



UNIVERSITAS INDONESIA

**PENGEMBANGAN METODE PEMENUHAN KENDALA
MODEL REPRESENTASI *FLEXIBLE BAY STRUCTURE*
DALAM PENYELESAIAN TATA LETAK FASILITAS**

SKRIPSI

**GERSEN SAMUEL SUMARDI
0706274672**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INDUSTRI
DEPOK
JUNI 2011**



UNIVERSITAS INDONESIA

**PENGEMBANGAN METODE PEMENUHAN KENDALA
MODEL REPRESENTASI *FLEXIBLE BAY STRUCTURE*
DALAM PENYELESAIAN TATA LETAK FASILITAS**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar
Sarjana Teknik**

**GERSEN SAMUEL SUMARDI
0706274672**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INDUSTRI
DEPOK
JUNI 2011**

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri, dan semua sumber,
baik yang dikutip maupun dirujuk, telah saya nyatakan dengan benar.

Nama : Gersen Samuel Sumardi

NPM : 0706274672

Tanda Tangan : 

Tanggal : 20 Juni 2011

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Gersen Samuel Sumardi
NPM : 0706274672
Program Studi : Teknik Industri
Judul Skripsi : Pengembangan Metode Pemenuhan Kendala Model Representasi *Flexible Bay Structure* dalam Penyelesaian Tata Letak Fasilitas

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Industri, Fakultas Teknik, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Armand Omar Moeis, ST., M.Sc. ()
Penguji : Akhmad Hidayatno, ST., MBT. ()
Penguji : Ir. M. Dachyar, M.Sc. ()
Penguji : Ir. Yadrifil, M.Sc. ()

Ditetapkan di : Depok
Tanggal : 20 Juni 2011

UCAPAN TERIMA KASIH

Segala hormat, puji, dan syukur saya panjatkan kepada Tuhan Yesus Kristus atas kasih, berkat, bimbingan, penyertaan, serta hikmat yang hanya dari pada-Nya, sehingga saya dapat menyelesaikan skripsi ini dengan baik dan tepat pada waktunya.

Penulisan skripsi ini dilaksanakan sebagai salah satu syarat mahasiswa tingkat akhir untuk mendapatkan gelar Sarjana Teknik jurusan Teknik Industri pada Fakultas Teknik Universitas Indonesia.

Saya menyadari bahwa penulisan skripsi ini tidak lepas dari dukungan, motivasi, bantuan, dan kerja sama berbagai pihak, sehingga saya pada akhirnya dapat menyelesaikan skripsi saya dengan baik. Oleh karena itu, pada kesempatan ini saya ingin mengucapkan terima kasih kepada:

1. Bapak Komarudin, ST., M.Eng., selaku dosen pembimbing skripsi, yang telah meluangkan banyak waktunya untuk mengajar, membimbing, membantu, serta memberikan ide, saran, kritik, nasihat, maupun pengetahuan kepada saya dalam rangka penyusunan skripsi ini;
2. Bapak Armand Omar Moeis, ST., M.Sc., atas segala masukan, nasihat, kritik, dan saran kepada saya dalam pengerjaan dan penyelesaian skripsi ini;
3. Bapak Akhmad Hidayatno, ST., MBT., selaku Kepala Laboratorium SEMS TIUI, atas dukungan, motivasi, kritik, saran, serta kesediaannya dalam memfasilitasi para mahasiswa RAP SEMS menyelesaikan skripsi mereka;
4. Seluruh dosen dan staff Departemen Teknik Industri Universitas Indonesia, atas ilmu pengetahuan, bimbingan, dukungan, dan bantuan yang telah diberikan selama masa perkuliahan hingga penyelesaian skripsi;
5. Orang tua, papa dan mama tercinta, Sujatno Sumardi dan Linyani Arievida, serta adik saya, Isakh Salomon Sumardi, atas segala kasih sayang, doa, dukungan, motivasi, nasihat, dan bantuan yang telah diberikan;

6. Rekan-rekan Research Assistant SEMS seperjuangan di bawah bimbingan Pak Komar, yaitu Ariel, Alan, Oscar, Paul, dan Lucy, atas kebersamaan, diskusi, dan kerjasamanya selama menyusun skripsi ini;
7. Rekan-rekan Research Assistant SEMS yang lain, Gersi, Daril, Tulus, Rangga, dan Berry, atas kebersamaan dan kerja samanya dalam rangka penyelesaian skripsi kita masing-masing;
8. Seluruh rekan-rekan Teknik Industri angkatan 2007, atas persahabatan yang terjalin begitu indah, serta atas seluruh waktu dan kebersamaan yang telah kita jalani bersama selama 4 tahun yang tak mungkin terlupakan;
9. Teman-teman dari Persekutuan Oikumene FTUI, atas rasa kekeluargaan yang tercipta, segala dukungan, motivasi, serta doanya;
10. Seluruh mahasiswa Teknik Industri Universitas Indonesia, yang telah mendukung dan menjadi teman-teman yang baik, khususnya bagi angkatan 2007;
11. Seluruh pihak yang terlibat, baik secara langsung maupun tidak langsung, yang tidak dapat saya sebutkan satu per satu.

Skripsi ini tentu tidak sempurna karena ditulis dan disusun oleh seseorang yang tidak sempurna, karena itu saya mohon maaf bila ada kekurangan. Segala kritik dan saran sangat saya harapkan demi perbaikan untuk masa yang akan datang. Mudah-mudahan skripsi ini dapat memberikan pengetahuan dan wawasan yang baru bagi para pembaca. Saya juga sungguh mengharapkan bahwa skripsi ini dapat memberikan manfaat yang baru bagi pengembangan ilmu pengetahuan di masa yang akan datang.

Depok, 20 Juni 2011

Gersen Samuel Sumardi

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai civitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Gersen Samuel Sumardi
NPM : 0706274672
Program Studi : Teknik Industri
Departemen : Teknik Industri
Fakultas : Teknik
Jenis Karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Non-eksklusif (*Non-exclusive Royalty Free Right*)** atas karya ilmiah saya yang berjudul:

Pengembangan Metode Pemenuhan Kendala Model Representasi *Flexible Bay Structure* Dalam Penyelesaian Tata Letak Fasilitas

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-eksklusif ini Universitas Indonesia berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di: Depok

Pada tanggal: 20 Juni 2011

Yang menyatakan



(Gersen Samuel Sumardi)

ABSTRAK

Nama : Gersen Samuel Sumardi
Program Studi : Teknik Industri
Judul : Pengembangan Metode Pemenuhan Kendala Model Representasi *Flexible Bay Structure* dalam Penyelesaian Tata Letak Fasilitas

Masalah tata letak fasilitas, khususnya dengan luas yang tidak sama (*UAFLP*), merupakan salah satu masalah yang dipelajari dalam *combinatorial optimization* dan telah menerima banyak perhatian peneliti. Penelitian ini akan fokus dan mengembangkan suatu metode berdasarkan pada *Flexible Bay Structure*, salah satu model representasi yang umum digunakan dalam *UAFLP*. Metode ini akan memenuhi kendala-kendala yang muncul, langkah demi langkah, sepanjang proses pembentukan hasil akhir. Tujuannya untuk mengurangi solusi yang tidak *feasible*, sehingga akan mengurangi kompleksitas solusi yang mungkin. Pengujian dilakukan menggunakan masalah-masalah yang telah diketahui secara umum dan telah digunakan pada penelitian-penelitian sebelumnya. Pencarian solusi terbaik akan dibantu dengan algoritma *Simulated Annealing*. Sebagian besar hasil dari penelitian ini mendekati hasil terbaik yang diketahui dan hanya satu masalah yang berhasil diperbaiki.

Kata kunci:

Optimasi, *Unequal Area Facility Layout Problem*, *Flexible Bay Structure*, Pemenuhan Kendala, *Simulated Annealing*

ABSTRACT

Name : Gersen Samuel Sumardi
Study Program : Industrial Engineering
Title : Developing Constraint Satisfaction Method on Flexible Bay Structure Representation in Solving Facility Layout Problem

Facility layout problem with unequal area (*UAFLP*) is one of the best-studied problems in *combinatorial optimization* and has received many researchers' attentions. This paper will focus and develop a method based on the *Flexible Bay Structure*, one of the common *UAFLP* model representations, which will satisfy the constraints that occur step by step in the construction process. It will reduce the non-feasible solutions that occur, thus reducing the complexity of possible solution. This method is tested using well-known problems using simulated annealing in exploring solution. Some results obtained are close with the best-known and with one problem solved.

Key words:

Optimization, *Unequal Area Facility Layout Problem*, *Flexible Bay Structure*, Constraint Satisfaction, *Simulated Annealing*

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN	iii
UCAPAN TERIMA KASIH.....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	vi
ABSTRAK.....	vii
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	x
DAFTAR LAMPIRAN.....	xi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Diagram Keterkaitan Masalah.....	3
1.3 Perumusan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Ruang Lingkup/Batasan Penelitian	4
1.6 Metodologi Penelitian	5
1.6.1 Diagram Alir Metodologi Penelitian	6
1.7 Sistematika Penulisan.....	7
BAB 2 DASAR TEORI.....	8
2.1 Tata Letak Fasilitas	8
2.1.1 <i>Unequal Area Facility Layout Problem</i>	8
2.2 <i>Simulated Annealing</i>	15
BAB 3 PENGUMPULAN DATA	18
BAB 4 PENGOLAHAN DATA DAN ANALISA.....	21
4.1 Pengolahan Data.....	21
4.1.1 Penyusunan Program	21
4.1.1.1 Menentukan <i>Feasible Subsets</i>	22
4.1.1.2 Menentukan Kombinasi yang Mungkin.....	24
4.1.1.3 Mencari Koordinat tiap Departemen dalam Kombinasi	25
4.1.1.4 Mencari <i>Objective Function</i> terbaik	26
4.1.2 Verifikasi dan Validasi Program	28
4.1.3 Hasil.....	28
4.2 Analisa Hasil	32
BAB 5 KESIMPULAN DAN SARAN.....	37
5.1 Kesimpulan.....	37
5.2 Saran.....	37
REFERENSI.....	39
LAMPIRAN.....	42

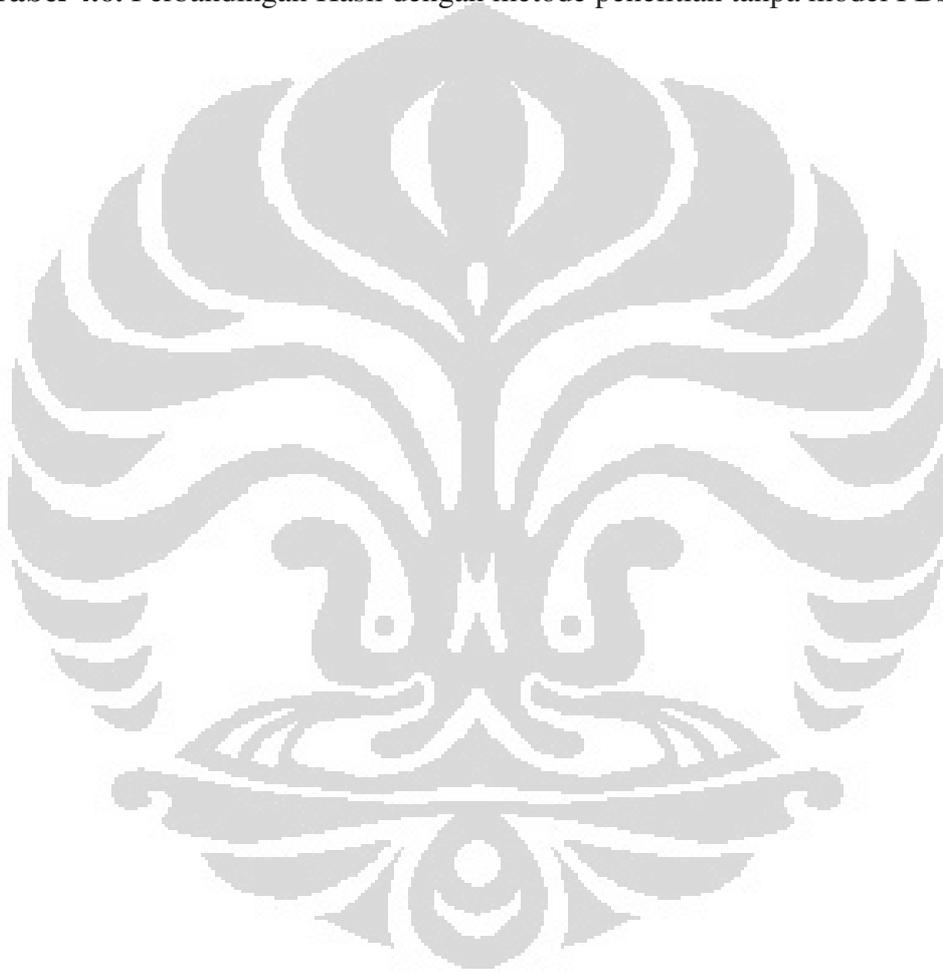
DAFTAR GAMBAR

Gambar 1.1. Diagram Keterkaitan Masalah	3
Gambar 1.2. Diagram Alir Metodologi Penelitian	6
Gambar 2.1. Contoh tampilan (a) <i>Block Layout</i> dan (b) <i>Detailed Layout</i>	10
Gambar 2.2. Penghitungan (a) jarak <i>Euclidean</i> dan (b) jarak <i>Rectilinear</i>	11
Gambar 2.3. Contoh Solusi menggunakan representasi Kontinu.....	12
Gambar 2.4. Contoh solusi dengan model QAP oleh Hardin dan User (2005)...	13
Gambar 2.5. Contoh solusi UA-FLP dengan model <i>Flexible Bay Structure</i>	14
Gambar 2.6. Contoh solusi UA-FLP dengan model STS.....	15
Gambar 2.7. Proses Penurunan Suhu Algoritam SA.....	16
Gambar 4.1. Pergerakan dalam <i>Local Search</i>	27



DAFTAR TABEL

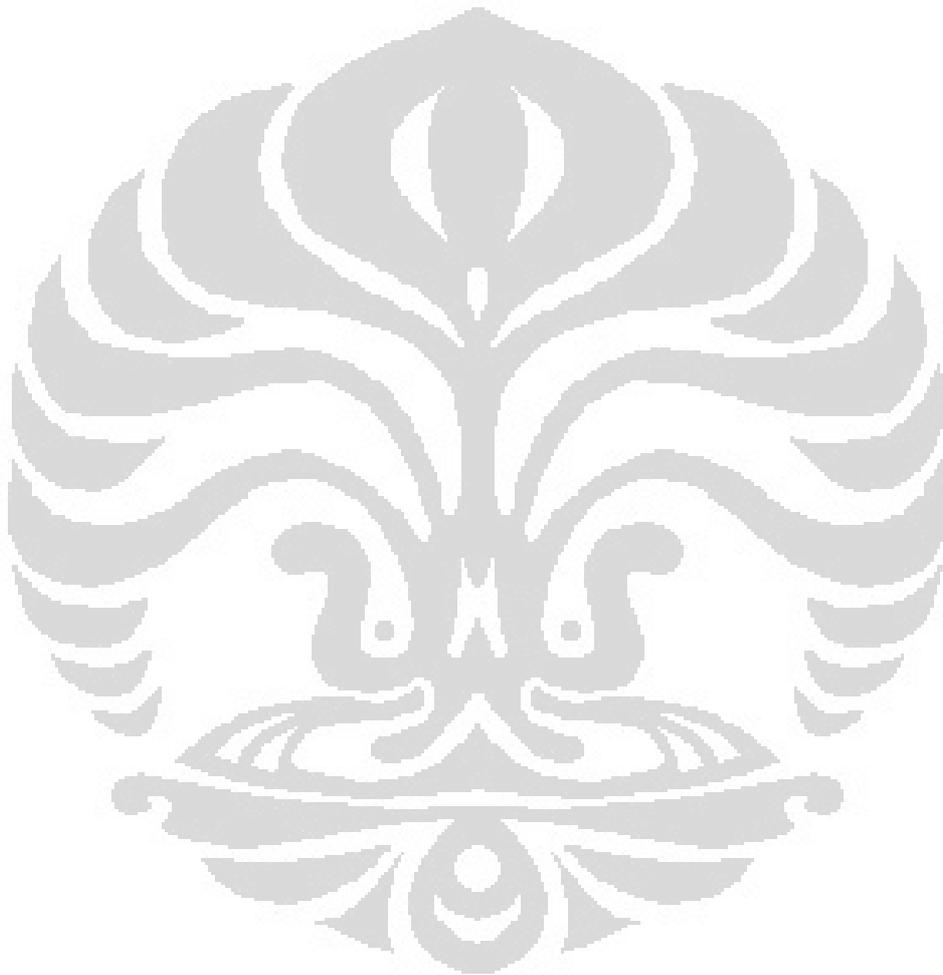
Tabel 3.1. <i>Problem Sets Unequal Area Facility Layout Problem (UAFLP)</i>	18
Tabel 4.1. Hasil Fungsi Tujuan dengan Replikasi	29
Tabel 4.2. Hasil Statistik dari Fungsi Tujuan dan <i>Computational Time</i>	30
Tabel 4.3. Parameter algoritma <i>Simulated Annealing</i> dalam Iterasi	31
Tabel 4.4. Perbandingan Hasil dengan <i>best-known</i> dan <i>best-known</i> FBS.....	33
Tabel 4.5. Perbandingan Penelitian menggunakan Model Representasi FBS	35
Tabel 4.6. Perbandingan Hasil dengan metode penelitian tanpa model FBS.....	36



DAFTAR LAMPIRAN

Lampiran 1 UA-FLP *Problem Sets*

Lampiran 2 Tampilan Tata Letak Fasilitas Terbaik



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Permasalahan tata letak fasilitas berkaitan erat dengan penentuan dan desain susunan fisik dari elemen-elemen dalam sebuah sistem manufaktur, seperti departemen, mesin, tempat penyimpanan, dan lain-lain. Tujuannya adalah menemukan sebuah tata letak fasilitas dengan biaya *material handling* paling minimal. Seperti yang dikemukakan oleh Apple (1977), bahwa sangat krusial untuk mengikutsertakan keputusan mengenai *material handling* dalam perencanaan dan desain suatu fasilitas. Tompkins *et al.* (1996) mengemukakan bahwa 20% - 50% dari total *operating expenses* dialokasikan kepada perencanaan fasilitas dan *material handling*. Sehingga diharapkan bahwa total *operating expenses* dapat dikurangi dan pada akhirnya berkontribusi pada efisiensi keseluruhan system produksi (Asef-Vaziri dan Laporte, 2004).

Banyak penelitian yang sudah dilakukan oleh para peneliti sejak permasalahan tata letak fasilitas ditemukan oleh Armour dan Buffa (1963). Di dalam laporan penelitiannya, Meller & Gau (1996) dan Singh & Sharma (2005), masing-masing melaporkan bahwa telah melihat kembali hampir 100 *paper* yang diterbitkan sebelum tahun 1996 dan hampir 140 *paper* dalam 20 tahun sebelum 2005. Kedua pengarang ini membahas tentang kecenderungan di dalam penelitian tata letak fasilitas untuk mengintegrasikan desain dari suatu fasilitas dengan desain dari *material handling*.

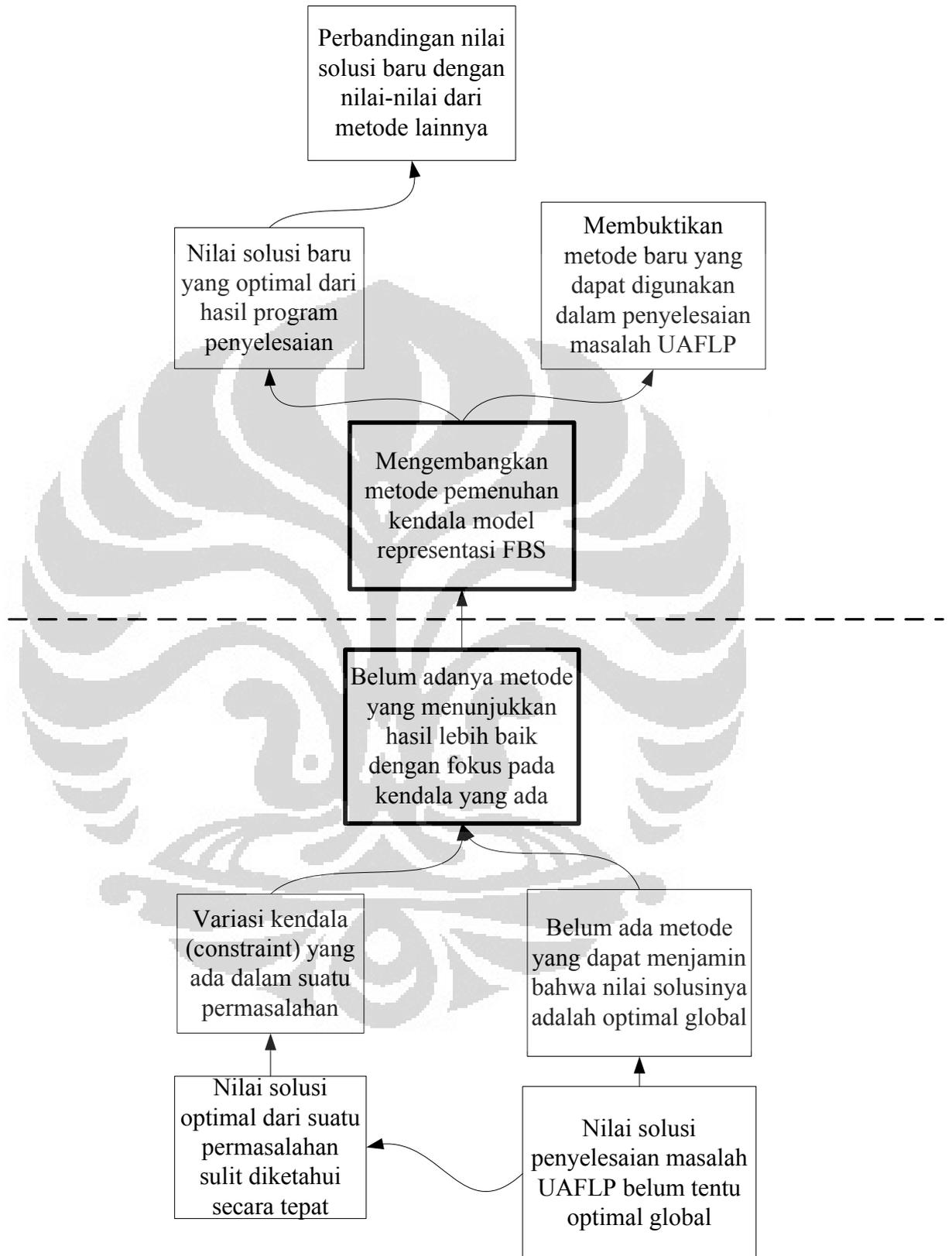
Salah satu masalah tata letak fasilitas yang diketahui secara umum adalah UAFLP (*Unequal Area Facility Layout Problem*). Masalah FLP seringkali dikaitkan secara langsung dengan UAFLP, atau dengan kata lain FLP identik dengan UAFLP. Masalah ini telah mendapat perhatian dari banyak peneliti dalam satu dekade terakhir, seperti penelitian yang dilakukan oleh Meller *et al.* (1998), Gau dan Meller (1999), Sherali *et al.* (2003), Castillo *et al.* (2005), Castillo dan Westerlund (2005), Konak *et al.* (2006), Liu dan Meller (2007), Scholz *et al.* (2009), Komarudin dan Wong (2010), Wong dan Komarudin (2010), Kulturel-Konak dan Konak (2010, 2011).

Tujuan dari UAFLP adalah untuk menempatkan departemen-departemen ke dalam suatu fasilitas, guna meminimalkan biaya penanganan material antar departemen di dalam fasilitas tersebut. Luas tiap departemen tidak perlu sama, tetapi dibatasi dengan adanya dimensi rasio atau dimensi minimum.

Salah satu model representasi yang sukses dalam menyelesaikan masalah UAFLP adalah *Flexible Bay Structure*. Model representasi ini pertama kali dicetuskan oleh Tong (1991) dan akhir-akhir ini mendapat banyak perhatian dari para peneliti. Ini dapat dilihat dari penelitian yang dilakukan oleh Tate dan Smith (1995), Konak *et al.* (2006), Wong dan Komarudin (2010), Kulturel-Konak and Konak (2010, 2011). Hasil terbaik untuk masalah-masalah UAFLP sebagian besar menggunakan model representasi ini (Kulturel-Konak and Konak, 2011).

Karena model ini masih dianggap sebagai model yang baik merepresentasikan masalah tata letak fasilitas, maka penelitian ini ingin fokus lebih lagi dan mempelajari setiap kendala yang mungkin muncul. Penelitian ini akan mengembangkan suatu metode yang memenuhi kendala yang ada, langkah demi langkah, pada proses pemebentukan hasil akhir, sehingga akan mengurangi solusi yang tidak *feasible* dan mengurangi kompleksitas dari solusi yang mungkin. Melalui metode ini diharapkan suatu hasil yang baik dan setidaknya dapat menyamai hasil dari *best-known* FBS dan akan mengurangi waktu pencarian dari solusi yang optimal. Penelitian ini akan menggunakan algoritma *Simulated Annealing* dalam pencarian solusi terbaik.

1.2 Diagram Keterkaitan Masalah



Gambar 1.1. Diagram Keterkaitan Masalah

1.3 Perumusan Masalah

Model Representasi *Flexible Bay Structure* merupakan suatu model yang sering digunakan di dalam penelitian beberapa tahun terakhir dan terbukti menghasilkan hasil yang terbaik untuk masalah-masalah UAFLP. Belum adanya metode yang fokus pada kendala dan mengurangi kemungkinan solusi sehingga penelitian ini bermaksud untuk fokus pada model representasi tersebut dan membuat sebuah metode yang dapat memenuhi kendala yang dihasilkan, langkah demi langkah, mulai dari proses awal pembentukan hingga hasil akhir, sehingga dapat mengurangi kemungkinan solusi dan mengurangi kompleksitas.

1.4 Tujuan Penelitian

Penelitian ini dilakukan dengan tujuan untuk:

- Mengembangkan suatu metode pemenuhan kendala model representasi *Flexible Bay Structure* dalam penyelesaian masalah tata letak fasilitas
- Mendapatkan nilai solusi optimal yang baru dan membandingkan dengan hasil dari metode lainnya dalam penyelesaian masalah UAFLP
- Melihat apakah metode yang diteliti ini lebih baik daripada metode-metode sebelumnya dalam memecahkan masalah yang sama

1.5 Ruang Lingkup/Batasan Penelitian

Agar penelitian ini dapat berjalan secara lancar dan dalam koridor yang jelas guna memenuhi tujuan, maka diberikan batasan masalah sebagai berikut:

- Penelitian dikhususkan pada masalah tata letak fasilitas dengan luas yang tidak sama
- Penelitian hanya memperhitungkan luas tiap fasilitas dan mengabaikan isi dari tiap fasilitas tersebut, isi tiap fasilitas dapat dimodifikasi sendiri sesuai keinginan setelah hasil *layout* terbaik didapatkan
- Penelitian hanya akan mengembangkan metode yang berlaku untuk fasilitas secara umum, metode ini dapat dikembangkan lebih lanjut jika ingin diaplikasikan dalam keadaan yang benar-benar *real* dan spesifik
- Penelitian akan melihat biaya total sebagai *output* dari fungsi tujuan dalam menentukan hasil terbaik

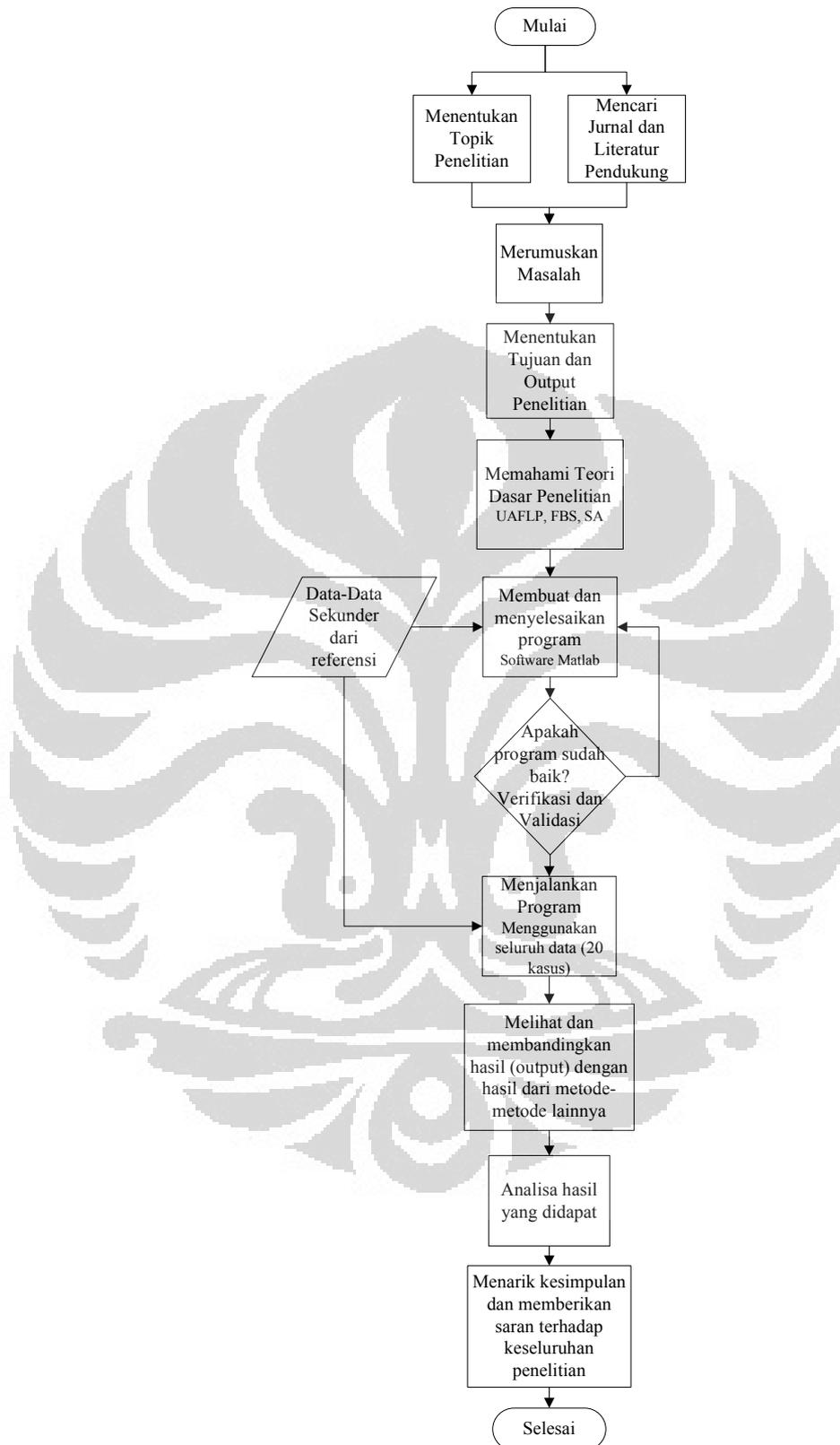
- Biaya (*cost*) dalam penelitian ini dianggap sama dengan 1 (satu), hal ini untuk membuat perhitungan menjadi sederhana dan memang ditetapkan demikian dari penelitian-penelitian sebelumnya.
- Masalah yang digunakan sebagai data pada penelitian ini hanya terbatas pada 20 (dua puluh) kasus, yang juga digunakan oleh penelitian-penelitian sebelumnya dari berbagai referensi, masalah-masalah terlampir pada bagian **Lampiran 1**.

1.6 Metodologi Penelitian

Penelitian ini dilakukan dengan tahap-tahap sebagai berikut:

- **Penentuan Topik dan Perumusan Masalah**
Merupakan langkah awal dalam penulisan tugas akhir, dimana akan ditetapkan topik penelitian dan perumusan masalah, sehingga dapat diketahui tujuan dan hasil akhir yang diinginkan
- **Pemahaman Dasar Teori**
Pemahaman teori-teori yang mendukung seperti *Flexible Bay Structure*, *Facility Layout Problem*, *Unequal Area Facility Layout Problem*, *Simulated Annealing*
- **Pembuatan Program**
Program yang dibuat menggunakan software MatLab dan akan menghasilkan program dengan data *input* berupa Panjang dan Tinggi Fasilitas, Luas Area Tiap Departemen, *Shape Constraint*, dan Matriks *Material Flow*
- **Pengolahan Data dan Analisa**
Data yang digunakan adalah 20 (dua puluh) kasus yang didapat dari referensi, yang akan dicari satu per satu nilai optimalnya menggunakan program yang sudah diselesaikan dan akan dilakukan perbandingan dengan hasil dari metode lainnya
- **Kesimpulan dan Saran**
Kesimpulan yang didapat dari penelitian ini dan saran dari penulis mengenai penelitian ini, yang mungkin dapat dilanjutkan untuk dikembangkan.

1.6.1 Diagram Alir Metodologi Penelitian



Gambar 1.2. Diagram Alir Metodologi Penelitian

1.7 Sistematika Penulisan

Laporan tugas akhir ini terdiri dari beberapa bab yang saling berhubungan satu dengan yang lainnya, sehingga baiknya dibaca mulai dari bagian awal hingga ke akhir. Penulisan diawali **Bab 1**, yaitu **Pendahuluan**, berisi latar belakang permasalahan, diagram keterkaitan masalah, perumusan masalah, tujuan penelitian, ruang lingkup penelitian, metodologi penelitian, dan sistematika penulisan. Bab ini merupakan pengantar dari seluruh laporan dan diharapkan melalui bab ini pembaca dapat mengerti alasan, tujuan, dan metode/cara penelitian ini dilakukan.

Setiap penelitian yang dilakukan harus dilandasi oleh **Dasar Teori** yang ada di **Bab 2**. Bab ini akan membahas teori mengenai *Facility Layout Problem* dan UAFLP, teori *Flexible Bay Structure*, dan algoritma *Simulated Annealing*.

Proses pengumpulan data akan dibahas pada **Bab 3, Pengumpulan Data**. Pada bab ini akan dijelaskan mengenai data-data yang digunakan, yang berasal dari berbagai referensi berdasarkan penelitian sebelumnya. Setelah data didapatkan, maka akan dilakukan proses pengolahan data dan analisa pada **Bab 4, Pengolahan Data dan Analisa**. Di sini cara pengolahan data hingga mendapatkan hasil akhir sesuai tujuan penelitian dibahas. Proses analisa hasil akhir akan dilakukan, begitu juga dengan membandingkan hasil akhir dengan hasil lainnya berdasarkan referensi. Dari proses analisa yang dilakukan akan didapatkan kesimpulan yang dibahas pada **Bab 5, Kesimpulan dan Saran**. Bab ini merupakan bab terakhir yang berisi kesimpulan dan saran dari penulis. Kesimpulan ini akan menjawab setiap tujuan penelitian yang diharapkan pada awal laporan, sedangkan saran merupakan opini pribadi penulis untuk penelitian lebih lanjut yang mungkin akan dilakukan.

BAB 2 DASAR TEORI

2.1 Tata Letak Fasilitas

Masalah tata letak fasilitas (*Facility Layout Problem*) terdiri dari berbagai masalah optimasi, termasuk di dalamnya suatu masalah optimasi yang membagi suatu fasilitas yang diketahui dimensinya (umumnya persegi panjang) menjadi departemen-departemen dengan luas tertentu, dengan tujuan untuk meminimalkan biaya yang berhubungan dengan interaksi antara departemen-departemen di dalamnya. Biaya-biaya ini dapat berupa biaya transportasi atau kecenderungan pemilihan kedekatan antara departemen.

Jika luas-luas departemen yang diketahui memiliki luas yang sama, maka akan menjadi mudah untuk menentukan beberapa tata letak yang dapat dihasilkan oleh departemen-departemen tersebut. Kondisi seperti ini yang dapat dimodelkan dengan model representasi *Quadratic Assignment Problem* (lihat **Gambar 2.4**).

Namun jika luas-luas departemen yang diketahui memiliki luas yang tidak sama, departemen-departemen yang ada tidak dapat begitu saja diletakkan pada beberapa titik pusat sebuah fasilitas. Letak titik pusat departemen tersebut akan tergantung kepada suatu konfigurasi yang tepat. Kesulitan utama yang dihadapi masalah ini adalah banyaknya kemungkinan *layout* yang dapat dihasilkan oleh departemen-departemen tersebut (Tate dan Smith, 1995). Masalah seperti ini yang biasa disebut dengan masalah tata letak fasilitas dengan luas yang tidak sama, atau biasa dikenal dengan singkatan UA-FLP (*Unequal Area Facility Layout Problem*).

2.1.1 Unequal Area Facility Layout Problem

UA-FLP merupakan salah satu masalah yang sejak lama telah dikembangkan oleh para peneliti. Meskipun UA-FLP termasuk ke dalam kategori FLP, namun FLP sudah identik dengan UA-FLP sehingga masalah ini menjadi suatu masalah yang menarik dan dikenal oleh banyak orang. Cukup banyak penelitian yang telah dilakukan terkait dengan UA-FLP dalam sepuluh (10) tahun terakhir.

UA-FLP pertama kali dikembangkan oleh Armour dan Buffa (1963). Mereka menggambarkan bahwa ada suatu fasilitas berbentuk persegi panjang dengan panjang (*Width*) dan lebar (*Height*) yang tetap dan terdapat beberapa (n) jumlah departemen yang perlu ditempatkan ke dalam fasilitas tersebut. Masing-masing departemen tersebut memiliki luas yang tetap dengan panjang dan lebar tertentu ($width \times height$). Arus perpindahan material antara departemen serta biaya perpindahannya diketahui.

Tujuannya adalah membagi luas fasilitas tersebut menjadi beberapa sub-wilayah sesuai dengan jumlah departemen, sehingga dapat meminimalkan biaya perpindahan material (*material handling*). Fungsi tujuan untuk meminimalkan biaya perpindahan material adalah sebagai berikut:

$$\text{Total Cost} = \sum_{i=1}^n \sum_{j=1, i \neq 1}^n d_{ij} f_{ij} c_{ij}$$

n = jumlah departemen

f_{ij} = jumlah arus material antara departemen i dan departemen j , dimana $i, j = 1, 2, \dots, n$

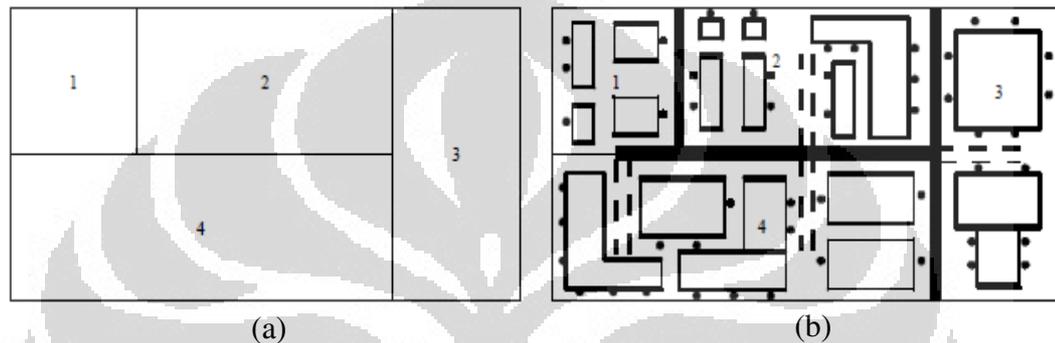
d_{ij} = jarak antara departemen i dan departemen j , dimana $i, j = 1, 2, \dots, n$

c_{ij} = biaya perpindahan satu unit material per unit jarak dari departemen i ke departemen j , dimana $i, j = 1, 2, \dots, n$

Luas tiap departemen tidak perlu sama, tetapi dimensinya dibatasi oleh kendala rasio maksimum (α_{\max}) atau minimum panjang (l_{\min}), untuk panjang (*width*) dan atau tinggi (*height*) tiap departemen. Kendala rasio maksimum (α_{\max}) merupakan suatu nilai maksimum yang tidak boleh dilanggar oleh hasil bagi antara panjang dengan tinggi atau tinggi dengan panjang (nilai yang lebih kecil menjadi pembagi).

$$\alpha_{\max} = \frac{\max(\text{width}, \text{height})}{\min(\text{width}, \text{height})}$$

Dalam penyelesaian masalah tata letak fasilitas akan dihasilkan suatu *block layout* (seperti pada **Gambar 2.1.(a)**), yang menunjukkan letak tiap departemen di dalam suatu fasilitas. Suatu *block layout* dapat dikembangkan lebih lanjut untuk mendapatkan *detailed layout*, yang menunjukkan letak departemen sebenarnya, struktur *aisle* (jalan), posisi *Input-Output*, dan tata letak dalam masing-masing departemen. Terdapat dua cara dalam perhitungan jarak yang digunakan dalam masalah tata letak fasilitas, yaitu:



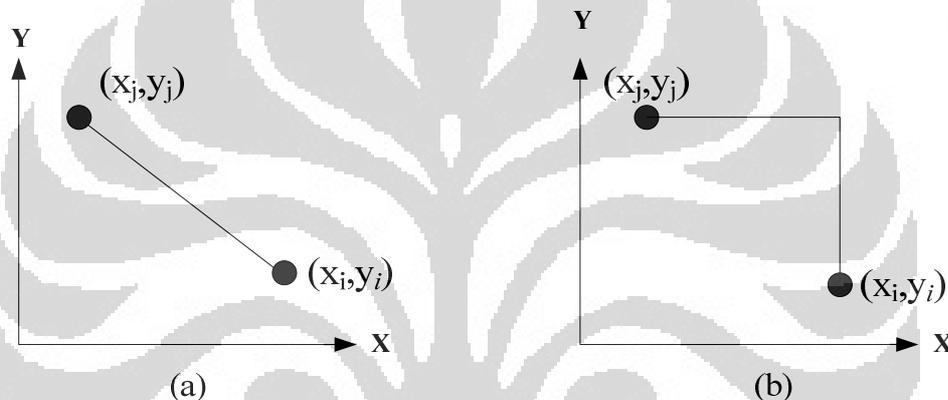
Gambar 2.1. Contoh tampilan (a) *Block Layout* dan (b) *Detailed Layout*

(Sumber : Komarudin, 2009)

- a. *Input-Output (I/O) Points Distance*, jarak diukur berdasarkan tempat *Input-Output* antara dua departemen dan dalam beberapa kasus diukur sepanjang *aisle* antara dua departemen yang bersangkutan (Trethewey dan Foote, 1994). Kekurangan utama dari cara ini adalah jika posisi dari tempat *Input-Output* dan/atau *aisle* antara dua departemen tidak diketahui secara jelas, sampai dibuat tata letak yang lebih detail (Meller dan Gau, 1996).
- b. *Centroid-to-Centroid (CTC) Distance*, jarak diukur antara titik pusat (*centroid*) dari dua departemen. Titik pusat departemen akan merepresentasikan tempat *Input-Output* departemen tersebut. Algoritma dengan dasar CTC berusaha untuk menempatkan titik pusat sedekat mungkin, sehingga jika tidak ada syarat kendala bentuk yang baik, bentuk dari departemen yang dihasilkan akan terlalu panjang dengan lebar (tinggi) yang pendek (Tate dan Smith, 1995). Kelemahan lainnya, departemen

dengan bentuk L mungkin dapat memiliki titik pusat yang jatuh di luar dari daerah departemen tersebut (Francis dan White, 1974).

Terdapat dua cara perhitungan untuk menghitung jarak antara dua departemen tersebut, yaitu *Rectilinear* dan *Euclidean*. Jarak *rectilinear* menghitung jarak dari suatu garis yang paralel dengan kedua sumbu (tegak lurus), yang menghubungkan dua titik. Jarak euclidean merupakan metode yang menghitung jarak suatu garis lurus yang menghubungkan dua titik. Jarak *rectilinear* merupakan metode penghitungan yang paling umum digunakan dalam penyelesaian masalah tata letak fasilitas.



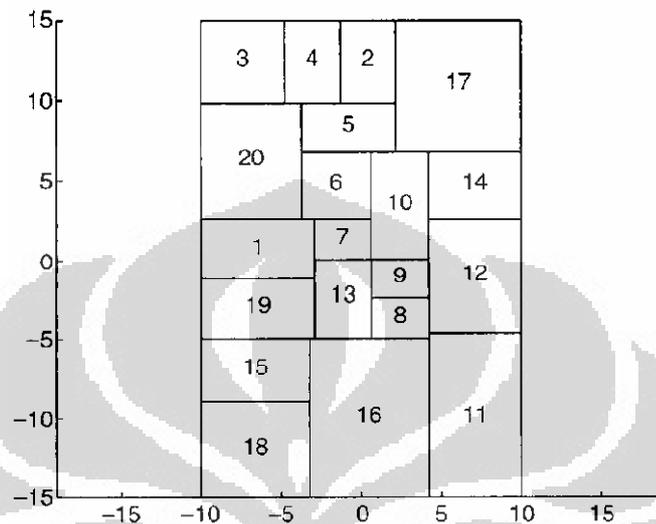
Gambar 2.2. Penghitungan (a) jarak *Euclidean* dan (b) jarak *Rectilinear*

Hasil *output* UA-FLP dapat dimodelkan menggunakan berbagai macam model representasi. Setiap model representasi mempunyai karakteristik dan kompleksitasnya sendiri. Terdapat empat model representasi berdasarkan karakteristik dari tata letak akhir (Komarudin, 2009), yaitu:

- Model Kontinu

Model kontinu memiliki karakteristik dimana penempatan setiap departemen dan dimensi pada fasilitas tidak dibatasi oleh apapun selain masalah itu sendiri. Hal ini membuat model memiliki kemungkinan untuk menghasilkan semua solusi tata letak yang mungkin, termasuk solusi optimal yang tepat. Meskipun demikian, model menjadi sulit untuk diselesaikan ketika solusi semakin kompleks. Hal ini membuat adanya pengembangan model lain yang lebih mudah untuk diselesaikan.

Gambar 2.3. merupakan contoh tata letak akhir dari model representasi kontinu yang tidak dapat dihasilkan oleh model representasi lainnya.



Gambar 2.3. Contoh Solusi menggunakan representasi Kontinu

(Sumber : Komarudin, 2009)

- Model *Quadratic Assignment Problem* (QAP)

Pada model QAP, seluruh departemen dibagi menjadi kotak-kotak kecil dengan luas yang sama. Kemudian algoritma akan mencoba mengumpulkan setiap kotak yang berasal dari departemen yang sama menjadi sebuah wilayah yang terkoneksi satu sama lain. Ini dapat dihasilkan dengan menambahkan arus material buatan yang sangat besar antar kotak dari departemen yang sama. **Gambar 2.4.** memberikan contoh hasil dari pemodelan ini. Model ini seringkali menghasilkan *blocks* yang tidak berbentuk persegi atau kotak.

8	8	8	8	7	7	7	7	7	7
8	8	8	8	7	7	7	7	10	10
9	9	8	3	3	3	7	7	10	10
9	9	3	3	3	3	10	10	10	10
1	1	3	3	3	3	10	10	10	10
1	1	3	6	6	6	5	10	10	10
1	1	6	6	6	6	5	5	5	5
1	1	1	6	6	6	4	4	2	2
1	1	1	6	6	6	4	4	2	2
1	1	1	6	6	6	4	4	4	2

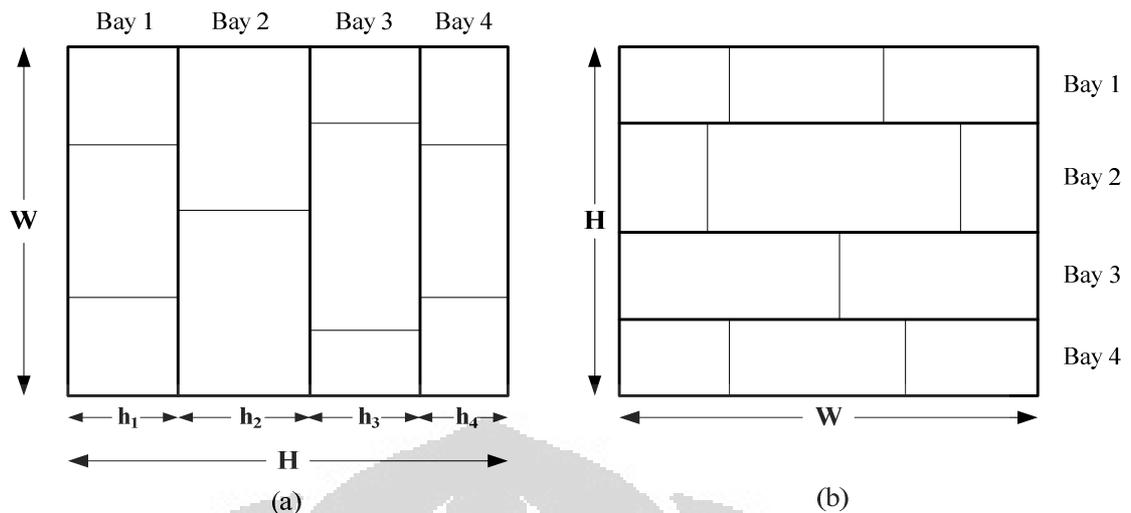
Gambar 2.4. Contoh solusi dengan model QAP oleh Hardin dan User (2005)

(Sumber : Komarudin, 2009)

- Model *Flexible Bay Structure* (FBS)

Merupakan suatu model representasi berbentuk *block layout* yang menempatkan departemen pada kolom-kolom atau *bays* dengan tinggi kolom yang berbeda-beda. Tiap kolom dibatasi oleh *aisle* lurus dan tiap departemen hanya diperbolehkan menempati satu kolom saja. Tiap *bay* diharapkan terisi secara penuh oleh departemen-departemen, karena itu diperlukan suatu kombinasi departemen yang tepat yang dapat mengisi tiap *bay* sekaligus memenuhi kendala bentuk yang ada.

Tinggi dari tiap kolom (h) akan ditentukan oleh jumlah departemen yang dimilikinya, sehingga secara otomatis tiap departemen yang ada di dalamnya akan memiliki tinggi yang sama.



Gambar 2.5. Contoh solusi UA-FLP dengan model *Flexible Bay Structure*

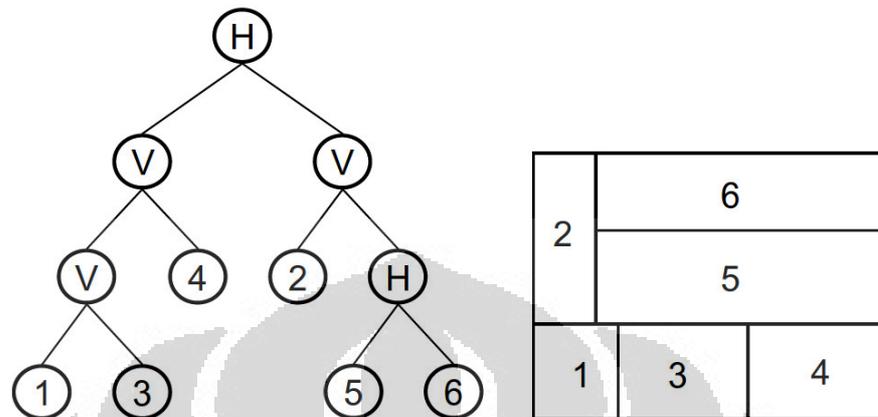
Dengan demikian masalah menjadi lebih sederhana dan lebih mudah untuk dipecahkan. Kompleksitas masalah berkurang dan hanya perlu menentukan urutan penempatan departemen dan jumlah total departemen yang dimiliki setiap kolom.

Model ini memiliki keuntungan dimana struktur pembatasnya akan menjadi kandidat untuk *aisle* dan dapat mengubah model menjadi suatu fasilitas yang *detail*. Model ini dapat memiliki *vertical bays*, seperti terlihat pada **Gambar 2.5.(a)** dan juga *horizontal bays* seperti yang terlihat pada **Gambar 2.5.(b)**.

- Model *Slicing Tree Structure* (STS)

Pada model STS, hasil akhir tata letak dapat dideskripsikan sebagai representasi sebuah pohon. Representasi pohon ini memiliki cabang-cabang yang merupakan departemen dan cabang-cabang penghubung. Cabang penghubung ini terdiri dari dua, yaitu Horizontal (H) dan Vertikal (V), yang akan berperan sebagai operator pemotong. Kedua operator ini akan menentukan letak departemen, jika horizontal maka letak departemen akan memiliki potongan secara horizontal dan begitu juga dengan pemotongan secara vertikal. Cabang-cabang departemen dan operator pemotong menentukan lokasi departemen dan letak departemen

sekitarnya. Model ini telah digunakan oleh Scholz *et al.* (2009) dan Komarudin & Wong (2010).



Gambar 2.6. Contoh solusi UA-FLP dengan model STS

(Sumber : Scholz *et al.*, 2009)

2.2 Simulated Annealing

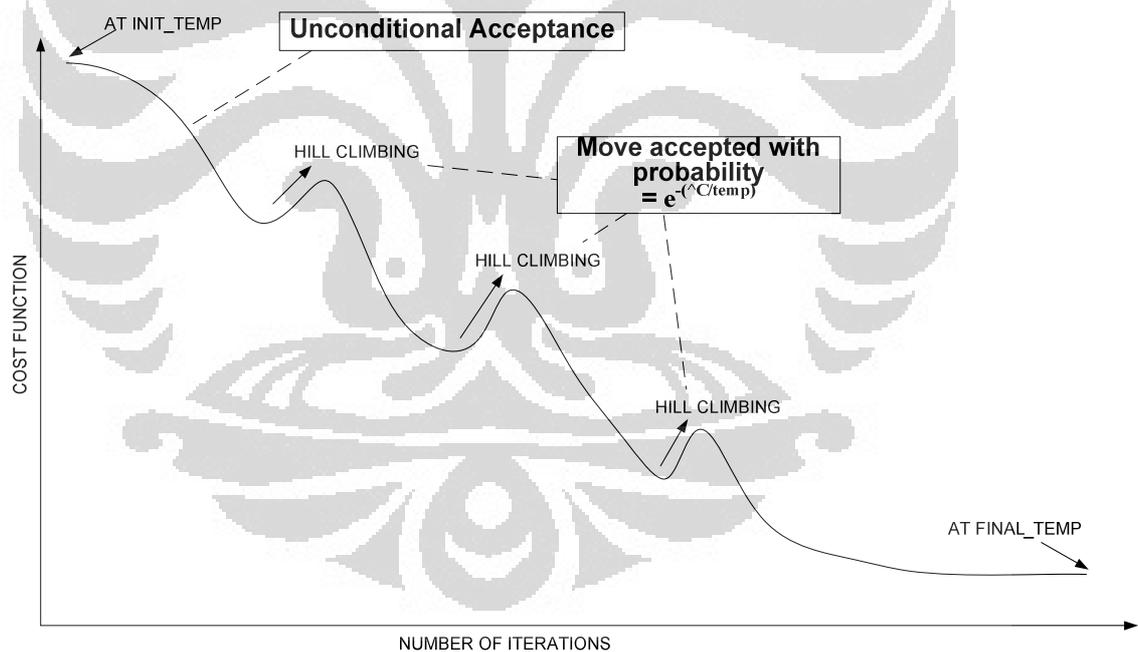
Simulated Annealing merupakan sebuah metode berbasis *Monte Carlo Simulation*, yang dapat digunakan untuk menyelesaikan masalah-masalah optimasi. *Simulated Annealing* diperkenalkan pertama kali oleh Metropolis *et al.* pada tahun 1953, dan aplikasinya dalam masalah optimasi dilakukan pertama kali oleh Kirkpatrick *et al.* tahun 1983. Nama metode ini diambil dari analogi dengan perilaku fisik suatu sistem, dimana suatu material benda padat dilelehkan dengan cara menurunkan suhu secara perlahan-lahan hingga mencapai *freezing point*.

Dari sisi ilmu fisika, tujuan sistem ini adalah untuk meminimalkan energi potensial. Fluktuasi kinematika acak menghalangi sistem untuk mencapai energi potensial yang minimum global, sehingga sistem dapat terperangkap dalam sebuah keadaan minimum lokal. Dengan menurunkan temperatur, diharapkan energi dapat berkurang ke suatu level yang relatif rendah. Semakin lambat laju pendinginan ini, semakin rendah pula energi yang dapat dicapai oleh sistem pada akhirnya.

Simulated annealing merupakan salah satu algoritma iteratif yang menggunakan ambang batas yang positif dan stokastik. Ambang batas ini semakin turun seiring dengan waktu. Dengan demikian, semakin mendekati akhir algoritma, solusi-solusi yang tidak melakukan perbaikan (*non-improving*) memiliki probabilitas yang semakin kecil untuk diterima.

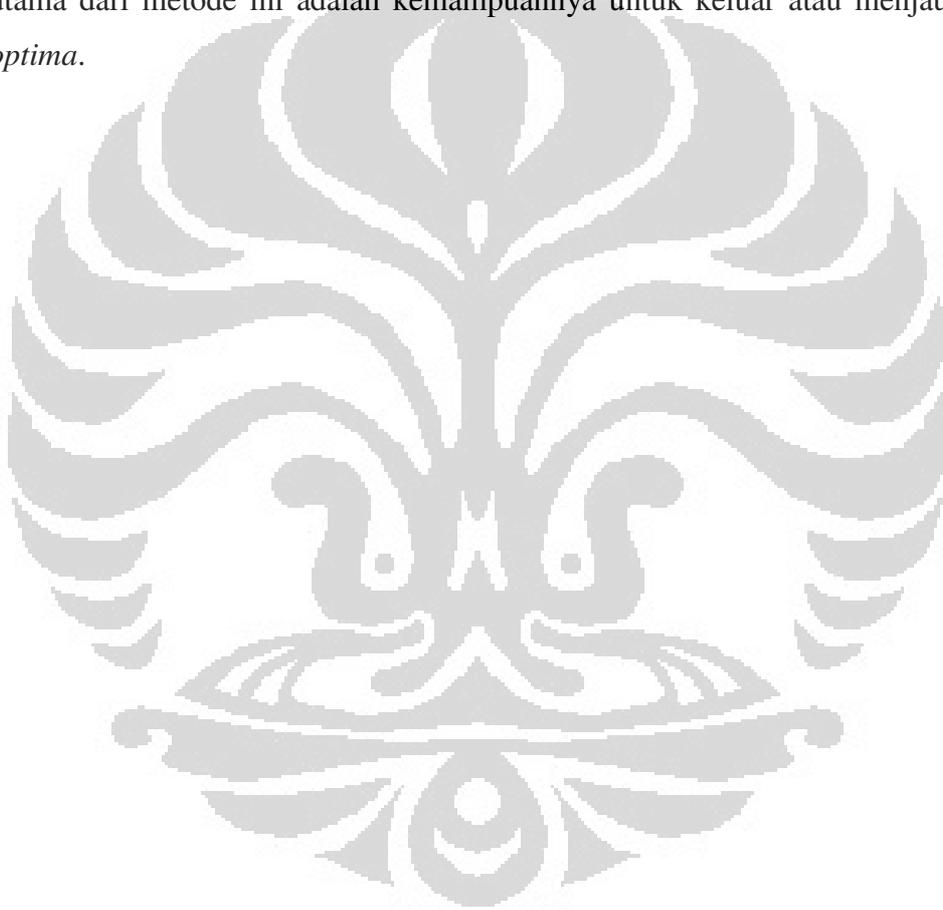
Simulated Annealing menggunakan konsep *neighbourhood search* atau *local search* dan pada setiap iterasi melakukan pencarian tersebut. Perubahan kondisi saat ini menjadi kondisi baru (tetangga) disebut dengan langkah (*move*).

Proses ini dimulai pada sebuah kondisi awal, dimana terjadi pergerakan secara acak ke sebuah kondisi yang baru, tetapi tetap pada lingkungan (tetangga) kondisi awal, dimana terjadi perubahan energi (ΔE) pada fungsi tujuan. Perubahan ΔE di dalam fungsi tujuan mengakibatkan perubahan level energi (di dalam analogi), yang muncul saat T menurun.



Gambar 2.7. Proses Penurunan Suhu Algoritma SA

Jika perubahan $\Delta E < 0$, bernilai negatif, maka kondisi baru tersebut diterima. Tetapi jika $\Delta E > 0$, dengan kata lain bernilai positif, maka kondisi baru tersebut akan diterima dengan probabilitas tertentu $p(\Delta E) = e^{\frac{-\Delta E}{k_B \cdot T}}$, dimana T merupakan parameter kontrol suhu. Semakin tinggi suhu (T), maka semakin tinggi probabilitas terjadinya “*hill climbing*”. Kirkpatrick *et al.* (1983) menyatakan bahwa dengan membiarkan terjadinya pergerakan “*hill climbing*”, yang menambah nilai fungsi tujuan, dapat menghindari konfigurasi menuju solusi lokal optimum. Dengan demikian hasil yang lebih baik dapat dicapai. Keuntungan utama dari metode ini adalah kemampuannya untuk keluar atau menjauhi *local optima*.



BAB 3 PENGUMPULAN DATA

Sebuah metode yang telah dikembangkan tidak akan dapat dinilai dan dievaluasi jika tidak digunakan dalam penyelesaian suatu masalah. Karena itu, metode dalam penelitian ini akan diaplikasikan dan digunakan dalam menyelesaikan masalah yang terkait dengan UAFLP. Seluruh masalah-masalah (seterusnya akan disebut dengan *problem sets*) yang digunakan dapat dilihat pada **Tabel 3.1.**, disertai juga dengan data-data pendukung lainnya. *Problem sets* yang digunakan sudah dikenal luas dan digunakan oleh para peneliti dalam penelitian-penelitian sebelumnya yang berkaitan dengan UA-FLP.

Tabel 3.1. *Problem Sets Unequal Area Facility Layout Problem (UAFLP)*

No.	Problem Set	Jumlah Departemen	Ukuran Fasilitas		Kendala Bentuk	Referensi
			Width	Height		
1.	O7	7	8.54	13.00	$\alpha_{\max} = 4$	Meller <i>et al.</i> (1999)
2.	FO7	7	8.54	13.00	$\alpha_{\max} = 4$	Meller <i>et al.</i> (1999)
3.	FO8	8	11.31	13.00	$\alpha_{\max} = 4$	Meller <i>et al.</i> (1999)
4.	O9	9	12.00	13.00	$\alpha_{\max} = 4$	Meller <i>et al.</i> (1999)
5.	V10s	10	25.00	51.00	$l_{\min} = 5$	van Camp <i>et al.</i> (1992)
6.	V10a	10	25.00	51.00	$\alpha_{\max} = 5$	van Camp <i>et al.</i> (1992)
7.	M11s	11	6.00	6.00	$l_{\min} = 1$	Meller (1992)
8.	M11a	11	6.00	6.00	$\alpha_{\max} = 5$	Meller (1992)
9.	M15s	15	15.00	15.00	$l_{\min} = 1$	Meller (1992)
10.	M15a	15	15.00	15.00	$\alpha_{\max} = 5$	Meller (1992)
11.	M25	25	15.00	5.00	$\alpha_{\max} = 5$	Meller (1992)
12.	NUG12	12	3.00	4.00	$\alpha_{\max} = 4$	Nugent <i>et al.</i> (1968)
13.	NUG15	15	3.00	5.00	$\alpha_{\max} = 4$	Nugent <i>et al.</i> (1968)
14.	BA12	12	10.00	6.00	$l_{\min} = 1$	Bazaraa (1975)
15.	BA12TS	16	10.00	6.00	$l_{\min} = 1$	Tate and Smith (1995)
16.	BA14	14	9.00	7.00	$l_{\min} = 1$	Bazaraa (1975)
17.	BA14TS	14	9.00	7.00	$l_{\min} = 1$	Tate and Smith (1995)
18.	AB20	20	2.00	3.00	$\alpha_{\max} = 4$	Armour and Buffa (1963)
19.	SC30	30	15.00	12.00	$\alpha_{\max} = 4$	Liu and Meller (2007)
20.	SC35	35	16.00	15.00	$\alpha_{\max} = 4$	Liu and Meller (2007)

Huruf alfabet dari tiap *problem set* (kecuali untuk 4 *problem sets* awal dan 2 *problem sets* akhir) menunjukkan inisial dari nama penemu atau pencetus masalah tersebut, dapat dilihat pada bagian referensi. Angka yang tertulis menunjukkan jumlah departemen yang dimiliki oleh tiap masalah tersebut. Disertakan juga data mengenai ukuran fasilitas, berupa panjang (W) dan tinggi (H). Untuk membatasi dimensi tiap departemen dalam menyusun suatu fasilitas diperlukan kendala bentuk (*common shape constraint*), dalam hal ini diberikan dalam *maximum aspect ratio* (α_{\max}) dan panjang minimum (l_{\min}). Penjelasan mengenai kedua kendala ini dapat dibaca lebih lengkap pada sub-bab 2.1.1. Data arus aliran material (*material flow*) dan luas tiap-tiap departemen pada masing-masing *problem set* dapat dilihat pada **Lampiran 1**.

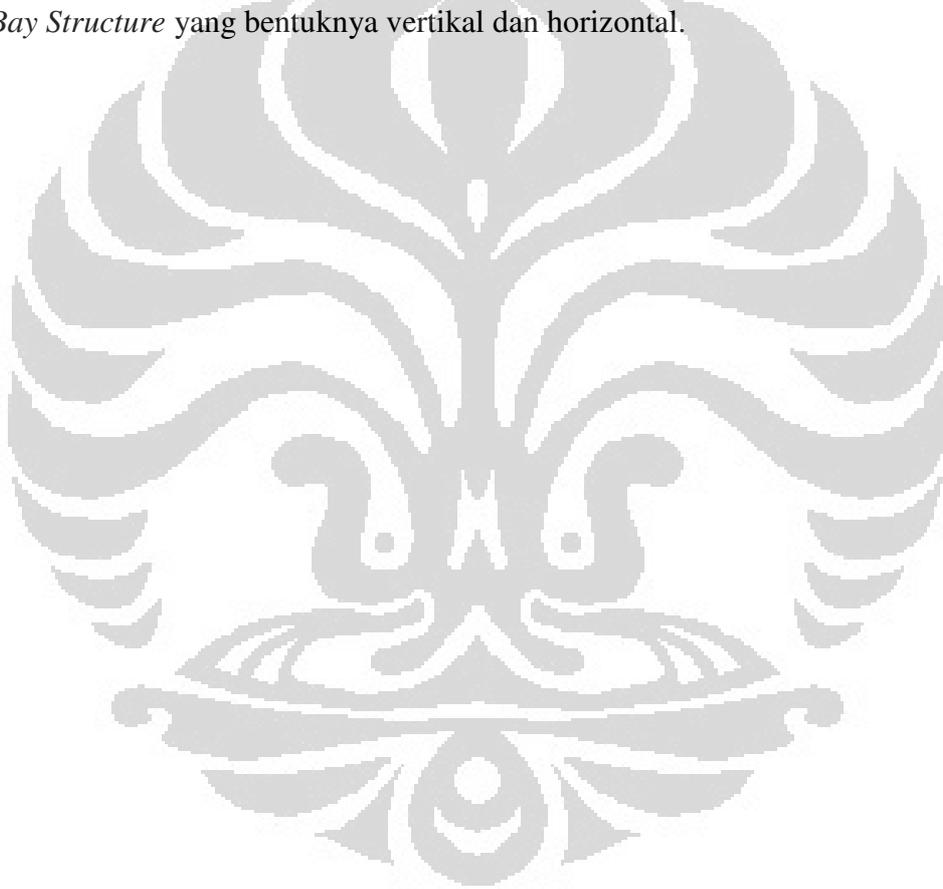
Seluruh data ini dikumpulkan secara langsung dari jurnal atau laporan akhir para pencetus atau pembuat *problem set* yang tertulis di referensi. Hal ini dilakukan, walaupun sebagian besar *problem sets* sudah cukup lama dicetuskan (lihat tahun), agar penelitian ini memiliki data-data yang lengkap dan benar, karena penelitian-penelitian sebelumnya, menggunakan data-data yang sama, tidak menyajikan keseluruhan data secara lengkap dan benar.

Khusus untuk data kendala bentuk, penelitian-penelitian yang telah dilakukan sebelumnya memodifikasi data ini. Ini memang dapat dilakukan karena kendala ini memang dapat dimodifikasi dengan tujuan mendapatkan hasil yang lebih baik. Data kendala bentuk yang ditampilkan pada **Tabel 3.1**, sebagian besar bukan merupakan data asli, tetapi mengikuti penelitian-penelitian sebelumnya yang menggunakan data kendala bentuk yang sama. Ini dilakukan agar hasil perhitungan penelitian ini dapat dibandingkan dengan hasil dari penelitian-penelitian sebelumnya.

Beberapa *problem set* memiliki suatu ruang kosong (*empty spaces*), yang muncul karena total luas departemen yang akan ditempatkan dalam fasilitas tidak sepenuhnya memenuhi luas fasilitas yang bersangkutan. Terdapat suatu ruang kosong yang akan terlihat pada tampilan hasil tata letak. Penelitian ini tidak mengikutsertakan ruang kosong di dalam perhitungan, walaupun beberapa penelitian sebelumnya mengikutsertakan perhitungan ruang kosong (salah satunya Kulturel-Konak dan Konak, 2011), yang seakan-akan menjadi *dummy*

department. Penggunaan *dummy department* hanya digunakan pada *problem BA12TS*, yang memang memiliki 4 (empat) *dummy department*, sesuai dengan yang dicetuskan oleh Tate dan Smith (1995).

Terdapat beberapa *problem* yang dimodifikasi dalam penelitian ini dengan tetap mempertahankan data yang asli. Yang dilakukan hanyalah menukar posisi Panjang (*Width*) dan Tinggi (*Height*) dari fasilitas. Ini dilakukan agar dapat menghasilkan model *Flexible Bay Structure* dengan *horizontal bays* dan juga *vertical bays*, seperti yang dijelaskan pada sub-bab 2.1.1. Hal ini terinspirasi dari penelitian Kulturel-Konak dan Konak (2011), yang memiliki hasil model *Flexible Bay Structure* yang bentuknya vertikal dan horizontal.



BAB 4 PENGOLAHAN DATA DAN ANALISA

4.1 Pengolahan Data

Pada bagian ini diberikan penjelasan mengenai konsep penyusunan program yang digunakan dalam pengembangan metode ini. Sebagai metode yang dikembangkan, program juga telah menguji *problem sets* yang telah digunakan pada penelitian-penelitian sebelumnya. Ini akan menjadi parameter baik tidaknya metode yang dikembangkan pada penelitian ini. Hasil dari pengujian ini akan ditampilkan pada sub-bab 4.1.3 bab ini.

Penelitian ini menggunakan software Matlab 7.9.0 529 (R2009b) dalam penyusunan program dan menggunakan Microsoft Excel 2007 untuk menghasilkan *output* gambar tata letak terbaik dari tiap *problem set* (gambar tata letak terbaik dapat dilihat pada **Lampiran 2**).

4.1.1 Penyusunan Program

Penyusunan program dilakukan dengan fokus pada model representasi *Flexible Bay Structure* dan mempertimbangkan semua kendala yang terdapat pada model tersebut. Program ini akan menjadi sebuah *constraint solver* yang memenuhi setiap kendala yang dihadapi dalam proses pembentukan model *Flexible Bay Structure*, langkah demi langkah, mulai dari tahap awal sampai akhir.

Berdasarkan pemahaman yang dimiliki terhadap model representasi *Flexible Bay Structure*, penelitian ini membagi langkah-langkah penyusunan program menjadi 4 (empat) bagian, yaitu menentukan *Feasible Subsets*, menentukan Kombinasi, mencari Koordinat, dan mencari *Objective Function*.

Untuk memudahkan pengertian dari pembaca, maka diberikan sebuah contoh kasus dari *problem O7* dengan urutan departemen yang ditentukan secara *random*, yakni 7 – 5 – 4 – 3 – 2 – 1 – 6. Kasus ini sekaligus menjadi kasus yang digunakan dalam proses penyusunan program, serta untuk melakukan verifikasi dan validasi program.

4.1.1.1 Menentukan *Feasible Subsets*

Dalam model representasi *Flexible Bay Structure*, penempatan departemen membentuk kolom-kolom atau *bays*. Sebuah *bay* mungkin dapat ditempati oleh satu departemen, dua departemen, tiga departemen, dan seterusnya, bahkan mungkin sampai seluruh jumlah departemen yang ada dalam suatu urutan departemen tertentu. Berdasarkan hal tersebut, dapat diketahui berapa jumlah *bay* yang dapat terbentuk oleh suatu urutan departemen tertentu, beserta departemen-departemen yang membentuk *bays* tersebut. Jumlah *bay* dalam penelitian ini dinamakan *subsets*.

Jumlah *subsets* yang mungkin dihasilkan oleh suatu urutan departemen dapat dihitung dengan persamaan berikut:

$$S_n = \frac{n}{2} (n+1)$$

Dalam penelitian ini hanya akan dicari *subsets* yang *feasible* dan dapat digunakan untuk mencari *objective function*. *Feasible subsets* artinya jumlah *bay* yang tiap anggota departemennya memenuhi kendala panjang minimal (l_{\min}) yang telah ditetapkan. Jika diketahui kendala *maximum aspect ratio* (α_{\max}), maka nilai panjang minimum dapat dihitung dengan menggunakan persamaan berikut:

$$\text{Minimum Side} = \sqrt{\frac{\text{Department Area}}{\text{Max Aspect Ratio } (\alpha_{\max})}}$$

Untuk menentukan dimensi dari departemen-departemen yang menyusun suatu *bay*, akan dilakukan penghitungan tinggi dari *bay*. Penghitungan *bay height* dilakukan dengan membagi jumlah total luas departemen anggota (dalam *bay*) dengan panjang fasilitas. Tinggi *bay* yang didapatkan sekaligus akan menjadi tinggi dari tiap departemen anggotanya.

$$\text{Bay Height} = \frac{\text{Total Department Area}}{\text{Facility Width}}$$

Setelah mendapatkan tinggi dari *bay* (sekaligus tinggi tiap departemen), maka panjang dari tiap-tiap departemen dalam *bay* dapat dihitung dengan persamaan berikut:

$$\text{Width} = \frac{\text{Total Department Area}}{\text{Bay Height}}$$

Setelah data panjang (*w*) dan lebar (*h*) masing-masing departemen didapatkan, maka diadakan pemeriksaan kendala panjang minimum. Jika salah satu saja departemen tidak dapat memenuhi kendala tersebut, maka *bay* tersebut menjadi tidak *feasible* dan tidak dapat digunakan. Hasil yang tidak *feasible* akan langsung dihilangkan sehingga hanya akan diambil *bay* atau *subset* yang *feasible*. Ini mengurangi kemungkinan solusi yang dapat muncul serta mengurangi kompleksitas.

Pada *problem 07*, dengan urutan departemen 7 – 5 – 4 – 3 – 2 – 1 – 6, hal pertama yang dilakukan adalah pengujian apakah satu atau lebih departemen dapat ditempatkan dalam sebuah *bay* atau tidak. Departemen 7, selaku urutan departemen pertama, akan dicoba apakah dapat menempati sendiri satu *bay*, ternyata dari hasil perhitungan tidak bisa. Lalu dicoba departemen 7 dan 5, hasilnya kedua departemen ini dapat membentuk *bay* dan memenuhi kendala yang ada sehingga membentuk sebuah *subset* yang *feasible*. Hal ini akan dilanjutkan dengan departemen 7, 5, dan 4, dan seterusnya sampai departemen 7 hingga 6 secara keseluruhan diuji penempatannya ke dalam satu *bay*. Kemudian pengujian akan dilanjutkan mulai dari urutan departemen kedua, yaitu departemen 5 dan seterusnya dengan cara yang sama sampai pengujian departemen terakhir, yaitu departemen 6 sendiri.

Perlu diketahui bahwa penambahan departemen ke dalam suatu *bay* tidak akan membuat *bay* tersebut menjadi *feasible*. Artinya jika departemen-departemen yang membentuk suatu *bay* tidak *feasible*, maka penambahan departemen lainnya tidak perlu dilakukan lagi, kecuali untuk departemen yang diuji sendiri. Hal ini dikarenakan nilai tinggi *bay* akan naik seiring dengan penambahan luas departemen, dan akan menurunkan nilai panjang (*w*) dari suatu departemen,

sehingga departemen tersebut tetap tidak akan memenuhi kendala panjang minimum, karena sebelumnya saja sudah tidak *feasible*.

Hasil *feasible subsets* yang didapatkan dari contoh kasus ini adalah sejumlah 9 (sembilan), artinya ada 9 *bay* yang mungkin dan akan yang digunakan untuk langkah selanjutnya. *Subsets* tersebut adalah {7 5}, {4}, {4 3}, {4 3 2}, {3 2}, {3 2 1}, {2 1}, {2 1 6}, dan {1 6}.

4.1.1.2 Menentukan Kombinasi yang Mungkin

Dari seluruh *feasible subset (bay)* yang dihasilkan akan dibuat kombinasi-kombinasi yang membentuk solusi yang *feasible* untuk sebuah model representasi *Flexible Bay Structure*. Jumlah kombinasi yang mungkin dihasilkan dari n *feasible subsets (bay)* dapat ditentukan dengan persamaan kombinasi berikut:

$$\sum_{i=1}^n {}_n C_i$$

dimana,

$${}_n C_r = \frac{n!}{r!(n-r)!}$$

n = jumlah *feasible subsets*

r = jumlah kombinasi yang dapat dilakukan ($r = 1, \dots, n$)

Kombinasi akan selalu dimulai dari *subset* atau *bay* yang memuat departemen pertama yang berada pada urutan departemen, dengan kata lain memiliki indeks 1. Setelah itu, kombinasi akan mencari *subset* atau *bay* yang memiliki urutan departemen dengan urutan setelah departemen terakhir pada *subset* atau *bay* sebelumnya. Proses ini akan berjalan terus sampai kombinasi selesai dihasilkan.

Dari kesembilan *subsets* yang ada, *bay* yang mungkin menjadi *bay* pertama adalah yang memiliki urutan departemen nomor satu, yaitu {7 5}. Jadi kombinasi yang muncul akan selalu diawali dengan *bay* tersebut. *Bay* berikutnya ditentukan oleh departemen 5, artinya *bay* selanjutnya akan dimulai dengan urutan

departemen ketiga, yaitu departemen 4. Hal ini akan berjalan terus sampai tercipta kombinasi yang *feasible* dan secara langsung mengeliminasi yang tidak mungkin.

Kombinasi yang *feasible* dan dapat digunakan dalam proses perhitungan dari kasus ini ada 3 (tiga), yaitu [7 5 | 4 | 3 2 | 1 6], [7 5 | 4 3 | 2 1 6], dan [7 5 | 4 3 2 | 1 6]. Garis lurus pada kombinasi menunjukkan pemisah atau *bay break*. Dari kombinasi ini akan dicari koordinat tiap departemen sehingga jarak antar departemen dapat dicari.

4.1.1.3 Mencari Koordinat tiap Departemen dalam Kombinasi

Data jarak menjadi data yang penting untuk diketahui dalam proses penentuan fungsi tujuan. Oleh karena itu perlu dicari koordinat titik pusat dari tiap departemen sehingga jarak dapat dihitung. Koordinat yang digunakan dalam pengembangan metode ini adalah *centroid*, karena *layout* yang digunakan adalah *block layout*, sehingga tidak diketahui keterangan mengenai tempat I/O dari departemen.

Penghitungan koordinat akan dimulai dengan menghitung kembali *bay height*, dengan membagi jumlah luas departemen dengan panjang fasilitas. Ini akan menjadi tinggi dari *bay*, dengan kata lain menjadi lebar (tinggi) dari masing-masing departemen yang terdapat di dalam *bay* tersebut.

Penentuan koordinat departemen akan selalu dimulai dari *subset* atau *bay* pertama. Panjang dari tiap departemen dalam satu *bay* akan dihitung dengan membagi luas departemen dengan lebar departemen. Setelah didapatkan data panjang dan lebar masing-masing departemen, maka panjang dan lebar nya masing-masing akan dibagi dua untuk mendapatkan titik pusat (*centroid*) dari departemen tersebut.

Penentuan koordinat Y untuk tiap departemen akan mengikuti ukuran tinggi *bay* dibagi dua. Perlu diingat bahwa untuk *bay* berikutnya, tinggi *bay* sekarang harus ditambahkan tinggi *bay* sebelumnya, baru dibagi dua untuk mendapatkan koordinat departemen pada *bay* berikutnya. Untuk penentuan koordinat X, penentuan koordinat departemen dengan urutan pertama pada tiap *bay* dapat langsung dilakukan dengan membagi panjang departemen menjadi dua. Untuk departemen setelah urutan departemen pertama, harus ditambahkan

panjang departemen sebelumnya. Hal ini perlu diingat karena proses penentuan koordinat tetap mengacu pada titik (0,0) yang terletak pada pojok kiri bawah, seperti pada sumbu Cartesius.

4.1.1.4 Mencari *Objective Function* terbaik

Dengan koordinat-koordinat dari tiap departemen, dapat dihitung jarak antar tiap departemen. Jarak yang digunakan adalah jarak *rectilinear* dari titik pusat departemen i ke titik pusat departemen j , yang dinyatakan dengan persamaan:

$$d_{ij} = |x_i - x_j| + |y_i - y_j|$$

Untuk menghitung fungsi tujuan, digunakan fungsi tujuan yang dibahas pada sub-bab 2.1.1. Jarak antar departemen akan dikalikan dengan *material flow* antar departemen tersebut, sehingga akan didapatkan biaya material. Keseluruhan biaya material akan dijumlahkan untuk mendapatkan biaya material model representasi *Flexible Bay Structure* untuk urutan dan kombinasi departemen tersebut.

Dalam contoh kasus yang digunakan, biaya *cost* yang terbaik dipilih dari ketiga kombinasi yang ada sebelumnya. Biaya *cost* yang terbaik adalah sebesar 183.205 dengan kombinasi [7 5 | 4 3 2 | 1 6]. Terdapat tiga *bay* pada solusi ini. Perlu diingat bahwa solusi ini hanya berdasarkan urutan departemen 7 – 5 – 4 – 3 – 2 – 1 – 6. Masih terdapat banyak kemungkinan urutan departemen lainnya.

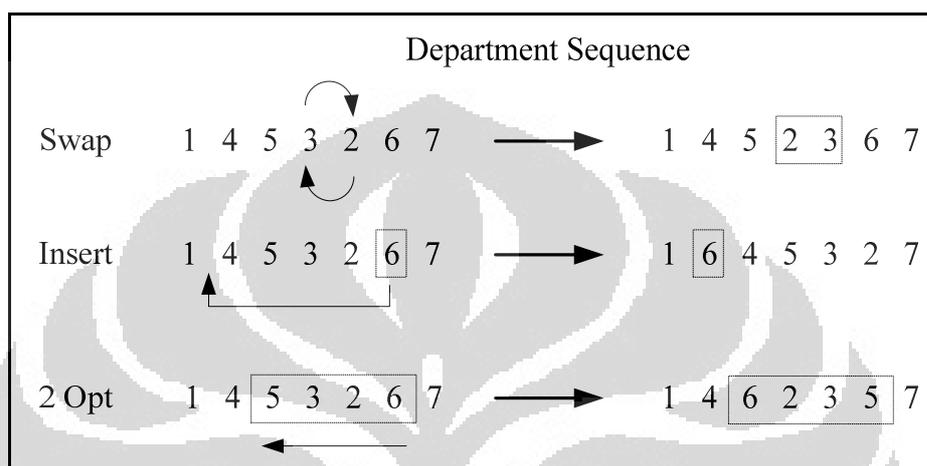
Untuk mencari nilai global optimum dari fungsi tujuan, digunakan algoritma *Simulated Annealing* untuk optimasi. Langkah-langkah iterasi dari algoritma *Simulated Annealing*, yang digunakan dalam program adalah sebagai berikut:

- Langkah pertama: *Initialize*

Kondisi awal dihasilkan dengan mengacak *random* urutan departemen secara permutasi. Hasilnya akan menjadi solusi awal urutan departemen. Hal ini dilakukan saat suhu awal yang tinggi.

- Langkah kedua: *Move*

Urutan departemen dari solusi akan mengalami pergerakan dengan konsep *neighbourhood search* atau *local search*. Pada metode ini, pergerakan yang diaplikasikan adalah *Swap*, *Insert*, dan *2Opt*. Pemilihan pergerakan dilakukan secara *random*.



Gambar 4.1. Pergerakan dalam *Local Search*

- Langkah ketiga: *Calculate Score*

Penghitungan fungsi tujuan kembali dilakukan berdasarkan urutan departemen baru, yang dihasilkan dari pergerakan pada langkah sebelumnya.

- Langkah keempat: *Choose*

Berdasarkan nilai fungsi tujuan yang baru akan dibandingkan dengan nilai fungsi tujuan sebelumnya. Jika lebih baik, maka nilai fungsi tujuan baru akan diterima. Jika lebih buruk, maka nilai fungsi tujuan baru akan diterima berdasarkan probabilitas yang ditentukan oleh suhu saat itu. Jika diterima, maka akan terjadi "*hill climbing*", keluar dari lingkungan *local search*.

- Langkah kelima: *Update and Repeat*

Solusi dicatat dan disimpan sebagai solusi terbaik, kemudian dilakukan penurunan suhu. Pergerakan urutan departemen kembali dilakukan, kembali ke langkah kedua. Proses ini akan berlangsung sampai suhu mencapai suhu akhir (*freezing point*) yang telah ditentukan.

4.1.2 Verifikasi dan Validasi Program

Verifikasi program dilakukan untuk memeriksa kesesuaian program yang penulis buat dengan apa yang sebenarnya penulis rencanakan dan harapkan, sesuai dengan diagram alur pembuatan program.

Proses verifikasi program ini sebenarnya telah dimulai sejak awal pembuatan program. Hal ini dikarenakan peneliti menyusun *syntax* pada *software* Matlab sesuai dengan langkah-langkah yang dijelaskan pada bagian sebelumnya. Tiap langkah masing-masing dibuat dalam sebuah perintah tersendiri (*m-file*), sehingga dapat diperiksa jalannya tiap *m-file*. Hal ini memang sengaja dilakukan guna memudahkan pembuatan *syntax* dan memudahkan pola berpikir. Tiap *m-file* saling berhubungan satu dengan yang lain, output *m-file* yang pertama menjadi input *m-file* berikutnya, dan demikian seterusnya.

Validasi program dilakukan guna memeriksa kebenaran program dalam mengolah data awal yang ada menjadi hasil akhir. Hal ini dilakukan agar peneliti mengetahui bahwa program berjalan dengan baik dan benar, serta dapat digunakan untuk penelitian ini. Dalam melakukan validasi, diperlukan sebuah pembandingan, yaitu hasil perhitungan secara manual. Dalam hal ini akan digunakan contoh kasus yang sebelumnya dan biaya *cost* yang terbaik adalah sebesar 183.205 untuk kombinasi [7 5 | 4 3 2 | 1 6]. Hal ini telah diuji dan dicoba dengan program yang telah dibuat, serta mendapatkan *layout* yang *feasible*.

4.1.3 Hasil

Proses penghitungan *problem sets* menggunakan *software* Matlab 7.9.0 529 (R2009b) dan menggunakan sebuah CPU dengan *processor* Intel ® Core™ 2 Duo CPU E7500 @ 2,936 Hz (2 CPUs) dan *total memory* 512MB.

Hasil fungsi tujuan merupakan biaya perpindahan material dalam satu unit *cost*, karena biaya pada penelitian ini dianggap satu. Hal ini untuk mempermudah dan agar sesuai dengan penelitian-penelitian sebelumnya yang dilakukan menggunakan *problem sets* ini.

Untuk setiap *problem set* dilakukan replikasi sebanyak 5 (lima) kali, agar dapat dilihat konsistensi dari hasil yang didapat. Hal ini dilakukan karena terdapat banyak kemungkinan urutan departemen yang dihasilkan (sesuai penjelasan pada

bagian sebelumnya), terutama untuk *problem sets* dengan jumlah departemen yang besar dan dengan karakteristik tertentu. Replikasi juga memastikan bahwa hasil terbaik, dari 5 kali replikasi yang didapat, bukan lokal optimum.

Hasil fungsi tujuan dengan replikasi sebanyak 5 kali diberikan pada **Tabel 4.1.** Hasil yang didapat untuk semua *problem sets* relatif cukup stabil, artinya hasil perhitungan konsisten dan program dapat mencari fungsi tujuan dengan hasil yang berdekatan satu dengan yang lain.

Tabel 4.1. Hasil Fungsi Tujuan dengan Replikasi

Problem Set ³	Fungsi Tujuan (Biaya)				
	1	2	3	4	5
O7 ¹	136.58	136.58	136.58	136.58	136.58
O7 ²	134.19	134.19	134.19	134.19	134.19
FO7(5) ¹	19.02	19.02	19.02	19.02	19.02
FO7(5) ²	18.88	18.88	18.88	18.88	18.88
FO7(4) ¹	23.12	23.12	23.12	23.12	23.12
FO7(4) ²	22.53	22.53	22.53	22.53	22.53
FO8(4)	22.39	22.39	22.39	22.39	22.39
O9(5)	242.43	238.11	241.32	238.11	243.26
O9(4)	248.97	247.03	247.29	241.06	252.14
V10s	22,899.65	22,899.65	22,899.65	22,899.65	22,899.65
V10a ¹	21,463.07	21,463.07	21,463.07	21,463.07	21,463.07
V10a ²	20,142.13	20,142.13	20,233.82	20,142.13	20,142.13
M11s	1,321.35	1,321.35	1,321.35	1,321.35	1,321.35
M11a	1,200.50	1,200.50	1,200.50	1,200.50	1,200.50
M15s	24,775.58	27,741.29	26,370.98	28,132.35	26,500.59
M15a	29,865.63	29,865.63	30,500.48	30,106.60	30,500.48
M25	1,499.29	1,544.88	1,548.56	1,499.74	1,273.24
NUG12	263.00	264.5	264.5	263	264.5
NUG15	565.75	591.08	604.46	580.75	582.25
BA12 ²	8,786.00	8,786.00	8,786.00	8,786.00	8,786.00
BA12TS ²	8,737.00	8,684.67	8,690.00	8,684.67	8,684.67
BA14	4,752.95	4,752.95	4,752.95	4,752.95	4,752.95
BA14TS	4,714.27	4,714.27	4,714.27	4,714.27	4,714.27
AB20	5,435.09	5,175.29	5,462.69	5,419.45	5,420.20
SC30a	4,595.53	4,187.09	4,318.13	4,396.31	3,967.76
SC35a	4,105.54	4,569.53	4,204.69	4,649.19	4,678.59

¹Ukuran fasilitas sesuai dengan data utama, yaitu Panjang (W) x Tinggi (H)

²Ukuran fasilitas diubah oleh penulis, yaitu Tinggi (H) x Panjang (W)

³Angka di dalam kurung (n) menunjukkan nilai maximum aspect ratio (α_{max})

Dari hasil replikasi tersebut, dibuat suatu analisa statistik yang, dapat dilihat pada **Tabel 4.2.**, mengenai hasil fungsi tujuan terbaik, rata-rata, dan terburuk. Hasil ini disertai juga dengan waktu perhitungan tiap *problem* untuk mendapatkan hasil terbaik dan diberikan juga total waktu yang dihabiskan untuk replikasi sebanyak 5 kali.

Tabel 4.2. Hasil Statistik dari Fungsi Tujuan dan *Computational Time*

Problem Set ³	Fungsi Tujuan			Waktu (sekon)	
	Terbaik	Rata-rata	Terburuk	Waktu	Total Waktu
O7 ¹	136.58	136.58	136.58	10.00	50.01
O7 ²	134.19	134.19	134.19	29.14	146.14
FO7(5) ¹	19.02	19.02	19.02	29.93	152.07
FO7(5) ²	18.88	18.88	18.88	30.16	151.27
FO7(4) ¹	23.12	23.12	23.12	29.39	147.09
FO7(4) ²	22.53	22.53	22.53	29.18	146.16
FO8(4)	22.39	22.39	22.39	49.24	247.22
O9(5)	238.11	240.65	243.26	48.45	243.57
O9(4)	241.06	247.30	252.14	73.39	367.99
V10s	22,899.65	22,899.65	22,899.65	100.00	500.05
V10a ¹	21,463.07	21,463.07	21,463.07	69.61	355.27
V10a ²	20,142.13	20,160.47	20,233.82	61.20	308.75
M11s	1,321.35	1,321.35	1,321.35	37.90	192.57
M11a	1,200.50	1,200.50	1,200.50	33.09	166.53
M15s	24,775.58	26,704.16	28,132.35	500.16	2,501.76
M15a	29,865.63	30,167.76	30,500.48	1,723.75	9,159.45
M25	1,273.24	1,473.14	1,548.56	407.03	2,039.82
NUG12	263.00	263.9	264.5	1,842.90	8872.893
NUG15	565.75	584.86	604.46	382.02	1,907.68
BA12 ²	8,786.00	8,786.00	8,786.00	60.04	301.14
BA12TS ²	8,684.67	8,696.20	8,737.00	220.46	1,207.31
BA14	4,752.95	4,752.95	4,752.95	81.17	411.88
BA14TS	4,714.27	4,714.27	4,714.27	92.31	465.13
AB20	5,175.29	5,382.55	5,462.69	419.40	2,077.27
SC30a	3,967.76	4,093.10	4,269.32	4,146.00	20,465.47
SC35a	4,105.54	4,441.51	4,678.59	3,970.96	22,207.55

¹Ukuran fasilitas sesuai dengan data utama, yaitu Lebar (W) x Tinggi (H)

²Ukuran fasilitas diubah oleh penulis, yaitu Tinggi (H) x Lebar (W)

³Angka di dalam kurung (n) menunjukkan nilai *maximum aspect ratio* (α_{max})

Semakin besar jumlah departemen yang ada, maka kecenderungan waktu yang dibutuhkan juga meningkat. Terdapat beberapa problem yang memiliki waktu unik tersendiri, seperti misalnya masalah NUG12. Waktu yang dibutuhkan cukup lama dibandingkan problem dengan jumlah departemen yang berdekatan. Hal ini mungkin saja terjadi karena parameter yang ditetapkan dalam perhitungan dan karena karakteristik *problem* itu sendiri. Begitu pula dengan masalah M15a, yang membutuhkan waktu yang cukup lama dibandingkan dengan M15s.

Tabel 4.3. Parameter algoritma *Simulated Annealing* dalam Iterasi

<i>Problem Sets</i>	Suhu Awal	Penurunan Suhu	Waktu Max.
O7	100	0.999	100
FO7(4)	100	0.999	100
FO8(4)	400	0.999	100
O9(5)	300	0.999	100
O9(4)	2,000	0.999	100
V10s	1,000	0.9999	100
V10a	1,000	0.999	100
M11s	200	0.999	100
M11a	100	0.999	100
M15s	1,000	0.999	500
M15a	15,000	0.9999	10,000
M25	200,000	0.999	10,000
NUG12	20,000	0.9999	5,000
NUG15	1,000	0.999	3,000
BA12	1,000	0.999	1,000
BA12TS	1,000	0.999	1,000
BA14	3,000	0.999	1,000
BA14TS	5,000	0.999	1,000
AB20	200,000	0.999	10,000
SC30a	100,000	0.9999	10,000
SC35a	5,000	0.9999	10,000

Tabel ini memperlihatkan parameter yang digunakan dalam proses perhitungan. Makin besar jumlah departemen, maka makin besar pula suhu awalnya, beserta waktu yang dibutuhkan. Penentuan parameter ini berdasarkan hasil yang didapat, ketika hasil yang didapat belum memuaskan, maka suhu awal akan dinaikkan. Tingkat penurunan suhu cenderung stabil pada 0.999, hanya untuk kasus tertentu menggunakan 0.9999. Makin besar suhu awal dan makin kecil nilai tingkat penurunan suhu, maka makin lama waktu yang dibutuhkan.

Penelitian ini menetapkan standar perhitungan dengan membiarkan iterasi berakhir sendiri tanpa memberikan batas waktu tertentu. Jadi waktu iterasi akan berada di bawah waktu maksimal yang dituliskan, artinya proses perhitungan telah selesai dilaksanakan dan mendapat nilai terbaik pada iterasi tersebut. Jika hasil belum baik, maka parameter suhu awal kembali dinaikkan.

4.2 Analisa Hasil

Jika dibandingkan dengan *best-known solution*, seperti yang ditunjukkan pada **Tabel 4.4.**, metode ini hanya berhasil memperbaiki satu masalah, yaitu *problem* BA14TS dengan peningkatan 2.74%.

Metode ini juga berhasil mendapatkan hasil yang sama untuk *problem* V10a. Hasil lain yang memiliki perbedaan persentase kecil (0% - 5%) adalah *problem* O7, FO8, O9, M11s, M11a, NUG12, BA12TS, BA14, dan AB20. Metode ini juga mendapatkan hasil dengan perbedaan yang besar (lebih dari 5%) untuk *problem* FO7(4), V10s, M15s, M15a, M25, NUG15, BA12, SC30, and SC35. Menarik untuk dilihat bagaimana perbedaan yang muncul merata pada kasus dengan jumlah departemen kecil, menengah, dan besar.

Hasil penelitian ini pada masalah M11s memiliki *output* dengan *layout* yang persis sama dan dengan urutan departemen yang sama dengan *best-known solution*. Namun, nilai fungsi tujuannya berbeda sedikit dengan persentase peningkatan -0.27%, dimana hasil dari penelitian Kulturel-Konak dan Konak (2011) berhasil mendapatkan hasil yang lebih baik. Untuk masalah V10s, *best-known solution* (Scholz *et al.*, 2009) menggunakan jarak *euclidean*, sementara penelitian ini menggunakan jarak *rectilinear*. Hal ini membuat hasil diantara keduanya sebenarnya tidak adil untuk dibandingkan.

Tabel 4.4. Perbandingan Hasil dengan *best-known* dan *best-known* FBS

Problem Set	Best-known	Reference	Best-known FBS	Reference	Author's Solution	Imp(%) over Best-Known	Imp(%) over Best-FBS
O7	131.57	Castillo and Westerlund (2005)			134.19	-1.99%	
FO7(5)	17.73	Castillo and Westerlund (2005)	23.12	Wong and Komarudin (2010)	18.88	-6.51%	22.43%
FO7(4)	19.56	Castillo and Westerlund (2005)			22.53	-15.19%	
FO8(4)	22.19	Castillo and Westerlund (2005)	22.39	Wong and Komarudin (2010)	22.39	-0.89%	0.00%
O9(5)	235.87	Castillo and Westerlund (2005)	241.06	Wong and Komarudin (2010)	238.11	-0.95%	1.24%
O9(4)	235.87	Castillo and Westerlund (2005)			241.06	-2.20%	
V10s	19,994.10	Scholz <i>et al.</i> (2009)	22,899.65	Kulturel-Konak and Konak (2011)	22,899.65	-14.53%	0.00%
V10a	20,142.13	Kulturel-Konak and Konak (2011)	20,142.13	Kulturel-Konak and Konak (2011)	20,142.13	0.00%	0.00%
M11s	1,317.79	Kulturel-Konak and Konak (2011)	1,317.79	Kulturel-Konak and Konak (2011)	1,321.35	-0.27%	-0.27%
M11a	1,185.20	Gau and Meller (1999)	1,201.12	Kulturel-Konak and Konak (2011)	1,200.50	-1.29%	0.05%
M15s	23,197.80	Wong and Komarudin (2010)	23,197.80	Wong and Komarudin (2010)	24,775.58	-6.80%	-6.80%
M15a	27,545.27	Kulturel-Konak and Konak (2011)	27,545.27	Kulturel-Konak and Konak (2011)	29,865.63	-8.42%	-8.42%
M25	1,132.59	Kulturel-Konak and Konak (2010)	1,132.59	Kulturel-Konak and Konak (2010)	1,273.24	-12.42%	-12.42%
NUG12	257.50	Kulturel-Konak and Konak (2011)	257.50	Kulturel-Konak and Konak (2011)	263.00	-2.14%	-2.14%
NUG15	524.75	Kulturel-Konak and Konak (2011)	524.75	Kulturel-Konak and Konak (2011)	565.75	-7.81%	-7.81%
BA12	8,129	Kulturel-Konak and Konak (2011)	8,129	Kulturel-Konak and Konak (2011)	9,798.73	-8.08%	-8.08%
BA12TS	8,587.04	Kulturel-Konak and Konak (2011)	8,587.04	Kulturel-Konak and Konak (2011)	8,740.00	-1.14%	-1.14%
BA14	4,712.33	Scholz <i>et al.</i> (2009)	4,780.91	Kulturel-Konak and Konak (2011)	4,752.95	-0.86%	0.59%
BA14TS	4,843.37	Kulturel-Konak and Konak (2011)	4,843.37	Kulturel-Konak and Konak (2011)	4,714.27	2.74%	2.74%
AB20	4,972.56	Komarudin and Wong (2010)	5,336.36	Kulturel-Konak and Konak (2011)	5,175.29	-4.08%	3.11%
SC30a	3,443.34	Kulturel-Konak and Konak (2011)	3,443.34	Kulturel-Konak and Konak (2011)	3,967.76	-15.23%	-15.23%
SC35a	3,604	Liu and Meller (2007)	3,700.75	Kulturel-Konak and Konak (2011)	4,105.54	-13.92%	-10.94%

$$\text{Improvement (\%)} = (\text{best-known solution} - \text{author's solution}) / \min(\text{best-known solution}, \text{author's solution}) \times 100$$

Pada masalah departemen kecil, hasil yang didapat hampir mendekati *best-known solution*, kecuali untuk masalah FO7. Ini tidak menjadi masalah karena model representasi yang digunakan berbeda. Untuk *best-known solution* yang tidak menggunakan model FBS (V10s, M11a, M15s, BA14, dan AB20), metode ini berhasil mendapatkan hasil yang tidak jauh berbeda dan mendekati, kecuali untuk masalah SC35a.

Karena penelitian ini menggunakan model *Flexible Bay Structure*, makaperbandingan akan lebih fokus dengan hasil penelitian yang menggunakan model yang sama. Untuk model yang sama, metode ini berhasil mendapatkan hasil yang lebih baik untuk masalah FO7(5), O9(5), M11a, BA14, BA14TS, dan AB20. Metode ini berhasil mendapat hasil yang sama untuk masalah FO8(4), V10s, dan V10a.

Seharusnya hasil penelitian ini setidaknya menghasilkan hasil yang sama dengan hasil terbaik FBS. Ini dikarenakan metode ini telah memenuhi semua kendala dan seharusnya dapat memberikan hasil yang terbaik. Ada kemungkinan terjadinya perbedaan data yang mengakibatkan perbedaan hasil akhir. Algoritma yang digunakan dalam penelitian ini juga mungkin kurang baik dan tidak dapat mendapat hasil yang terbaik untuk masalah-masalah yang kompleks. Ini dapat dilihat dari hasil yang tidak baik pada masalah M15s, M15a, M25, NUG12, NUG15, BA12, BA12TS, SC30a, dan SC35a. Masalah ini dapat dikatakan masalah dengan jumlah departemen yang cukup besar dan kompleks. Untuk NUG12 dan NUG15, masalah ini memang memiliki karakteristik dimana luas departemennya semua bernilai 1 (satu), sehingga memiliki tingkat kompleksitas yang tinggi.

Hal lain yang perlu diperhatikan adalah penggunaan *dummy departments* dalam memenuhi *empty spaces*. Beberapa penelitian menggunakan *dummy departments* sehingga membuat luas fasilitas benar-benar penuh, karena ada beberapa fasilitas yang tidak dipenuhi oleh semua departemen yang ada. Ini akan menambah konfigurasi dan variasi sehingga sangat mungkin didapatkan suatu hasil yang lebih baik. Perlu diingat bahwa masalah yang digunakan dalam penelitian ini tidak memakai *dummy departments* sama sekali, kecuali masalah BA12TS, yang memang dari pembuatnya sudah demikian.

Tabel 4.5. Perbandingan Penelitian menggunakan Model Representasi FBS

Problem Set	Tate and Smith (1995)	Konak <i>et al.</i> (2006)	Wong and Komarudin (2010)	Kulturel-Konak and Konak (2010)	Kulturel-Konak and Konak (2011)	Author's Solution
	Genetic Algorithm	Mix Integer Programming	Ant System	Ant Colony Optimization	Particle Swarm Optimization	FBS Constraint Satisfaction
O7						134.19
FO7(5)		23.12	23.12			18.88
FO7(4)						22.53
FO8(4)			22.39			22.39
O9(5)		241.06	241.06			238.11
O9(4)						241.06
V10s		22,899.65	22,899.65	22,899.65	22,899.65	22,899.65
V10a		21,463.10	21,463.10	21,463.10	20,142.13	20,142.13
M11s		1,317.79	1,321.35		1,317.79	1,321.35
M11a		1,225	1,204.15		1,201.12	1,200.50
M15s		27,781.95	23,197.80		23,197.84	24,775.58
M15a		31,779.09	27,545.27		27,545.27	29,865.63
M25			1,496.42	1,132.59	1,399.24	1,273.24
NUG12		265.6	262	262	257.50	263.00
NUG15		526.75	536.75	524.75	524.75	565.75
BA12	8,768.2	8,801.33	8,299.5	8,801.33	8,129	9,798.73
BA12TS	8,861	8,600.33	8,587.05	8,600.33	8,587.04	8,740.00
BA14		5,004.55	4,913.22	5,004.55	4,780.91	4,752.95
BA14TS	5,077	4,927.69	4,927.69	4,927.69	4,843.37	4,714.27
AB20	5,743.1			5,360.8	5,336.36	5,175.29
SC30a			3,679.85	3,794.82	3,443.34	3,967.76
SC35a			3,962.72	4,001.31	3,700.75	4,105.54

Untuk *problem* BA12, SC30a, dan SC35a, sebenarnya tidak adil jika dibandingkan, karena Kulturel-Konak dan Konak (2011) dalam penelitiannya menggunakan *dummy departments*, sehingga ini menjadi salah satu indikator lebih baiknya hasil yang didapat. Model yang digunakan dalam penelitian tersebut juga tidak menggunakan model FBS secara umum, tetapi menggunakan model FBS yang telah dimodifikasi (seperti Wong dan Komarudin, 2010), yaitu *relaxed-FBS*.

Pada **Tabel 4.5.** diberikan suatu perbandingan dengan hasil penelitian yang menggunakan model representasi yang sama (FBS) dengan metode yang berbeda. Perbandingan ini diberikan karena penelitian ini fokus kepada model representasi FBS, sehingga ingin dilihat apakah pendekatan atau metode yang digunakan dapat menghasilkan hasil yang lebih baik dibandingkan dengan metode lainnya dengan model yang sama.

Tabel 4.6. Perbandingan Hasil dengan metode penelitian tanpa model FBS

Problem Set	Castillo and Westerlund (2005)	Liu and Meller (2007)	Scholz <i>et al.</i> (2009)	Komarudin and Wong (2010)	Author's Solution
	MIP – e-Accurate model	MIP – Sequence-pair	Tabu Search - STS	Ant System - STS	FBS Constraint Satisfaction
O7	131.57	131.63	132	131.68	134.19
FO7(5)	17.73	17.75			18.88
FO7(4)	19.56	20.73			22.53
FO8(4)	22.19	22.31			22.39
O9(5)	235.87				238.11
O9(4)	235.87	235.95	236.14	236.14	241.06
V10s			19,994.1 ^b		22,899.65
V10a		19,997 ^a			20,142.13
BA12		8702	8,264	8,252.67	9,798.73
BA12TS					8,740.00
BA14		5,004	4,712.33	4,724.68	4,752.95
BA14TS					4,714.27
AB20		5,668	5,225.96	4,972.56	5,175.29
SC30		3,706.83		3,868.54	3,967.76
SC35		3,604.12		4,132.37	4,105.54

^aMasalah menggunakan *maximum aspect ratio* 4, penelitian ini 5

^bJarak yang digunakan adalah jarak *Euclidean*

Karena penelitian Kulturel-Konak dan Konak (2011) sebagian besar sudah menjadi *best-known* FBS, maka perbandingan akan dibahas hanya dengan empat penelitian sebelumnya. Untuk beberapa *problem*, penelitian ini berhasil mendapatkan hasil yang lebih baik, seperti untuk *problem* FO7, FO8, O9, V10a, M11a, BA14, BA14TS, dan AB20.

Untuk sekedar melihat perbandingan hasil dengan penelitian yang tidak menggunakan *Flexible Bay Structure*, diberikan Tabel 4.6., yang membandingkan penelitian ini dengan beberapa metode tertentu dalam kurun satu dekade terakhir. Terdapat dua buah hal yang harus diperhatikan, yaitu nilai *maximum aspect ratio* yang digunakan Liu dan Meller (2007), V10a, adalah 4, sedangkan penelitian ini adalah 5. Hal kedua adalah penghitungan jarak secara *Euclidean* pada masalah V10s pada penelitian Scholz *et al.* (2009).

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian dan analisa yang sudah dilakukan, maka dapat diambil beberapa kesimpulan mengenai penelitian ini, yaitu:

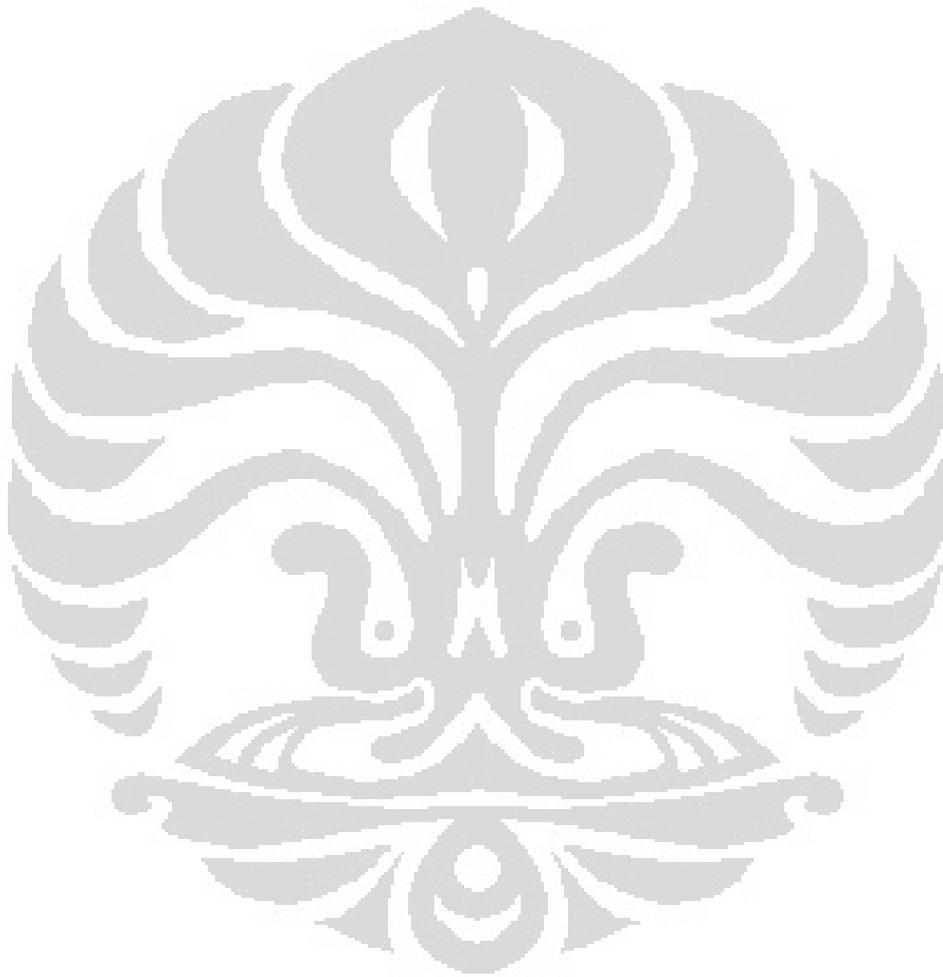
- Penelitian telah berhasil mengembangkan suatu metode yang memenuhi kendala-kendala yang dihasilkan dalam proses konstruksinya dengan mengurangi kemungkinan solusi pada model representasi *Flexible Bay Structure*;
- Metode ini fokus pada kendala yang ada sehingga hasil perhitungannya mempunyai nilai yang cukup baik dan stabil;
- Metode ini untuk beberapa masalah mendekati hasil *best-known solution* yang ada dan untuk kasus BA14TS mendapatkan hasil yang lebih baik dengan 2.74%;
- Metode ini mendapatkan hasil yang lebih baik daripada *best-known* FBS untuk 6 (enam) masalah, yaitu FO7(5), O9(5), M11a, BA14, BA14TS, dan AB20 dan mendapat hasil yang sama dengan *best-known* FBS untuk problem FO (8), V10s, dan V10a;
- Secara umum, perbedaan dengan *best-known solution* tidak berbeda jauh, cenderung mendekati, kecuali untuk beberapa masalah yang memang memiliki jumlah departemen yang besar dan yang kompleks, serta yang menggunakan *dummy departments*.

5.2 Saran

Berdasarkan hasil analisa, maka ada beberapa saran yang dapat dilakukan ke depannya untuk mengembangkan metode ini supaya lebih baik lagi. Saran tersebut adalah:

- Penelitian ini belum menggunakan *dummy departments* sehingga belum diketahui bagaimana hasil yang akan diterima, sehingga ada baiknya masalah yang ada dicoba dengan *dummy departments*, jika terdapat *empty spaces*, seperti penelitian-penelitian lainnya (khususnya untuk *problem* besar);

- Penelitian ini dapat mencoba menukar posisi *Width* dan *Height* fasilitas, *vertical bays* atau *horizontal bays*, karena untuk beberapa *problem*, didapatkan hasil yang lebih baik untuk suatu *layout* tertentu;
- Penggunaan algoritma meta-heuristik lain mungkin dapat membantu tercapainya hasil fungsi tujuan yang jauh lebih baik dibanding hasil yang didapat pada penelitian ini, seperti *Tabu Search*, dll.



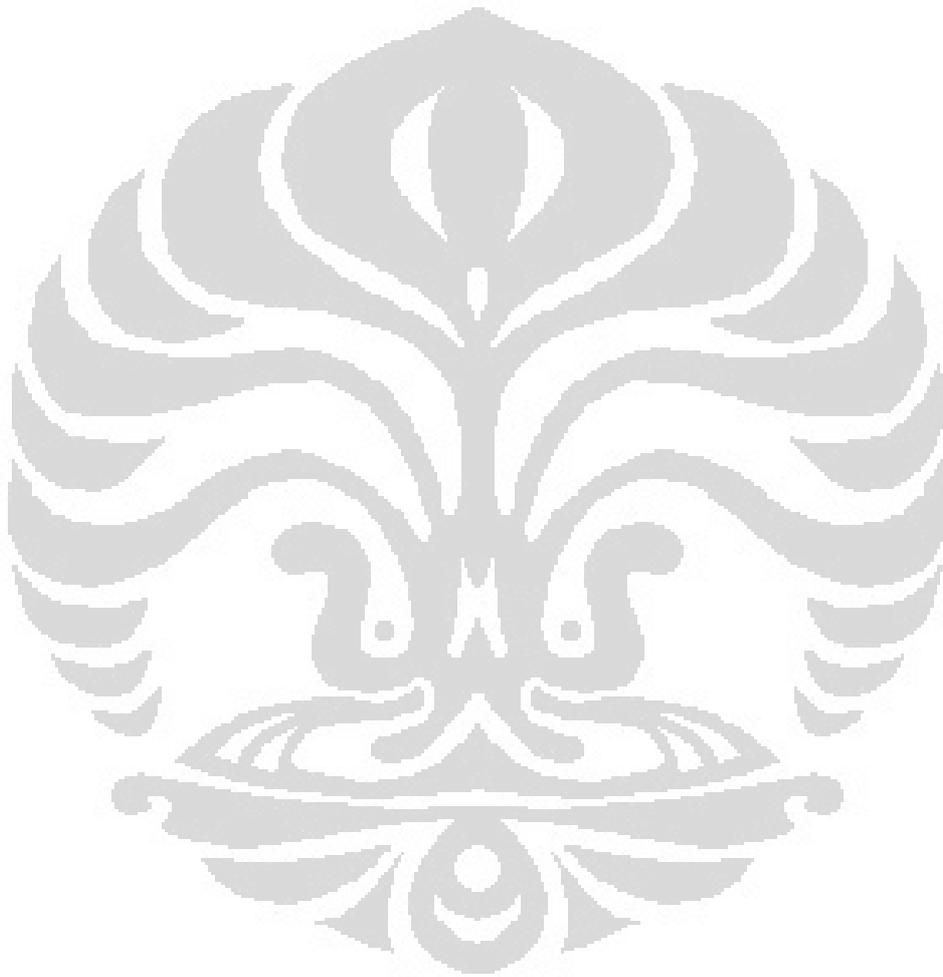
REFERENSI

- Armour, G.C., and Buffa, E.S. (1963). A Heuristic Algorithm and Simulation Approach to Relative Allocation of Facilities. *Management Science*. 9(2), 294-300.
- Asef-Vaziri, A. and Laporte, G. (2004). Loop based facility planning and material handling. *European Journal of Operational Research*, 164, 1–11.
- Bazaraa, M.S., 1975. Computerized layout design: a branch and bound approach. *IIE Transactions*, 7, 549–560.
- Castillo, I. and Westerlund, T. (2005). An e-accurate model for optimal unequal-area block layout design. *Computers and Operations Research*, 32, 429–447.
- Chwif, L., Baretto, M.R.P, and Moscato, L.A. (1998). A solution to the facility layout problem using simulated annealing. *Computers in Industry*, 36, 125-132.
- Gau, K.Y. and Meller, R.D. (1999). An iterative facility layout algorithm. *International Journal of Production Research*, 37, 3739–3758.
- Komarudin. (2009). An Improved Ant System Algorithm for Unequal Area Facility Layout Problems. Master of Engineering Thesis, Faculty of Mechanical Engineering, Universiti Teknologi Malaysia, Malaysia.
- Komarudin and Wong, K.Y. (2010). Applying ant system for solving unequal area facility layout problems. *European Journal of Operational Research*, 202, 730–746.
- Konak, A., Kulturel-Konak, S., Norman, B.A., and Smith, A.E. (2006). A New Mixed Integer Formulation for Optimal Facility Layout design. *Operation Research Letters*. 34(6), 660-672.
- Kulturel-Konak, S. and Konak, A. (2010). Unequal area flexible bay facility layout using ant colony optimisation. *International Journal of Production Research*, 49 (7), 1877–1902.
- Kulturel-Konak, S. and Konak, A. (2011). A new flexible bay structure representation and particle swarm optimization for the unequal area facility layout problem. *Engineering Optimization*, iFirst, 1-25.

- Liu, Q., and Meller, R.D. (2007). A sequence-pair representation and MIP modelbased heuristic for the facility layout problem with rectangular departments. *IIE Transactions*. 39(4), 377-394.
- Mavridou, T.D, Pardalos, P.M. (1997). Simulated Annealing and Genetic Algorithms for the facility layout problems: A survey. *Computational Optimization and Applications*, 7, 111–126.
- Meller, R.D. (1992). Layout Algorithms for Single and Multiple Floor Facilities. Doctor of Philosophy Thesis, Department of Industrial and Operations Engineering, University of Michigan, USA.
- Meller, R.D., and Bozer, Y.A. (1996). A New Simulated Annealing Algorithm for The Facility Layout Problem. *International Journal of Production Research*. 34(6), 1675-1692
- Meller, R.D., and Gau, K.Y. (1996). The facility layout problem: recent and emerging trends and perspectives. *Journal of Manufacturing Systems*. 15(5), 351-366.
- Meller, R.D., Narayanan, V., and Vance, P.H. (1998). Optimal Facility Layout Design. *Operations Research Letters*. 23(3-5), 117–127.
- Nugent, C.E., Vollman, T.E. and Ruml, J. (1968). An experimental comparison of techniques for the assignment of facilities to locations. *Operations Research*, 16, 150–173.
- Scholz, D., Petrick, A., and Domschke, W. (2009). STaTS: A slicing tree and tabu search based heuristic for the unequal area facility layout problem. *European Journal of Operational Research*. 197(1), 166-178.
- Sherali, H.D., Fraticelli, B.M.P., and Meller, R.D. (2003). Enhanced Model Formulations for Optimal Facility Layout. *Operations Research*. 51(4), 629-644.
- Tate, D.M., and Smith, A.E. (1995). Unequal-Area Facility Layout by Genetic Search. *IIE Transactions*. 27, 465-472.
- Tretheway, S.J. and Foote, B.L. (1994). Automatic Computation and Drawing of Facility Layouts with Logical Aisle Structures. *International Journal of Production Research*. 32(7), 1545-1556.

van Camp, D.J., Carter, M.W., and Vannelli, A. (1992). A Nonlinear Optimization Approach for Solving Facility Layout Problem. *European Journal of Operation Research*. 57(2), 174-189.

Wong, K.Y. and Komarudin, K. (2010). Solving facility layout problems using Flexible Bay Structure representation and Ant System algorithm. *Expert Systems with Applications*, 37, 5523–5527.



LAMPIRAN

1.1 Problem Sets UAFLP

No	Problem Set	Number of Departments	Facility Size		Common Shape Constraints	Reference
			Width	Height		
1.	O7	7	8.54	13.00	$\alpha_{\max} = 4$	Meller <i>et al.</i> (1999)
2.	FO7	7	8.54	13.00	$\alpha_{\max} = 4$	Meller <i>et al.</i> (1999)
3.	FO8	8	11.31	13.00	$\alpha_{\max} = 4$	Meller <i>et al.</i> (1999)
4.	O9	9	12.00	13.00	$\alpha_{\max} = 4$	Meller <i>et al.</i> (1999)
5.	V10s	10	25.00	51.00	$l_{\min} = 5$	van Camp <i>et al.</i> (1992)
6.	V10a	10	25.00	51.00	$\alpha_{\max} = 5$	van Camp <i>et al.</i> (1992)
7.	M11s	11	6.00	6.00	$l_{\min} = 1$	Meller (1992)
8.	M11a	11	6.00	6.00	$\alpha_{\max} = 5$	Meller (1992)
9.	M15s	15	15.00	15.00	$l_{\min} = 1$	Meller (1992)
10.	M15a	15	15.00	15.00	$\alpha_{\max} = 5$	Meller (1992)
11.	M25	25	15.00	5.00	$\alpha_{\max} = 5$	Meller (1992)
12.	NUG12	12	3.00	4.00	$\alpha_{\max} = 4$	Nugent <i>et al.</i> (1968)
13.	NUG15	15	3.00	5.00	$\alpha_{\max} = 4$	Nugent <i>et al.</i> (1968)
14.	BA12	12	10.00	6.00	$l_{\min} = 1$	Bazaraa (1975)
15.	BA12TS	16	10.00	6.00	$l_{\min} = 1$	Tate and Smith (1995)
16.	BA14	14	9.00	7.00	$l_{\min} = 1$	Bazaraa (1975)
17.	BA14TS	14	9.00	7.00	$l_{\min} = 1$	Tate and Smith (1995)
18.	AB20	20	2.00	3.00	$\alpha_{\max} = 4$	Armour and Buffa (1963)
19.	SC30	30	15.00	12.00	$\alpha_{\max} = 4$	Liu and Meller (2007)
20.	SC35	35	16.00	15.00	$\alpha_{\max} = 4$	Liu and Meller (2007)

1.2 Problem Set O7

Area Requirement

Dept.	Area	Max. Aspect Ratio
1	16	4
2	16	4
3	16	4
4	36	4
5	9	4
6	9	4
7	9	4

Matriks Material Flow

Dept.	1	2	3	4	5	6	7
1	0	0	0	5	0	0	1
2	0	0	0	3	0	0	1
3	0	0	0	2	0	0	1
4	0	0	0	0	4	4	0
5	0	0	0	0	0	0	2
6	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0

1.3 Problem Set FO7

Area Requirement

Dept.	Area	Max. Aspect Ratio
1	16	4
2	16	4
3	16	4
4	36	4
5	9	4
6	9	4
7	9	4

Matriks Material Flow

Dept.	1	2	3	4	5	6	7
1	0	1	0	0	0	0	0
2	0	0	1	0	0	0	0
3	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0
5	0	0	0	0	0	1	0
6	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0

1.4 Problem Set FO8

Area Requirement

Dept.	Area	Max. Aspect Ratio
1	16	4
2	16	4
3	16	4
4	36	4
5	36	4
6	9	4
7	9	4
8	9	4

Matriks Material Flow

Dept.	1	2	3	4	5	6	7	8
1	0	1	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0
3	0	0	0	1	0	0	0	0
4	0	0	0	0	1	0	0	0
5	0	0	0	0	0	1	0	0
6	0	0	0	0	0	0	1	0
7	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0

1.5 Problem Set O9

Area Requirement

Dept.	Area	Max. Aspect Ratio
1	16	4
2	16	4
3	16	4
4	36	4
5	36	4
6	9	4
7	9	4
8	9	4
9	9	4

Matriks Material Flow

Dept.	1	2	3	4	5	6	7	8	9
1	0	0	0	5	5	0	0	0	1
2	0	0	0	3	3	0	0	0	1
3	0	0	0	2	2	0	0	0	1
4	0	0	0	0	0	4	4	0	0
5	0	0	0	0	0	3	0	0	4
6	0	0	0	0	0	0	0	0	2
7	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0

1.6 Problem Set V10s

Area Requirement untuk *Problem Set V10s*

Dept.	Area	Minimum Side
1	238	5
2	112	5
3	160	5
4	80	5
5	120	5
6	80	5
7	60	5
8	85	5
9	221	5
10	119	5

Matriks Material Flow untuk *Problem Set V10s*

Dept.	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	218	0	0	0	0
2	0	0	0	0	0	148	0	0	296	0
3	0	0	0	28	70	0	0	0	0	0
4	0	0	0	0	0	28	70	140	0	0
5	0	0	0	0	0	0	0	210	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	28
8	0	0	0	0	0	0	0	0	0	888
9	0	0	0	0	0	0	0	0	0	59.2
10	0	0	0	0	0	0	0	0	0	0

1.7 Problem Set V10a

Area Requirement untuk *Problem Set V10a*

Dept.	Area	Max. Aspect Ratio
1	238	5
2	112	5
3	160	5
4	80	5
5	120	5
6	80	5
7	60	5
8	85	5
9	221	5
10	119	5

Matriks *Material Flow* untuk *Problem Set V10a*

Dept.	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	218	0	0	0	0
2	0	0	0	0	0	148	0	0	296	0
3	0	0	0	28	70	0	0	0	0	0
4	0	0	0	0	0	28	70	140	0	0
5	0	0	0	0	0	0	0	210	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	28
8	0	0	0	0	0	0	0	0	0	888
9	0	0	0	0	0	0	0	0	0	59.2
10	0	0	0	0	0	0	0	0	0	0

1.8 *Problem Set M11s*

Area Requirement untuk *Problem Set M11s*

Dept.	Area	Minimum Side
1	3	1
2	2	1
3	4	1
4	5	1
5	2	1
6	3	1
7	4	1
8	5	1
9	1	1
10	1	1
11	6	1

Matriks *Material Flow* untuk *Problem Set M11s*

Dept.	1	2	3	4	5	6	7	8	9	10	11
1	0	10	0	0	140	90	20	0	40	0	0
2	0	0	10	0	0	0	0	0	0	0	0
3	0	0	0	10	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	4
5	0	10	0	0	0	0	40	0	0	20	0
6	0	0	10	0	0	0	0	0	20	0	0
7	0	0	0	0	0	0	0	10	0	0	0
8	0	0	0	0	0	0	0	0	0	0	11
9	0	0	0	0	0	0	0	0	0	20	0
10	0	0	0	0	0	0	0	0	0	0	20
11	146	0	0	0	0	0	0	0	0	0	0

1.9 Problem Set M11a

Area Requirement untuk *Problem Set M11a*

Dept.	Area	Max. Aspect Ratio
1	3	5
2	2	5
3	4	5
4	5	5
5	2	5
6	3	5
7	4	5
8	5	5
9	1	5
10	1	5
11	6	5

Matriks *Material Flow* untuk *Problem Set M11a*

Dept.	1	2	3	4	5	6	7	8	9	10	11
1	0	10	0	0	140	90	20	0	40	0	0
2	0	0	10	0	0	0	0	0	0	0	0
3	0	0	0	10	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	4
5	0	10	0	0	0	0	40	0	0	20	0
6	0	0	10	0	0	0	0	0	20	0	0
7	0	0	0	0	0	0	0	10	0	0	0
8	0	0	0	0	0	0	0	0	0	0	11
9	0	0	0	0	0	0	0	0	0	20	0
10	0	0	0	0	0	0	0	0	0	0	20
11	146	0	0	0	0	0	0	0	0	0	0

1.10 *Problem Set M15s*

Area Requirement untuk *Problem Set M15s*

Dept.	Area	Minimum Side
1	15	1
2	10	1
3	9	1
4	7	1
5	9	1
6	25	1
7	25	1
8	15	1
9	10	1
10	25	1
11	10	1
12	15	1
13	6	1
14	19	1
15	25	1

Matriks *Material Flow* untuk *Problem Set M15s*

Dept.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	240
2	240	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1200	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	1200	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	600	0
6	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120
9	0	0	0	0	0	0	0	0	0	600	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	600	0	0	0
11	0	0	0	0	0	0	480	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	600
13	0	0	0	0	0	0	480	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	600	0	0	0
15	0	10	50	0	25	40	0	0	25	0	40	0	20	0	0

1.11 Problem Set M15a*Area Requirement* untuk *Problem Set M15a*

Dept.	Area	Max. Aspect Ratio
1	15	5
2	10	5
3	9	5
4	7	5
5	9	5
6	25	5
7	25	5
8	15	5
9	10	5
10	25	5
11	10	5
12	15	5
13	6	5
14	19	5
15	25	5

Matriks *Material Flow* untuk *Problem Set M15a*

Dept.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	240
2	240	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1200	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	1200	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	600	0
6	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120
9	0	0	0	0	0	0	0	0	0	600	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	600	0	0	0
11	0	0	0	0	0	0	480	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	600
13	0	0	0	0	0	0	480	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	600	0	0	0
15	0	10	50	0	25	40	0	0	25	0	40	0	20	0	0

1.12 *Problem Set M25*

Area Requirement untuk *Problem Set M25*

Dept.	Area	Max. Aspect Ratio	Dept.	Area	Max. Aspect Ratio
1	2	5	16	4	5
2	1	5	17	1	5
3	5	5	18	5	5
4	3	5	19	3	5
5	1	5	20	4	5
6	3	5	21	1	5
7	5	5	22	4	5
8	3	5	23	5	5
9	2	5	24	2	5
10	1	5	25	4	5
11	3	5			
12	2	5			
13	4	5			
14	5	5			
15	2	5			

Lampiran 1: UA-FLP *Problem Sets* (lanjutan)

Matriks *Material Flow* untuk *Problem Set M25*

Dept.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	0	0	0	6.67	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.5	0	0	0	0	0	0	0	0
3	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0
6	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	5	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0
19	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0
21	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16
23	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0
25	4	5	4	0	0	16	0	4	0	0	0	16	0	8	8	8	0	0	0	0	0	0	0	0	0

1.13 Problem Set NUG12

Area Requirement untuk *Problem Set NUG12*

Dept.	Area	Max. Aspect Ratio
1	1	4
2	1	4
3	1	4
4	1	4
5	1	4
6	1	4
7	1	4
8	1	4
9	1	4
10	1	4
11	1	4
12	1	4

Matriks Material Flow untuk *Problem Set NUG12*

Dept.	1	2	3	4	5	6	7	8	9	10	11	12
1	0	5	2	4	1	0	0	6	2	1	1	1
2	0	0	3	0	2	2	2	0	4	5	0	0
3	0	0	0	0	0	0	0	5	5	2	2	2
4	0	0	0	0	5	2	2	10	0	0	5	5
5	0	0	0	0	0	10	0	0	0	5	1	1
6	0	0	0	0	0	0	5	1	1	5	4	0
7	0	0	0	0	0	0	0	10	5	2	3	3
8	0	0	0	0	0	0	0	0	0	0	5	0
9	0	0	0	0	0	0	0	0	0	0	10	10
10	0	0	0	0	0	0	0	0	0	0	5	0
11	0	0	0	0	0	0	0	0	0	0	0	2
12	0	0	0	0	0	0	0	0	0	0	0	0

1.14 Problem Set NUG15

Area Requirement untuk *Problem Set NUG15*

Dept.	Area	Max. Aspect Ratio
1	1	4
2	1	4
3	1	4
4	1	4
5	1	4
6	1	4
7	1	4
8	1	4
9	1	4
10	1	4
11	1	4
12	1	4
13	1	4
14	1	4
15	1	4

Matriks Material Flow untuk *Problem Set NUG15*

Dept.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	10	0	5	1	0	1	2	2	2	2	0	4	0	0
2	0	0	1	3	2	2	2	3	2	0	2	0	10	5	0
3	0	0	0	10	2	0	2	5	4	5	2	2	5	5	5
4	0	0	0	0	1	1	5	0	0	2	1	0	2	5	0
5	0	0	0	0	0	3	5	5	5	1	0	3	0	5	5
6	0	0	0	0	0	0	2	2	1	5	0	0	2	5	10
7	0	0	0	0	0	0	0	6	0	1	5	5	5	1	0
8	0	0	0	0	0	0	0	0	5	2	10	0	5	0	0
9	0	0	0	0	0	0	0	0	0	0	10	5	10	0	2
10	0	0	0	0	0	0	0	0	0	0	0	4	0	0	5
11	0	0	0	0	0	0	0	0	0	0	0	5	0	5	0
12	0	0	0	0	0	0	0	0	0	0	0	0	3	3	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	10	2
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1.15 Problem Set BA12

Area Requirement untuk *Problem Set BA12*

Dept.	Area	Minimum Side
1	9	1
2	8	1
3	10	1
4	6	1
5	4	1
6	3	1
7	3	1
8	4	1
9	2	1
10	2	1
11	1	1
12	1	1

Matriks Material Flow untuk *Problem Set BA12*

Dept.	1	2	3	4	5	6	7	8	9	10	11	12
1	0	288	180	54	72	180	27	72	36	0	0	9
2	0	0	240	54	72	24	48	160	16	64	8	16
3	0	0	0	120	80	0	60	120	60	0	0	30
4	0	0	0	0	72	18	18	48	24	48	12	0
5	0	0	0	0	0	12	12	64	16	16	4	8
6	0	0	0	0	0	0	18	24	6	12	3	3
7	0	0	0	0	0	0	0	0	6	6	3	6
8	0	0	0	0	0	0	0	0	16	16	16	4
9	0	0	0	0	0	0	0	0	0	4	4	2
10	0	0	0	0	0	0	0	0	0	0	2	2
11	0	0	0	0	0	0	0	0	0	0	0	2
12	0	0	0	0	0	0	0	0	0	0	0	0

1.16 Problem Set BA12TS

Area Requirement untuk *Problem Set* BA12TS

Dept.	Area	Minimum Side
1	9	1
2	8	1
3	10	1
4	6	1
5	4	1
6	3	1
7	3	1
8	4	1
9	2	1
10	2	1
11	2	1
12	2	1
13	2	0
14	1	0
15	1	0
16	1	0

Matriks Material Flow untuk *Problem Set* BA12TS

Dept.	1	2	3	4	5	6	7	8	9	10	11	12
1	0	288	180	54	72	180	27	72	36	0	0	9
2	0	0	240	54	72	24	48	160	16	64	8	16
3	0	0	0	120	80	0	60	120	60	0	0	30
4	0	0	0	0	72	18	18	48	24	48	12	0
5	0	0	0	0	0	12	12	64	16	16	4	8
6	0	0	0	0	0	0	18	24	6	12	3	3
7	0	0	0	0	0	0	0	0	6	6	3	6
8	0	0	0	0	0	0	0	0	16	16	16	4
9	0	0	0	0	0	0	0	0	0	4	4	2
10	0	0	0	0	0	0	0	0	0	0	2	2
11	0	0	0	0	0	0	0	0	0	0	0	2
12	0	0	0	0	0	0	0	0	0	0	0	0

1.17 Problem Set BA14*Area Requirement* untuk *Problem Set BA14*

Dept.	Area	Minimum Side
1	9	1
2	8	1
3	9	1
4	10	1
5	6	1
6	3	1
7	3	1
8	3	1
9	2	1
10	3	1
11	2	1
12	1	1
13	1	0
14	1	0

Matriks Material Flow untuk *Problem Set BA14*

Dept.	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	72	162	90	108	27	0	0	18	27	18	0	0	0
2	0	0	72	80	0	48	0	48	32	0	16	8	0	0
3	0	0	0	45	54	27	27	27	0	27	0	9	18	0
4	0	0	0	0	30	0	30	30	20	0	20	10	10	0
5	0	0	0	0	0	18	0	18	12	18	24	0	0	0
6	0	0	0	0	0	0	9	9	0	0	6	6	6	0
7	0	0	0	0	0	0	0	9	12	9	6	3	0	0
8	0	0	0	0	0	0	0	0	6	9	0	3	0	0
9	0	0	0	0	0	0	0	0	0	6	4	6	2	0
10	0	0	0	0	0	0	0	0	0	0	6	3	6	0
11	0	0	0	0	0	0	0	0	0	0	0	2	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	4	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1.18 Problem Set BA14TS

Area Requirement untuk *Problem Set* BA14TS

Dept.	Area	Minimum Side
1	9	1
2	8	1
3	9	1
4	10	1
5	6	1
6	3	1
7	3	1
8	3	1
9	2	1
10	3	1
11	2	1
12	1	1
13	1	0
14	3	0

Matriks Material Flow untuk *Problem Set* BA14TS

Dept.	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	72	162	90	108	27	0	0	18	27	18	0	0	0
2	0	0	72	80	0	48	0	48	32	0	16	8	0	0
3	0	0	0	45	54	27	27	27	0	27	0	9	18	0
4	0	0	0	0	30	0	30	30	20	0	20	10	10	0
5	0	0	0	0	0	18	0	18	12	18	24	0	0	0
6	0	0	0	0	0	0	9	9	0	0	6	6	6	0
7	0	0	0	0	0	0	0	9	12	9	6	3	0	0
8	0	0	0	0	0	0	0	0	6	9	0	3	0	0
9	0	0	0	0	0	0	0	0	0	6	4	6	2	0
10	0	0	0	0	0	0	0	0	0	0	6	3	6	0
11	0	0	0	0	0	0	0	0	0	0	0	2	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	4	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1.19 Problem Set AB20*Area Requirement* untuk *Problem Set AB20*

Dept.	Area	Max. Aspect Ratio
1	0.27	4
2	0.18	4
3	0.27	4
4	0.18	4
5	0.18	4
6	0.18	4
7	0.09	4
8	0.09	4
9	0.09	4
10	0.24	4
11	0.6	4
12	0.42	4
13	0.18	4
14	0.24	4
15	0.27	4
16	0.75	4
17	0.64	4
18	0.41	4
19	0.27	4
20	0.45	4

Lampiran 1: UA-FLP *Problem Sets* (lanjutan)

Matriks *Material Flow* untuk *Problem Set AB20*

Dept.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	18	12	0	0	0	0	0	0	10.4	11.2	0	0	12	0	0	0	0	0	0
2	18	0	9.6	244.5	7.8	0	139.5	0	12	13.5	0	0	0	0	0	0	0	0	69	0
3	12	9.6	0	0	0	22.1	0	0	31.5	39	0	0	0	130.5	0	0	0	0	136.5	0
4	0	244.5	0	0	10.8	57	75	0	23.4	0	0	14	0	0	0	0	0	15	157.5	0
5	0	7.8	0	10.8	0	0	22.5	13.5	0	15.6	0	0	0	0	13.5	0	0	0	0	0
6	0	0	22.1	57	0	0	61.5	0	0	0	0	4.5	0	0	0	0	0	10.5	0	0
7	0	139.5	0	75	22.5	61.5	0	240	0	18.7	0	0	0	9.6	0	0	0	16.5	0	37.5
8	0	0	0	0	13.5	0	240	0	0	0	0	0	6	0	0	0	0	0	75	334.5
9	0	12	31.5	23.4	0	0	0	0	0	0	0	0	0	75	0	0	75	0	0	0
10	10.4	13.5	39	0	15.6	0	18.7	0	0	0	3.6	120	0	186	19.2	0	0	0	52.5	0
11	11.2	0	0	0	0	0	0	0	0	3.6	0	22.5	0	30	9.6	225	0	0	0	0
12	0	0	0	14	0	4.5	0	0	0	120	22.5	0	0	0	16.5	0	150	0	84	0
13	0	0	0	0	0	0	0	6	0	0	0	0	0	80	10.4	60	0	0	0	0
14	12	0	130.5	0	0	0	9.6	0	75	186	30	0	80	0	97.5	0	0	9	0	0
15	0	0	0	0	13.5	0	0	0	0	19.2	9.6	16.5	10.4	97.5	0	0	52.5	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	120	0	0	0
17	0	0	0	0	0	0	0	0	75	0	0	150	0	0	52.5	120	0	0	75	0
18	0	0	0	15	0	10.5	16.5	0	0	0	0	0	0	9	0	0	0	0	46.5	0
19	0	69	136.5	157.5	0	0	0	75	0	52.5	0	84	0	0	0	0	75	46.5	0	0
20	0	0	0	0	0	0	37.5	334.5	0	0	0	0	0	0	0	0	0	0	0	0

1.20 Problem Set SC30*Area Requirement* untuk *Problem Set SC30*

Dept.	Area	Max. Aspect Ratio
1	3	4
2	4	4
3	4	4
4	16	4
5	4	4
6	5	4
7	2	4
8	3	4
9	5	4
10	6	4
11	2	4
12	24	4
13	5	4
14	3	4
15	11	4
16	6	4
17	2	4
18	8	4
19	4	4
20	5	4
21	4	4
22	3	4
23	1	4
24	3	4
25	1	4
26	4	4
27	6	4
28	1	4
29	14	4
30	4	4

Matriks *Material Flow* untuk *Problem Set* SC30

From	To	<i>Material Flow</i>	From	To	<i>Material Flow</i>
1	24	2.95	10	12	38.03
1	25	6.32	11	12	8.84
1	26	1.26	12	13	12.97
1	27	2.11	12	14	4.67
1	28	1.26	12	15	53.42
1	30	13.91	12	16	1.83
2	6	20.38	13	15	12.97
2	19	4.5	14	15	4.67
2	20	3.63	15	4	63.95
2	21	2.93	15	18	190.74
2	22	1.29	16	15	4.58
2	23	1.43	17	15	0.76
4	3	394.11	18	4	190.74
5	11	4.09	19	29	9.18
5	12	33.09	20	29	7.4
6	5	8.92	21	29	5.97
6	7	2.07	22	29	2.63
6	8	4.48	23	29	2.92
6	10	4.56	24	29	5.9
6	12	1.83	25	29	12.65
6	17	0.61	26	29	2.53
7	8	12.97	27	29	4.22
8	9	43.23	28	29	2.53
9	11	4.76	29	4	190.13
9	12	38.47	30	29	59.64

1.21 Problem Set SC35*Area Requirement untuk Problem Set SC35*

Dept.	Area	Max. Aspect Ratio	Dept.	Area	Max. Aspect Ratio
1	3	4	21	4	4
2	5	4	22	3	4
3	4	4	23	1	4
4	14	4	24	3	4
5	4	4	25	1	4
6	5	4	26	4	4
7	2	4	27	6	4
8	3	4	28	1	4
9	5	4	29	18	4
10	6	4	30	4	4
11	2	4	31	9	4
12	6	4	32	14	4
13	5	4	33	10	4
14	3	4	34	4	4
15	13	4	35	3	4
16	6	4			
17	2	4			
18	10	4			
19	4	4			
20	5	4			

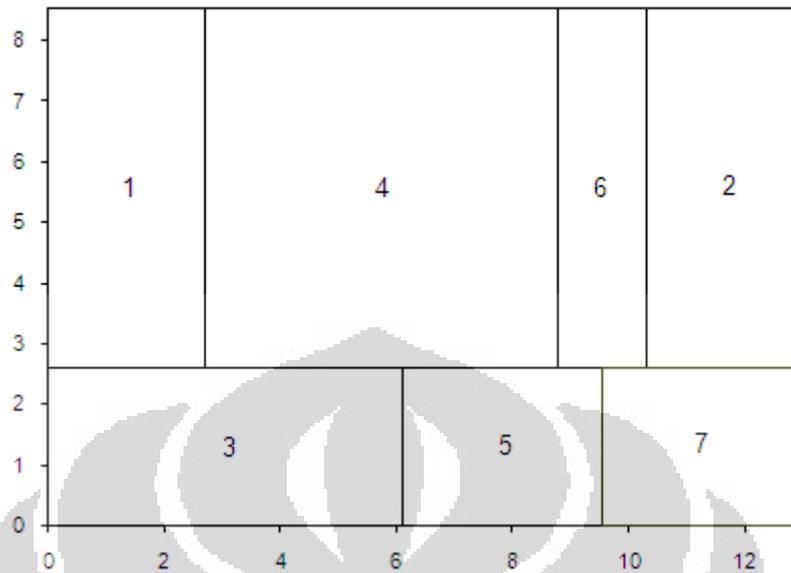
Matriks *Material Flow* untuk Problem Set SC35

From	To	<i>Material Flow</i>	From	To	<i>Material Flow</i>
1	24	2.95	11	12	8.84
1	25	6.32	12	32	72.89
1	26	1.26	13	15	12.97
1	27	2.11	14	15	4.67
1	28	1.26	15	4	63.95
1	30	13.91	15	18	190.74
2	6	20.38	16	15	4.58
2	19	4.5	17	15	0.76
2	20	3.63	18	4	190.74
2	21	2.93	19	29	9.18
2	22	1.29	20	29	7.4
2	23	1.43	21	29	5.97
4	3	225.65	22	29	2.63
5	11	4.09	23	29	2.92
5	12	33.09	24	29	5.9
6	5	8.92	25	29	12.65
6	7	2.07	26	29	2.53
6	8	4.48	27	29	4.22
6	10	4.56	28	29	2.53
6	12	1.83	29	33	190.13
6	17	0.61	30	29	59.64
6	31	3.93	31	32	22.92
7	8	12.97	32	13	12.97
8	9	43.23	32	14	4.67
9	11	4.76	32	15	53.42
9	12	38.47	32	16	1.83
10	12	38.03	33	34	168.45

Lampiran 2: Tampilan Tata Letak Fasilitas Terbaik

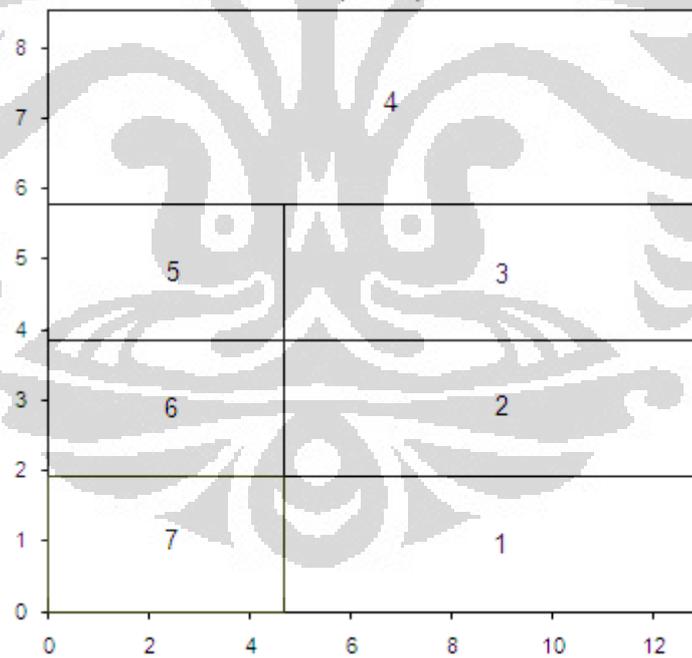
Tata Letak Terbaik untuk *Problem Set O7*

(134.19)



Tata Letak Terbaik untuk *Problem Set FO7*

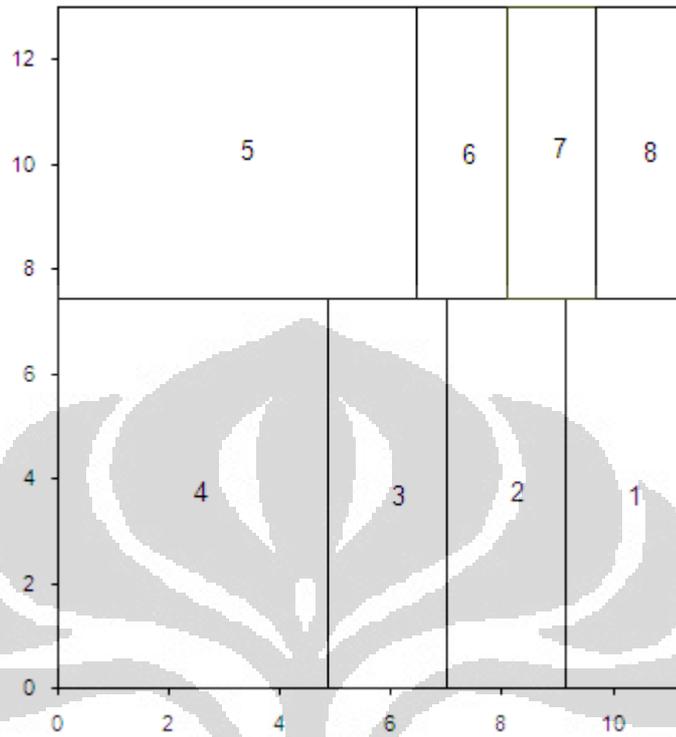
(18.88)



Lampiran 2: Tampilan Tata Letak Fasilitas Terbaik (lanjutan)

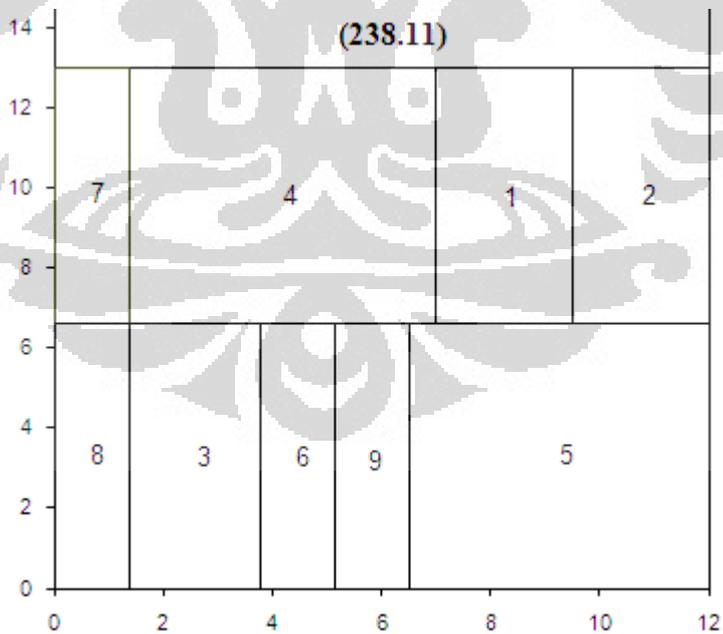
Tata Letak Terbaik untuk *Problem Set FO8*

(22.39)



Tata Letak Terbaik untuk *Problem Set O9*

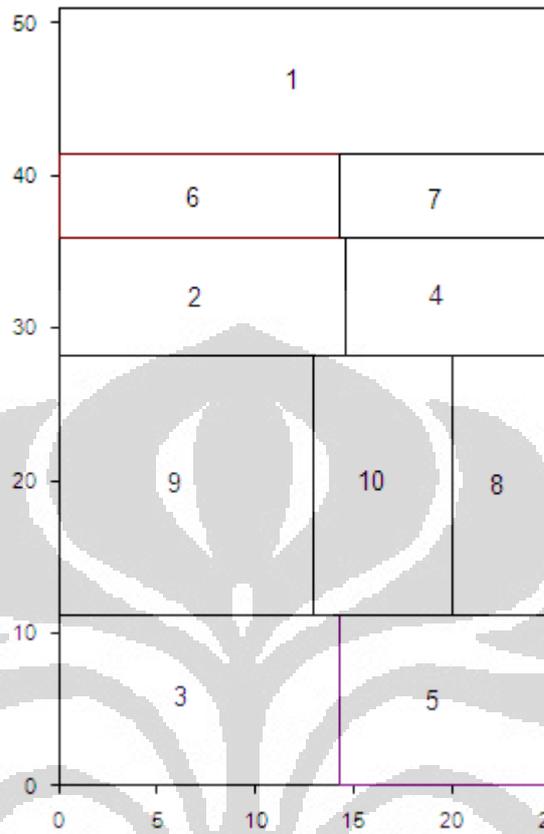
(238.11)



Lampiran 2: Tampilan Tata Letak Fasilitas Terbaik (lanjutan)

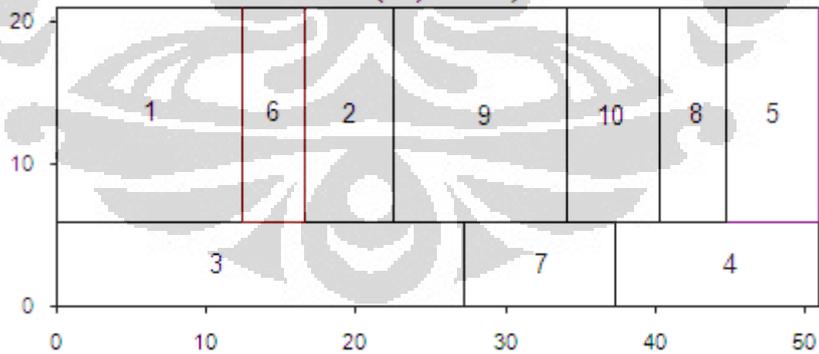
Tata Letak Terbaik untuk *Problem Set V10s*

(22,899.65)



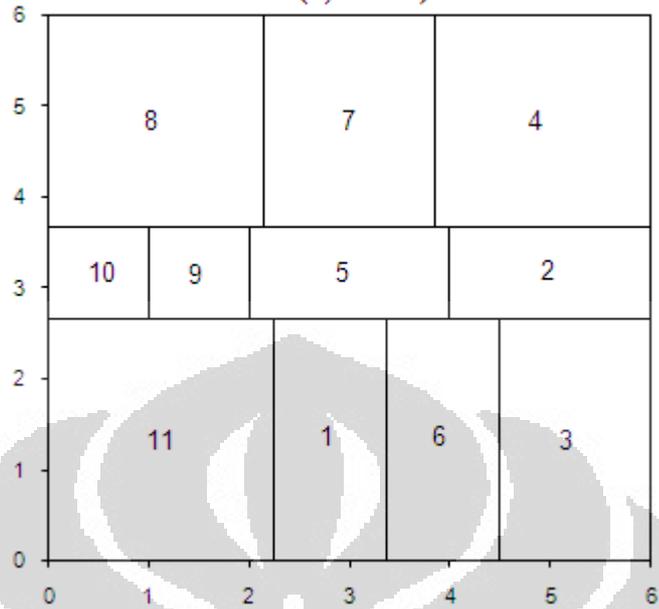
Tata Letak Terbaik untuk *Problem Set V10a*

(20,142.13)

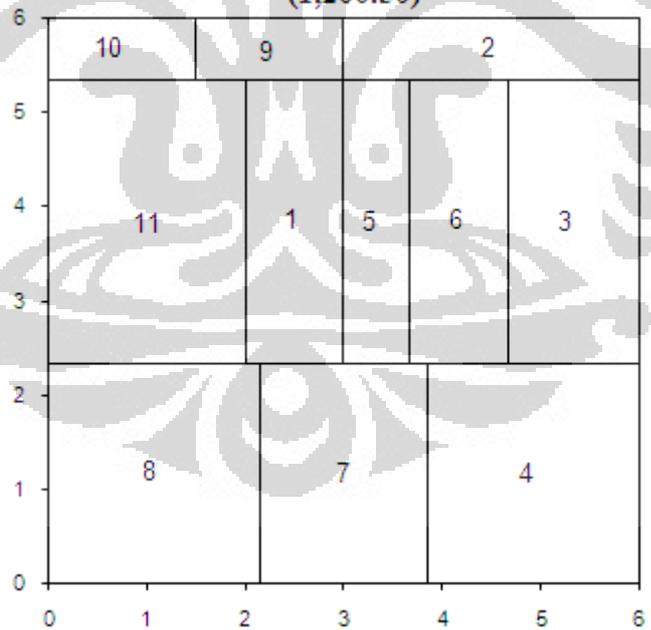


Lampiran 2: Tampilan Tata Letak Fasilitas Terbaik (lanjutan)

Tata Letak Terbaik untuk *Problem Set M11s*
(1,321.35)



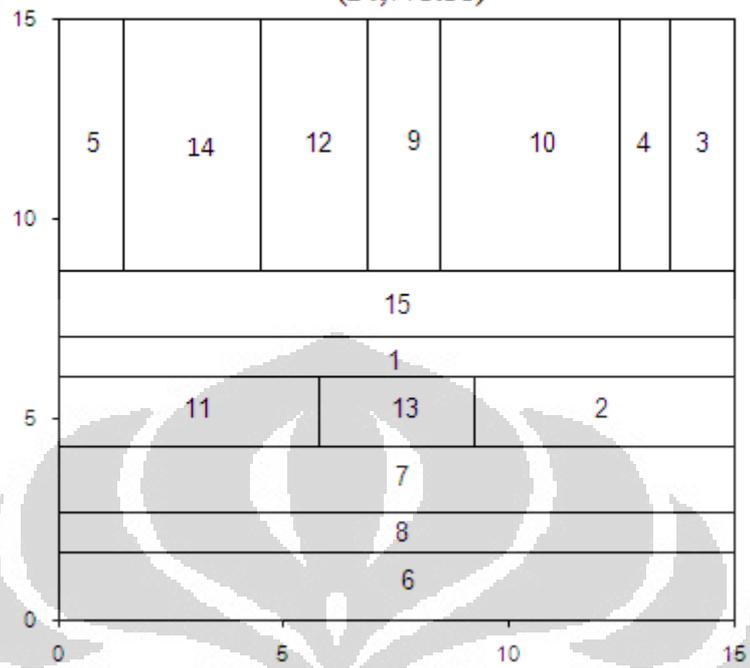
Tata Letak Terbaik untuk *Problem Set M11a*
(1,200.50)



Lampiran 2: Tampilan Tata Letak Fasilitas Terbaik (lanjutan)

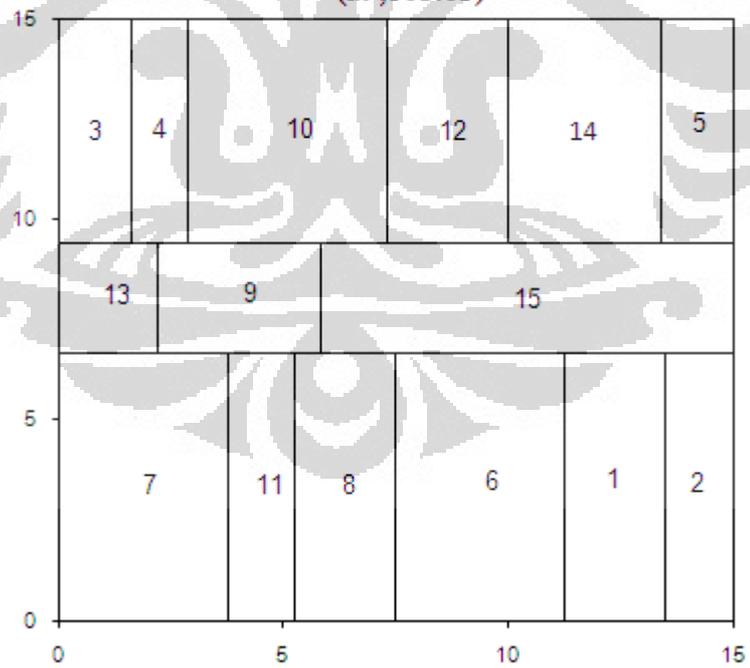
Tata Letak Terbaik untuk *Problem Set M15s*

(24,775.58)



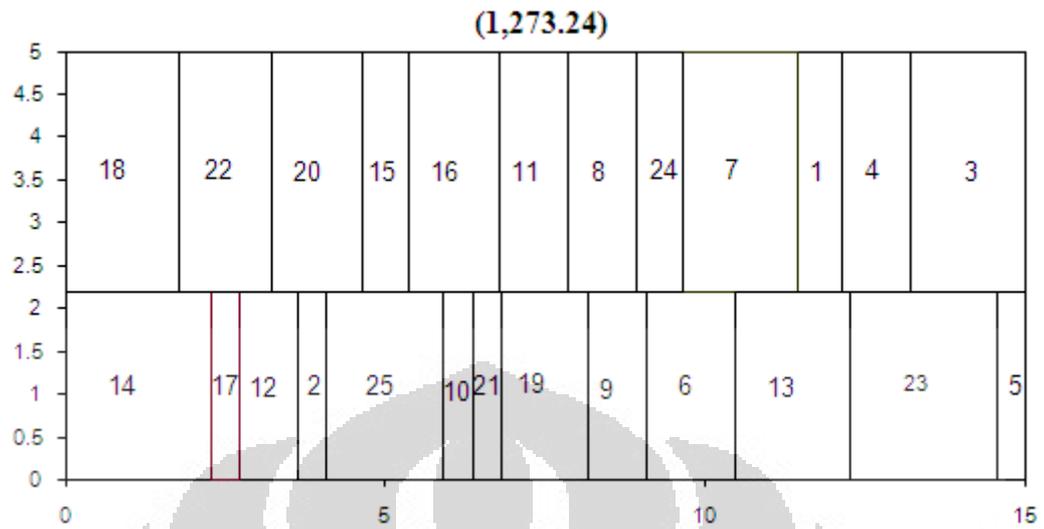
Tata Letak Terbaik untuk *Problem Set M15a*

(29,865.63)

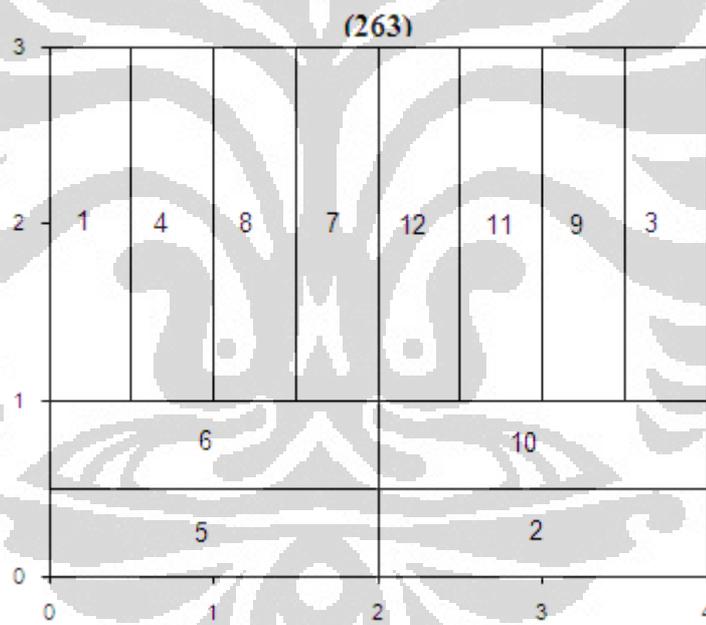


Lampiran 2: Tampilan Tata Letak Fasilitas Terbaik (lanjutan)

Tata Letak Terbaik untuk *Problem Set M25*

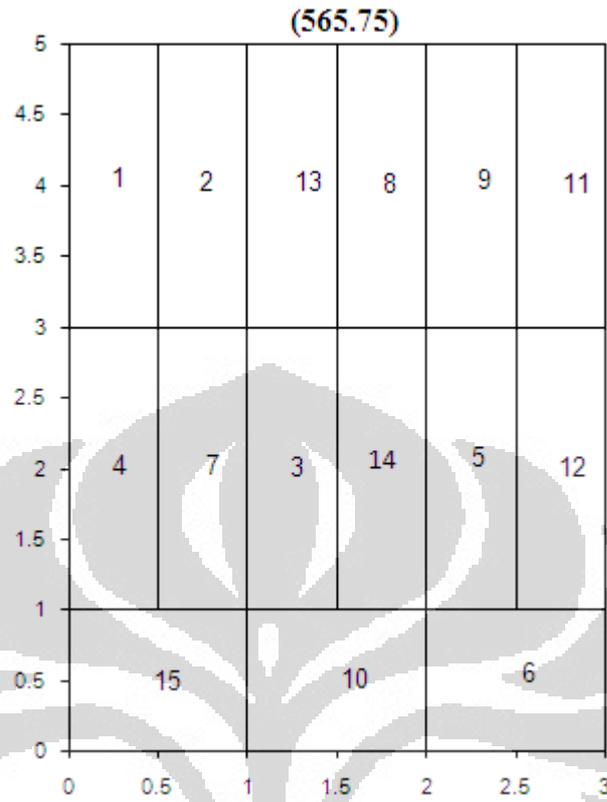


Tata Letak Terbaik untuk *Problem Set NUG12*

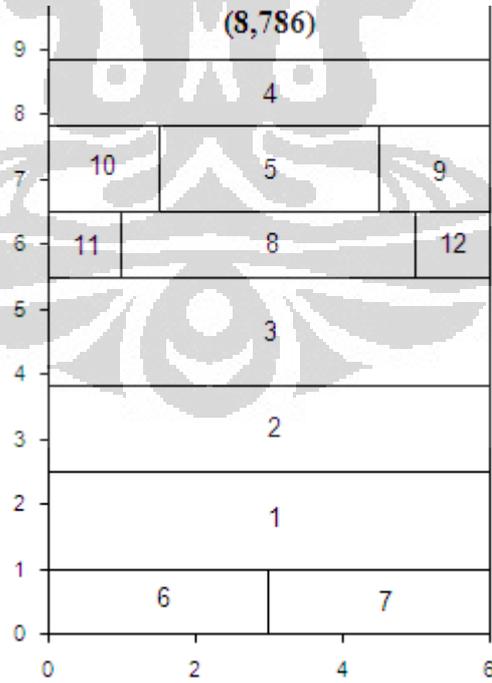


Lampiran 2: Tampilan Tata Letak Fasilitas Terbaik (lanjutan)

Tata Letak Terbaik untuk *Problem Set* NUG15

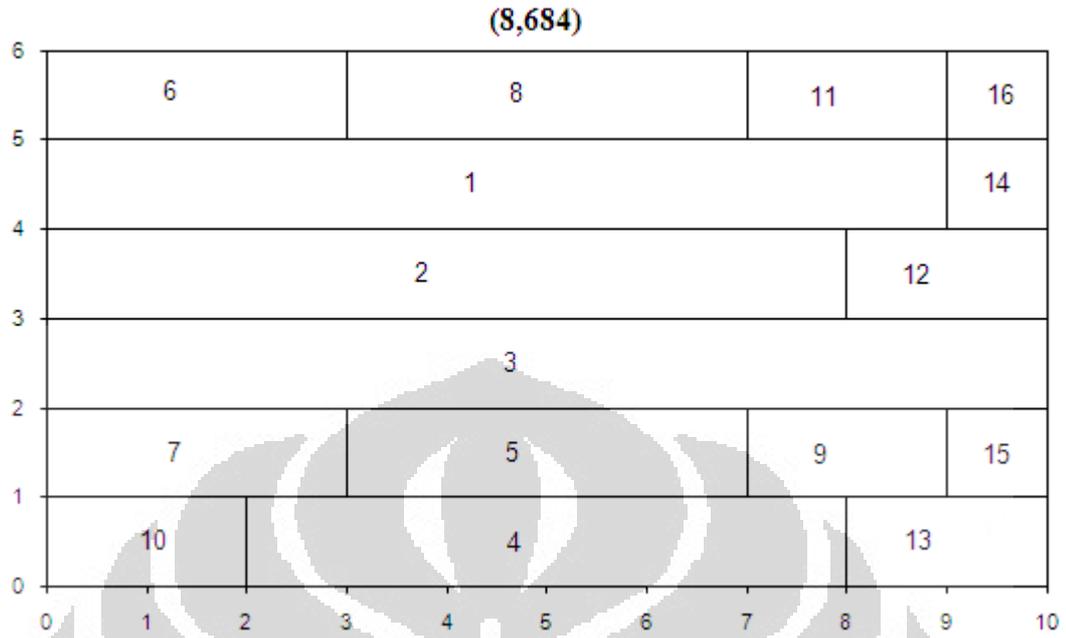


Tata Letak Terbaik untuk *Problem Set* BA12

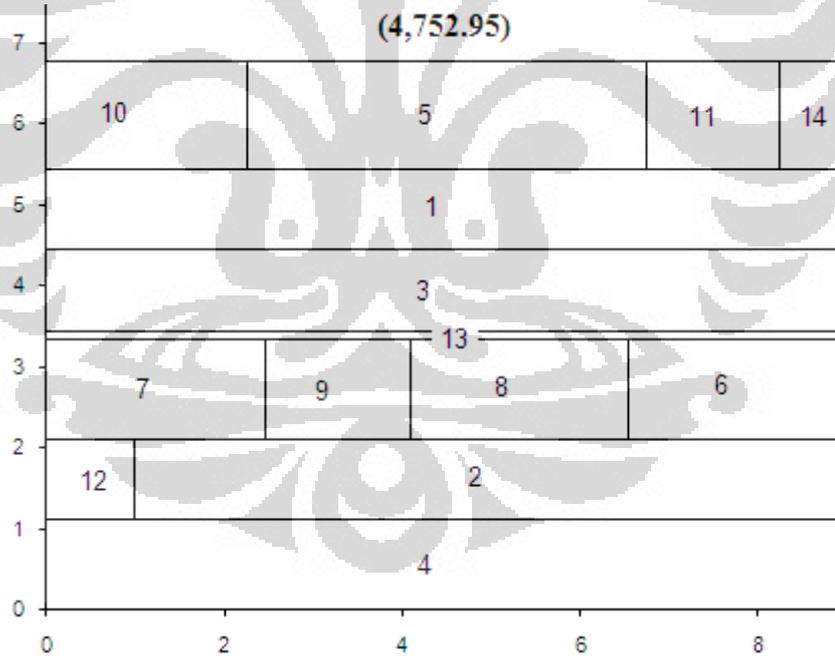


Lampiran 2: Tampilan Tata Letak Fasilitas Terbaik (lanjutan)

Tata Letak Terbaik untuk *Problem Set* BA12TS

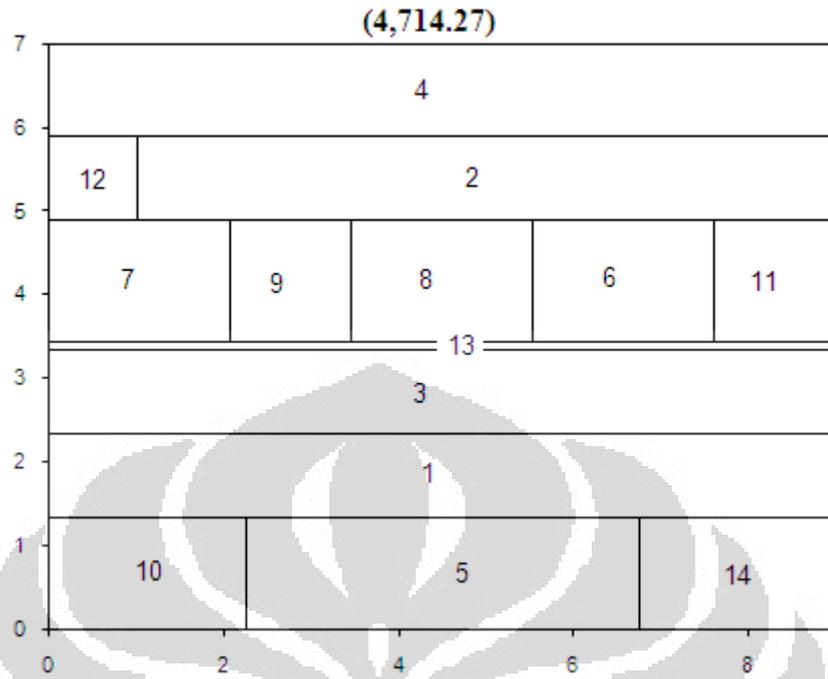


Tata Letak Terbaik untuk *Problem Set* BA14

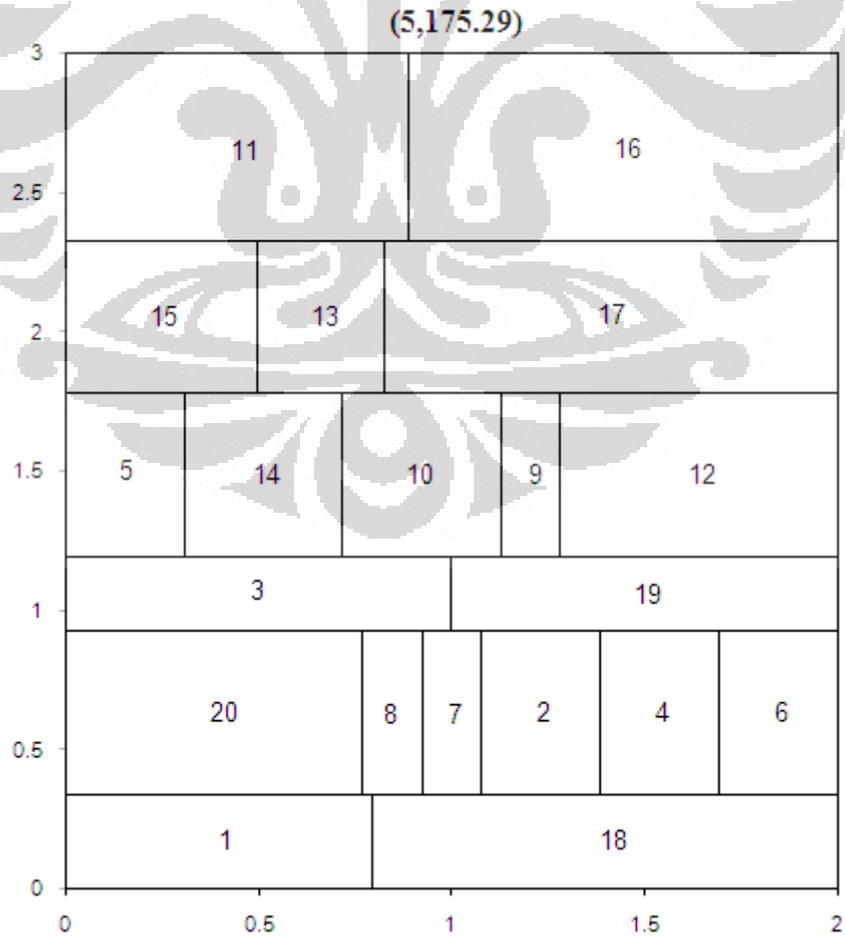


Lampiran 2: Tampilan Tata Letak Fasilitas Terbaik (lanjutan)

Tata Letak Terbaik untuk *Problem Set* BA14TS



Tata Letak Terbaik untuk *Problem Set* AB20



Lampiran 2: Tampilan Tata Letak Fasilitas Terbaik (lanjutan)

Tata Letak Terbaik untuk *Problem Set SC30*
(3,967.76)

