**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# **IDENTITAS PENGUSUL**

Nama : **I Wayan Suartika Eka Putra**

NRP : **5109100096**

Dosen Wali : **Waskitho Wibisono, S.Kom., M.Eng., Ph.D.**

1. **JUDUL TUGAS AKHIR**

***“Penyelesaian Permasalahan Penjadwalan pada Rumah Sakit dengan menggunakan Algoritma Implementor dan Adversary pada Cyclic dan Robust Schedule”***

***“Completion Scheduling Problems in Hospitals using Algorithm Implementor / Adversary on Cyclic and Robust Schedule”***

1. **ABSTRAKSI**

Masalah umum dalam rumah sakit terdiri dalam mengalokasikan beberapa sumber daya medis yang langka, seperti kamar operasi atau staf medis, untuk spesialisasi medisdimana agar antrian pasien terjaga sesingkat mungkin. Kesulitan utama pada permasalahan ini berada pada alokasi yang harus ditetapkan beberapa bulan di muka, dan jumlah yang tepat dari masing-masing pasien yang memerlukan penanganan khusus merupakan parameter pasti.

Selain masalah dalam pengalokasian terdapat juga masalah jadwal *cyclic* (berulang), dimana alokasi didefinisikan dalam waktu yang singkat, misalnya seminggu, dan kemudian diulang selama *time horizon* atau *planning horizon* (titik waktu tertentu di masa depan dimana titik proses tersebut akan dievaluasi). Namun, permintaan biasanya bervariasi dari minggu ke minggu, bahkan jika diketahui terlebih dahulu kebutuhan yang tepat untuk setiap minggu, jadwal mingguan tidak dapat disesuaikan.

Dari kedua permasalahan tersebut dimodelkan ke dalam masalah alokasi *cyclic* sebagai *adjustable* masalah penjadwalan *robust scheduling*.Dalam permasalahan ini dikembangkan juga algoritma *row and column generation* untuk memecahkan permasalahan ini dan menunjukkan bahwa hal itu sesuai dengan algoritma *implementor/adversary* untuk *robust optimization* yang baru-baru ini diperkenalkan. Penerapan *general model* untuk menghitung jadwal MSS(*Master Surgery Schedule*) di kehidupan nyata pada suatu rumah sakit besar.

1. **PENDAHULUAN**
   1. **LATAR BELAKANG**

Suatu rumah sakit dikatakan memiliki performa yang baik apabila antrian pasien pada rumah sakit tersebut sedikit. Dimana peningkatan performa dari suatu rumah sakit sangat diperlukan dalam efisiensi kinerja pada suatu rumah sakit. Permasalahan yang sering terjadi pada rumah sakit besar saat ini adalah penjadwalan pada pasien dan sumber daya dari rumah sakit itu sendiri. Penjadwalan yang kurang efisien akan menyebabkan penanganan suatu pasien menjadi lama dikarenakan antrian dan prioritas setiap pasien kurang tepat, selain itu jadwal dari staf medis yang tidak dinamis menjadikan permasalahan tersendiri dari pihak rumah sakit sehingga terjadi antrian panjang pada pasien yang memerlukan penanganan khusus dari tenaga medis special(dokter spesialis).

Proses alokasi sudah dilakukan lebih awal dari jadwal yang ditetapkan namun jumlah pasien yang memiliki penanganan special secara substansial berbeda dari dugaan awal. Selain jadwal sering dibuat dengan menggunakan refrensi *planning horizon* dan jadwal yang secara *cyclic* terus diterapkan. Jadwal *cyclic* harus dipastikan memiliki antrian yang pendek ketika sesuatu yang tidak diinginkan terjadi seperti penanganan pasien yang tidak terduga. Masalah seperti ini dapat ditangani oleh *robust optimization*. Terdapat 2 kriteria *robust* yang dapat diterapkan disini, yaiut *absolute robustness* dan *robust deviation*. *Absolute robustness* merupakan *robust* yang dapat mencari solusi dengan minimum cost. *Robust deviation* atau bisa dikatakan *min-max regret*.

* 1. **RUMUSAN MASALAH**

Adapun rumusan masalah untuk tugas akhir ini adalah :

1. Bagaimana konsep penerapan MILP(Mixed-Integer Linear Programming) dalam penyesuaian *robust model* untuk menentukan MS(Master Schedule)?
2. Bagaimana konsep penerapan algoritma implementor/adversary untuk *robust model* dalam menentukan MS?
3. Bagaimana implementasi konsep tersebut pada Matlab?
4. Bagaimana melakukan uji coba metode diatas untuk menyelesaikan penjadwalan pada pasien dan sumber daya yang ada di rumah sakit ?
   1. **BATASAN MASALAH**

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, yaitu sebagai berikut :

1. Sistem perangkat lunak dibangun dengan menggunakan perangkat lunak MATLAB 7.6.0 dan TOMLAB Optimization.
2. Data uji coba adalah simulasi data dari rumah sakit.
3. Penjadwalan diterapkan untuk pasien yang akan dioperasi.
4. Variable penjadwalan bergantung pada pasien, team dokter spesialisasi, time-slots.
   1. **TUJUAN DAN MANFAAT TUGAS AKHIR**

Tugas akhir ini bertujuan untuk :

1. Memenuhi tuntutan jam kerja pada ketersediaan pegawai dengan total jam kerja yang minimum.
2. Memenuhi target jam kerja untuk setiap pegawai.
3. Mengimplemetasikan *robust model* dengan MILP untuk penyesuaian pada MS.
4. Mengimplemetasikan *robust model* dengan *implementor/adversary algorithm* untuk penyesuaian pada permasalahan MS.
5. **TINJAUAN PUSTAKA**
   1. **Mixed-Integer Linear Programming**

Permasalahan *Linear Programming(LP)* adalah optimasi permasalahan yang mencoba memaksimalkan ataupun meminimalkan fungi linear dari variabel variabel keputusan [[1](#Hil01)]. Fungsi tersebut disebut fungsi objektif. Nilai dari variabel keputusan yang akan dicari harus memenuhi batasan batasan yang telah ditentukan. Batasan batasan tersebut harus berupa persamaan atau pertidaksamaan linear.

*Integer Linear Programming* (*ILP*) adalah pengembangan dari permasalahan Linear Programming, yaitu variabel yang digunakan harus berupa integer [[2](#Pin08)].

*Mixed-Integer Linear Programming* (*MILP*) adalah kombinasi variable yang digunakan pada Linear Programming, yaitu variable yang digunakan berupa integer dan binary. Dalam pengerjaan Tugas Akhir ini akan digunakan *MILP* untuk menyelesaikan permasalahan penjadwalan pegawai paruh waktu.

* 1. **Scheduling**

Permasalahan penjadwalan sangat berhubungan alokasi sumber daya yang tersedia untuk melakukan serangkaian pekerjaan tertentu dalam periode tertentu dengan tujuan utama yaitu untuk mengoptimalkan satu atau lebih tujuan yang harus dicapai dengan batasan batasan yang ada [[2](#Pin08)]. Penjadwalan sebagai proses pengambilan keputusan memainkan peranan yang sangat penting dalam hampir semua industri sehingga mampu meningkatkan produktivitas mereka.

Permasalahan yang diangkat dalam Tugas Akhir ini, adalah permasalahan penjadwalan untuk rumah makan makanan cepat saji**,** dimana terdapat tuntutan jam kerja yang harus dipenuhi oleh para pegawai yang tersedia. Terdapat sekumpulan pegawai yang memiliki kemampuan yang berbeda, dan kecenderungan untuk bekerja pada waktu yang berbeda, dan bekerja pada total jam kerja yang berbeda setiap minggunya. Disinilah, penjadwalan memiliki peranan yang sangat peting dalam masalah ini, yaitu untuk memenuhi tuntutan jam kerja dengan berbagai batasan yang dimiliki oleh para pegawai, sehingga meningkatkan produktivitas kerja.

* 1. **Pattern Formulation**

Formulasi tersebut didasarkan pada konsep *l-pattern*, yang mana himpunan dari *l*, dimana superslotnya berdekatan sesuai dengan hari yang sama dan sumber daya medis yang sama. Contoh, jika terdapat panjang slot 2h dan jam kerja 8h, maka kita memiliki 4 time-slots. Seperti yang dijelaskan oleh contoh kita memiliki 4 pola, *1-pattern*(1 slot atau 2h) 3 distinct, *2-pattern*(4h) 2 distinct, *3-pattern*(6h) 1 distinct dan 4-pattern(8h) [[3](#Man12)].

* 1. **MATLAB**

MATLAB adalah lingkungan pengembangan yang ditujukan untuk komputasi teknis,visualisasi,dan pemrograman [[4](#Mat12)]. MATLAB adalah system interaktif yang mempunyai basis data array yang tidak membutuhkan dimensi. Ini memungkinkan kita dapat menyelesaikan banyak masalah komputasi teknis, khususnya yang berkaitan dengan formulasi matrik dan vector.

Pemilihan MATLAB sebagai lingkungan pengembangan dalam pengerjaan Tugas Akhir ini karena MATLAB sangat handal untuk pemodelan data berupa matriks dan vektor, serta keberadaan TOMLAB yang dikembangkan pada MATLAB.

* 1. **TOMLAB Optimization**

TOMLAB adalah kakas kerja yang dikembangkan pada Matlab untuk pemecahan permasalahan optimasi. Pemodelan pada TOMLAB sangat fleksibel, mudah digunakan, dan sangat handal untuk menyelesaikan berbagai permasalahan optimasi.

TOMLAB menyediakan berbagai macam data tes, serta beberapa *solver* untuk melakukan pemecahan masalah optimasi, seperti: CPLEX, MINLP, MINOS, SOL dan lain sebagainya[[5](#TOM13)]. Dalam TOMLAB kita hanya perlu melakukan pemodelan satu kali, kemudian dapat menggunakan beberapa solver untuk membandingkan performanya.

1. **METODOLOGI**

Penjadwalan pada rumah sakit yang bertujuan untuk meningkatkan performa dari rumah sakit dengan cara mengurangi antrian pada pasien. Permasalahan ini sangat sulit diselesaikan karena membutuhkan MILP yang sangat besar. Oleh karena itu, permasalahan ini akan diselesaikan dalam 2 tahap, yaitu: penggunaan *robust model* dan penerapan algoritma *implementor/adversary* pada *robust model*, dengan menggunakan MILP untuk menyelesaikan setiap tahapan tersebut [[6](#Hoj11)].

Data uji coba terdiri dari:

* Data ruang operasi yang tersedia pada rumah sakit
* Data dokter spesialisasi yang tersedia
* Data pasien yang akan dioperasi

### MINIMALISASI QUEUE COST PADA RUMAH SAKIT

Untuk menentukan penjadwalan yang tepat dan efisien, diperlukan minimalisasi antrian pada pasien dan penggunaan ruang operasi agar *time-slots* yang ada dapat dengan efisien memberikan waktu untuk setiap pasien di operasi.

Misal:

G = himpunan dokter spesialis

D = himpunan hari pada *planning horizon*

R = himpunan ketersediaan *medical resources* (*integer* *number of* *time-slots*)

*Block* = himpunan *time-slots* yang berdekatan

= untuk setiap dokter spesialis

= untuk setiap panjang *block* yang memungkinkan

= *non-negative integer*

Hasil dari permisalan diatas didapat merupakan persamaan untuk jumlah slot yang didapat untuk setiap *g* selama D.

Untuk menyederhanakan notasi di atas diperkenalkan *superslots* S:

*r* = *medical resources*

*s* = *time-slot*

*d* = *day*

Untuk permasalahan antrian yang panjang berdampak pada *social cost*, dimana hal tersebut dapat memengaruhi kesehatan. Maka kita model permasalahan tersebut kedalam fungsi (1).

(1), dimana fungsi linear yang tidak dapat diturunkan, *convex*, dan *piece-wise*

Dikarenakan perkiraan tentang panjang dari sebuah antrian bergantung pada MS dan *demand b* secara inheren tidak pasti dan kita berasumsi bahwa, untuk setiap dan kuantitas dari

(2)

Walau pun semua prinsip *demand*  dapat bervariasi dalam interval yang sesuai, jumlah keseluruhan slot yang diperlukan oleh pasien yang memerlukan penanganan dari dokter spesialis tidak melebihi ambang batas K

(3)

Dari persamaan (2) dan (3) di atas, disimpulkan menggunakan persamaan dibawah dimana permsaan (2) dan (3) merupakan batasan untuk mendapakan minimum *queue cost*.

### PENYELESAIAN MS PROBLEM DENGAN ADJUSTABLE-ROBUST MODEL

Pada tahap ini *Master Schedule* diselesaikan dengan MILP. Permasalahan ini diadaptasi dengan *pattern formulation*. Setiap *pattern* yang sesuai untuk interval waktu yang tepat seperti dari pukul 10.00 s/d 16.00. Misal dibuktikan bahwa P merupakan himpunan dari setiap panjang *block* *l* yang memungkinkan dan , E merupakan himpunan dari *pattern* *non-disjoint pair*.

semua *pattern* yang ada;

Hal ini mungkin terjadi, ketika dokter spesialis membutuhkan peralatan khusus namun tidak tersedia disetiap ruang operasi. Misal be *two pattern*. Jika *p* dan *q* memotong, maka mereka tidak dapat diberikan secara bersamaan. Ini diwakili oleh *packing constraint*:

(4)

Jika hanya 1 tim operasi yang ada untuk *specialty g*, maka , untuk semua . Vector terpenuhi untuk semua pertidaksamaan MS. Kendala seperti ini dapat diperkuat dengan mempertimbangkan setiap himpunan pola C yang saling berpotongan. Sehingga kita dapat mengganti persamaan (4) dengan pertidaksamaan (5):

(5)

merupakan himpunan maximal dari pola yang salin berpotongan. Sekarang misalkan adalah himpunan pola dari panjang *l*. Jumlah adalah jumlah block dari panjang *l* dimana diterapkan pada specialty *g*: yang pada prinsipnya dapat melebihi jumlah . Maka persyaratan dari queue :

(6)

*Cost Convex piece-wise linear* berkaitan dengan *q* dapat dengan mudah berubah menjadi 1 *linear* dengan memperkenalkan *variable* dan *constraint* yang sesuai. Untuk kesederhanaan pada kasus fungsi *linear cost*, khususnya pada *cost* dari antrian q agar menjadi sebanding dengan total panjang antrian.

(7)

Dari persamaan (5) kita kaitkan dengan *variable* bilangan *real non-negative* dengan , untuk , maka *MS problem* dapat diformulasikan menjadi MILP:

Min

s.t (i)

(ii) (8)

(iii)

**Cyclic dan Robust Schedules**

Kita asumsikan *demand b* berada pada himpunan *Y* yang telah ditentukan sebelumnya, sehingg dapat dituliskan seperti ini :

(9)

Mengingat permaslahan aR-MS, yang terdiri dari MS x, dimana dapat meminimalkan *queue cost* sesuai dengan permintaan terburuk yang mungkin terjadi di Y, dapat dinyatakan sebagai *min-max problem* :

(10)

Untuk setiap , dibuktikan , queue pada saat *vector* dimana . Pada kasus seperti ini, dapat diasosiasikan *variable* dengan dimana keduanya merupakan bilangan *Real* *non-negative*. Maka permasalahan aR-MS dapat direpresentasikan kedalam MILP :

Min

s.t (i)

(ii) (11)

(iii)

(iv)

**PENERAPAN ALGORITMA IMPLEMENTOR/ADVERSARY PADA aR-MS PROBLEM**

Diadaptasi dari pendekatan dimana memiliki batasan optimal solution untuk memecahkan MILP (10). Yakni, kita mulai dengan mempertimbangkan *subset* dari *vector demand* yang mungkin sesuai dengan himpunan *index* dari :

Min

s.t (i)

(ii) (12)

(iii)

(iv)

Max

s.t (i) (13)

Persamaan (12.i) merupakan *constraint non-linear* dimana untuk memastikan . merupakan queue yang berhubungan dengan vector demand . Untuk men-*linear*-kan constrain (12.i) tidak bisa dilanjutkan, karena bentuk specific dari tujuan fungsi pada (12). Sebagai gantinya diperkenalkan , variable binary yang mana bernilai 1 jika atau samadengan 0. Jadi setiap constraint (12.i) diganti sebagai berikut :

1. (13)

Karena Q memiliki nilai konstanta yang besar, jika maka nilai (bisa dilihat pada (13ii) dan (13iii)). Sebaliknya jika maka (13ii) akan menjadi *redundant*, jadi (13i) dirubah menjadi :

**The implementor/adversary algorithm**

Untuk penerapan *robust optimization* seperti yang disarankan[[7](#Dan07)] diterapkan pendekatan *column and row generation* dan algoritma *implementor/adversary*(I/A).

**Output:** value L dan U dengan , dan sehingga .

**Initialization:** , , .

**Iterate:**

1. **Implementor Problem:** *solve* , dengan solusi . *Reset*
2. **Adversary Problem:** *solve* , dengan solusi . *Reset*
3. **Test:** If *is small, exit, else reset* *and go to* **1**.

Gambar Diagram Alir Algoritma Implementor/Adversary



1. **JADWAL KEGIATAN**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Tahapan | Bulan | | | | | | | | | | | | | | | | | | | |
| Maret | | | | April | | | | Mei | | | | Juni | | | | Juli | | | |
| 1. | Penyusunan Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2. | Studi Literatur |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3. | Implementasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4. | Pengujian dan Evaluasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5. | Penyusunan Buku Tugas Akhir |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. **DAFTAR PUSTAKA**

x

|  |  |
| --- | --- |
| [1] | Frederick S Hillier and Gerald J Lieberman, *Introduction to Operations Research*, 7th ed., 2001. |
| [2] | Michael L Pinedo, *Scheduling: Theory , Algorithm , and Systems*, 3rd ed.: Prentice Hall, 2008. |
| [3] | Carlo Mannino, Eivind J. Nilssen, and Tomas Eric Nordlander, "A pattern based, robust approach to cyclic master surgery," *J Sched*, vol. 15, pp. 553-563, May 2012. |
| [4] | (2012, February) MATLAB- The language of technical computing. [Online]. <http://www.mathworks.com/products/matlab/> |
| [5] | (2013, February) TOMLAB OPTIMIZATION. [Online]. <http://tomopt.com/tomlab/about/> |
| [6] | Mehran Hojati and Ashok S Patil, "An integer linear programming-based heuristic for scheduling heterogeneous part-time service employees," *European Journal of Operational Research 209*, pp. 37-50, 2011. |
| [7] | Daniel Bienstock, "Histogram Models for Robust Portfolio Optimization," *Journal Of Computational Finance*, vol. 11, pp. 1-64, 2007. |

x