

Perancangan Tata Letak Fasilitas Pada PT. Lima Jaya

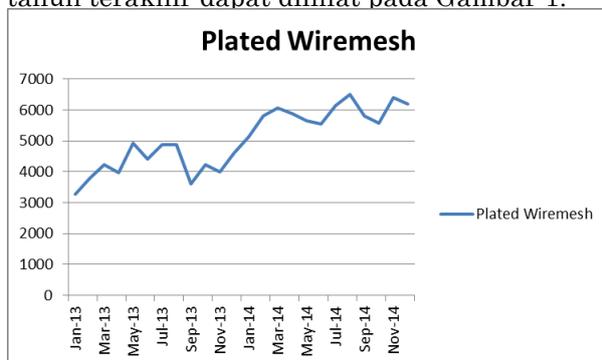
Jonathan Wijaya Gunawan¹, Tanti Octavia², Felecia³

Abstract: PT.Lima Jaya is a company that engaged in the field of steel and metalworks. Currently, this company wants to build a new company that specialized in wiremesh production. Therefore, this research tries to design their facility layout for minimizing moment. Three algorithms are applied in order to get the best result. The algorithms are corelap algorithm, aldep algorithm, and blocplan algorithm. The result shows that the blocplan algorithm gives the best result with the moment 13579,4.

Keywords: Facility Layout Design, Corelap Algorithm, Aldep Algorithm, Blocplan Algorithm

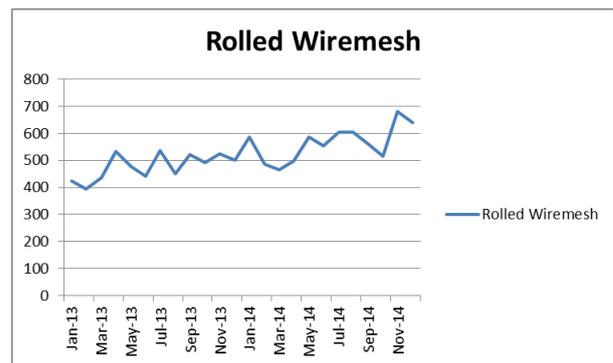
Pendahuluan

PT. Lima Jaya adalah perusahaan yang bergerak di bidang pembuatan besi beton dimana produk yang di produksi adalah besi *wiremesh*. Produk utama dari perusahaan adalah *rolled wiremesh* dan *plated wiremesh*. *Plated wiremesh* adalah *wiremesh* yang berbentuk lembaran sedangkan *rolled wiremesh* adalah *wiremesh* yang berbentuk roll. Saat ini, PT.Lima Jaya belum melakukan produksi *wiremesh* sendiri. Selama ini, perusahaan mendapatkan *wiremesh* dari perusahaan lain. Permintaan konsumen seringkali melebihi dari kapasitas yang dapat disediakan oleh perusahaan yang melakukan produksi *wiremesh* sehingga PT.Lima Jaya ingin melakukan produksi *wiremesh* sendiri. Data penjualan *plated wiremesh* dalam dua tahun terakhir dapat dilihat pada Gambar 1.



Gambar 1. Data Penjualan *Plated Wiremesh*

dengan penjualan dari *rolled wiremesh*. Data penjualan *rolled wiremesh* dalam dua tahun terakhir dapat dilihat pada Gambar 2.



Gambar 2. Data Penjualan *Rolled Wiremesh*

Perusahaan saat ini sudah menentukan lokasi untuk produksi pada Jalan Kedamean KM7, Gresik. Mesin-mesin untuk keperluan produksi pun sudah disurvei. Seiring dengan pembuatan lokasi baru, perusahaan menginginkan perancangan tata letak fasilitas yang baik dan tentunya dapat meminimumkan momen.

Metode Penelitian

Metode penelitian yang digunakan untuk menyelesaikan permasalahan yang ada dibahas pada bab ini. Tata letak fasilitas dibentuk menggunakan suatu algoritma. Algoritma memiliki beberapa macam jenis dan kegunaannya.

Jenis Algoritma

Algoritma dapat diklasifikasikan menjadi dua jenis algoritma yang ada yaitu algoritma optimal dan algoritma heuristik. Tabel 1 membandingkan kelebihan dan kekurangan dari algoritma optimal dan algoritma heuristik

^{1,2} Fakultas Teknologi Industri, Program Studi Teknik Industri, Universitas Kristen Petra. Jl. Siwalankerto 121-131, Surabaya 60236. Email: jonathanwgunawan@hotmail.com, tanti@petra.ac.id, felecia@petra.ac.id

Penjualan dari *plated wiremesh* meningkat pesat dalam dua tahun terakhir, begitu juga

Tabel 1. Perbandingan Algoritma Optimal dan Algoritma Heuristik

Algoritma	Kelebihan	Kekurangan
Optimal	Memberikan solusi terbaik	Perhitungan memakan waktu yang lama Kapasitas jumlah fasilitas sedikit
Heuristik	Memberikan banyak solusi Waktu perhitungan lebih cepat Banyak jenis algoritma heuristik	Solusi yang dihasilkan tidak selalu paling optimal

Algoritma Menurut Kegunaannya

Ada tiga macam algoritma yang memiliki kegunaan yang berbeda-beda antara lain adalah algoritma *construction*, algoritma *improvement*, dan yang terakhir adalah algoritma *hybrid*. Algoritma *construction* adalah algoritma yang digunakan untuk membuat perancangan tata letak fasilitas yang benar benar baru, jadi pembuatannya dimulai dari *layout* kosong yang ada dan diisi dengan fasilitas-fasilitas yang diperlukan. Algoritma *improvement* adalah Algoritma yang digunakan untuk membuat perancangan tata letak fasilitas dengan *layout* yang sudah ada dan ingin diperbarui. Algoritma *hybrid* adalah algoritma yang bisa digunakan untuk perancangan tata letak fasilitas yang benar-benar baru maupun untuk *improvement* dari *layout* yang sudah ada.

Algoritma yang dipilih untuk digunakan dalam makalah ini adalah algoritma *construction*. Tiga algoritma digunakan untuk membandingkan hasil terbaik yang dapat dihasilkan. Tiga algoritma yang digunakan kali ini adalah algoritma *corelap*, algoritma *aldep*, dan algoritma *blocplan*.

Algoritma Corelap

Algoritma *corelap* adalah salah satu algoritma *construction* yang dikembangkan oleh Lee dan Moore pada tahun 1967 yang mengubah data kualitatif menjadi data kuantitatif untuk menentukan fasilitas pertama untuk diletakkan didalam *layout* yang ada (Heragu, [1]). Dalam penggunaan algoritma ini, ada penentuan hubungan kedekatan antar fasilitas dengan tanda A, E, I, O, U, dan X. Setiap huruf memiliki nilai sendiri, dimana A memiliki nilai terbesar hingga X yang memiliki nilai minus. Langkah pertama yang dilakukan dalam pengerjaan algoritma *corelap* adalah perhitungan nilai *Total Closeness Rating*. Fasilitas yang memiliki nilai *Total Closeness Rating* tertinggi akan menjadi fasilitas pusat dalam lokasi yang tersedia. Penempatan fasilitas selanjutnya dilakukan sesuai

nilai kedekatan antar fasilitas yang ada dengan fasilitas pusat. Terakhir dilakukan perhitungan momen melalui *jarak* dan *flow* yang ada.

Algoritma Aldep

Algoritma Aldep sering dikenal juga dengan *Automated Layout Design Program* dimana algoritma ini biasa digunakan untuk melakukan *construct layout*. Data-data yang dibutuhkan oleh algoritma Aldep antara lain adalah ukuran fasilitas, *relationship chart*, dan ukuran lokasi yang ada. Langkah awal yang dilakukan dalam algoritma *aldep* adalah menempatkan sebuah lokasi pada ujung lokasi secara *random*. Fasilitas yang selanjutnya diletakkan secara mengular mengikuti lokasi pertama yang telah diletakkan. Fasilitas dipilih untuk diletakkan sesuai nilai kedekatan dengan fasilitas yang terakhir diletakkan. Terakhir dilakukan perhitungan momen melalui data jarak dan *flow* yang ada.

Algoritma Blocplan

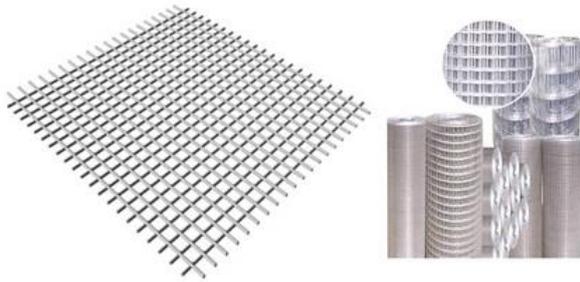
Algoritma *blocplan* adalah suatu algoritma *hybrid* yang dikembangkan oleh Donaghey dan Pire pada tahun 1991 dimana algoritma ini dapat menyelesaikan permasalahan *single story* maupun *multi story layout* (Heragu, [1]). Algoritma ini dapat digunakan untuk perancangan tata letak fasilitas yang sifatnya *construction* maupun *improvement*. Algoritma *blocplan* merupakan algoritma heuristik yang menggunakan data kuantitatif maupun data kualitatif. Ada tiga macam data yang dapat digunakan untuk menyediakan *flow data* yang diperlukan. Pertama secara kualitatif dengan diagram ARC, kedua secara kuantitatif dengan *flow matrix*, dan ketiga dengan informasi jenis dan jumlah produk yang diproduksi dengan urutan proses pembuatan untuk tiap produknya. Apabila pengguna memilih untuk menyediakan data dengan cara kedua atau ketiga, maka *blocplan* akan mengubah *flow matrix* menjadi diagram hubungan.

Algoritma *blocplan* adalah algoritma *hybrid* sehingga saat digunakan untuk melakukan *construction*, algoritma ini membutuhkan suatu *initial layout* dimana *initial layout* bisa diperoleh dari peletakan fasilitas secara random pada lokasi yang tersedia maupun dari iterasi sebelumnya. Pertama dilakukan perhitungan *Rel-dist score*, *upper score*, dan *lower score* dari *initial layout*. *Rscore* dapat dihitung setelah menghitung semua data-data tersebut. *Rscore* yang baik memiliki nilai yang mendekati 1. Proses iterasi dilakukan setelah *Rel-dist score* dan *Rscore* dari *initial layout* diketahui. *Rel-dist score* dan *Rscore* dari hasil iterasi dibandingkan dengan hasil dari *initial layout* agar diketahui bentuk *layout* yang dapat menghasilkan momen paling minimum.

Hasil dan Pembahasan

Sekilas tentang *Plated* dan *Rolled Wiremesh*

Plated wiremesh memiliki bentuk seperti lembaran sedangkan *rolled wiremesh* berbentuk seperti gulungan dari *plated wiremesh*. Bentuk dari *plated wiremesh* dan *rolled wiremesh* dapat dilihat pada Gambar 3.



Gambar 3. *Plated* dan *Rolled Wiremesh*

Proses Produksi

Bahan baku yang digunakan oleh perusahaan adalah besi roll. Proses pembuatan *plated wiremesh* dimulai dengan membawa besi roll dari gudang bahan baku ke mesin *cutting*. Besi roll pertama diletakkan di bagian depan mesin *cutting*. Mesin ini memiliki seperti penarik untuk menarik besi sampai batas maksimal yang ditentukan yaitu 2,1 meter. Setelah itu tuas pemotong akan memotong besi secara otomatis dan besi hasil potong diletakkan di sebelah mesin *cutting*. Besi potong dari mesin *cutting* ini diambil dan dibawa ke mesin *welding*.

Besi roll dari gudang bahan baku juga dibawa ke mesin *welding*. Besi roll ini disusun secara horizontal dengan jarak 15 sentimeter. Setelah ditata, mesin *welding* akan diatur untuk mengelas besi potong sepanjang 2,1 meter hasil dari mesin *cutting* diatas susunan besi roll yang telah ditata sebelumnya. Hasil las ini dibawa ke mesin *plate cutting*. Lembaran besi yang besar itu dipotong kedalam bentuk 2,1 meter x 5,4 meter. Hasil potongan ini dibawa ke gudang barang jadi untuk disimpan.

Proses untuk memproduksi *rolled wiremesh* hampir sama dengan proses produksi *plated wiremesh*. Perbedaan proses terletak pada proses yang dilakukan setelah pengelasan. Berbeda dengan *plated wiremesh*, *rolled wiremesh* dibawa ke mesin *rolling* setelah mengalami pengelasan. Mesin *rolling* ini akan

mengerol besi lembaran tadi sampai menghasilkan *rolled wiremesh* yang berdiameter 70 sentimeter.

Pengukuran Luas Gudang

Cara mencari luas gudang yang dibutuhkan adalah memperhitungkan kapasitas maksimal dan mengalikannya dengan ukuran tiap barangnya. Pertama untuk gudang bahan baku dimana gudang ini hanya memiliki satu jenis barang didalamnya yaitu besi roll yang berukuran 1 meter x 1,15 meter. Panjang besi tiap roll adalah 6741,57 meter. Kapasitas produksi bulanan maksimum dari perusahaan adalah menggunakan 3628800 meter besi sehingga besi roll yang dibutuhkan adalah sejumlah 539 roll. Luas gudang bahan baku yang diperlukan adalah sejumlah $539 \times 1 \text{ m} \times 1,15 \text{ m} = 619,85 \text{ m}^2$. Asumsi sementara ukuran gudang bahan baku adalah $15 \text{ m} \times 41,42 \text{ m} + \text{luas jalan selebar } 2 \text{ meter menjadi } 17 \text{ m} \times 41,42 \text{ m} = 704,14 \text{ m}^2$. Gudang barang jadi memiliki dua jenis barang didalamnya yaitu *plated wiremesh* yang berukuran 5,4 meter x 2,1 meter dan *rolled wiremesh* yang berukuran 0,7 meter x 2,1 meter. Kapasitas maksimal produksi *plated wiremesh* adalah 12000 dan tumpukan maksimum dalam gudangnya adalah 150 sehingga lahan yang diperlukan untuk *plated wiremesh* adalah $12000 \times 5,4 \text{ m} \times 2,1 \text{ m} / 150 = 907,2 \text{ m}^2$. Kapasitas maksimal produksi *rolled wiremesh* adalah 1200 dan tumpukan maksimum dalam gudangnya adalah 3 sehingga lahan yang diperlukan untuk *plated wiremesh* adalah $1200 \times 0,7 \text{ m} \times 2,1 \text{ m} / 3 = 588 \text{ m}^2$. Luas yang dibutuhkan oleh gudang barang jadi adalah sejumlah $1495,2 \text{ m}^2$. Asumsi sementara ukuran gudang bahan baku adalah $30 \text{ m} \times 49,84 \text{ m} + \text{luas jalan selebar } 6 \text{ meter menjadi } 36 \text{ m} \times 49,84 \text{ m} = 1794,24 \text{ m}^2$.

Fasilitas

Perusahaan memiliki total 16 fasilitas dan untuk mempermudah dilakukan pengelompokan mesin untuk mesin *cutting* dan mesin *welding*. Melalui luasan total yang ada, luas fasilitas dibagi menjadi kotak-kotak dimana setiap kotak memiliki ukuran 2,5m x 2,5m. Berikut adalah daftar fasilitas, luasan tambahan yang dibutuhkan, oleh dan pembagian dalam kotak dapat dilihat pada Tabel 2.

Tabel 2. Fasilitas dan Kebutuhan Luas Tambahan

Fasilitas	Ukuran (meter)	Luas Tambahan (meter)	Keterangan Tambahan	Luas Total (meter)	Jumlah Kotak
-----------	-------------------	--------------------------	---------------------	-----------------------	-----------------

<i>Cutting</i>	10 x 3,5	7 x 6	<i>Forklift</i> , hasil potong, bahan baku	17 x 9,5	28
<i>Welding</i>	24,5 x 6,5	8 x 5	<i>Forklift</i> , hasil las	32,5 x 11,5	65
<i>Plate cutting</i>	22 x 4	11x5	Hasil las, <i>forklift</i> , hasil potong	33 x 9	56
<i>Rolling</i>	7,5 x 4	10 x 5	Hasil las, <i>forklift</i> , hasil rol	17,5 x 9	28
GBB	15 x 41,42	2 x 41,42	<i>Aisle</i>	17 x 41,42	119
GBJ	30 x 49,84	6 x 49,84	<i>Aisle</i>	36 x 49,84	300
Parkiran	50 x 15			50 x 15	120
Ruang Satpam	4,5 x 2,5			4,5 m x 2,5 m	2
Kantor	17 x 4			17 m x 4 m	14
Ruang Tunggu Supir	7 x 4			7 m x 4 m	6
<i>Loading Dock</i>	5 x 15			5 m x 15 m	12
<i>Receiving Buffer</i>	5 x 5			5 m x 5 m	4

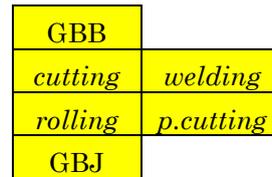
Penempatan Fasilitas dengan Algoritma Corelap

Pertama dalam corelap dilakukan perhitungan nilai TCR untuk memilih fasilitas mana yang menjadi pusat. Berikut adalah nilai TCR dari fasilitas produksi dapat dilihat pada Tabel 3.

Tabel 3. Nilai TCR Fasilitas Produksi

	1	2	3	4	5	6	TCR
1	-	A	O	O	A	U	37
2	A	-	A	A	E	U	57
3	O	A	-	E	U	A	43
4	O	A	E	-	U	A	43
5	A	E	U	U	-	O	28
6	U	U	A	A	O	-	36

Fasilitas 1 adalah fasilitas *cutting*, 2 adalah fasilitas *welding*, 3 adalah *plate cutting*, 4 adalah *rolling*, 5 adalah gudang bahan baku, dan 6 adalah gudang barang jadi. Fasilitas yang memiliki TCR paling tinggi adalah fasilitas nomor 2 yaitu fasilitas *welding* sehingga fasilitas ini ditempatkan pada pusat lokasi. Fasilitas yang diletakkan selanjutnya adalah *plate cutting*, *rolling*, *cutting*, gudang barang jadi dan terakhir gudang bahan baku. Bentuk fasilitas produksi hasil algoritma *corelap* dapat dilihat pada Gambar 4.



Gambar 4. Fasilitas Produksi Hasil Algoritma Corelap

Penempatan fasilitas produksi dilanjutkan dengan penempatan fasilitas pendukung lainnya. Berikut adalah nilai TCR dari fasilitas pendukung dapat dilihat pada Tabel 4.

Tabel 4. Nilai TCR Fasilitas Pendukung

	1	2	3	4	5	6	7	TCR
1	-	E	O	A	U	E	A	51
2	E	-	O	E	A	A	O	52
3	O	O	-	U	E	U	U	15
4	A	E	U	-	U	E	A	52
5	U	A	E	U	-	U	U	28
6	E	A	U	E	U	-	A	52
7	A	O	U	A	U	A	-	52

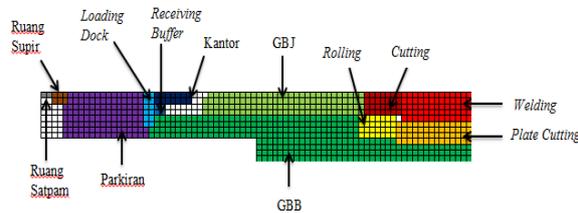
Fasilitas 1 adalah fasilitas produksi, 2 adalah fasilitas parkir, 3 adalah ruang satpam, 4 adalah kantor, 5 adalah ruang supir, 6 adalah *loading dock*, dan 7 adalah *receiving buffer*. Fasilitas yang memiliki TCR terbesar ada 4 fasilitas tetapi dipilih fasilitas parkir untuk diletakkan pertama karena luasnya paling besar. Fasilitas yang diletakkan

selanjutnya adalah *loading dock*, ruang supir, *receiving buffer*, kantor, produksi, dan terakhir ruang satpam. Bentuk fasilitas pendukung hasil algoritma *corelap* dapat dilihat pada Gambar 5.

R.Satpam	R.Supir	R.Buffer	Kantor
	Parkiran	L.dock	Produksi

Gambar 5. Fasilitas Pendukung Hasil Algoritma *Corelap*

Hasil dari algoritma *corelap* mengalami penyesuaian dengan lokasi kenyataan yang dimiliki oleh perusahaan. Berikut adalah hasil dua dimensi dari algoritma *corelap* dapat dilihat pada Gambar 6.



Gambar 6. Fasilitas Hasil Algoritma *Corelap*

Titik berat dapat dihitung setelah bentuk dua dimensi dihasilkan oleh algoritma. Berikut adalah titik berat fasilitas hasil algoritma *corelap* dapat dilihat pada Tabel 5.

Tabel 5. Titik berat Fasilitas Hasil Algoritma *Corelap*

Fasilitas	Titik Berat Sumbu X	Titik Berat Sumbu Y
Produksi	136.81	15.62
Parkiran	29.74	19.9
Ruang Satpam	2.5	28.75
Kantor	61.25	27.5
Ruang Tunggu Supir	8.75	27.5
Loading Dock	50	22.5
Receiving Buffer	55	22.5

Titik berat dihitung untuk mencari jarak antar fasilitas. Jarak antar fasilitas hasil algoritma *corelap* dapat dilihat pada Tabel 6.

Tabel 6. Jarak antar Fasilitas Algoritma *Corelap*

	1	2	3	4	5	6	7
1	-	111.35	147.44	87.44	139.94	93.69	88.69
2	111.35	-	36.09	39.11	28.59	22.86	27.86

3	147.44	36.09	-	60	7.5	53.75	58.75
4	87.44	39.11	60	-	52.5	16.25	11.25
5	139.94	28.59	7.5	52.5	-	46.25	51.25
6	93.69	22.86	53.75	16.25	46.25	-	5
7	88.69	27.86	58.75	11.25	51.25	5	-

Momen membutuhkan data jarak yang ada, juga memerlukan data *flow* antar fasilitas yang ada. Berikut data *flow* antar fasilitas dari algoritma *corelap* dapat dilihat pada Tabel 7.

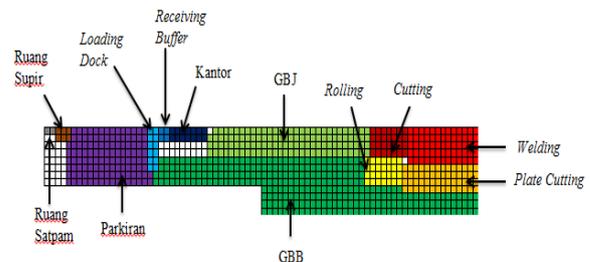
Tabel 7. *Flow* antar Fasilitas Algoritma *Corelap*

	1	2	3	4	5	6	7
1	-	8	2	16	1	8	16
2	8	-	2	8	16	16	2
3	2	2	-	1	8	1	1
4	16	8	1	-	1	8	16
5	1	16	8	1	-	1	1
6	8	16	1	8	1	-	16
7	16	2	1	16	1	16	-

Setelah mengetahui data jarak dan *flow*, dilakukan perhitungan momen untuk algoritma *corelap*. Momen total yang dihasilkan oleh algoritma *corelap* adalah sebesar 13859,4.

Penempatan Fasilitas dengan Algoritma *Aldep*

Algoritma *aldep* memilih secara acak untuk penempatan fasilitas pertama tetapi kali ini, dipilih untuk menempatkan fasilitas yang berhubungan dengan produksi terlebih dahulu. Fasilitas yang diletakkan kedua dan selanjutnya dipilih melalui hubungan kedekatannya. Peletakan fasilitas dari ujung dimulai dengan *welding*, *plate cutting*, *rolling*, GBJ, GBB, dan *cutting*. Setelah fasilitas produksi selesai diletakkan, dilanjutkan dengan peletakan kantor, *receiving buffer*, *loading dock*, parkiran, ruang tunggu supir, dan terakhir ruang satpam. Berikut adalah hasil dua dimensi dari algoritma *aldep* dapat dilihat pada Gambar 7.



Gambar 7. Fasilitas Hasil Algoritma *Aldep*

Titik berat dapat dihitung setelah bentuk dua dimensi dihasilkan oleh algoritma. Berikut adalah titik berat fasilitas hasil algoritma *corelap* dapat dilihat pada Tabel 8.

Tabel 8. Titik berat Fasilitas Hasil Algoritma Aldep

Fasilitas	Titik Berat	
	Sumbu X	Sumbu Y
Produksi	136.81	15.62
Parkiran	29.74	19.9
Ruang Satpam	2.5	28.75
Kantor	66.25	27.5
Ruang Tunggu	8.75	27.5
Supir		
Loading Dock	50	22.5
Receiving Buffer	55	27.5

Titik berat dihitung untuk mencari jarak atau jarak antar fasilitas. Jarak antar fasilitas hasil algoritma aldep dapat dilihat pada Tabel 9.

Tabel 9. Jarak antar Fasilitas Algoritma Aldep

	1	2	3	4	5	6	7
1		111.35	147.44	82.44	139.94	93.69	93.69
2	111.35		36.09	44.11	28.59	22.86	32.86
3	147.44	36.09		65	7.5	53.75	53.75
4	82.44	44.11	65		57.5	21.25	11.25
5	139.94	28.59	7.5	57.5		46.25	46.25
6	93.69	22.86	53.75	21.25	46.25		10
7	93.69	32.86	53.75	11.25	46.25	10	

Momen membutuhkan data jarak yang ada, juga memerlukan data flow antar fasilitas yang ada. Berikut data flow antar fasilitas dari algoritma aldep dapat dilihat pada Tabel 10.

Tabel 10. Flow antar Fasilitas Algoritma Aldep

	1	2	3	4	5	6	7
1	-	8	2	16	1	8	16
2	8	-	2	8	16	16	2
3	2	2	-	1	8	1	1
4	16	8	1	-	1	8	16
5	1	16	8	1	-	1	1
6	8	16	1	8	1	-	16
7	16	2	1	16	1	16	

Setelah mengetahui data jarak dan flow maka dilakukan perhitungan momen untuk algoritma aldep. Momen total yang dihasilkan oleh algoritma aldep adalah sebesar 14.199,4.

Penempatan Fasilitas dengan Algoritma Blocplan

Algoritma blocplan membutuhkan sebuah *initial layout* pada awalnya sehingga dipilih untuk menggunakan hasil algoritma corelap untuk dijadikan *initial layout* karena momen yang dihasilkan lebih kecil daripada momen yang dihasilkan algoritma aldep. Berikut adalah flow antar fasilitas dari algoritma blocplan dapat dilihat pada Tabel 11.

Tabel 11. Flow antar Fasilitas Algoritma Blocplan

	1	2	3	4	5	6	7
1	-	8	2	16	1	8	16
2	8	-	2	8	16	16	2
3	2	2	-	1	8	1	1
4	16	8	1	-	1	8	16
5	1	16	8	1	-	1	1
6	8	16	1	8	1	-	16
7	16	2	1	16	1	16	

Upper Bound dan Lower Bound

Upper bound dapat dikatakan sebagai *worst case* atau kemungkinan terburuk dari momen yang dapat diperoleh. *Upper bound* dapat dicari dengan cara mengalikan jarak terbesar dengan flow terbesar hingga jarak terkecil dengan flow terkecil. Berbeda dengan *upper bound*, *lower bound* adalah *best case* atau kemungkinan terbaik dari momen yang dapat diperoleh. *Lower bound* dapat dicari dengan cara mengalikan jarak terkecil dengan flow terbesar hingga jarak terbesar dengan flow terkecil yang ada. Setelah mendapatkan nilai *upper bound* dan *lower bound* yang ada, kedua hasil ini dipakai sebagai batas atas dan batas bawah saat perhitungan Rscore. *Upper bound* didapati bernilai 26538.02 dan *Lower bound* didapati bernilai 8240.78.

Perhitungan Rscore

Res-dist score atau momen awal didapati bernilai 13.859,4. Data ini digunakan untuk melakukan perhitungan Rscore. $Rscore = 1 - (Res-dist\ score - lower\ bound) / (upper\ bound - lower\ bound)$ sehingga R-score awal didapati bernilai 0.692925.

Iterasi Algoritma Blocplan

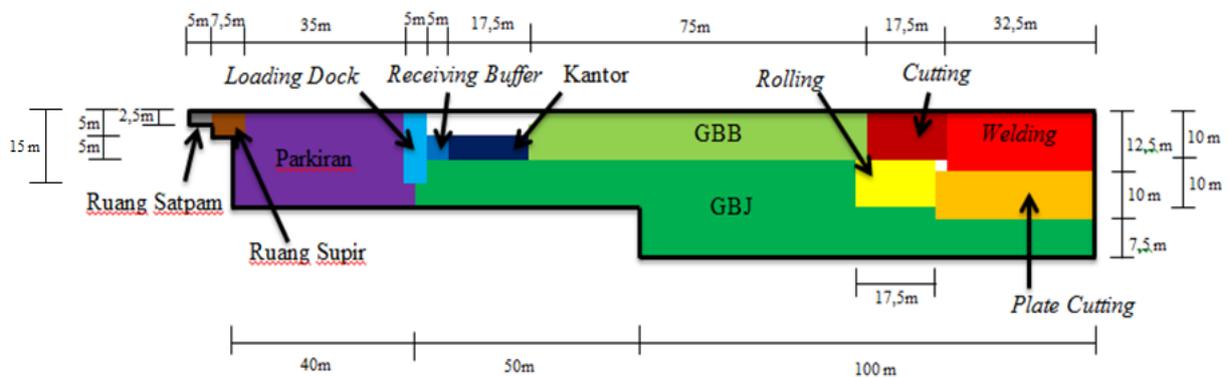
Setiap kali iterasi dilakukan, dilakukan perhitungan momen dan Rscore untuk dibandingkan dengan momen dan Rscore yang dimiliki oleh *initial layout*. Hal ini dilakukan untuk memperoleh layout yang menghasilkan momen paling minimum. Iterasi dilakukan sebanyak 5 kali. Hasil momen dan Rscore dari setiap iterasi dapat dilihat pada Tabel 12.

Tabel 12. Hasil Iterasi Algoritma Blocplan

Iterasi	Rscore	Momen
Awal	0,692925	13859,4
1	0,714924	14319,4
2	0,70978	13579,4
3	0,721803	13945,48
4	0,672565	13694,4
5	0,689465	13879,4

Iterasi *blocplan* dilakukan sampai data yang dihasilkan tidak mengalami perubahan yang

signifikan atau dibatasi sepuluh kali maksimal dalam penelitian kali ini. Berhubung data tidak mengalami perubahan cukup signifikan dalam beberapa iterasi, maka iterasi dihentikan pada iterasi ke 5. Melalui hasil lima iterasi yang ada, hasil terbaik adalah hasil dari iterasi ke 2 yaitu hasil pertukaran antara kantor dan area perluasan. Melalui hasil iterasi ke 2, didapat nilai *Rscore* dengan nilai 0,70978. Momen yang dihasilkan oleh iterasi ini adalah paling rendah yaitu 13579,4. Berikut adalah tampak dua dimensi hasil iterasi ke 2 *blocplan* dapat dilihat pada Gambar 8.



Gambar 8. Fasilitas Hasil Algoritma Blocplan

Simpulan

Melalui beberapa metode yang telah dilakukan, didapati beberapa *layout* yang berbeda. Jarak dan momen yang dihasilkan pun beragam. Hasil terbaik didapatkan dengan algoritma *blocplan* yang menghasilkan momen paling minimum sejumlah 13579,4. Bentuk akhir dari lokasi yang menghasilkan momen paling minimum adalah fasilitas yang diletakkan pada lokasi paling depan adalah ruang tunggu supir dan tempat satpam. Setelah itu fasilitas yang berada selanjutnya adalah parkiran. Setelah parkiran, terdapat *loading dock*, *receiving buffer*, dan kantor. Pada sisi belakang kantor, baru ditemukan fasilitas produksi antara lain adalah gudang bahan baku, gudang barang jadi, area *cutting*, area *welding*, *rolling*, dan *plate cutting*.

Daftar Pustaka

1. Heragu, S. (1997). *Facilities Design*. Boston: PWS Publishing Company.

