masalah tata letak fasilitas diartikan sebagai penentuan subgraf rentang berbobot maksimum G' dari graf planar G . Gambar berikut adalah sebuah graf dengan tata letak fasilitas yang berkorespondensi disertai dengan tabel yang menyatakan hubungan antar fasilitas.



	1	2	3	4	5	6	7	8
1	---	(a,d)	(c,d)	(b,c)	(a,b)	---	---	---
2		---	(d,e)	---	(a,e)	---	---	---
3			---	(c,f)	(e,g)	(f,g)	---	---
4				---	(b,l)	(f,i)	---	(i,l)
5					---	(g,h)	(h,k)	(k,l)
6						---	(h,j)	(i,j)
7							---	(j,k)
8								---

Gambar 1. Sebuah graf dan Matriks-M yang berkorespondensi.

Sejumlah heuristik konstruktif yang membentuk subgraf planar maksimal sudah banyak yang dikemukakan. Umumnya heuristik-heuristik tersebut diawali dengan K_3 atau K_4 dan membentuk penyelesaian melalui penyisipan simpul, dengan tetap menjaga planaritas dalam setiap langkahnya. Foulds *et al.* (1985) menyatakan bahwa tes planaritas, meskipun mudah implementasinya tapi sangat memperlambat algoritma.

Beberapa heuristik konstruktif yang tidak memerlukan uji planaritas umumnya berbasis teori graf, misalnya Metode Deltahedron (Foulds dan Robinson, 1978), Metode Wheel Expansion (Eades *et al.*, 1982), Algoritma Green-Al Hakim (Green and Al-Hakim, 1985), Heuristik Konstruktif Leung (Leung, 1992), TESSA (Boswell, 1992), dan Algoritma Kim-Kim (Kim dan Kim, 1995).

4. Algoritma Pemrosesan String

Seppanen dan Moore (1975) serta Moore (1976) memperkenalkan suatu algoritma untuk menyelesaikan masalah desain tata letak fasilitas, dengan lambang string yang berkorespondensi dengan sebuah graf yang dimanipulasi untuk memperoleh suatu penyelesaian.

Untuk menerapkan algoritma ini, pertama-tama kita membentuk pohon rentang maksimal (graf bagian dari sebuah pohon), dengan menggunakan algoritma Kruskal. Kemudian dilanjutkan dengan membuat bahasa string lambang untuk menyatakan sebuah kelas graf planar dan pohon.

Misalkan $G=(V, E)$ adalah graf terhingga dengan n simpul. Suatu aturan penulisan dalam bahasa string adalah sebuah relasi dalam bentuk $\phi \rightarrow \varphi$, dengan ϕ dan φ sebagai string lambang dari V . Ini berarti bahwa jika string ϕ dibentuk, maka string φ juga dapat dibentuk.

a. Batas Pohon

Misalkan $T=(V, E^*)$ adalah sebuah pohon dengan n simpul. Batas dari T didefinisikan sebagai sebuah string lambang yang diperoleh dengan cara sebagai berikut. Kita ambil sebuah simpul dalam T sebagai titik awal. Lambang dari simpul tersebut diambil sebagai lambang pertama dari string batas. Untuk memperoleh string yang lainnya, pohon tersebut ditelusuri dalam arah tertentu (misalnya searah jarum jam atau berlawanan arah jarum jam). Di saat sebuah simpul ditelusuri, lambangnya disisipkan pada string tadi. Penelusuran ini dilanjutkan terus sampai titik awal dicapai.

b. Gramatika Pohon dan Graf Planar

Untuk menumbuhkan pohonnya dan untuk memperoleh sebuah graf planar dengan manipulasi string, kita gunakan beberapa aturan. Untuk membuat pohon awal, kita buat sebuah sisi tunggal dengan menggunakan aturan berikut.

Aturan 1: $T \rightarrow ab$.

Segera setelah sebuah pohon ditanam, maka untuk lambang c yang ada dalam pohon tersebut dan suatu lambang baru x , maka pohonnya dapat dikembangkan dengan menggunakan Aturan 2:

Aturan 2: $c \rightarrow cxc$.

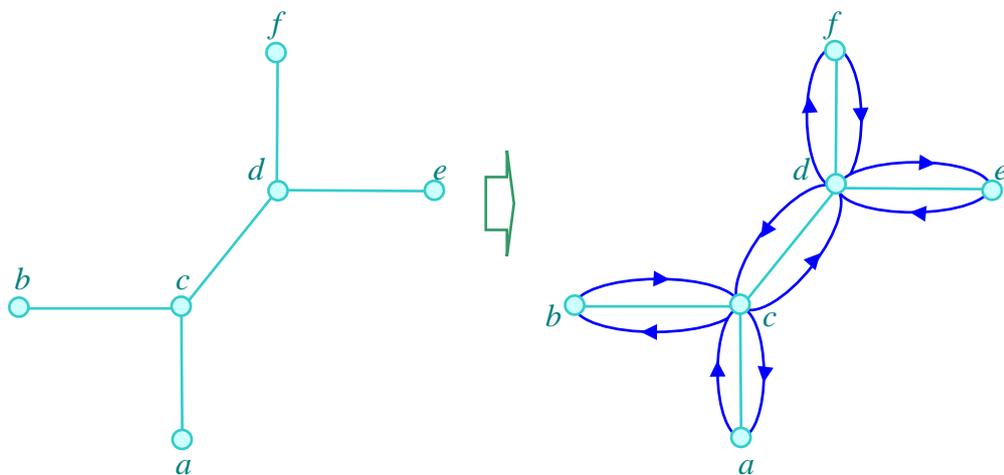
Untuk membentuk sebuah graf planar, sejumlah sisi ditambahkan pada pohon rentang yang ada, dengan menempelkan sisi tersebut pada representasi planar grafnya. Misalkan T adalah sebuah pohon dan $S = aAbB$ adalah sebuah string yang dibentuk oleh G_T yang menyatakan batas dari T dengan dua unsur khusus a dan b . Lambang a dan b dikatakan membatasi substring A dan B dalam string batas sirkular S . Kita dapat mengubah string menjadi dua string yang berlainan, dengan menggunakan aturan penulisan berikut:

Aturan 3: $\{aAbB \rightarrow aAb bBa\}$

Aturan ini mempartisikan string S menjadi 2 string yang berlainan, masing-masing mengandung sebuah substring yang memberi batas.

Contoh 1.

Perhatikan pohon pada gambar berikut.

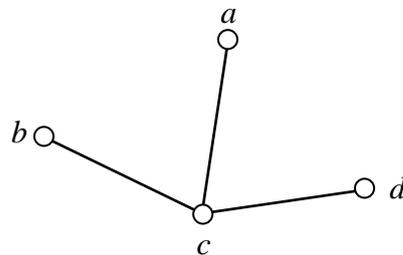


Gambar 2. Sebuah pohon dan batasnya.

Dengan mendefinisikan simpul a sebagai titik awal, kita dapat menyatakan batas pohon sebagai sebuah string $S = acbcdedc$.

Contoh 2.

Misalkan terdapat sebuah pohon rentang T seperti yang diilustrasikan dalam gambar di bawah ini.



Gambar 3. Pohon Rentang.

Misalkan kita ambil simpul c sebagai titik awal dan sisi (c,b) sebagai pohon awal. Untuk memperoleh sebuah batas pohon dari pohon ini kita bentuk sebuah string sebagai berikut (fon yang dicetak tebal menyatakan simpul sebagai titik pusat).

$$\begin{aligned}
 T &\rightarrow \mathbf{cb} \\
 &\rightarrow \mathbf{cacb} \\
 &\rightarrow \mathbf{cdcacb}
 \end{aligned}$$

Untuk membentuk sebuah graf planar maksimal dengan menggunakan sebuah representasi string kita partisikan string ini menjadi beberapa string terpisah yang berlainan sebagai berikut:

$$\begin{aligned}
 \mathbf{cdcacb} &\rightarrow \mathbf{cdcab} \mathbf{acb} && \text{(Menggabungkan simpul } a \text{ dengan simpul } b \text{ sehingga diperoleh muka } (a,c,b)) \\
 &\rightarrow \mathbf{cdb} \mathbf{dcab} \mathbf{acb} && \text{(Menggabungkan simpul } d \text{ dengan simpul } b \text{ sehingga diperoleh muka } (c,d,b)) \\
 &\rightarrow \mathbf{cdb} \mathbf{dab} \mathbf{dca} \mathbf{acb} && \text{(Menggabungkan simpul } d \text{ dengan simpul } a \text{ sehingga diperoleh muka } (d,a,b), \text{ dan } (d,c,a)).
 \end{aligned}$$

Baris terakhir menyatakan sebuah graf planar maksimal yang memuat muka-muka (c,d,b) , (d,a,b) , (d,c,a) , and (a,c,b) . Di sini kita mengasumsikan bahwa urutan seleksi sisinya didasarkan pada bobot sisi. Dalam conoh ini urutannya adalah (a,b) , (d,b) , (d,a) .

Contoh 3.

Untuk memahami bagaimana algoritma pemrosesan string, kita ambil sebuah contoh yang terdiri atas 6 simpul dengan menggunakan bobot sisi dari Boswell (1992).

	a	b	c	d	e	f
a	0	55	45	45	83	91
b	55	0	78	38	72	16
c	45	78	0	52	43	69
d	45	38	52	0	85	41
e	83	72	43	85	0	50
f	91	16	69	41	50	0

Tabel 1. Bobot Sisi dalam Diagram Hubungan (Boswell, 1992).

Pertama kita bentuk sebuah pohon rentang maksimal dengan menggunakan algoritma Kruskal. Kita susun dalam urutan tak naik:

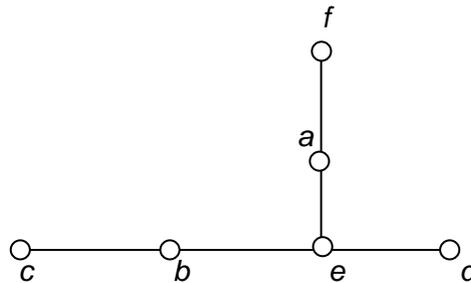
	Edges	Weight
1	(a,f)	91
2	(d,e)	85
3	(a,e)	83
4	(b,c)	78
5	(b,e)	72
6	(c,f)	69
7	(a,b)	55
8	(c,d)	52

	Edges	Weight
9	(e,f)	50
10	(a,c)	45
11	(a,d)	45
12	(c,e)	43
13	(d,f)	41
14	(b,d)	38
15	(b,f)	16

Tabel 2. Daftar Bobot Sisi dengan Urutan Tak Naik.

Berdasarkan pada urutan ini dan bobot masing-masing sisi, kita memiliki pohon rentang maksimal berikut:

$$(a,f) \rightarrow (a,e) \rightarrow (d,e) \rightarrow (b,e) \rightarrow (b,c).$$



Gambar 4. Pohon rentang maksimal.

Untuk membuat sebuah string dari pohon rentang maksimal di atas, kita perlu memilih sebuah simpul (verteks) dan mengambil sebuah string yang menyatakan simpul ini sebagai titik awal. Tanpa kehilangan sifat keumuman, kita ambil simpul a dan mulai dari af (yang menyatakan sisi (a,f)).

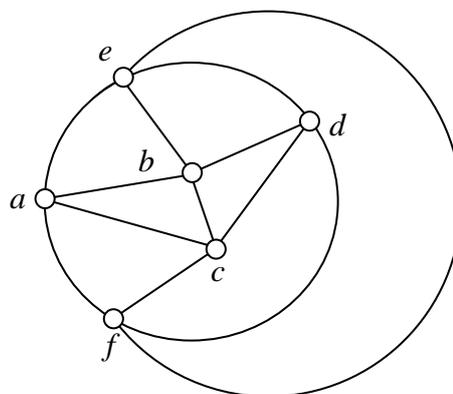
$af \rightarrow aeaf$ (penambahan ae)
 $\rightarrow aedeaf$ (penambahan de)
 $\rightarrow aeदेbeaf$ (penambahan be)
 $\rightarrow aeदेbcbeaf$ (penambahan bc).

Sekarang kita membentuk sebuah graf planar maksimal dengan membuat triangulasi pohon tersebut. Kita awali dengan menyisipkan sisi dengan bobot tertinggi (c,f) pada pohon rentang maksimal yang ada. Untuk melakukannya, kita pecah string $S = aeदेbcbeaf$ menjadi menjadi $S_1 = cbeaf$ dan $S_2 = aeदेbcf$. Berdasarkan bobotnya, (a,b) merupakan sisi berikutnya yang akan disipkan. Dari S_1 kita peroleh sebuah string baru $S_3 = bea$ dan $S_4 = cbaf$. Seleksi berikutnya adalah (c,d) . Maka kita tempelkan cd pada string S_2 yang menghasilkan $S_5 = aeदcf$ dan $S_6 = देbc$. Dengan menempelkan ef pada S_5 memberikan $S_7 = edcf$ dan $S_8 = aef$, dan menempelkan ca pada S_4 menghasilkan $S_9 = cba$ dan $S_{10} = caf$. Seleksi berikutnya adalah df , maka kita tempelkan pada S_7 , yang menghasilkan $S_{11} = dcf$, dan $S_{12} = edf$. Penempelan terakhir adalah bd pada S_6 , yang menghasilkan $S_{13} = देb$ dan $S_{14} = dbc$.

Dari proses di atas kita peroleh muka-muka berikut yang membentuk graf planar maksimal sebagai penyelesaian:

(b,e,a) , (c,b,a) , (c,a,f) , (a,e,f) , (d,c,f) , (e,d,f) , (d,e,b) , and (d,b,c) .

Graf planar maksimal ini berbobot total 759, dan dapat digambarkan sebagai berikut:



Gambar 5. Penyelesaian akhir dengan Algoritma Pemrosesan String.

Referensi

- Boswell, S. G. (1992). TESSA-A new greedy heuristic for facilities layout planning. *International Journal of Production Research*, **30**, 1957-1968.
- Burkard, R. E. (1984). Some recent advances in quadratic assignment problems. *Mathematical Programming*, edited by R. W. Cottle *et al.* (Amsterdam: North Holland).
- Eades, P., Foulds L. R, and Giffin, J. W. (1982). An efficient heuristic for identifying a maximum Weight Planar Subgraph. In *Lecture Notes in Mathematics No. 952* (Combinatorial Mathematics IX). Springer-Verlag, Berlin.
- Foulds, L. R., and Robinson, D. F. (1978). Graph theoretic heuristic for the plant layout problem. *International Journal of Production Research*, **16**, 27-37.
- Foulds, L. R., Gibbons, P.B., and Giffin, J. W. (1985). Facilities Layout Adjacency Determination: An Experimental Comparison of Three Graph Theoretic Heuristics. *Operations Research*, **33**, 1091-1106.
- Gilmore, P. C. (1962). Optimal and Suboptimal algorithms for the quadratic assignment problem. *Journal of the Society for Industrial and Applied Mathematics*, **10**, 305-313.
- Green, R. H., and Al-Hakim, L. (1985). A heuristic for facility layout planning. *Omega*, **13**, 469.
- Kaku, B. K., and Thompson, G. L. (1986). An exact algorithm for the general quadratic assignment problem. *European Journal of Operational Research*, **23**, 382-390.
- Kim, J. Y., and Kim, Y. D. (1995). Graph Theoretic Heuristics for Unequal-sized Facility Layout Problems. *Omega, International Journal of Management Science*, **23**, 391-401.
- Koopmans, T. C., and Beckman, M. (1957). Assignment problems and the location of economic activities. *Econometrica*, **25**, 53-76.
- Kusiak, A., and Heragu, S. S. (1987). Invited Review. The Facility layout problem. *European Journal of Operational Research*, **29(3)**. 229-251.
- Lawler, E. L. (1963) The quadratic assignment problem. *Management Science*, **9(4)**, 586-599.
- Leung, J. (1992). A new graph-theoretic heuristic for the facility layout. *Management Science*, **38(4)**, 594-605.
- Moore, J. M. (1976). Facilities design with graph theory and strings, *Omega, The International Journal of Management Science*, **4(2)**, 193-203.
- Seppanen, J., and Moore, J. M. (1970). Facilities Planning with graph theory. *Management Science*, **17(4)**, B242-B253.