

ANALISA ALGORITMA LAYOUT HEURISTIK UNTUK MEMINIMASI TOTAL MATERIAL HANDLING DALAM PERENCANAAN TATA LETAK FASILITAS

Annisa Kesy G

ABSTRACT

This research has goal to compare there alternative layout performances : CRAFT, MULTIPLE, and BLOCPAN. Using the problem of layout which include forward flow material and fordward backward flow material can be found result : Multiple Algorithm has total distance and material cost minimally but the problems that have lower departemen, CRAFT algorithm can be used better than MULTIPLE algorithm because almost gives the same result with easier input process.

Key Words : Heuristic, multiple algoritm, layout

PENDAHULUAN

Pengaturan fasilitas produksi , stasiun kerja serta alokasi sumber daya lainnya di dalam suatu pabrik memegang peranan penting dalam mewujudkan suatu aliran kerja dan kondisi kerja yang teratur, aman, dan nyaman. Pada umumnya pengaturan tata letak fasilitas yang terencana dengan baik secara tidak langsung akan turut menjaga kelangsungan hidup dan kesuksesan kerja dalam suatu industri. Mengingat arti pentingnya pengaturan tata letak fasilitas tersebut, maka bagi sebuah pabrik yang akan berdiri atau melakukan ekspansi dan atau melakukan renovasi perlu melakukan design “Tata Letak Fasilitas Produksi/Tata Letak Pabrik”.

Pada dasarnya, proses design layout akan melibatkan penetapan hubungan keterkaitan antara departemen-departemen atau antara stasiun kerja-stasiun kerja dan

suatu evaluasi menyeluruh dari hubungan tersebut. Hubungan keterkaitan antara pasangan-pasangan departemen yang ada ini bisa dalam bentuk hubungan kualitatif yang diwujudkan dalam derajat kedekatan dengan skala A, E, I, O, U, X dan hubungan kuantitatif yang diwujudkan dengan total aliran bahan (material handling) dalam bentuk from to chart. Selanjutnya dilakukan evaluasi terhadap hubungan-hubungan tersebut untuk mendapatkan alternatif-alternatif tata letak / layout.

Pada cara tradisional, dengan menggunakan metode trial error (secara analisa grafis) akan dilakukan pembentukan beberapa alternatif layout dan selanjutnya dari masing-masing akan dihitung besarnya total material handling yang ada. Metode ini mudah dilaksanakan untuk layout baru yang melibatkan departemen yang sedikit. Namun untuk relayout atau layout yang melibatkan banyak departemen akan menjadi rumit

terutama pada proses analisa grafisnya. Permasalahan ini mendorong dikembangkan model-model penyelesaian layout, diharapkan dengan menggunakan model ini permasalahan yang dihadapi model tradisional dalam relayout yang melibatkan banyak departemen dapat diselesaikan dalam waktu yang lebih singkat dan hasil alternatif layout yang lebih baik. Selanjutnya untuk menyelesaikan model-model permasalahan layout tersebut, telah dikembangkan algoritma-algoritma sebagai suatu prosedur untuk mendapatkan solusi bagi model yang ada.

Untuk dapat memilih dan menentukan algoritma mana yang tepat dalam membantu penyelesaian permasalahan layout perlu dikembangkan metode analisis guna mengetahui performansi masing-masing algoritma yang ada, sehingga akan didapatkan tata letak fasilitas/plant layout yang memberikan total material handling yang paling minimal serta pertimbangan kelebihan dan kekurangan untuk tiap metode. Penelitian ini mencoba membandingkan beberapa algoritma sub optimal diantaranya MULTIPLE, CRAFT, dan BLOCPAN.

TINJAUAN PUSTAKA

Pengertian Tata Letak Fasilitas Pabrik (Plant Layout)

Tata letak pabrik atau tata letak fasilitas definisikan sebagai tata cara pengaturan fasilitas-fasilitas pabrik guna menunjang kelancaran proses produksi.

Pengaturan tersebut akan mencoba memanfaatkan luas area untuk penempatan mesin atau fasilitas penunjang produksi lainnya, kelancaran gerakan perpindahan material, penyimpanan material baik yang bersifat temporer maupun permanen, personel pekerja dan sebagainya.

Pendekatan - Pendekatan Untuk Menyelesaikan Permasalahan Tata Letak Fasilitas.

Sampai saat ini ada dua pendapat utama mengenai karakteristik dari permasalahan tata letak fasilitas yaitu pendapat pertama yang menyatakan bahwa problem layout sebagai problem desain, sedangkan pendapat kedua menyatakan sebagai problem optimasi.

1. Problem Layout sebagai Problem Desain.

Alasan utama mengapa problem layout dimasukkan sebagai problem desain, karena para desainer yang terlibat dalam merancang solusi layout lebih cenderung untuk mendapatkan sebuah solusi yang memenuhi kondisi dan batasan-batasan yang ada dalam pabrik yang akan diatur layoutnya daripada sebuah solusi layout yang terbaik. Karena termasuk dalam problem desain, desainer atau analis layout akan cenderung untuk menggunakan pendapat pribadinya dalam mengembangkan prosedur atau metode dalam menyelesaikan permasalahan tata letak fasilitas. Beberapa pendekatan yang telah dikembangkan untuk menyelesaikan problem layout dengan

mendasarkan bahwa problem layout adalah problem desain, diantaranya adalah :

- Systematic Layout planning yang dikembangkan oleh Muther.
- Apple's Plant Layout Procedure.
- Reed's Plant Layout Procedure.

2. Problem Layout sebagai Problem Optimasi.

Hal ini disebabkan karena dalam problem layout fasilitas selalu diupayakan untuk mendapatkan solusi yang terbaik. Jika problem layout dikategorikan sebagai problem optimasi, maka langkah pertama yang harus dilakukan untuk menyelesaikan problem layout adalah membangun model. Langkah selanjutnya adalah mengembangkan teknik solusi atau algoritma yang sesuai untuk mendapatkan solusi yang terbaik dari model yang dibuat.

Sampai saat ini terdapat beberapa model yang telah dikembangkan untuk menyelesaikan problem layout diantaranya adalah QAP (Quadratic Assignment Problem), ABS MODELS, dan LMIP (Linear Mixed Integer Models). Dan model yang paling banyak digunakan sampai saat ini adalah QAP yang dikembangkan oleh Koopmans dan Beckmann pada tahun 1957.

Untuk mendapatkan solusi dari model-model diatas, telah banyak algoritma atau pendekatan yang dikembangkan. Pada dasarnya, algoritma-algoritma tersebut dapat diklasifikasikan sebagai :

1. Algoritma Optimal

Dengan menggunakan algoritma optimal untuk menyelesaikan model layout

akan diperoleh setidaknya satu solusi rancangan layout yang terbaik. Beberapa algoritma optimal yang telah dikembangkan untuk menyelesaikan model QAP untuk problem layout adalah Branch and Bound , Bender's Decomposition , dan Cutting Plane .

Algoritma optimal ini meskipun dapat menghasilkan sedikitnya satu solusi yang terbaik, memiliki beberapa kelemahan diantaranya yaitu waktu komputasi dan memory yang dibutuhkan akan semakin besar bila ukuran problemnya semakin besar sehingga algoritma ini hanya dapat menghasilkan solusi optimal untuk problem-problem dengan ukuran kecil dimana jumlah departemen kurang dari atau sama dengan 15 (Sunderesh Heragu, 1997).

2. Algoritma Sub Optimal atau Heuristik.

Dengan adanya kelemahan dari algoritma optimal dalam hal waktu komputasi dan memori yang dibutuhkan, hal ini mendorong peneliti untuk mengembangkan algoritma-algoritma sub optimal. Pertimbangan lain mengapa algoritma sub optimal dikembangkan dan akhirnya lebih banyak digunakan adalah : pertama, karena problem layout sangat kompleks sedangkan algoritma optimal yang efisien dalam menyelesaikan problem dengan ukuran besar nampaknya tidak ada dan kedua, model yang dikembangkan saat ini adalah formulasi tidak sempurna dari problem real sehingga tidak terlalu perlu menggunakan solusi optimal untuk model yang tidak sempurna. Dalam perkembangannya, algoritma-algoritma sub

optimal ini kemudian dibagi menjadi empat yaitu (Heragu S.S dan Kusiak A., 1987) :

- *Algoritma Konstruksi*, algoritma ini berawal dari masalah dasar untuk membangun solusi dalam cara iterasi. Awalnya, tak sebuah fasilitas pun yang diletakkan pada satu lokasi. Pada langkah selanjutnya dalam proses konstruksi, akan didapat satu set fasilitas yang telah ditugaskan pada lokasi-lokasi. Algoritma-algoritma yang termasuk dalam algoritma konstruksi adalah HC 66, ALDEP, CORELAP, FLAT, FATE, RMA, dan MAT.
- *Algoritma Perbaikan*, algoritma ini pada dasarnya memodifikasi solusi awal yang merupakan input data hingga biaya material handling dapat dikurangi. Dalam algoritma perbaikan, selalu ada solusi awal yang dipilih secara random untuk kemudian dilakukan pertukaran di antara fasilitas-fasilitas dari solusi awal tersebut hingga didapat solusi akhir yang optimal. Algoritma yang termasuk : CRAFT, COFAD, MULTIPLE, MCRAFT, LOGIC, dan FRAT.
- *Algoritma Hybrid*, algoritma ini pada dasarnya mengklasifikasikan algoritma-algoritma yang memiliki karakteristik optimal dan sub optimal, atau algoritma yang memiliki karakteristik sebagai algoritma konstruksi dan juga sebagai algoritma perbaikan. Algoritma yang termasuk : DISCON, FLAC, BLOCPLAN, dan

algoritma heuristik Burkard dan Stratman yang menggunakan konsep algoritma branch and bound dan algoritma perbaikan.

- *Graph Teoritic Algorithm*.

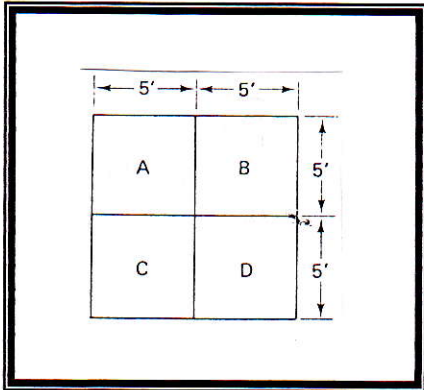
METODOLOGI PENELITIAN

Setelah dirumuskan sebuah tujuan penelitian dan studi literatur terhadap ketiga algoritma yang akan dibandingkan, diperoleh kesimpulan bahwa harus ada software atau program agar ketiga algoritma tersebut memberikan hasil layout maupun biaya material handling. Maka langkah awal yang dilakukan pada penelitian ini adalah mencari atau membuat program. Dari ketiga algoritma tersebut, algoritma CRAFT ternyata telah digunakan pada menu Facility Layout di software Quantitative System,

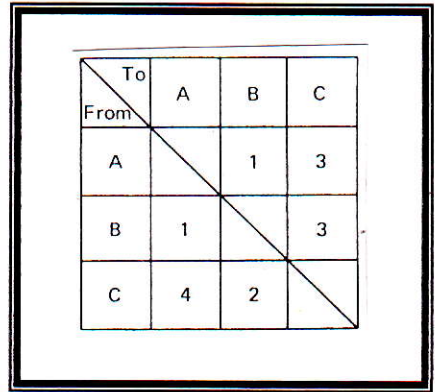
Selanjutnya dilakukan pengumpulan data yang akan digunakan sebagai input dari program meliputi jumlah departemen/fasilitas, luasan masing-masing departemen, bentuk dari departemen, from to chart/matriks aliran antar departemen serta SFC (Space Filling Curve) yang digunakan khusus pada algoritma MULTIPLE. Input tersebut akan dibuat dalam beberapa skenario.

Dalam penelitian ini menggunakan 7 problem layout yang dapat dilihat sebagai berikut :

1. Problem 1

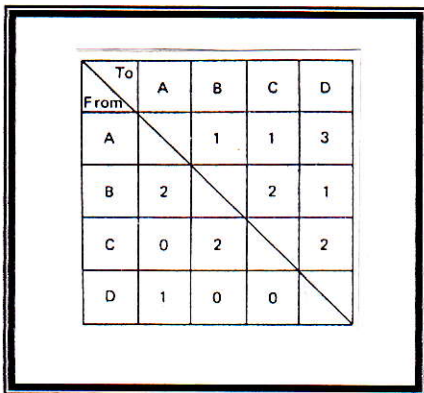


Gambar 1
Initial layout P-1

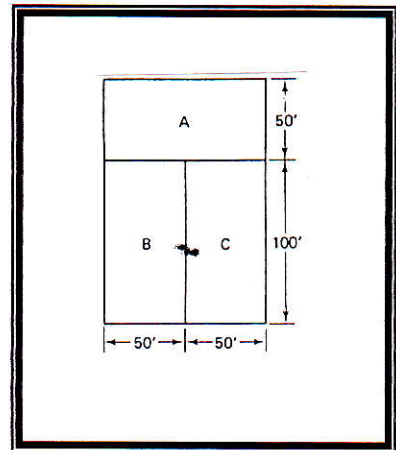


Gambar 4
Matriks Aliran P-2

3. Problem 3

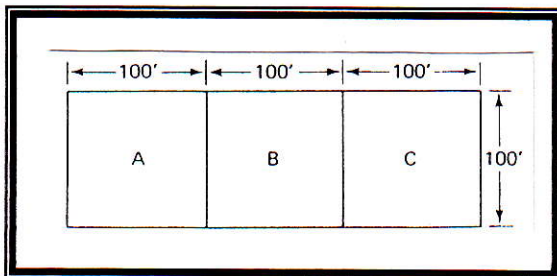


Gambar 2
Matriks Aliran P-1

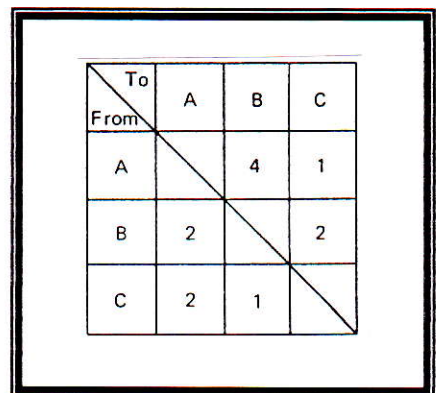


Gambar 5
Initial Layout P-3

2. Problem 2

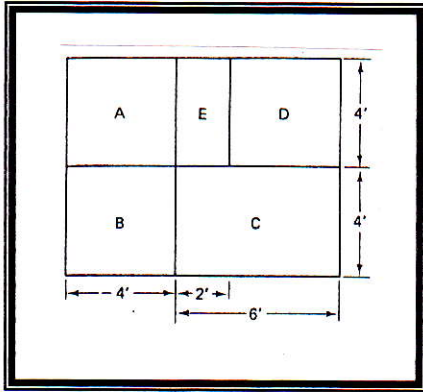


Gambar 3
Initial Layout P-2



Gambar 6
Matriks Aliran P-3

4. Problem 4



Gambar 7
Initial Layout P-4

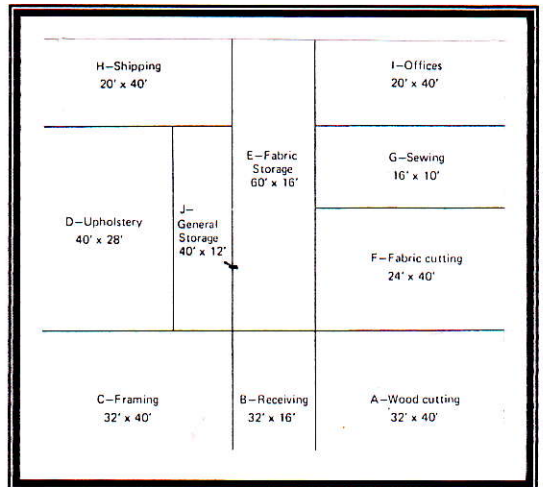
	A	B	C	D	E	F	G	H	I
A		15	1	3	2	4	3	2	2
B	10		2	5	3	15	20	30	9
C	1	4		40	6	1	14	5	6
D	4	8	25		10	5	15	7	10
E	6	5	2	20		1	20	2	5
F	7	40	1	3	1		20	10	0
G	2	8	4	25	15	4			15
H	5	30	6	3	2	30	5		15
I	0	0	0	0	0	0	0	0	0

Gambar 10
Matriks Aliran P-5

6. Problem 6

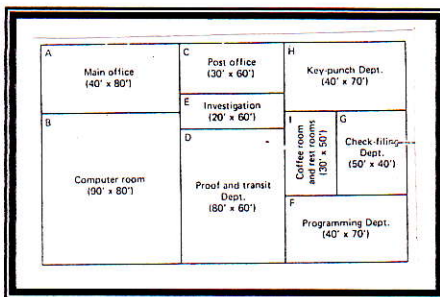
To	From	A	B	C	D	E
A			1	2	2	0
B		0		2	0	0
C		2	0		0	0
D		3	0	1		0
E		0	0	0	0	

Gambar 8
Matriks Aliran P-4



Gambar 11
Initial Layout P-6

5. Problem 5

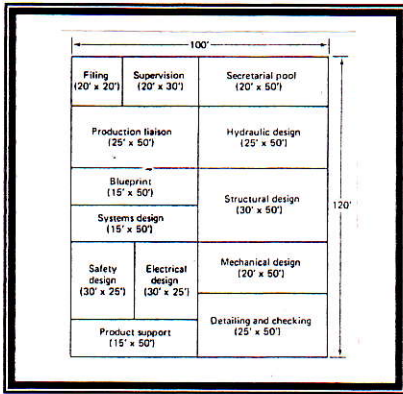


Gambar 9
Initial Layout P-5

	A	B	C	D	E	F	G	H	I	J
A		2	1							2
B	6				6	2				2
C	1	2		4					2	
D	1	1	4			4	2	3	2	2
E						5	3		2	
F				4	5		4		2	
G				2	2	4			2	
H					3					
I	1	1	1	1	1	1	1	1		1
J				2			1	1	2	

Gambar 12
Matriks Aliran P-6

7. Problem 7



Gambar 13
Initial Layout P-7

To From	01	02	03	04	05	06	07	08	09	10	11	12	13
01	-	15	0	0	0	5	10	15	10	10	0	15	
02	40	-	25	40	100	90	80	70	40	50	160	85	60
03	0	20	-	0	0	0	0	0	0	0	0	0	0
04	10	80	0	-	0	0	20	30	210	100	280	0	10
05	50	120	600	0	-	40	0	60	0	50	0	340	50
06	5	90	350	0	40	-	0	20	60	40	160	270	60
07	10	80	400	40	20	0	-	30	0	10	140	320	45
08	10	70	350	30	10	10	60	-	10	20	150	310	40
09	15	110	215	60	10	15	10	20	-	60	210	180	110
10	30	85	75	85	120	115	105	135	60	-	180	150	105
11	30	160	0	450	160	350	380	300	400	100	-	0	680
12	0	300	700	40	550	510	400	410	250	370	0	-	20
13	20	90	0	10	50	60	50	30	410	320	680	30	-

Gambar 14
Matriks Aliran P-7

Sedangkan untuk input SFC dibantu dengan membuat sebuah program.

Setelah diperoleh solusi layout untuk masing-masing skenario dilakukan analisa perbandingan antara hasil yang diperoleh dari masing-masing program. Dimana perbandingan akan ditinjau dari total material handling yang akan didapat dengan bentuk layout tersebut, CPU second, serta total iterasi yang dilakukan.

HASIL DAN PEMBAHASAN

Dari input yang telah dimasukkan, selanjutnya dilakukan running pada masing-masing program baik CRAFT, MULTIPLE MAUPUN BLOCPLAN. Selanjutnya pengolahan data ini dibagi menjadi dua bagian yaitu : **Dengan ada gerakan balik/backward**, dimana disini yang akan dibandingkan adalah hasil dari program MULTIPLE dan CRAFT (dikarenakan program BLOCPLAN hanya memungkinkan untuk aliran material forward/maju saja), sedangkan yang kedua adalah **Tanpa ada gerakan balik**, dimana perbandingan akan dilakukan pada ketiga program yang ada.

Dari hasil pengolahan data yang telah dilakukan diperoleh final layout untuk masing-masing problem beserta biaya material handlingnya. Untuk menghitung biaya material handling disini digunakan asumsi bahwa biaya material handling antar departemen baik untuk gerakan forward maupun backward adalah sama dengan 1 dan jarak antar departemen dihitung secara rectilinear.

Secara lengkap biaya material handling initial dan final layout, jumlah iterasi, dan CPU second yang diperoleh dari ketiga program yang ada untuk 7 problem dengan asumsi tanpa ada gerakan balik dapat dilihat pada tabel berikut :

Tabel 1
Hasil Iterasi 1 dari Ketujuh Problem dengan Blocplan,

Craft dan Multiple

Problem	CRAFT	MULTIPLE	BLOCPLAN
1			
- Biaya initial layout	- $30 \times 2,5 = 75$	- 75	- 75
- Biaya final layout	- $24 \times 2,5 = 60$	- 60	- 58,3
- Jumlah Iterasi	- 1	- 1	- 6
- CPU second	- 2,359	- 2,1	- 4,3
2			
- Biaya initial layout	- $20 \times 50 = 1000$	- 1000	- 1000
- Biaya final layout	- $16 \times 50 = 800$	- 800	- 823,8
- Jumlah Iterasi	- 1	- 1	- 4
- CPU second	- 3,129	- 2,1	- 3,5
3			
- Biaya initial layout	- $24 \times 25 = 600$	- 600	- 600
- Biaya final layout	- $20 \times 25 = 500$	- 400	- 535,8
- Jumlah Iterasi	- 1	- 1	- 4
- CPU second	- 2,691	- 2,1	- 3,5
4			
- Biaya initial layout	- 44	- 44	- 44
- Biaya final layout	- 26,625	- 31,67	- 38,4
- Jumlah Iterasi	- 2	- 3	- 10
- CPU second	- 3,57	- 4,5	- 9,6
5			
- Biaya initial layout	- $3447 \times 10 = 34470$	- 34470	- 34470
- Biaya final layout	- 28225,03	- 26027,4	- 28263
- Jumlah Iterasi	- 11,367	- 5	- 20
- CPU second		- 7,7	- 21,3
6			
- Biaya initial layout	- $674 \times 4 = 2697,6$	- 2697,6	- 2697,6
- Biaya final layout	- 2485,4	- 2431,9	- 2505,9
- Jumlah Iterasi	- 3	- 5	- 18
- CPU second	- 8,839	- 8,6	- 22,05
7			
- Biaya initial layout	- $79862,5 \times 5 = 399312,5$	- 399312,5	- 399312,5
- Biaya final layout	- 305545	- 289370,9	- 3060555
- Jumlah Iterasi	- 4	- 11	- 20
- CPU second	- 7,309	- 11,56	- 27,4

Sedangkan hasil layout untuk ketujuh problem dengan menggunakan data aliran bahan dengan arah maju dan mundur dapat dilihat pada tabel berikut :

Tabel 2
Hasil Iterasi 2 dari Ketujuh Problem dengan
Craft dan Multiple

Problem	Program	Biaya Initial Layout	Biaya Final Layout	Jumlah Iterasi	CPU Second
1	- Multiple	- 115	- 85	- 1	- 2,5
	- Craft	- $46 \times 2,5 = 115$	- $34 \times 2,5 = 85$	- 1	- 3,082
2	- Multiple	- 2100	- 1600	- 1	- 2,6
	- Craft	- $42 \times 50 = 2100$	- $32 \times 50 = 1600$	- 1	- 3,621
3	- Multiple	- 1050	- 750	- 1	- 2,5
	- Craft	- $42 \times 25 = 1050$	- $36 \times 25 = 900$	- 1	- 2,3
4	- Multiple	-	- 53,67	- 1	- 3,3
	- Craft	-	- 56,2	- 2	- 4
5	- Multiple	- 66680	- 49686,9	- 7	- 8,5
	- Craft	- $6668 \times 10 = 66680$	- 59151,15	- 6	- 6,648
6	- Multiple	- 5070	- 4618,4	- 10	- 13,5
	- Craft	- $1267,5 \times 4 = 5070$	- $1127,12 \times 4 = 4508,5$	- 3	- 6,98
7	- Multiple	- 1239437,5	- 957018,2	- 10	- 16,8
	- Craft	- $247887,5 \times 5 = 1239437,5$	- 985050	- 3	- 6,98

Dari tabel diatas dapat dilihat bahwa hampir secara keseluruhan layout yang menghasilkan biaya material handling yang paling minimal dihasilkan oleh algoritma MULTIPLE baik mulai problem 1 sampai 7. Sedangkan algoritma CRAFT menghasilkan solusi terbaik jika permasalahan hanya melibatkan **beberapa departemen saja** (kurang dari 10) hal ini dapat dibuktikan bahwa pada problem 5 dengan banyak departemen 9, problem 6 dengan banyak departemen 10 dan problem 7 dengan banyak departemen 13, hasil final layout Craft hampir selalu lebih besar dari pada hasil dengan Multiple atau bila **luasan antar departemen yang dipertukarkan banyak yang sama** (sebagai contoh pada problem 1 dan 2 tiap departemen mempunyai luasan yang sama dan hasil biaya final layout yang dihasilkan Craft dan Multiple sama, sedangkan pada problem 3 dimana ada 3 departemen yang memiliki luasan sama tetapi hasil yang diperoleh dengan CRAFT lebih besar dibanding MULTIPLE dikarenakan disini digunakan 2 bentuk SFC yang masing-masing memberi final layout (1) 500 dan 2(400)).

Kelebihan yang paling baik dari Craft adalah kemampuan untuk menangkap initial layout dengan sebaik-baiknya sesuai dengan layout yang sesungguhnya karena matriks untuk initial layout sudah tersedia. Dalam tabel 5.1 dan 5.2 diatas semua biaya material handling initial layout dihitung dari running pada program Craft. Blocplan tidak dapat menangkap initial layout yang sesungguhnya dikarenakan input yang

dimasukkan hanya luas departemen dan blocplan akan merandom sendiri initial layoutnya sehingga Blocplan sangat cocok digunakan sebagai alat bantu untuk membangun sebuah layout baru, demikian pula dengan Multiple dimana pada program Multiple, peletakan posisi departemen dan pertukaran antar departemen sangat tergantung pada SFC yang digunakan, sehingga SFC yang berbeda akan menghasilkan bentuk dan biaya final layout yang berbeda. Sebagai contoh pada problem 1 dimana urutan SFC yang digunakan adalah 1 2 6 5 11 13 14 10 11 15 16 12 8 7 3 4 dengan urutan departemen yang masuk adalah $3 \rightarrow 1 \rightarrow 2 \rightarrow 4$, maka departemen 3 dengan luas area 4 akan menempati grid 1 2 6 5, diikuti departemen 1 dengan luas area 4 akan menempati grid 11 13 14 10, demikian seterusnya dan ternyata peletakan posisi departemen sesuai dengan SFC diatas sama dengan initial layout pada problem 1 sehingga biaya material handling pada problem 1 dengan Multiple sama dengan yang dihasilkan Craft. Sedangkan contoh yang lain pada problem 4 dengan urutan departemen 2 1 5 3 4 dan SFC yang diberikan diperoleh initial layout sebagai berikut :

1	1	5	3	4
1	1	5	3	4
2	2	3	3	4
2	2	3	3	4

Gambar 15
Initial Layout yang Dihasilkan Multiple pada Problem 4

1	1	5	4	4
1	1	5	4	4
2	2	3	3	3
2	2	3	3	3

Gambar 16
Initial Layout yang Sesungguhnya
pada Problem 4

Dari gambar diatas dapat dilihat bahwa initial layout yang dihasilkan dengan Multiple ternyata agak berbeda dengan initial layout yang sebenarnya, sehingga kemampuan Multiple dalam menangkap initial layout akan sangat tergantung pada SFC yang digunakan dan sering kali akan menghasilkan bentuk dan biaya final layout yang berbeda. Dan sebagai akibatnya dalam problem 4 tanpa gerakan balik diperoleh final solusi Craft (26,625) ternyata lebih baik dibanding Multiple (31,67), salah satu hal yang paling mempengaruhi adalah karena multiple tidak mampu menangkap initial layout diatas dengan baik dan menghasilkan biaya awal yang lebih besar yaitu sebesar 47,33 sedangkan biaya material handling initial layout hanya sebesar 44. Sehingga untuk mendapatkan solusi yang lebih baik pada Multiple mungkin dapat dicoba beberapa alternatif urutan SFC dan urutan departemen yang berbeda.

Secara keseluruhan untuk semua problem, Blocplan kurang dapat memberi solusi layout yang lebih baik kecuali untuk problem 1 dan 2. Untuk problem 1 dan 2, Blocplan mampu memberikan solusi yang

lebih baik dibanding yang lain. Namun dapat dilihat bahwa final layout yang dihasilkan benar-benar berbeda dengan initial layout yang ada dimana terjadi perubahan letak dan ukuran masing-masing departemen yang drastis (lain dengan Craft dan Multiple , dimana perubahan hanya cenderung pada letaknya sedang ukurannya tidak akan jauh berbeda), sehingga bila sifatnya relayout maka Blocplan akan sangat kurang cocok untuk digunakan karena ukuran departemen banyak berubah. Sehingga untuk relayout, craft dan Multiple lebih cocok untuk digunakan. Untuk mendapatkan solusi yang baik pada blocplan mungkin dapat dicoba dengan beberapa kali random layout dan dicari yang nilainya paling baik baru kemudian dilakukan improvement.

Dari tabel 5.1 dan 5.2, dapat disimpulkan dengan makin bertambahnya banyak departemen dalam sebuah layout dimana memungkinkan pertukaran yang semakin banyak maka jumlah iterasi akan bertambah dan secara otomatis akan menambah waktu CPU second. Untuk kondisi sekarang, jumlah iterasi dan CPU second sudah kurang akurat untuk dijadikan patokan dikarenakan komputer yang ada sekarang sudah sangat canggih sehingga untuk problem diatas (misal problem 7 dengan banyak departemen 13 maksimal hanya membutuhkan CPU second sebesar 22 detik). Dari tabel diatas dapat dilihat bahwa antara Blocplan, Multiple dan Craft mempunyai waktu untuk running program yang hampir sama/tidak terlalu jauh berbeda. Sehingga dari sini, jika problem

layout yang ada memiliki jumlah departemen yang sedikit atau departemen dengan luasan area yang hampir sama maka gunakan algoritma Craft atau Blocplan saja karena akan menghasilkan solusi yang baik dengan waktu dan cara memasukkan input yang cepat dan mudah dibanding dengan Multiple, namun bila jumlah departemen semakin banyak maka akan lebih baik untuk mendapatkan solusi yang lebih optimal dengan menggunakan Multiple.

KESIMPULAN

Dari analisa hasil dapat disimpulkan beberapa hal yaitu :

1. Algoritma Multiple mampu menghasilkan solusi yang cukup optimal untuk hampir semua problem dibandingkan dengan Algoritma Craft maupun Blocplan dikarenakan SFC yang digunakan untuk mempertukarkan posisi antar departemen.
2. Kelemahan yang dimiliki oleh algoritma Multiple adalah kemampuan untuk menangkap initial layout kurang baik, sehingga untuk menghitung biaya initial layout yang sesungguhnya dapat menggunakan algoritma Craft karena pada programnya sudah tersedia menu matriks initial layout.
3. Blocplan lebih cocok digunakan sebagai algoritma konstruksi/membuat sebuah layout baru daripada sebagai algoritma improvement karena mempunyai kemampuan untuk merandom layout dengan hanya berdasarkan pada luas departemen. Sedangkan Craft dan Multiple selalu membutuhkan initial layout dan terbukti solusi layoutnya jauh lebih baik dibanding dengan Blocplan sehingga cocok sebagai algoritma improvement.
4. Untuk problem dengan sedikit departemen atau banyak departemen dengan luasan antar departemen hampir sama sebaiknya cukup menggunakan algoritma Craft atau Blocplan karena proses penginputan data jauh lebih cepat dibanding dengan Multiple dan hasil solusi yang diperoleh juga baik.

DAFTAR PUSTAKA

- Apple James M., 1977, **Plant Layout and Material Handling**, edisi ketiga, John Wiley & Sons.
- James A. Tompkins and John A. White, 1996, **Facilities Planning**, edisi kedua, John Wiley & Sons, Canada.
- James M. Moore, 1962, **Plant Layout and Design**, New York
- Kusiak & S.S. Heragu, 1987, **“The Facility Layout Problem”**, European Journal of Operation Research, vol.. 29.
- Richards L. Francis and John A. White, 1974, **Facility Layout and Location An Analytical Approach**, Prentice Hall International, New Jersey.
- Sritomo Wignjosoebroto, 1996 , **“ Tata Letak Pabrik dan Pemindahan Bahan”**, edisi ketiga, PT. Guna Widya, Jakarta.