



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - KS141501**

**OPTIMASI PENJADWALAN STAF DENGAN  
MENGUNAKAN ALGORITMA REINFORCEMENT  
LEARNING HYPER-HEURISTICS STUDI KASUS  
RUMAH SAKIT IBU DAN ANAK KENDANGSARI**

***OPTIMIZATION ON STAFF SCHEDULLING PROBLEM  
USING REINFORCEMENT LEARNING ALGORITHM  
HYPER-HEURISTIC. STUDY CASE IN  
KENDANGSARI'S MOM AND CHILD HOSPITAL***

**FATA HIRZI ABI KARAMI  
NRP 5214100180**

**Dosen Pembimbing :  
Ahmad Muklason, S.Kom., M.Sc., Ph.D.**

**DEPARTEMEN SISTEM INFORMASI  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017**





**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - KS141501**

**OPTIMASI PENJADWALAN STAF DENGAN  
MENGUNAKAN ALGORITMA REINFORCEMENT  
LEARNING HYPER-HEURISTICS STUDI KASUS RUMAH  
SAKIT IBU DAN ANAK KENDANGSARI**

**FATA HIRZI ABI KARAMI**  
NRP 5214100180

**Dosen Pembimbing :**  
Ahmad Muklason, S.Kom., M.Sc., Ph.D.

**DEPARTEMEN SISTEM INFORMASI**  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017





**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**FINAL PROJECT – KS141501**

***OPTIMIZATION ON STAFF SCHEDULLING PROBLEM  
USING REINFORCEMENT LEARNING ALGORITHM  
HYPER-HEURISTIC. STUDY CASE IN KENDANGSARI'S  
MOM AND CHILD HOSPITAL***

**FATA HIRZI ABI KARAMI**  
**NRP 5214100180**

**Supervisor :**  
**Ahmad Muklason, S.Kom., M.Sc., Ph.D.**

**INFORMATION SYSTEM DEPARTMENT**  
**Information Technology and Communication Faculty**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya 2017**



## LEMBAR PENGESAHAN

### OPTIMASI PENJADWALAN STAF DENGAN MENGUNAKAN ALGORITMA REINFORCEMENT LEARNING HYPER-HEURISTICS STUDI KASUS RUMAH SAKIT IBU DAN ANAK KENDANGSARI

#### TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat Memperoleh  
Gelar Sarjana Komputer  
Pada  
Departemen Sistem Informasi  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember

Oleh:

**Fata Hirzi Abi Karami**  
5214100180

Surabaya, Januari 2018

Plh Kepala  
Departemen Sistem Informasi



**Edwin Riksakomara, S.Kom, MT.**  
NIP. 196907252003121001



**LEMBAR PERSETUJUAN**

**OPTIMASI PENJADWALAN STAFF DENGAN  
MENGUNAKAN ALGORITMA REINFORCEMENT  
LEARNING HYPER-HEURISTICS STUDI KASUS  
RUMAH SAKIT IBU DAN ANAK KENDANGSARI**

**TUGAS AKHIR**

Disusun untuk Memenuhi Salah Satu Syarat Memperoleh  
Gelar Sarjana Komputer

Pada

Departemen Sistem Informasi  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember

Oleh:

**Fata Hirzi Abi Karami**

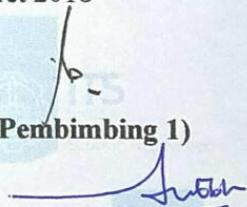
**05211440000180**

Surabaya, Januari 2018

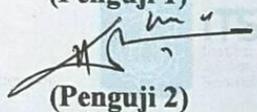
Disetujui Tim Penguji : Tanggal Ujian : 12 Januari 2018  
Periode Wisuda : Maret 2018

**Ahmad Muklason, S.Kom., M.Sc., Ph.D (Pembimbing 1)**

**Wiwik Anggraeni, S.Si, M.Kom**

  
**(Penguji 1)**

**Faizal Mahananto, S.Kom., M.Eng., Ph.D**

  
**(Penguji 2)**





**OPTIMASI PENJADWALAN STAF DENGAN  
MENGUNAKAN ALGORITMA REINFORCEMENT  
LEARNING HYPER-HEURISTICS STUDI KASUS  
RUMAH SAKIT IBU DAN ANAK KENDANGSARI**

**Nama Mahasiswa** : Fata Hirzi Abi Karami  
**NRP** : 5214100180  
**Jurusan** : Sistem Informasi FTIf-ITS  
**Pembimbing I** : Ahmad Muklason, S.Kom., M.Sc.,  
**Ph.D.**

**ABSTRAK**

*Penjadwalan staf merupakan suatu permasalahan yang sering dihadapi oleh setiap Rumah Sakit. Untuk itu, optimasi penjadwalan staf sangat diperlukan oleh pihak Rumah Sakit untuk mendapatkan jadwal yang sesuai dengan kebutuhan dan sumber daya manusia Rumah Sakit. Sebelum melakukan penjadwalan, ada batasan-batasan yang harus dipertimbangkan seperti aturan penjadwalan dan pembagian shift pada Rumah Sakit. Hal-hal tersebut nantinya akan digolongkan menjadi dua yaitu hard constraint dan soft constraint. Pada penelitian ini membahas tentang implementasi algoritma Reinforcement Learning Hyper-Heuristic untuk menyelesaikan permasalahan penjadwalan staf di Rumah Sakit Ibu dan Anak Kendangsari. Untuk mengukur tingkat optimasi dari penjadwalan akan dihitung dari nilai Jain Fairness Index (JFI) dari masing-masing unit pada rumah sakit. Unit-unit tersebut adalah unit farmasi, bayi/NICU, gizi, IGD, Kamar Operasi, dan SIM & RM. Nilai JFI berkisar antara 0 sampai 1 dan jadwal akan semakin optimal jika nilai JFI mendekati 1. Kemudian algoritma Reinforcement Learning Hyper-Heuristic akan diterapkan untuk memilih low-level heuristic yang memiliki solusi terbaik. Low-level heuristic yang digunakan pada tugas akhir ini adalah Move dan Swap. Hasil dari optimasi ini adalah berupa perbandingan nilai JFI dari jadwal eksisting rumah sakit dengan jadwal hasil optimasi. Nilai JFI unit farmasi yang awalnya bernilai 0,80 berubah menjadi 0,97, unit bayi/NICU dari 0,67 menjadi 0,96, unit gizi dari 0,90*

*menjadi 0,84, unit IGD dari 0,91 menjadi 0,96, unit Kamar Operasi dari 0,80 menjadi 0,98, dan unit SIM & RM dari 0,74 menjadi 1. Hasil optimasi ini diharapkan dapat memberikan manfaat terutama bagi pihak Rumah Sakit dan staf untuk mendapatkan jadwal yang paling optimal berdasarkan tingkat keadilan untuk tiap staf.*

***Kata kunci : Optimasi, Penjadwalan Staf, Reinforcement Learning, Hyper-Heuristic***

**OPTIMIZATION ON STAFF SCHEDULLING  
PROBLEM USING REINFORCEMENT LEARNING  
ALGORITHM HYPER-HEURISTIC. STUDY CASE IN  
KENDANGSARI'S MOM AND CHILD HOSPITAL**

**Name** : Fata Hirzi Abi Karami  
**NRP** : 5214100180  
**Jurusan** : Information System FTIf-ITS  
**Pembimbing I** : Ahmad Muklason, S.Kom., M.Sc., Ph.D.

**ABSTRACT**

*Staff scheduling is a problem that is often faced by every hospital. For that, the optimization of staff scheduling is needed by the Hospital to get a schedule that suits the needs and the availability of human resources at the Hospital. Before scheduling, there are some constraints to be considered such as scheduling rules and the distribution of shifts in the Hospital. These constraints are also will be classified into two category, namely hard constraint and soft constraint. This study discuss about the implementation of Reinforcement Learning Hyper-Heuristic algorithm to solve the staff scheduling problem at Kendangsari's Maternity and Child Hospital. To measure the optimization level of staff scheduling, value of Jain Fairness Index (JFI) will be calculated for each division in the hospital. These divisions are pharmacy, infant/NICU, nutrition, IGD, Operation Khamer, and Registration. JFI value ranges from 0 to 1 and the schedule will be more optimal if the value of JFI close to 1. Then the Reinforcement Learning Hyper-Heuristic algorithm will be applied to choose the low-level heuristic that has the best solution. Low-level heuristic used in this final project are Move and Swap. The result of this optimization is a comparison of JFI value from the existing schedule of the hospital and the optimization schedule. The value of the JFI pharmacy division initially valued at 0.80 changed to 0.97, the infant/NICU division from 0.67 to 0.96, the nutrition division*

*from 0.90 to 0.84, the IGD division from 0.91 to 0.96 , Operational Khamer division from 0.80 to 0.98, and Registration division from 0.74 to 1. This result is expected to provide benefits, especially for the Hospital and staff to get the most optimal schedule based on the fairness level for each staff.*

***Kata kunci : Optimization, Staff Scheduling, Reinforcement Learning, Hyper-Heuristic***

## KATA PENGANTAR

Puji syukur kepada Tuhan yang Maha Esa sehingga penulis dapat menyelesaikan buku tugas akhir dengan judul **”OPTIMASI PENJADWALAN STAF DENGAN MENGGUNAKAN ALGORITMA REINFORCEMENT LEARNING HYPER-HEURISTICS STUDI KASUS RUMAH SAKIT IBU DAN ANAK KENDANGSARI”** yang merupakan salah satu syarat kelulusan pada Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Secara khusus penulis ingin memberikan ucapan terima kasih sedalam-dalamnya kepada :

1. Allah SWT, yang telah memberikan kesehatan, kemudahan, kelancaran, dan kesempatan untuk penulis hingga dapat menyelesaikan Tugas Akhir ini.
2. Bapak Dr. Ir. Aris Tjahyanto, M.Kom, selaku Ketua Jurusan Sistem Informasi ITS, yang telah menyediakan fasilitas terbaik untuk kebutuhan penelitian mahasiswa.
3. Bapak Ahmad Muklason, S.Kom., M.Sc., Ph.D. Selaku Dosen Pembimbing yang telah banyak meluangkan waktu untuk membimbing, mengarahkan dan mendukung dalam penyelesaian Tugas Akhir
4. Ibu Wiwik Anggraeni, S.Si., M.Kom selaku dosen wali yang telah memberikan arahan terkait perkuliahan di Jurusan Sistem Informasi
5. Seluruh dosen pengajar beserta staff dan karyawan di Jurusan Sistem Informasi, FTIf ITS Surabaya yang telah memberikan ilmu dan bantuan kepada penulis selama 7 semester ini
6. Kedua orang tua, adik, dan seluruh keluarga yang selalu hadir dan senantiasa mendoakan dan memberikan kasih sayang serta semangat tiada henti untuk menyelesaikan Tugas Akhir ini

7. Teman-teman bimbingan bapak Ahmad Muklason yaitu Zuli Maulidati, Rizka Pordella, Indriarti Kusumanita, dan Fachrur Zaffrinda Prayogo yang selalu bekerja sama untuk menyelesaikan Tugas Akhir ini dengan baik
8. Teman-teman seperjuangan laboratorium RDIB, ADDI, dan OSIRIS (SI-2014) yang selalu memberikan dukungan positif hingga penulis bisa menyelesaikan Tugas Akhir ini
9. Para teman dekat (Warung Squad) yang selalu memotivasi, memberikan dukungan serta membantu penulis selama di perkuliahan sehingga bisa menyelesaikan penelitian Tugas Akhir ini
10. Serta semua pihak yang telah membantu dalam pengerjaan Tugas Akhir ini yang belum mampu penulis sebutkan diatas.

Penulis menyadari bahwa Tugas Akhir ini masih belum sempurna dan memiliki banyak kekurangan di dalamnya. Dan oleh karena itu, penulis meminta maaf atas segala kesalahan yang dibuat penulis dalam buku Tugas Akhir ini. Penulis membuka pintu selebar-lebarnya kepada pihak-pihak yang ingin memberikan kritik, saran, masukan, dan penelitian selanjutnya yang ingin menyempurnakan karya, dan Tugas Akhir ini. Semoga buku Tugas Akhir ini bermanfaat bagi seluruh pembaca.

Surabaya, Januari 2018  
Penulis,

(Fata Hirzi Abi Karami)

## DAFTAR ISI

ABSTRAK.....	vi
<i>ABSTRACT</i> .....	viii
KATA PENGANTAR.....	x
DAFTAR ISI.....	xii
DAFTAR GAMBAR .....	xvi
DAFTAR TABEL .....	xix
DAFTAR KODE PROGRAM .....	xxii
BAB I PENDAHULUAN .....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah .....	3
1.3 Batasan masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	4
1.6 Relevansi .....	4
BAB II TINJAUAN PUSTAKA .....	5
2.1 Penelitian Terdahulu .....	5
2.2 Dasar teori.....	9
2.2.1 RSIA Kendangsari .....	9
2.2.2 Nurse Rostering Problem (NRP).....	9
2.2.3 Reinforcement Learning .....	12
2.2.4 Hill Climbing .....	13
2.2.5 Hyper-Heuristic .....	14
2.2.6 <i>Jain Fairness Index</i> .....	15
BAB III METODOLOGI PENELITIAN .....	17
3.1 Diagram Metodologi.....	17
3.2 Uraian Metodologi.....	18
3.2.1 Identifikasi Masalah di RSIA Kendangsari .....	18
3.2.2 Studi Literatur .....	18

3.2.3	Pengumpulan data-data yang dibutuhkan.....	18
3.2.4	Penyusunan Model Matematis .....	18
3.2.5	Implementasi Algoritma Reinforcement Learning .....	20
3.2.6	Pengujian Performa Algoritma .....	21
3.2.7	Perbandingan Jadwal Eksisting Dengan Hasil Penjadwalan Otomatis dan Kesimpulan .....	21
3.2.8	Penyusunan Tugas Akhir .....	21
<b>BAB IV DATA MASUKAN DAN PEMODELAN.....</b>		<b>23</b>
4.1	Hasil Pengumpulan Data.....	23
4.2	Kebijakan dan Regulasi Tiap Unit .....	24
4.2.1	Unit Farmasi .....	24
4.2.2	Unit Bayi/NICU .....	27
4.2.3	Unit Gizi .....	30
4.2.4	Unit IGD .....	34
4.2.5	Unit Kamar Operasi .....	37
4.2.6	Unit SIM & RM.....	39
4.3	Proses Pembuatan Model .....	41
4.3.1	Batasan .....	42
4.3.2	Asumsi dan Notasi .....	44
4.3.3	Variabel Keputusan.....	44
4.3.4	Pemodelan Hard Constraint .....	44
4.3.5	Pemodelan Soft Constraint.....	50
4.3.6	Fungsi Tujuan .....	51
4.4	Pemodelan Algoritma Reinforcement Learning.....	51
4.5	Pemodelan Algoritma Hill Climbing .....	53
<b>BAB V IMPLEMENTASI.....</b>		<b>55</b>

5.1	Lingkungan Uji Coba.....	55
5.2	Tahapan Otomasi Penjadwalan .....	55
5.2.1	Unit Farmasi .....	56
5.2.2	Unit Bayi/NICU.....	56
5.2.3	Unit Gizi .....	57
5.2.4	Unit IGD .....	58
5.2.5	Unit Kamar Operasi .....	58
5.2.6	Unit SIM & RM.....	58
5.3	Algoritma Otomasi dan Cek Constraint Penjadwalan	59
5.3.1	Otomasi Unit Farmasi .....	60
5.3.2	Otomasi Unit Bayi/NICU.....	63
5.3.3	Otomasi Penjadwalan Gizi.....	64
5.3.4	Otomasi Unit IGD.....	68
5.3.5	Otomasi Unit Kamar Operasi.....	69
5.3.6	Otomasi Unit SIM & RM.....	72
5.4	Algoritma Optimasi Penjadwalan .....	73
5.4.1	Jain Fairness Index.....	74
5.4.2	Low Level Heuristic .....	75
5.4.3	Reinforcement Learning .....	77
5.4.4	Hill Climbing .....	79
<b>BAB VI HASIL DAN PEMBAHASAN .....</b>		<b>81</b>
6.1	Jain Fairness Index Jadwal Eksisting .....	81
6.2	Hasil Otomasi Jadwal dan Jain Fairness Index.....	85
6.2.1	Hasil Otomasi Unit Farmasi.....	86
6.2.2	Hasil Otomasi Unit Bayi/NICU .....	86
6.2.3	Hasil Otomasi Unit Gizi.....	88
6.2.4	Hasil Otomasi Unit IGD .....	89

6.2.5 Hasil Otomasi Unit Kamar Operasi .....	90
6.2.6 Hasil Otomasi Unit SIM & RM .....	90
6.3 Hasil Optimasi Jadwal .....	92
6.3.1 Analisis Hasil Optimasi .....	98
6.3.2 Diagram Trajectory .....	106
6.3.3 Perbandingan hasil algoritma reinforcement learning dan hill climbing .....	119
6.3.4 Statistik .....	123
6.3.5 Box Plot .....	126
6.4 Perbandingan Nilai Jain Fairness Index .....	131
<b>BAB VII KESIMPULAN DAN SARAN</b> .....	<b>135</b>
7.1 Kesimpulan .....	135
7.2 Saran .....	136
<b>DAFTAR PUSTAKA</b> .....	<b>137</b>
<b>BIODATA PENULIS</b> .....	<b>139</b>
<b>LAMPIRAN</b> .....	<b>140</b>
Lampiran A: Interview Protocol .....	140
Lampiran B: Hasil Pengujian Algoritma .....	145

## DAFTAR GAMBAR

Gambar 2.1 Framework Hyper-Heuristic. A. Muklason, "Hyper-Heuristics And Fairness In Examination Timetabling Problems," 2017 .....	15
Diagram Metodologi 3.1 Diagram Metodologi .....	17
4.1 Pseudocode Algoritma Reinforcement Learning Extended Great Deluge base Hyper-heuristics. A. Muklason, "Hyper-Heuristics And Fairness In Examination Timetabling Problems," 2017 .....	52
Gambar 4.2 Pseudocode Algoritma Hill Climbing .....	54
Gambar 6.1 Jadwal eksisting unit farmasi bulan November..	81
Gambar 6.2 Jadwal eksisting unit Bayi/NICU bulan November .....	82
Gambar 6.3 Jadwal eksisting unit gizi bulan November.....	83
Gambar 6.4 Jadwal eksisting unit IGD bulan November.....	84
Gambar 6.5 Jadwal eksisting unit Kamar Operasi bulan November.....	84
Gambar 6.6 Jadwal eksisting unit SIM & RM bulan November .....	85
Gambar 6.7 Otomasi jadwal staf unit farmasi .....	86
Gambar 6.8 Otomasi jadwal dan Jain Fairness Index hasil otomasi unit Farmasi .....	86
Gambar 6.9 Otomasi jadwal staf unit Bayi/Nicu .....	87
Gambar 6.10 Jain Fairness Index hasil otomasi unit Bayi/Nicu .....	87
Gambar 6.11 Otomasi jadwal staf unit gizi .....	88
Gambar 6.12 Jain Fairness Index hasil otomasi unit gizi.....	88
Gambar 6.13 Otomasi jadwal staf unit IGD .....	89
Gambar 6.14 Jain Fairness Index hasil otomasi unit IGD.....	89
Gambar 6.15 Otomasi jadwal staf unit OK.....	90
Gambar 6.16 Jain Fairness Index hasil otomasi unit OK.....	90
Gambar 6.17 Otomasi jadwal staf unit SIM & RM .....	90
Gambar 6.18 Jain Fairness Index hasil otomasi unit SIM & RM .....	91
Gambar 6.19 Diagram trajectory unit farmasi menggunakan algoritma reinforcement learning .....	107

Gambar 6.20 Diagram trajectory unit farmasi menggunakan algoritma hill climbing .....	108
Gambar 6.21 Diagram trajectory unit Bayi/NICU menggunakan algoritma reinforcement learning .....	109
6.22 Diagram trajectory unit Bayi/NICU menggunakan algoritma hill climbing .....	110
Gambar 6.23 Diagram trajectory unit gizi menggunakan algoritma reinforcement learning .....	111
Gambar 6.24 Diagram trajectory unit gizi menggunakan algoritma reinforcement learning .....	112
Gambar 6.25 Diagram trajectory unit IGD menggunakan algoritma reinforcement learning .....	113
Gambar 6.26 Diagram trajectory unit IGD menggunakan algoritma hill climbing .....	114
Gambar 6.27 Diagram trajectory unit Kamar Operasi menggunakan algoritma reinforcement learning .....	115
Gambar 6.28 Diagram trajectory unit Kamar Operasi menggunakan algoritma hill climbing .....	116
Gambar 6.29 Diagram trajectory unit SIM & RM menggunakan algoritma reinforcement learning .....	117
Gambar 6.30 Diagram trajectory unit SIM & RM menggunakan algoritma hill climbing .....	118
Gambar 6.31 Perbandingan best solution algoritma reinforcement learning dan hill climbing unit farmasi.....	119
Gambar 6.32 Perbandingan best solution algoritma reinforcement learning dan hill climbing unit Bayi/NICU	120
Gambar 6.33 Perbandingan best solution algoritma reinforcement learning dan hill climbing unit gizi.....	120
Gambar 6.34 Perbandingan best solution algoritma reinforcement learning dan hill climbing unit IGD .....	121
Gambar 6.35 Perbandingan best solution algoritma reinforcement learning dan hill climbing unit Kamar Operasi .....	122
Gambar 6.36 Perbandingan best solution algoritma reinforcement learning dan hill climbing unit SIM & RM ..	122
Gambar 6.37 Box Plot unit farmasi .....	127
Gambar 6.38 Box Plot unit Bayi/NICU .....	127

Gambar 6.39 Box Plot unit gizi .....	128
Gambar 6.40 Box Plot unit IGD .....	129
Gambar 6.41 Box Plot unit Kamar Operasi .....	129
Gambar 6.42 Box Plot unit SIM & RM .....	130

## DAFTAR TABEL

Tabel 2.1 Penjelasan Penelitian Terdahulu .....	5
Tabel 2.2 Set. Shahriar Asta, Ender Özcan , Tim Curtois, “A tensor based hyper-heuristic for nurse rostering”, 2016 .....	10
Tabel 2.3 Parameter. Shahriar Asta, Ender Özcan , Tim Curtois, “A tensor based hyper-heuristic for nurse rostering”, 2016... ..	10
Tabel 2.4 Variabel. Shahriar Asta, Ender Özcan , Tim Curtois, “A tensor based hyper-heuristic for nurse rostering”, 2016... ..	11
Tabel 2.5 Constraint. Shahriar Asta, Ender Özcan , Tim Curtois, “A tensor based hyper-heuristic for nurse rostering”, 2016... ..	12
Tabel 2.6 Fungsi Tujuan. Shahriar Asta, Ender Özcan , Tim Curtois, “A tensor based hyper-heuristic for nurse rostering”, 2016 .....	12
Tabel 4.1 Kodifikasi tipe skill unit farmasi .....	25
Tabel 4.2 Jumlah staf pada setiap shift unit farmasi .....	26
Tabel 4.3 Jumlah shift tiap staf unit farmasi.....	27
Tabel 4.4 Kodifikasi tipe skill unit bayi/NICU.....	27
Tabel 4.5 Jumlah staf pada setiap shift unit bayi/NICU .....	28
Tabel 4.6 Jumlah shift tiap staf unit bayi/NICU .....	29
Tabel 4.7 Kodifikasi tipe skill unit gizi .....	31
Tabel 4.8 Jumlah staf pada setiap shift unit gizi .....	32
Table 4.9 Jumlah shift tiap staf unit gizi.....	33
Tabel 4.10 Kodifikasi tipe skill unit IGD .....	34
Tabel 4.11 Jumlah staf pada setiap shift unit IGD .....	35
Tabel 4.12 Jumlah shift tiap staf unit IGD .....	36
Tabel 4.13 Kodifikasi tipe skill unit Kamar Operasi .....	37
Tabel 4.14 Jumlah staf pada setiap shift unit Kamar Operasi .....	37
Tabel 4.15 Jumlah shift tiap staf unit Kamar Operasi.....	38
Tabel 4.16 Kodifikasi tipe skill unit SIM & RM .....	39
Tabel 4.17 Jumlah staf pada setiap shift unit SIM & RM.....	40
Tabel 4.18 Jumlah shift tiap staf unit SIM & RM .....	41
Tabel 4.19 Kodifikasi Tipe Shift .....	41
Table 4.20 Daftar variabel yang digunakan. A. Muklason, "Hyper-Heuristics And Fairness In Examination Timetabling Problems," 2017 .....	53
Tabel 5.1 Perangkat Keras dan Lunak yang digunakan.....	55
Tabel 6.1 Jadwal hasil optimasi unit farmasi.....	92

Tabel 6.2 Jadwal hasil optimasi unit bayi/NICU .....	93
Tabel 6.3 Jadwal hasil optimasi unit gizi.....	94
Tabel 6.4 Jadwal hasil optimasi unit IGD.....	95
Tabel 6.5 Jadwal hasil optimasi unit Kamar Operasi.....	96
Tabel 6.6 Jadwal hasil optimasi unit SIM & RM .....	97
Tabel 6.7 Jumlah shift tiap staf hasil optimasi unit farmasi ...	98
Tabel 6.8 Jumlah hari libur tiap staf hasil optimasi unit farmasi .....	98
Tabel 6.9 Jumlah shift tiap staf hasil optimasi unit bayi/NICU .....	99
Tabel 6.10 Jumlah hari libur tiap staf hasil optimasi unit bayi/NICU .....	100
Tabel 6.11 Jumlah shift tiap staf hasil optimasi unit gizi .....	101
Tabel 6.12 Jumlah hari libur tiap staf hasil optimasi unit gizi .....	101
Tabel 6.13 Jumlah shift tiap staf hasil optimasi unit IGD ....	103
Tabel 6.14 Jumlah hari libur tiap staf hasil optimasi unit IGD .....	103
Tabel 6.15 Jumlah shift tiap staf hasil optimasi unit Kamar Operasi .....	104
Tabel 6.16 Jumlah hari libur tiap staf hasil optimasi unit Kamar Operasi .....	104
Tabel 6.17 Jumlah shift tiap staf hasil optimasi unit SIM & RM .....	105
Tabel 6.18 Jumlah hari libur tiap staf hasil optimasi unit SIM & RM .....	106
Tabel 6.19 Statistik Algoritma Reinforcement Learning .....	123
Tabel 6.20 Statistik Algoritma Hill Climbing.....	123
Tabel 6.21 Statistik Algoritma Reinforcement Learning Unit Bayi/Nicu .....	123
Tabel 6.22 Statistik Algoritma Hill Climbing Unit Bayi/Nicu .....	124
Tabel 6.23 Statistik Algoritma Reinforcement Learning Unit Gizi .....	124
Tabel 6.24 Statistik Algoritma Hill Climbing Unit Gizi .....	124
Tabel 6.25 Statistik Algoritma Reinforcement Learning Unit IGD .....	125
Tabel 6.26 Statistik Algoritma Hill Climbing Unit IGD.....	125

Tabel 6.27 Statistik Algoritma Reinforcement Learning Unit Kamar Operasi .....	125
Tabel 6.28 Statistik Algoritma Hill Climbing Unit Kamar Operasi .....	125
Tabel 6.29 Statistik Algoritma Reinforcement Learning Unit SIM & RM .....	126
Tabel 6.30 Statistik Algoritma Hill Climbing Unit SIM & RM .....	126
Tabel 6.31 Perbandingan JFI unit farmasi .....	131
Tabel 6.32 Perbandingan JFI unit Bayi/NICU.....	131
Tabel 6.33 Perbandingan JFI unit gizi .....	132
Tabel 6.34 Perbandingan JFI unit IGD .....	132
Tabel 6.35 Perbandingan JFI unit Kamar Operasi .....	132
Tabel 6.36 Perbandingan JFI unit SIM & RM.....	133

## DAFTAR KODE PROGRAM

Kode Program 5.1 Deklarasi Variabel Pada Parent Class .....	59
Kode Program 5.2 Penjadwalan otomatis unit farmasi ..	61
Kode Program 5.3 Cek constraints unit farmasi.....	62
Kode Program 5.4 Penjadwalan Otomatis Unit Bayi/NICU .....	63
Kode Program 5.5 Cek constraint unit Bayi/NICU .....	64
Kode Program 5.6 Penjadwalan Otomatis Unit Gizi .....	66
Kode Program 5.7 Cek constraint unit Gizi .....	67
Kode Program 5.8 Penjadwalan Otomatis Unit IGD.....	68
Kode Program 5.9 Cek constraint unit IGD .....	69
Kode Program 5.10 Penjadwalan Otomatis Unit Kamar Operasi.....	70
Kode Program 5.11 Cek constraint unit Kamar Operasi	71
Kode Program 5.12 Penjadwalan Otomatis Unit SIM & RM.....	72
Kode Program 5.13 Cek constraint unit SIM &RM .....	73
Kode Program 5.14 Perhitungan Jain Fairness Index.....	74
Kode Program 5.15 Low Level Heuristic Move .....	76
Kode Program 5.16 Low Level Heuristic Swap.....	77
Kode Program 5.17 Algoritma Reinforcement Learning	79
Kode Program 5.18 Algoritma Hill Climbing .....	80

*(Halaman sengaja dikosongkan)*



# BAB I

## PENDAHULUAN

Pada bab pendahuluan ini akan membahas terkait latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan relevansi terhadap pengerjaan tugas akhir.

### 1.1 Latar belakang

RSIA Kendangsari merupakan salah satu Rumah Sakit Ibu dan Anak yang ada di Surabaya. Dilihat dari namanya, dapat diketahui bahwa RSIA Kendangsari mengutamakan pelayanan kebutuhan untuk ibu dan anak. RSIA Kendangsari Surabaya merupakan bentuk dari satu kesatuan visi dan misi yang sama oleh beberapa dokter spesialis kebidanan dan kandungan dalam upaya meningkatkan “kualitas kesehatan reproduksi kaum perempuan khususnya ibu hamil dan janin yang dikandungnya” [1].

Penjadwalan staf merupakan salah satu masalah yang biasa terjadi pada Rumah Sakit. Dalam proses penjadwalan daftar jaga staf ada beberapa komponen yang digunakan yaitu staf, ruang, hari, dan shift (Tjajo, 2008). Permasalahan staf atau dikenal sebagai *Nurse Scheduling Problem (NSP)* menjadi menarik untuk diselesaikan karena dengan sumber daya rumah sakit yang terbatas pihak rumah sakit dituntut untuk mendapatkan jadwal yang bisa memperlakukan semua staf secara adil dengan memenuhi batasan-batasan yang ada. Ada beberapa batasan yang perlu diperhatikan atau bisa juga disebut *constrain*. Pada penelitian ini, *constraint* tersebut dibagi menjadi dua yaitu *hard constraints* dan *soft constraints*. *Hard constraints* adalah batasan-batasan yang harus dipenuhi seperti regulasi rumah sakit, sedangkan *soft constraints* adalah batasan yang tidak harus dipenuhi, tetapi

jika dipenuhi lebih baik seperti ukuran optimality pada rumah sakit [2]. Pemenuhan semua batasan penjadwalan sering kali terhambat ketika satu batasan terpenuhi, namun ternyata batasan lain terlanggar [3]. Batasan-batasan tersebut juga nantinya akan berbeda-beda karena tiap unit di rumah sakit memiliki regulasinya sendiri. Berdasarkan permasalahan tersebut, maka dalam tugas akhir ini penjadwalan staf pada RSIA Kendangsari akan dioptimalkan dengan menggunakan algoritma *Reinforcement Learning Hyper-Heuristic*.

Saat ini penjadwalan staf di RSIA Kendangsari Surabaya dilakukan dengan cara manual sehingga dapat menghabiskan waktu yang lama dan tenaga yang tidak sedikit. Penjadwalan staf juga dilakukan tanpa memikirkan faktor pribadi para staf sehingga staf harus dengan suka rela menerima jadwal yang sudah ada. Hal ini tentu dapat mengakibatkan performa kinerja staf menurun, oleh karena itu alasan pribadi staf juga harus di pertimbangkan untuk tetap menjaga performa kinerja staf. Dalam penelitian ini, masalah keadilan antar staf akan dihitung dengan menggunakan Jain Fairness Index (JFI). JFI akan menghitung tingkat keadilan berdasarkan hari libur yang didapatkan oleh setiap staf. Jika staf mendapatkan libur di hari Minggu maka akan diberi bobot sebesar 4, jika di hari Sabtu akan diberi bobot sebesar 2, dan sebesar 1 jika mendapatkan libur di hari kerja.

Algoritma *reinforcement learning* digunakan menghasilkan penjadwalan dengan solusi yang paling optimal dengan memilih *low-level heuristic* terlebih dahulu. *Reinforcement learning* sendiri merupakan pengembangan dari algoritma *hill climbing*, dimana algoritma *hill climbing* merupakan algoritma sederhana yang mampu menghasilkan solusi yang baik. *Hill climbing* akan menghasilkan sebuah solusi yang akan dibandingkan dengan solusi terbaik yang sudah didapatkan.

Perbedaannya adalah pada algoritma *reinforcement learning* diterapkan parameter *score* untuk masing-masing *low-level heuristic* agar dapat dilihat *low-level heuristic* mana yang memiliki solusi terbaik.

## 1.2 Rumusan masalah

Berdasarkan latar belakang yang telah diuraikan, didapatkan perumusan permasalahan yaitu:

1. Bagaimana model matematis penjadwalan staf pada RSIA Kendangsari?
2. Bagaimana hasil jadwal staf yang optimal dengan menerapkan algoritma *Reinforcement Learning Hyper-Heuristic* di RSIA Kendangsari?
3. Bagaimana perbandingan hasil penjadwalan staf secara manual dengan penjadwalan otomatis menggunakan algoritma *Reinforcement Learning Hyper-Heuristic*?

## 1.3 Batasan masalah

Batasan masalah dari penelitian ini adalah :

1. Studi kasus yang digunakan pada tugas akhir ini adalah permasalahan optimasi penjadwalan staf di RSIA Kendangsari Surabaya
2. Data yang digunakan sebagai *hard constraint* adalah regulasi staf pada RSIA Kendangsari
3. Data yang digunakan sebagai *soft constraint* adalah ukuran *optimality* pada RSIA Kendangsari

## 1.4 Tujuan

1. Membuat model matematis Nurse Rostering *Problem* dari permasalahan yang sudah didefinisikan dan batasan-batasan yang ada
2. Membuat jadwal staf otomatis dengan menggunakan algoritma *Reinforcement Learning Hyper-Heuristic*

3. Membandingkan hasil penjadwalan staf secara otomatis dengan jadwal yang sudah ada di RSIA Kendangsari

### **1.5 Manfaat**

Hasil dari tugas akhir ini diharapkan dapat membantu RSIA Kendangsari dalam menyusun penjadwalan staf untuk meningkatkan performa staf, meningkatkan tingkat *fairness* tiap staf, dan meningkatkan citra positif rumah sakit.

### **1.6 Relevansi**

Lab Rekayasa Data dan Intelegensi Bisnis (RDIB) merupakan salah satu lab yang ada di departemen Sistem Informasi ITS. Lab ini menaungi 4 mata kuliah pilihan bagi mahasiswa semester akhir. Salah satunya adalah Riset Operasi Tingkat Lanjut yang mempelajari mengenai berbagai metode untuk pengambilan keputusan secara optimal. Untuk itu pada tugas akhir ini saya mengambil topik penelitian terhadap optimasi penjadwalan staf di RSIA Kendangsari.

Tugas akhir ini layak dijadikan sebagai tugas akhir pada tingkat S1, karena tugas ini dapat membantu rumah sakit dalam mengoptimalkan penjadwalan staf yang selama ini masih dilakukan secara manual. Dengan menemukan titik optimal untuk penjadwalan diharapkan rumah sakit dapat membuat jadwal yang adil bagi seluruh staf dan memenuhi semua regulasi yang ada. Dalam bidang keilmuan, penelitian ini mendukung pengembangan ilmu dalam optimasi terutama optimasi penjadwalan.

## BAB II TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan mengenai penelitian sebelumnya dan dasar teori yang akan dijadikan acuan atau landasan dalam pengerjaan tugas akhir ini.

### 2.1 Penelitian Terdahulu

Pada sub bab ini akan dijelaskan tentang penelitian terdahulu yang digunakan dalam pengerjaan tugas akhir ini :

**Tabel 2.1 Penjelasan Penelitian Terdahulu**

<b>Judul Paper</b>	<i>A TWO-STAGE MODELING WITH GENETIC ALGORITHMS FOR THE NURSE SCHEDULING PROBLEM</i>
<b>Penulis; Tahun</b>	Chang-Chun Tsai , Sherman H.A. Li; 2008
<b>Deskripsi Umum Penelitian</b>	Pada penelitian ini, penyelesaian <i>Nurse Scheduling Problem</i> (NSP) dilakukan dengan menggunakan dua tahap <i>genetic algorithm</i> dengan studi kasus rumah sakit di Taiwan. Pada tahap pertama, jadwal kerja dan libur staf sudah diatur dan akan diperiksa apabila masih terdapat pelanggaran terhadap regulasi rumah sakit. Pada tahap kedua, jadwal staf diatur dan <i>genetic algorithm</i> digunakan untuk mendapatkan jadwal yang paling optimal. Hasil dari penelitian ini berupa tabel jumlah staf di tiap shift yang ada dan jumlah staf yang libur pada hari tersebut.

<b>Keterkaitan Penelitian</b>	Penelitian penyelesaian NSP di Taiwan ini dapat menjadi referensi penelitian yang pernah dilakukan terkait optimasi penjadwalan dalam mengerjakan tugas akhir.
-------------------------------	--

<b>Judul Paper</b>	<i>A HARMONY SEARCH ALGORITHM FOR NURSE ROSTERING PROBLEMS</i>
<b>Penulis; Tahun</b>	Mohammed Hadwan, Masri Ayo, Nasser R. Sabar, Roug Qu; 2013
<b>Deskripsi Umum Penelitian</b>	Pada penelitian ini, dilakukan optimasi <i>Nurse Rostering Problem</i> (NRP) dengan menggunakan algoritma <i>harmony search</i> . Optimasi dilakukan pada dua set NRP yaitu yang pertama merepresentasikan dataset yang berasal dari rumah sakit di Malaysia dan yang kedua adalah dataset NRP yang sudah terkenal dan biasa digunakan untuk penelitian. Hasilnya algoritma <i>harmony search</i> mendapatkan hasil yang lebih baik dibandingkan dengan <i>genetic algorithm</i> pada dataset yang pertama. Pada dataset yang kedua juga mendapatkan hasil yang bagus saat dibandingkan dengan literatur sebelumnya yang menggunakan algoritma <i>meta-heuristic</i> .
<b>Keterkaitan Penelitian</b>	Literatur ini dapat digunakan sebagai acuan dalam melakukan pengerjaan tugas akhir karena memiliki studi kasus yang mirip sehingga banyak

	persamaan pada langkah-langkah penjadwalannya.
<b>Judul Paper</b>	OPTIMASI NURSE SCHEDULLING PROBLEM (STUDY KASUS RSUD DR. SOETOMO SURABAYA
<b>Penulis; Tahun</b>	Aditya Pratama Hidayatullan, Budi santosa; 2014
<b>Deskripsi Umum Penelitian</b>	Pada penelitian ini, penyelesaian Nurse Schedulling Problem dilakukan dengan menggunakan algoritma <i>Simulated Annealing</i> (SA). Lalu pada saat validasi model, dilakukan juga dengan menggunakan metode eksak yaitu branch and bound dengan menggunakan bantuan software LINGO. Hasilnya akan ada dua perbandingan yaitu perbandingan antara perhitungan menggunakan SA dengan kondisi eksisting dan yang kedua yaitu perbandingan antara perhitungan menggunakan SA dengan metode eksak. Jumlah deviasi pelanggaran dengan menggunakan metaheuristik sebesar 1140 dan dengan metode eksak sebesar 275.
<b>Keterkaitan Penelitian</b>	Literatur ini dapat digunakan sebagai acuan dalam melakukan pengerjaan tugas akhir dan akan menjadi referensi dalam metodologi pengerjaan tugas akhir

<b>Judul Paper</b>	OPTIMASI PENJADWALAN PERAWAT MENGGUNAKAN GABUNGAN INTEGER LINEAR PROGRAMMING DAN VARIABLE NEIGHBORHOOD SEARCH. STUDI KASUS INSTALASI GAWAT DARURAT RUMAH SAKIT IBNU SINA MAKASSAR
<b>Penulis; Tahun</b>	Muhammad Asrar Amir; 2017
<b>Deskripsi Umum Penelitian</b>	<p>Pada penelitian ini, dilakukan penyelesaian NSP dengan menggunakan metode <i>Integer Linear Programming</i> dan <i>Variable Neighborhood Search</i>. Dalam penyelesaian masalah tersebut, dilakukan dua kali proses pengerjaan. Pertama dengan menggunakan <i>Integer Linear Programming</i> dan yang kedua dengan kombinasi antara <i>Integer Linear Programming</i> dengan <i>Variable Neighborhood Search</i>. Output dari perhitungan dengan <i>Integer Linear Programming</i> menggunakan aplikasi LINGO nantinya akan menjadi input dalam perhitungan <i>Variable Neighborhood Search</i> menggunakan VBA Excel. Kemudian dari kedua metode tersebut akan diambil hasil yang paling optimal.</p>
<b>Keterkaitan Penelitian</b>	<p>Penelitian ini digunakan untuk melihat optimasi terkait penjadwalan perawat yang pernah dilakukan oleh peneliti Indonesia dan dapat</p>

	dijadikan referensi sebagai bahan acuan dalam metodologi pengerjaan tugas akhir.
--	--

## 2.2 Dasar teori

Bagian ini menjelaskan dasar teori yang digunakan di dalam penelitian tugas akhir.

### 2.2.1 RSIA Kendangsari

Menurut situs resmi RSIA Kendangsari, rumah sakit ini adalah Rumah Sakit Ibu dan Anak Surabaya yang dirancang unik berfokus untuk melayani kebutuhan ibu dan anak. Dalam menjalankan fungsinya RSIA Kendangsari Surabaya memberikan pelayanan kesehatan yang paripurna untuk wanita dan anak. Pelayanan kesehatan diberikan secara prima dan komprehensif bagi pasien, keluarga pasien dan provider baik perusahaan maupun asuransi [1].

### 2.2.2 Nurse Rostering Problem (NRP)

*Nurse Rostering Problem* (NRP) adalah masalah penjadwalan perawat yang sering terjadi pada rumah sakit. Alasan utama NRP menjadi sangat penting untuk diselesaikan oleh pihak rumah sakit adalah agar rumah sakit dapat mengalokasikan staf pada beberapa shift yang tersedia dalam hari tersebut selama 24 jam penuh [6]. Pembagian shift yang tidak adil tentu akan berdampak buruk dari sisi staf, rumah sakit, dan pasien. Maka dari itu perlu dibuatkan jadwal yang dapat menyelesaikan masalah NRP seperti itu.

Sebagai contoh ilustrasi terhadap masalah NRP ini, anggap ada sebuah jadwal staf untuk sebuah rumah sakit. Shift yang tersedia adalah shift pagi, siang, sore, dan malam. Lalu ada regulasi rumah sakit yang mengharuskan tiap shift diisi minimal oleh lima staf dan staf tidak boleh mengambil shift malam dan pagi secara berturut-turut. Untuk

mendapatkan jadwal yang baik, maka batasan-batasan seperti itu harus dipenuhi. Batasan-batasan atau *constraints* seperti diatas merupakan contoh dari *hard constraints* atau *constraints* yang harus dipenuhi [10].

Sedangkan untuk *soft constraints* adalah batasan yang tidak berpengaruh kepada validitas jadwal, tetapi tetap berpengaruh pada kualitas jadwal yang dihasilkan [2]. Contohnya adalah pandangan tersendiri dari staf yang menginginkan waktu libur yang tidak sesuai dengan jadwal. Hal-hal seperti itu termasuk kedalam *soft constraints* dan menjadi faktor terbesar yang mempengaruhi kualitas jadwal yang sudah dibuat.

#### 2.2.2.1 Pemodelan Matematis Nurse Rostering Problem

- Set

**Tabel 2.2 Set. Shahriar Asta, Ender Özcan , Tim Curtois, “A tensor based hyper-heuristic for nurse rostering”, 2016**

$E$ = Staf yang akan dijadwalkan, $e \in E$ $T$ = Tipe shift untuk setiap staf, $t \in T$ $D$ = Hari dalam jangka perencanaan, $d \in \{ 1, \dots   D \}$ $R_e$ = Regular expressions untuk staf $e$ , $r \in R_e$ $W_e$ = Batas waktu kerja untuk staf $e$ , $w \in W_e$
---

- Parameter

**Tabel 2.3 Parameter. Shahriar Asta, Ender Özcan , Tim Curtois, “A tensor based hyper-heuristic for nurse rostering”, 2016**

$r_{er}^{max}$ = Angka maksimal kecocokan antara <i>regular expression</i> $r$ di dalam jadwal kerja staf $e$ . $r_{er}^{min}$ = Angka minimal kecocokan antara <i>regular expression</i> $r$ di dalam jadwal kerja staf $e$ . $a_{er}$ = Bobot yang diasosiasikan dengan <i>regular expression</i> $r$ untuk staf $e$ .
--

$v_{ew}^{max}$	= Angka maksimal dari jam yang dialokasikan untuk staf $e$ selama dalam periode waktu yang didefinisikan oleh batasan waktu kerja.
$v_{ew}^{min}$	= Angka minimal dari jam yang dialokasikan untuk staf $e$ selama dalam periode waktu yang didefinisikan oleh Batasan waktu kerja.
$b_{ew}$	= Bobot yang diasosiasikan dengan Batasan waktu kerja $w$ untuk staf $e$ .
$s_{td}^{max}$	= Angka maksimum dari tipe shift $t$ yang dibutuhkan pada hari $d$ .
$s_{td}^{min}$	= Angka minimum dari tipe shift $t$ yang dibutuhkan pada hari $d$ .
$c_{td}$	= Bobot yang diasosiasikan dengan kebutuhan minimal tipe shift $t$ dalam hari $d$ .

- Variabel

**Tabel 2.4 Variabel.** Shahriar Asta, Ender Özcan , Tim Curtois, "A tensor based hyper-heuristic for nurse rostering", 2016

$x_{etd}$	= bernilai 1 jika staf $e$ diberikan tipe shift $t$ pada hari $d$ , dan bernilai 0 bila tidak ada pemberian kerja pada shift tersebut.
$n_{er}$	= Angka kecocokan dari <i>regular expression</i> $r$ di dalam jadwal kerja dari staf $e$ .
$p_{ew}$	= Jumlah jam kerja yang dialokasikan untuk staf $e$ selama periode waktu yang didefinisikan dengan batas kerja $w$ .
$q_{td}$	= Jumlah tipe shift $t$ yang dialokasikan untuk hari $d$ .

- Constraint

Contoh batasan disini adalah staf hanya bisa dialokasikan ke dalam satu *shift* per hari.

**Tabel 2.5 Constraint.** Shahriar Asta, Ender Özcan , Tim Curtois, “A tensor based hyper-heuristic for nurse rostering”, 2016

$$\sum_{t \in T} x_{etd} \leq 1, \forall e \in E, d \in D \quad (1)$$

▪ *Fungsi Tujuan*

**Tabel 2.6 Fungsi Tujuan.** Shahriar Asta, Ender Özcan , Tim Curtois, “A tensor based hyper-heuristic for nurse rostering”, 2016

$$\text{Min } f(s) = \sum_{e \in E} \sum_{i=1}^4 f_{e,i}(x) + \sum_{t \in T} \sum_{d \in D} \sum_{i=5}^6 f_{t,d,i}(x) \quad (2a)$$

where

$$f_{e,1}(x) = \sum_{e \in R_e} \max \{0, (n_{er} - r_{er}^{\max}) a_{er}\} \quad (2b)$$

$$f_{e,2}(x) = \sum_{e \in R_e} \max \{0, (r_{er}^{\min} - n_{er}) a_{er}\} \quad (2c)$$

$$f_{e,3}(x) = \sum_{e \in R_e} \max \{0, (p_{ew} - v_{ew}^{\max}) b_{ew}\} \quad (2d)$$

$$f_{e,4}(x) = \sum_{e \in R_e} \max \{0, (v_{ew}^{\min} - p_{ew}) b_{ew}\} \quad (2e)$$

$$f_{e,5}(x) = \max \{0, (s_{td}^{\min} - q_{td}) c_{td}\} \quad (2f)$$

$$f_{e,6}(x) = \max \{0, (q_{td} - s_{td}^{\max}) c_{td}\} \quad (2g)$$

### 2.2.3 Reinforcement Learning

*Reinforcement Learning* merupakan salah satu metode optimasi yang termasuk dalam framework Hyper Heuristic. Konsep dasar *Reinforcement Learning* diambil dari suatu teori dalam ilmu psikologi yang disebut dengan *Reinforcement Theory*. Teori ini menjelaskan bagaimana seseorang dapat menentukan, memilih, dan mengambil keputusan dalam dinamika kehidupan. *Reinforcement*

*Theory* ini mengatakan bahwa tingkah laku manusia itu adalah merupakan hasil kompilasi dari pengalaman-pengalaman sebelumnya [11].

Ide dari *Reinforcement Learning* berdasarkan mekanisme untuk *low-level heuristic selection* adalah dengan memberikan tiap *low-level heuristic* penghargaan dan hukuman [9]. Awalnya, tiap *low-level heuristic* akan diberikan *score* yang sama. Lalu jika heuristic terpilih dan memberikan hasil pada solusi yang diterima, maka *score* akan bertambah 1. Sebaliknya, jika heuristic terpilih tetapi memberikan hasil pada solusi yang ditolak, maka *score* akan berkurang 1 sampai batas bawah *score* tercapai. Jika ada lebih dari satu *low-level heuristics* dengan *score* tertinggi, salah satunya akan dipilih secara acak.

#### 2.2.4 Hill Climbing

*Hill Climbing Algorithm (HCA)* adalah proses pengujian yang dilakukan dengan menggunakan fungsi heuristik[15]. Dinamakan hill climbing karena pada metode ini iterasi akan terus dilakukan hingga mencapai titik teratas yang dinamakan best solution. Teknik *hill climbing* juga mampu menghasilkan data uji yang optimal dengan kualitas yang cukup baik dalam jangka waktu tertentu. *Hill climbing* dilakukan dengan cara mengeksekusi setiap *low-level heuristic* secara random. *Low-level heuristic* yang dipakai pada Tugas Akhir ini adalah move dan swap.

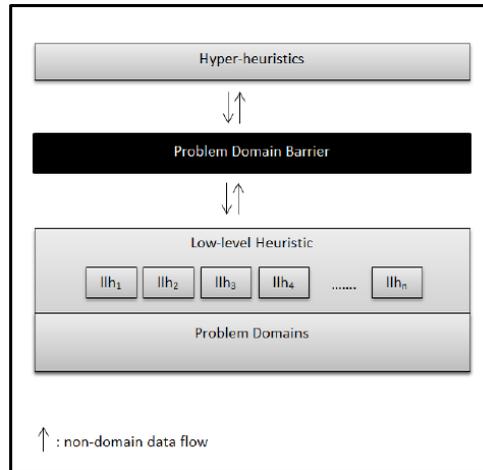
Algoritma Hill Climbing merupakan salah satu teknik optimasi matematis yang termasuk ke dalam kategori local search. [16]. Disebut local search karena hanya melakukan evaluasi terhadap kemungkinan-kemungkinan *state* yang saat ini sedang dihadapi. Ketika telah memilih salah satu *state* yang dianggap terbaik, maka *Hill Climbing* akan melanjutkan pencarian hanya berdasar *state* yang telah dipilih tersebut, hingga mencapai kondisi goalnya. Sehingga ketika telah dipilih satu jalur, maka jalur yang lain akan diabaikan. Itulah mengapa *Hill Climbing* sering

dianggap sebagai cara pencarian heuristic yang tercepat, karena hanya melakukan simple evaluation terhadap beberapa kemungkinan state yang dianggap terbaik, lalu memilihnya, dan melupakan kemungkinan lain yang berada di luar kondisi evaluatifnya.

HCA bekerja pada sebuah fungsi,  $f(x)$ , yang disebut sebagai fungsi tujuan. HCA akan mulai dengan mempertimbangkan fungsi tujuan dan bekerja dalam langkah-langkah berulang. Pada setiap langkah, algoritma akan mengubah satu elemen dalam vektor  $x$ . Kemudian akan menghitung dan menganalisa pengembangan pada fungsi tujuan  $f(x)$ . Jika perkembangan ini positif, perubahan dalam vektor akan dipertimbangkan. Sampai perubahan vektor tidak lagi membawa peningkatan apapun dalam  $f(x)$ .

#### 2.2.5 Hyper-Heuristic

Kondisi saat ini dari penelitian mengenai *hyper-heuristic* adalah kumpulan dari beberapa pendekatan yang membagikan tujuan utama yaitu mengotomasi desain dan adaptasi dari metode *heuristic* untuk menyelesaikan masalah pencarian komputasi yang sulit [11]. Atau dapat dikatakan bahwa *hyper-heuristic* adalah kombinasi dari *heuristic* dengan *machine learning*. *Hyper-heuristic* bertujuan untuk memilih dan mengkombinasikan *heuristic* yang lebih sederhana atau menghasilkan *heuristic* baru dengan komponen *heuristic* yang sudah ada.



**Gambar 2.1 Framework Hyper-Heuristic. A. Muklason, "Hyper-Heuristics And Fairness In Examination Timetabling Problems," 2017**

Dari gambar diatas, terlihat bahwa *Hyper-heuristic* merupakan sebuah framework yang digunakan untuk memilih *low-level heuristic* terbaik yang nantinya akan diangkat sehingga bersifat lebih *generic* dibandingkan pendekatan *metaheuristic*. *Generic* yang dimaksud disini adalah pada *hyper-heuristic* hanya diperlukan parameter tuning yang lebih simple/sedikit dibandingkan *metaheuristic* untuk menyelesaikan domain problem yang ada karena *metaheuristic* berhubungan langsung dengan domain problem tersebut.

### 2.2.6 Jain Fairness Index

*Jain Fairness Index* (JFI) merupakan sebuah metode yang digunakan untuk menangani masalah keadilan dengan menggunakan pendekatan kuantitatif. Keadilan pada *Jain Fairness Index* pada kasus ini adalah kesetaraan alokasi sumber daya pada penjadwalan rumah sakit. Sehingga harapannya dengan menggunakan metode *Jain Fairness*

*Index*, semua tenaga kerja dapat merasa diperlakukan secara adil. Berikut merupakan rumus dari JFI, dengan asumsi T merupakan solusi dari penjadwalan, dan  $X_i$  merupakan nilai libur yang diasumsikan untuk masing-masing orang [14]:

$$JFI(T) = \frac{(\sum_{s=1}^S X_i)^2}{(S * \sum_{s=1}^S (X_i)^2)} \quad (2.1)$$

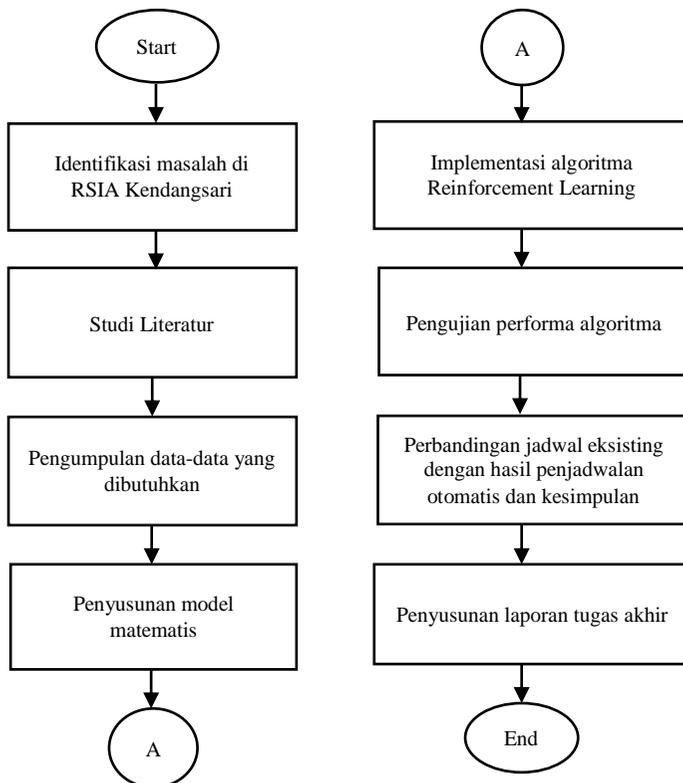
Perhatikan bahwa nilai JFI dibatasi antara 0 dan 1, dengan nilai yang lebih besar berarti solusinya akan menjadi lebih adil. Pada Tugas akhir ini, ditentukan bobot libur di hari Minggu bernilai 4, Sabtu bernilai 2, dan 1 untuk libur di hari kerja

## BAB III METODOLOGI PENELITIAN

Dalam bab ini, akan dijelaskan metodologi yang akan digunakan sebagai panduan untuk menyelesaikan tugas akhir ini.

### 3.1 Diagram Metodologi

Berikut merupakan alur dari diagram metodologi untuk pengerjaan tugas akhir.



**Diagram Metodologi 3.1 Diagram Metodologi**

## **3.2 Uraian Metodologi**

Dibawah ini adalah penjelasan dari setiap proses alur metodologi berdasarkan diagram alur metodologi pada sub bab sebelumnya.

### **3.2.1 Identifikasi Masalah di RSIA Kendangsari**

Tahap ini merupakan pertama yang dilakukan dalam pengerjaan tugas akhir ini. Pada tahapan ini, dilakukan identifikasi terhadap permasalahan pada rumah sakit yang akan diangkat menjadi masalah pada tugas akhir ini. Identifikasi masalah dilakukan dengan mengetahui proses bisnis RSIA Kendangsari, terutama permasalahan pada Nurse Rostering Problem

### **3.2.2 Studi Literatur**

Studi literatur dilakukan dengan mengumpulkan beberapa referensi yang berasal dari buku pustaka dan beberapa penelitian sebelumnya yang memiliki keterkaitan dengan permasalahan yang ingin diselesaikan. Pustaka yang digunakan yaitu dengan mencari paper, jurnal, laporan penelitian ataupun tugas akhir terkait permasalahan yang ada. Dari studi literatur yang dilakukan maka didapatkan faktor-faktor yang mempengaruhi permasalahan dan juga batasan dari permasalahan.

### **3.2.3 Pengumpulan data-data yang dibutuhkan**

Pada tahapan ini, dilakukan setelah didapatkan permasalahan dan diketahui mengenai konsep dan model yang dapat dilakukan dalam menyelesaikan permasalahan.. Selanjutnya dilakukan pengumpulan data yang berkaitan dengan permasalahan yang dijelaskan untuk menjawab tujuan dari tugas akhir.

### **3.2.4 Penyusunan Model Matematis**

Data-data yang sudah terkumpul berupa regulasi staf, peraturan staf, dan nurse preference dikonversikan menjadi sebuah model matematis yang nantinya bisa diselesaikan menggunakan metode yang dipilih. Masalah yang telah didefinisikan juga

dibentuk dalam model matematis seperti contoh yang ada pada bab 2.2.2.1. Adapun acuan bentuk model yang nantinya akan dibuat sebagai berikut :

#### 3.2.4.1 Constraints

*Constraint* atau batasan yang digunakan sebagai acuan dalam membuat model matematis nya terbagi menjadi dua yaitu *Hard Constraint* dan *Soft Constraint*. Berikut merupakan daftar *constraint* tersebut:

##### *Hard Constraints:*

- Unit Farmasi
  1. Jumlah staff di shift pagi tiap harinya harus sama dengan 3
  2. Jumlah staff di shift siang tiap harinya harus sama dengan 2
  3. Jumlah staff di shift malam tiap harinya harus sama dengan 1
- Unit Gizi
  1. Jumlah staff di shift pagi sore (PS) harus sama dengan 2 di hari Minggu
  2. Jumlah staff di shift pagi sore (PS) harus sama dengan 2 di hari Senin
  3. Jumlah staff di shift malam selalu sama dengan 1 tiap harinya
- Unit NICU/Bayi
  1. Jumlah staff di shift pagi setiap hari harus lebih dari sama dengan dua dan kurang dari sama dengan empat
  2. Jumlah staff di shift siang setiap hari harus lebih dari sama dengan satu dan kurang dari sama dengan 2
  3. Jumlah staff di shift malam setiap hari harus lebih dari sama dengan dua dan kurang dari sama dengan tiga
- Unit SIM & RM
  1. Jumlah staff di shift pagi tiap harinya lebih dari sama dengan 1 atau kurang dari sama dengan 2
  2. Jumlah staff di shift siang tiap harinya lebih dari sama dengan 1 atau kurang dari sama dengan 2

3. Jumlah staff di shift malam tiap harinya harus sama dengan 1

- Unit IGD

1. Jumlah staff di shift pagi tiap harinya lebih dari sama dengan satu atau kurang dari sama dengan dua

2. Jumlah staff di shift siang setiap hari harus lebih dari samadengan dua dan kurang dari sama dengan empat

3. Jumlah staff di shift malam setiap hari harus berjumlah satu atau dua

- Unit Kamar Operasi

1. Jumlah staff di shift pagi setiap hari harus lebih dari samadengan dua dan kurang dari sama dengan empat

2. Jumlah staff di shift siang setiap hari harus lebih dari samadengan dua atau kurang dari samadengan tiga

3. Jumlah staff di shift pagi pada hari Minggu harus terdiri dari empat

*Soft Constraint:*

- Ukuran optimality dari Jain Fairness Index untuk setiap unit pada RSIA Kendangsari

### 3.2.4.2 Variabel Keputusan

- *Set of staff* yang bekerja pada setiap shift per hari nya

### 3.2.4.3 Fungsi Tujuan

- Tujuan dari pemodelan permasalahan ini adalah untuk memaksimalkan hasil perhitungan *Jain's Fairness Index*

### 3.2.5 Implementasi Algoritma Reinforcement Learning

Setelah pembuatan model selesai, maka berikutnya adalah implementasi menggunakan algoritma *Reinforcement Learning Hyper-Heuristic*. Pada tahap ini, algoritma *Reinforcement Learning Hyper-Heuristic* akan diimplementasikan menyesuaikan dengan model matematis yang telah dibuat sebelumnya dan tujuan dari permasalahan yang ingin diselesaikan.

### 3.2.6 Pengujian Performa Algoritma

Pengujian dilakukan dengan menggunakan perhitungan *Jain Fairness Index* (JFI) yang dijalankan menggunakan java pada Netbeans IDE. JFI digunakan untuk menghitung tingkat *fairness* pada jadwal yang dihasilkan dari implementasi algoritma *reinforcement learning*.

### 3.2.7 Perbandingan Jadwal Eksisting Dengan Hasil Penjadwalan Otomatis dan Kesimpulan

Pada tahap ini dilakukan penarikan kesimpulan yang dilakukan dengan membandingkan hasil analisis dari perhitungan yang telah dilakukan. Analisis yang dilakukan yaitu dengan melakukan perbandingan antara kondisi eksisting rumah sakit dengan hasil optimasi yang telah dilakukan. Dari analisa yang dilakukan akan terlihat kekurangan ataupun kelebihan dari metode yang telah digunakan rumah sakit, sehingga model dapat menjadi masukan dan pengembangan bagi rumah sakit.

### 3.2.8 Penyusunan Tugas Akhir

Penyusunan laporan merupakan tahap akhir dari proses-proses yang telah dilakukan sebelumnya. Disini dilakukan dokumentasi terhadap proses-proses yang telah dilakukan dan kesimpulan dari permasalahan yang didapatkan. Seluruh pelaksanaan ataupun pengerjaan tugas akhir ini akan didokumentasikan dengan mengikuti format yang telah ditetapkan oleh laboratorium Rekayasa Data dan Intelegensia Bisnis (RDIB) serta yang berlaku di Jurusan Sistem Informasi ITS.

Di dalam laporan Tugas Akhir akan mencakup :

- a. BAB I Pendahuluan  
Dalam bab ini dijelaskan mengenai latar belakang, rumusan dan batasan permasalahan, tujuan, manfaat, serta relevansi dalam pengerjaan tugas akhir ini.
- b. BAB II Tinjauan Pustaka  
Pada bab ini akan dijelaskan mengenai penelitian-penelitian sebelumnya yang telah dilakukan serta teori-teori yang menunjang permasalahan yang dibahas pada tugas akhir ini

c. BAB III Metodologi

Pada bab ini akan dijelaskan alur proses dari pengerjaan tugas akhir mulai dari identifikasi permasalahan sampai pembuatan laporan tugas akhir.

d. BAB IV Perancangan

Pada bab ini akan dijelaskan proses pengumpulan dan juga deskripsi data-data yang digunakan dalam penyelesaian tugas akhir.

e. BAB V Implementasi

Bab ini berisi tentang implementasi dan penjelasan setiap alur proses yang telah dijabarkan sebelumnya pada metodologi yang digunakan dalam tugas akhir.

f. BAB VI Hasil dan Pembahasan

Bab ini berisi analisis dan pembahasan dalam penyelesaian permasalahan yang dibahas pada pengerjaan tugas akhir.

g. BAB VII Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran yang ditujukan untuk kelengkapan penyempurnaan tugas akhir ini.

## **BAB IV**

### **DATA MASUKAN DAN PEMODELAN**

Pada bab ini akan dijelaskan mengenai batasan-batasan dan fungsi tujuan yang didapatkan melalui pengumpulan data pada Rumah Sakit Ibu dan Anak Kendangsari.

#### **4.1 Hasil Pengumpulan Data**

Pada Rumah Sakit Ibu dan Anak Kendangsari, penjadwalan staf ditentukan oleh setiap kepala unit yang bertanggung jawab pada bagian tersebut. Masing-masing unit memiliki regulasi sendiri mengenai penjadwalan stafnya. Regulasi tersebut didapatkan berdasarkan Peraturan Menteri Kesehatan Republik Indonesia, kebutuhan tiap unit, dan preferensi dari tiap staf yang ada di unit tersebut. Berdasarkan interview yang kami lakukan kepada Ibu Silvi (Kepala Unit Farmasi) ada 6 (enam) unit yang akan dioptimalkan penjadwalannya. Keenam unit tersebut adalah Gizi, SIM (Pendaftaran) & RM (Rekam Medis), NICU (*Neonatal Intensive Care Unit*)/Bayi, Poli/IGD (Instalasi Gawat Darurat), Farmasi, dan OK (Kamar Operasi).

Regulasi umum rumah sakit adalah bahwa setiap tenaga kerja harus memenuhi minimal 42 jam kerja untuk tiap minggu dan jika diturunkan menjadi 7 jam per hari. Setiap minggu, tiap tenaga kerja berhak mendapatkan 1 kali libur dan jika diakumulasikan menjadi 4 kali libur per bulannya. Secara umum, ada 5 (lima) shift yang ada pada Rumah Sakit Ibu dan Anak Kendangsari, yaitu P (Pagi), S (Sore), M (Malam), LM (Lepas Malam), dan MS (Middle). Shift pagi dimulai dari pukul 07:00 hingga pukul 14:00, shift sore dari pukul 14:00 hingga pukul 21:00, shift malam dari pukul 21:00 hingga pukul 07:00, middle shift dari pukul 10:00 hingga pukul 17:00. Penentuan pola middle dapat terjadi pada tanggal tanggal yang diprediksi akan ada banyak pasien seperti tanggal cantik, jadwal praktik dokter, dan sisa personel yang belum mendapatkan shift yang dijadwalkan. Kelima shift tersebut berlaku untuk semua tenaga kerja kecuali Kepala Unit yang hanya mendapatkan shift pagi karena kepala unit staff yang berurusan langsung dengan pihak

manajemen. Sehingga kehadirannya dibutuhkan setaip shift pagi yang bersamaan jam kantor bagian manajemen. Khusus untuk unit gizi, tidak semua tenaga kerja mengikuti kelima shift tersebut. Untuk lebih jelasnya akan dibahas pada bab selanjutnya.

Karena penjadwalan yang dilakukan masih secara manual, untuk membuat jadwal biasanya kepala unit pertama-tama memasukkan terlebih dahulu hari libur dan cuti yang diinginkan oleh staf. Setelah itu barulah akan disusun jadwal untuk keseluruhannya. Untuk membuat jadwal, ada pola pokok yang diterapkan oleh Rumah Sakit yaitu pola M, M, LM, L. Hal ini berarti setelah satu staf mendapatkan shift malam dua hari berturut-turut, maka untuk keesokan harinya staf tersebut berhak untuk mengambil shift lepas malam atau bisa disebut juga shift kosong karena shift malam terjadwal mulai dari jam 21.00 hingga 07.00 keesokan harinya. Untuk itu demi menjaga kesehatan dan fokus bekerja staf, rumah sakit membuat regulasi tersebut. Pola pokok tersebut juga bersifat akumulatif atau disesuaikan dengan jadwal pada bulan sebelumnya. Misalnya jika ada staf yang baru mendapatkan 3 kali libur pada bulan sebelumnya, maka untuk bulan selanjutnya akan dijadwalkan staf tersebut berhak mendapatkan 5 kali libur.

## **4.2 Kebijakan dan Regulasi Tiap Unit**

Kebijakan dalam penjadwalan rumah sakit ditetapkan oleh setiap kepala unit melalui berbagai pertimbangan. Nantinya beberapa pertimbangan tersebut akan dimasukkan ke dalam jadwal. Lalu berikut merupakan kebijakan dan regulasi yang didapatkan untuk setiap unitnya.

### **4.2.1 Unit Farmasi**

- Shift terdiri dari 5 bagian yaitu shift P (Pagi) dari pukul 07.00 sampai 14.00, lalu shift S (sore) dari pukul 14.00 sampai 21.00, shift M (malam) dari pukul 21.00 hingga pukul 07.00, shift MS (middle) dari pukul 10.00 hingga pukul 17.00, dan shift LM (lepas malam)

- Unit terdiri dari 8 tenaga kerja dengan rincian 1 Apoteker, 1 staff senior, 1 petugas obat, dan 5 anggota
- Setiap staf hanya boleh mengisi 1 shift per hari nya
- Setelah selesai 2 kali shift malam, maka harus melanjutkan pola pokok dan mendapatkan lepas malam dan libur untuk 2 hari kedepan
- Selalu ada staf minimal 1 di setiap shift (kecuali shift middle) setiap harinya
- Staff senior hanya boleh mendapatkan shift sore dan middle shift
- Apoteker dan petugas gudang hanya bisa mendapatkan shift pagi dan jadwal libur di hari minggu

**Tabel 4.1 Kodifikasi tipe skill unit farmasi**

<b>No</b>	<b>Kode Staff</b>	<b>Skill</b>	<b>Tipe Skill</b>
1	101	Apoteker	1
2	102	Staff Senior	2
3	103	Anggota	4
4	104	Anggota	4
5	105	Anggota	4
6	106	Anggota	4
7	107	Anggota	4
8	108	Petugas Gudang Obat	3

Tabel 4.2 Jumlah staf pada setiap shift unit farmasi

Kode Shift	Tanggal									
	1	2	3	4	5	6	7	8	9	10
M	1	1	1	1	1	1	1	1	1	1
P	3	3	3	3	1	3	3	3	3	3
S	2	2	2	2	1	2	2	2	2	2
MS	0	1	0	1	1	0	1	1	1	0
CT	0	0	0	0	0	0	0	0	0	1
L	1	1	1	1	3	2	0	1	0	1
LM	1	0	1	0	1	0	1	0	1	0
Kode Shift	Tanggal									
	11	12	13	14	15	16	17	18	19	20
M	1	1	1	1	1	1	1	1	1	1
P	3	1	3	3	3	3	3	3	1	3
S	2	1	2	2	2	2	2	2	1	2
MS	0	1	0	0	1	1	1	0	1	1
CT	1	0	1	1	0	0	0	0	1	0
L	0	4	0	1	0	1	0	2	2	1
LM	1	0	1	0	1	0	1	0	1	0
Kode Shift	Tanggal									
	21	22	23	24	25	26	27	28	29	30
M	1	1	1	1	1	1	1	1	1	1
P	3	3	3	3	3	2	3	3	3	3
S	2	2	2	2	2	2	2	2	2	2
MS	1	0	1	1	0	0	0	1	0	1
CT	0	1	0	0	0	0	0	0	0	0
L	0	1	0	1	1	3	1	1	1	1
LM	1	0	1	0	1	0	1	0	1	0

Tabel 4.3 Jumlah shift tiap staf unit farmasi

Kode Staff	Kode Shift						
	P	S	MS	L	CT	M	LM
101	26	0	0	4	0	0	0
102	1	15	10	4	0	0	0
103	7	5	3	4	2	6	3
104	8	7	1	4	1	5	3
105	7	7	1	4	1	6	4
106	3	12	0	4	2	6	3
107	5	11	2	4	0	6	2
108	26	0	0	4	0	0	0

#### 4.2.2 Unit Bayi/NICU

- Shift terdiri dari 5 bagian yaitu shift P (pagi) dari pukul 07.00 sampai 14.00, lalu shift S (sore) dari pukul 14.00 sampai 21.00, shift M (malam) dari pukul 21.00 hingga pukul 07.00, shift MS (middle) dari pukul 10.00 hingga pukul 17.00, dan shift LM (lepas malam)
- Unit terdiri dari 13 tenaga kerja dengan rincian 1 kepala unit dan 12 anggota
- Setiap staf hanya boleh mengisi 1 shift per hari nya
- Setelah selesai 2 kali shift malam, maka harus melanjutkan pola pokok dan mendapatkan lepas malam dan libur untuk 2 hari kedepan
- Selalu ada staf minimal 1 di setiap shift setiap harinya
- Kepala Unit hanya bisa mendapatkan shift pagi dan jadwal libur dihari minggu

Tabel 4.4 Kodifikasi tipe skill unit bayi/NICU

No	Kode Staff	Skill	Tipe Skill
1	201	Kepala Unit	1



<b>L</b>	2	1	1	3	2	2	2	1	1	1
<b>LM</b>	2	1	2	1	1	0	1	1	1	1
<b>Mid</b>	2	1	2	1	2	1	2	1	2	1
<b>CT</b>	0	1	0	0	0	0	0	1	1	0
<b>Kode Shift</b>	<b>Tanggal</b>									
	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>
<b>P</b>	2	2	2	2	2	2	2	2	2	2
<b>S</b>	2	2	2	2	2	2	2	2	2	2
<b>M</b>	3	3	3	3	3	3	3	3	3	3
<b>L</b>	2	3	1	3	1	3	1	2	1	2
<b>LM</b>	2	1	1	0	2	0	2	0	2	1
<b>Mid</b>	2	1	2	1	2	1	2	1	2	1
<b>CT</b>	0	0	1	0	0	0	0	0	1	1
<b>Kode Shift</b>	<b>Tanggal</b>									
	<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>	<b>25</b>	<b>26</b>	<b>27</b>	<b>28</b>	<b>29</b>	<b>30</b>
<b>P</b>	2	2	2	2	2	2	2	2	2	2
<b>S</b>	2	2	2	2	2	2	2	2	2	2
<b>M</b>	3	3	3	3	3	3	3	3	3	3
<b>L</b>	1	2	2	2	1	2	2	2	0	1
<b>LM</b>	2	1	2	1	1	1	2	1	2	1
<b>Mid</b>	2	1	2	2	2	1	3	1	3	1
<b>CT</b>	0	0	0	0	0	0	0	1	2	1

Tabel 4.6 Jumlah shift tiap staf unit bayi/NICU

<b>Kode Staff</b>	<b>Kode Shift</b>						
	<b>P</b>	<b>S</b>	<b>M</b>	<b>L</b>	<b>LM</b>	<b>MS</b>	<b>CT</b>
<b>201</b>	26	0	0	4	0	0	0
<b>202</b>	16	1	4	6	2	0	1
<b>203</b>	6	8	6	5	4	0	1

<b>Kode Staff</b>	<b>Kode Shift</b>						
	<b>P</b>	<b>S</b>	<b>M</b>	<b>L</b>	<b>LM</b>	<b>MS</b>	<b>CT</b>
<b>204</b>	7	3	8	6	4	1	1
<b>205</b>	4	8	8	6	3	0	1
<b>206</b>	3	8	6	6	4	1	2
<b>207</b>	2	8	6	6	4	2	2
<b>208</b>	3	4	10	5	5	2	1
<b>209</b>	2	5	8	5	4	5	1
<b>210</b>	7	3	6	6	3	5	0
<b>211</b>	1	7	10	4	5	3	0
<b>212</b>	5	4	8	5	4	4	0
<b>213</b>	4	1	10	6	6	3	0

#### 4.2.3 Unit Gizi

- Shift terdiri dari 7 bagian yaitu shift P1 (Pagi 1) dari pukul 04:30 sampai 11:30, shift P2 (Pagi 2) dari pukul 05:00 sampai 12:00, shift P (Pagi) dari pukul 07.00 sampai 14.00, shift PS (Pagi Sore) dari pukul 06:00 sampai 02:00, lalu shift S (sore) dari pukul 14.00 sampai 21.00, shift M (malam) dari pukul 21.00 hingga pukul 07.00 dan shift MS (middle) dari pukul 10.00 hingga pukul 17.00.
- Unit terdiri dari 19 tenaga kerja dengan rincian 1 Ahli gizi, 4 chef, 4 helper, 6 penyaji, 2 café, dan 2 driver
- Setiap staf hanya boleh mengisi 1 shift per hari nya
- Selalu ada staf minimal 1 di setiap shift (kecuali shift middle) setiap harinya
- Shift malam khusus untuk Penyaji
- Shift P1 khusus untuk tenaga kerja laki-laki
- Ahli gizi hanya bisa mendapatkan shift pagi dan jadwal libur dihari minggu

**Tabel 4.7 Kodifikasi tipe skill unit gizi**

<b>No</b>	<b>Kode Staff</b>	<b>Skill</b>	<b>Tipe Skill</b>
1	401	Ahli Gizi	1
2	402	Chef	5
3	403	Chef	5
4	404	Chef	5
5	405	Chef	5
6	406	Helper	6
7	407	Helper	6
8	408	Helper	6
9	409	Helper	6
10	410	Penyaji	4
11	411	Penyaji	4
12	412	Penyaji	4
13	413	Penyaji	4
14	414	Penyaji	4
15	415	Penyaji	4
16	416	Café	7
17	417	Café	7

No	Kode Staff	Skill	Tipe Skill
18	418	Driver	8
19	419	Driver	8

Tabel 4.8 Jumlah staf pada setiap shift unit gizi

Kode Shift	Tanggal									
	1	2	3	4	5	6	7	8	9	10
P	5	4	4	5	3	2	5	5	5	6
P1	1	1	1	1	1	1	1	1	1	1
P2	2	1	1	0	1	1	1	1	0	1
PS	0	0	0	0	2	2	0	0	0	0
S	5	6	6	5	3	3	7	6	7	5
M	1	1	1	1	1	1	1	2	2	1
LM	0	0	1	0	0	1	0	0	1	1
L	2	2	0	1	6	6	2	0	1	2
MS	2	2	3	3	2	2	1	2	0	1
CT	1	2	2	3	0	0	1	2	2	1
Kode Shift	Tanggal									
	11	12	13	14	15	16	17	18	19	20
P	5	1	2	4	5	5	5	4	3	3
P1	1	1	1	1	1	1	1	1	1	1
P2	1	1	1	1	1	1	1	1	1	1
PS	0	2	2	0	0	0	0	1	1	2
S	6	3	3	6	5	6	6	5	5	4
M	1	1	1	1	1	1	1	1	1	1
LM	1	0	1	0	1	0	1	0	1	0
L	2	8	6	2	0	3	1	3	2	4

<b>MS</b>	2	2	1	1	2	2	2	3	2	2
<b>CT</b>	0	0	1	3	3	0	1	0	1	1
<b>Kode Shift</b>	<b>Tanggal</b>									
	<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>	<b>25</b>	<b>26</b>	<b>27</b>	<b>28</b>	<b>29</b>	<b>30</b>
<b>P</b>	4	6	4	5	3	3	2	5	5	5
<b>P1</b>	1	1	1	1	1	1	1	1	1	1
<b>P2</b>	1	1	1	1	1	1	1	1	1	1
<b>PS</b>	0	0	0	0	1	2	2	0	0	0
<b>S</b>	6	4	5	4	5	3	4	6	5	6
<b>M</b>	1	1	1	1	1	1	1	1	1	1
<b>LM</b>	1	0	1	0	1	0	1	0	1	0
<b>L</b>	2	1	0	3	3	6	3	2	1	1
<b>MS</b>	2	1	1	1	1	1	2	2	2	2
<b>CT</b>	1	4	5	3	2	1	2	1	2	2

Table 4.9 Jumlah shift tiap staf unit gizi

<b>Kode Staf</b>	<b>Kode Shift</b>									
	<b>P</b>	<b>P1</b>	<b>P2</b>	<b>PS</b>	<b>S</b>	<b>M</b>	<b>LM</b>	<b>L</b>	<b>MS</b>	<b>CT</b>
<b>401</b>	25	0	0	0	0	0	0	4	0	0
<b>402</b>	6	7	2	0	10	0	0	4	0	1
<b>403</b>	2	4	0	0	5	0	0	4	13	2
<b>404</b>	2	5	1	0	8	0	0	4	6	4
<b>405</b>	2	2	0	0	9	0	0	4	8	5
<b>406</b>	6	0	5	0	16	0	0	3	0	0
<b>407</b>	0	12	0	0	9	0	0	4	2	3
<b>408</b>	4	0	12	0	7	0	0	4	0	3
<b>409</b>	2	0	9	0	13	0	0	4	0	2
<b>410</b>	5	0	0	0	5	7	3	4	4	2
<b>411</b>	11	0	0	0	5	0	0	4	5	5

Kode Staf	Kode Shift									
	P	P1	P2	PS	S	M	LM	L	MS	CT
412	2	0	0	0	9	7	3	4	1	4
413	7	0	0	0	6	6	3	4	4	0
414	6	0	0	0	4	6	3	4	3	4
415	8	0	0	0	4	6	2	4	6	0
416	10	0	0	5	7	0	0	4	0	4
417	10	0	0	4	11	0	0	4	0	1
418	10	0	0	4	12	0	0	4	0	0
419	5	0	0	4	10	0	0	4	0	7

#### 4.2.4 Unit IGD

- Shift terdiri dari 6 bagian yaitu shift P (pagi) dari pukul 07.00 sampai 14.00, lalu shift S (sore) dari pukul 14.00 sampai 21.00, shift M (malam) dari pukul 21.00 hingga pukul 07.00, shift MS (middle) dari pukul 10.00 hingga pukul 17.00, shift MD (middle 2) dari pukul 12.00 hingga 19.00, dan shift LM (lepas malam)
- Unit terdiri dari 8 Staf
- Setiap staf hanya boleh mengisi 1 shift per hari nya
- Setelah selesai 2 kali shift malam, maka harus melanjutkan pola pokok dan mendapatkan lepas malam dan libur untuk 2 hari kedepan
- Selalu ada staf minimal 1 di setiap shift (kecuali shift middle) setiap harinya

**Tabel 4.10 Kodifikasi tipe skill unit IGD**

No	Kode Staff	Skill	Tipe Skill
1	501	Anggota	4
2	502	Anggota	4

No	Kode Staff	Skill	Tipe Skill
3	503	Anggota	4
4	504	Anggota	4
5	505	Anggota	4
6	506	Anggota	4
7	507	Anggota	4
8	508	Anggota	4

Tabel 4.11 Jumlah staf pada setiap shift unit IGD

Kode Shift	Tanggal									
	1	2	3	4	5	6	7	8	9	10
<b>P</b>	1	1	1	2	1	1	1	1	1	1
<b>S</b>	3	4	3	1	1	3	3	3	2	3
<b>MD</b>	1	2	1	1	0	0	1	1	0	0
<b>L</b>	1	0	1	2	3	1	0	0	1	0
<b>M</b>	1	1	1	1	1	1	1	1	1	1
<b>LM</b>	1	0	1	0	1	0	1	0	1	0
<b>MS</b>	0	0	0	1	0	1	1	1	2	0
<b>CT</b>	0	0	0	0	0	1	0	1	0	1
Kode Shift	Tanggal									
	11	12	13	14	15	16	17	18	19	20
<b>P</b>	2	1	2	1	1	1	1	2	1	1
<b>S</b>	1	1	3	4	2	2	2	1	1	4
<b>MD</b>	1	0	0	1	0	0	0	1	0	1
<b>L</b>	1	4	0	0	1	1	1	2	3	0

<b>M</b>	1	1	1	1	1	1	1	1	1	1
<b>LM</b>	1	0	1	0	1	0	1	0	1	0
<b>MS</b>	1	0	1	0	2	2	1	1	0	0
<b>CT</b>	0	0	0	0	0	1	1	0	0	1
<b>Kode Shift</b>	<b>Tanggal</b>									
	<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>	<b>25</b>	<b>26</b>	<b>27</b>	<b>28</b>	<b>29</b>	<b>30</b>
<b>P</b>	1	1	1	1	2	1	1	1	1	1
<b>S</b>	3	3	3	4	1	1	3	3	2	2
<b>MD</b>	0	1	1	1	1	0	0	1	0	0
<b>L</b>	0	1	1	1	1	2	0	1	0	2
<b>M</b>	1	1	1	1	1	1	1	1	1	1
<b>LM</b>	1	0	0	0	1	0	1	0	1	0
<b>MS</b>	1	1	1	0	1	0	1	1	2	2
<b>CT</b>	0	0	0	0	0	2	1	0	1	0

Tabel 4.12 Jumlah shift tiap staf unit IGD

<b>Kode Staff</b>	<b>Kode Shift</b>							
	<b>P</b>	<b>S</b>	<b>MD</b>	<b>L</b>	<b>M</b>	<b>LM</b>	<b>CT</b>	<b>MS</b>
<b>501</b>	5	8	5	4	2	1	2	3
<b>502</b>	6	7	1	4	4	2	2	3
<b>503</b>	2	11	2	4	4	1	2	3
<b>504</b>	3	10	3	4	4	2	1	3
<b>505</b>	6	8	2	4	4	1	2	3
<b>506</b>	3	9	0	4	6	3	1	3
<b>507</b>	5	9	3	4	2	2	1	2
<b>508</b>	5	10	1	3	4	2	0	4

#### 4.2.5 Unit Kamar Operasi

- Shift hanya terdiri dari 3 bagian yaitu shift P (pagi) dari pukul 07.00 sampai 14.00, lalu shift S (sore) dari pukul 14.00 sampai 21.00, dan shift M (malam) sesuai panggilan (on call) dari pukul 21.00 hingga pukul 07.00.
- Unit terdiri dari 6 Staff
- Setiap staf hanya boleh mengisi 1 shift per hari nya
- Selalu ada staf minimal 1 di setiap shift (kecuali shift middle) setiap harinya

**Tabel 4.13 Kodifikasi tipe skill unit Kamar Operasi**

No	Kode Staff	Skill	Tipe Skill
1	601	Anggota	4
2	602	Anggota	4
3	603	Anggota	4
4	604	Anggota	4
5	605	Anggota	4

**Tabel 4.14 Jumlah staf pada setiap shift unit Kamar Operasi**

Kode Shift	Tanggal									
	1	2	3	4	5	6	7	8	9	10
<b>P</b>	4	3	4	3	4	3	2	3	4	3
<b>S</b>	1	2	2	2	0	2	2	2	2	2
<b>M</b>	0	0	0	0	0	0	0	0	0	0
<b>MS</b>	0	0	0	0	0	0	0	0	0	0
<b>LM</b>	0	0	0	0	0	0	0	0	0	0
<b>L</b>	1	1	0	1	2	0	1	1	0	1

Kode Shift	Tanggal									
	1	2	3	4	5	6	7	8	9	10
CT	0	0	0	0	0	1	1	0	0	0
Kode Shift	Tanggal									
	11	12	13	14	15	16	17	18	19	20
P	3	4	3	3	3	4	3	3	4	3
S	2	0	2	2	2	2	2	2	0	2
M	0	0	0	0	0	0	0	0	0	0
MS	0	0	0	0	0	0	0	0	0	0
LM	0	0	0	0	0	0	0	0	0	0
L	0	2	1	0	1	0	1	1	2	1
CT	1	0	0	1	0	0	0	0	0	0
Kode Shift	Tanggal									
	21	22	23	24	25	26	27	28	29	30
P	4	3	3	3	3	4	3	4	3	3
S	2	2	2	2	2	0	2	2	2	2
M	0	0	0	0	0	0	0	0	0	0
MS	0	0	0	0	0	0	0	0	0	0
LM	0	0	0	0	0	0	0	0	0	0
L	0	1	1	1	0	2	1	0	1	0
CT	0	0	0	0	1	0	0	0	0	1

Tabel 4.15 Jumlah shift tiap staf unit Kamar Operasi

Kode Staff	Kode Shift						
	P	S	M	MS	LM	L	CT
601	7	9	5	3	2	5	0
602	9	9	4	2	2	5	0
603	9	7	4	2	3	6	0
604	9	6	6	2	3	5	0

Kode Staff	Kode Shift						
	P	S	M	MS	LM	L	CT
605	7	7	6	3	3	5	0
606	6	9	6	3	3	4	0

#### 4.2.6 Unit SIM & RM

- Shift terdiri dari 5 bagian yaitu shift P (pagi) dari pukul 07.00 sampai 14.00, lalu shift S (sore) dari pukul 14.00 sampai 21.00, shift M (malam) dari pukul 21.00 hingga pukul 07.00, shift MS (middle) dari pukul 10.00 hingga pukul 17.00, dan shift lepas malam
- Unit terdiri dari 6 staff
- Setiap staf hanya boleh mengisi 1 shift per hari nya
- Setelah selesai 2 kali shift malam, maka harus melanjutkan pola pokok dan mendapatkan lepas malam dan libur untuk 2 hari kedepan
- Selalu ada staf minimal 1 di setiap shift (kecuali shift middle) setiap harinya

**Tabel 4.16 Kodifikasi tipe skill unit SIM & RM**

No	Kode Staff	Skill	Tipe Skill
1	301	Anggota	4
2	302	Anggota	4
3	303	Anggota	4
4	304	Anggota	4
5	305	Anggota	4
6	306	Anggota	4



Tabel 4.18 Jumlah shift tiap staf unit SIM &amp; RM

Kode Staff	Kode Shift						
	P	S	M	MS	LM	L	CT
301	8	8	5	2	2	4	1
302	8	8	5	2	3	4	0
303	7	8	5	2	2	5	1
304	8	7	5	2	3	4	1
305	8	7	5	3	3	4	0
306	7	8	5	3	3	4	0

### 4.3 Proses Pembuatan Model

Masalah penjadwalan staf pada RSIA Kendangsari adalah menjadwalkan staf dalam kurun waktu 1 bulan yaitu 31 hari atau 4 minggu 3 hari yang telah memenuhi waktu kerja dan batasan-batasan staf lainnya. Lalu dalam penyelesaian permasalahan ini diusahakan untuk memenuhi tujuan yang telah ditentukan.

Tabel 4.19 Kodifikasi Tipe Shift

Type Shift	Kode	Shift Id	Waktu
Pagi	P	1	07:00 – 14:00
Pagi 1	P1	2	04:30 – 11:30
Pagi 2	P2	3	05:00 – 12:00
Pagi Sore	PS	4	06:00 – 02:00
Sore	S	5	14:00 – 21:00
Middle	MS	6	10:00 – 17:00
Middle 2	MD	7	12:00 – 19:00
Malam	M	8	21:00 – 07:00
Lepas Malam	LM	9	-
Libur	L	10	-
Cuti	CT	11	-

Keterangan:

1. Shift pada unit farmasi :  
Pagi, sore, malam, middle, lepas malam, dan shift libur
2. Shift pada unit bayi/NICU :

- Pagi, sore, malam, middle, lepas malam, dan shift libur
3. Shift pada unit gizi :  
Pagi, pagi 1, pagi 2, sore, malam, middle, lepas malam, dan shift libur
  4. Shift pada unit IGD :  
Pagi, sore, malam, middle, middle 2, lepas malam, dan shift libur
  5. Shift pada unit Kamar Operasi :  
Pagi, sore, malam, dan shift libur
  6. Shift pada unit SIM &RM :  
Pagi, sore, malam, middle, lepas malam, dan shift libur

#### 4.3.1 Batasan

Penyelesaian permasalahan penjadwalan staf pada unit Instalasi Farmasi ini didasarkan atas batasan sesuai dengan aturan dan regulasi yang berlaku. Batasan yang berlaku dibagi menjadi dua bagian yaitu *Hard Constraint* yaitu batasan yang tidak dapat dilanggar dan *Soft Constraint* yaitu batasan yang dapat dilanggar. Berikut merupakan batasan-batasan yang berlaku pada unit instalasi Farmasi RSIA Kendangsari.

#### ***Hard Constraint :***

- Unit Farmasi
  1. Jumlah staff di shift pagi tiap harinya harus sama dengan 3
  2. Jumlah staff di shift siang tiap harinya harus sama dengan 2
  3. Jumlah staff di shift malam tiap harinya harus sama dengan 1
- Unit Gizi
  1. Jumlah staff di shift pagi sore (PS) harus sama dengan 2 di hari Minggu
  2. Jumlah staff di shift pagi sore (PS) harus sama dengan 2 di hari Senin
  3. Jumlah staff di shift malam selalu sama dengan 1 tiap harinya
- Unit NICU/Bayi

1. Jumlah staff di shift pagi setiap hari harus lebih dari sama dengan dua dan kurang dari sama dengan empat
  2. Jumlah staff di shift siang setiap hari harus lebih dari sama dengan satu dan kurang dari sama dengan 2
  3. Jumlah staff di shift malam setiap hari harus lebih dari sama dengan dua dan kurang dari sama dengan tiga
- Unit SIM & RM
    1. Jumlah staff di shift pagi tiap harinya lebih dari sama dengan 1 atau kurang dari sama dengan 2
    2. Jumlah staff di shift siang tiap harinya lebih dari sama dengan 1 atau kurang dari sama dengan 2
    3. Jumlah staff di shift malam tiap harinya harus sama dengan 1
  - Unit IGD
    1. Jumlah staff di shift pagi tiap harinya lebih dari sama dengan satu atau kurang dari sama dengan dua
    2. Jumlah staff di shift siang setiap hari harus lebih dari samadengan dua dan kurang dari sama dengan empat
    3. Jumlah staff di shift malam setiap hari harus berjumlah satu atau dua
  - Unit Kamar Operasi
    1. Jumlah staff di shift pagi setiap hari harus lebih dari samadengan dua dan kurang dari sama dengan empat
    2. Jumlah staff di shift siang setiap hari harus lebih dari samadengan dua atau kurang dari samadengan tiga
    3. Jumlah staff di shift pagi pada hari Minggu harus terdiri dari empat

### ***Soft Constraint***

- Ukuran optimality dari Jain Fairness Index untuk setiap unit pada RSIA Kendangsari
- Jumlah libur tiap staf dalam 1 bulan sama dengan jumlah hari minggu pada 1 bulan

#### 4.3.2 Asumsi dan Notasi

Penjadwalan otomatis diasumsikan dimulai pada hari pertama bulan Desember 2017 yaitu pada hari Jumat dimana bulan Desember 2017 terdiri atas 31 hari. Berikut adalah notasi-notasi yang digunakan.

<b>i</b>	= Staf yang dijadwalkan
<b>d</b>	= Hari yang dijadwalkan
<b>t</b>	= Tipe shift yang akan dialokasikan,
<b>n</b>	= Jumlah staf

#### 4.3.3 Variabel Keputusan

Berikut adalah variabel keputusan dari penelitian ini:

**X<sub>idt</sub>** = bernilai 1 jika staf **i** ditugaskan pada shift **t** di hari **d**, bernilai 0 jika tidak.

Dimana:

**i** = 1,2,3, ... **n** **n** adalah jumlah staf

**d** = 1,2,3, ... **k**; **k** adalah jumlah hari periode penjadwalan

**t** = 1,2,3, ... **z**; **z** adalah tipe shift (lihat Tabel 4.19)

#### 4.3.4 Pemodelan Hard Constraint

##### 1. Pemodelan Unit Farmasi

Unit Farmasi di RSIA Kendangsari Merr Surabaya ini terdiri atas 8 orang personil yang terbagi menjadi 1 orang Apoteker, 1 orang staff senior, 5 orang anggota dan 1 orang petugas gudang obat. Dalam penjadwalan instalasi farmasi terdapat 6 jenis shift baik shift kerja maupun shift libur, yaitu shift pagi, shift sore, shift malam, middle shift, libur, dan lepas malam. Batasan yang dirumuskan berdasarkan batasan sebagai berikut:

##### **Hard Constraint 1**

Jumlah staff di shift pagi tiap harinya harus sama dengan 3. Dirumuskan: Untuk setiap nilai d.

$$\sum_{i=1}^n X_{id1} = 3 \quad (4.1)$$

### **Hard Constraint 2**

Jumlah staff di shift siang tiap harinya harus sama dengan 2. Dirumuskan: Untuk setiap nilai d.

$$\sum_{i=1}^n X_{id5} = 2 \quad (4.2)$$

### **Hard Constraint 3**

Jumlah staff di shift malam tiap harinya harus sama dengan 1 Dirumuskan: Untuk setiap nilai d.

$$\sum_{i=1}^n X_{id8} = 1 \quad (4.3)$$

## 2. Pemodelan Unit Bayi/NICU

Unit NICU/Bayi di RSIA Kendangsari Merr Surabaya ini terdiri atas 13 tenaga kerja dengan rincian 1 kepala unit dan 12 anggota. Dalam penjadwalan instalasi NICU/Bayi terdapat 5 jenis shift baik shift kerja maupun shift libur, yaitu shift pagi, shift sore shift malam, shift lepas malam dan shift middle. Batasan yang dirumuskan berdasarkan batasan sebagai berikut:

### **Hard Constraint 1**

Jumlah staff di shift pagi setiap hari harus lebih dari sama dengan dua dan kurang dari sama dengan empat. Dirumuskan: Untuk setiap nilai d.

$$2 \leq \sum_{i=1}^n X_{id1} \leq 4 \quad (4.4)$$

### Hard Constraint 2

Jumlah staff di shift siang setiap hari harus lebih dari sama dengan satu dan kurang dari sama dengan 2. Dirumuskan: Untuk setiap nilai d.

$$1 \leq \sum_{i=1}^n X_{id5} \leq 2 \quad (4.5)$$

### Hard Constraint 3

Jumlah staff di shift malam setiap hari harus lebih dari sama dengan dua dan kurang dari sama dengan tiga. Dirumuskan: Untuk setiap nilai d.

$$2 \leq \sum_{i=1}^n X_{id8} \leq 3 \quad (4.6)$$

### 3. Pemodelan Unit Gizi

Unit Gizi di RSIA Kendangsari Merr Surabaya ini terdiri atas 19 orang personil yang terbagi menjadi 1 Ahli gizi, 4 chef, 4 helper, 6 penyaji, 2 café, dan 2 driver. Dalam penjadwalan instalasi gizi terdapat 7 jenis shift baik shift kerja maupun shift libur, yaitu shift P1 (Pagi 1), shift P2 (Pagi 2), shift P (Pagi), shift PS (Pagi Sore), shift malam dan shift middle. Batasan yang dirumuskan berdasarkan batasan sebagai berikut:

### Hard Constraint 1

Jumlah staff di shift pagi sore (PS) harus sama dengan 2 di hari Minggu. Dirumuskan: Untuk setiap nilai d, jika d adalah hari Minggu

$$\sum_{i=1}^n X_{id4} = 2 \quad (4.7)$$

**Hard Constraint 2**

Jumlah staff di shift pagi sore (PS) harus sama dengan 2 di hari Senin. Dirumuskan: Untuk setiap nilai  $d$ , jika  $d$  adalah hari Senin.

$$\sum_{i=1}^n X_{id4} = 2 \quad (4.8)$$

**Hard Constraint 3**

Jumlah staff di shift malam selalu sama dengan 1 tiap harinya. Dirumuskan: Untuk setiap nilai  $d$ .

$$\sum_{i=1}^n X_{id8} = 1 \quad (4.9)$$

4. **Pemodelan Unit IGD**

Unit IGD di RSIA Kendangsari Merr Surabaya ini terdiri atas 8 staff. Dalam penjadwalan instalasi IGD terdapat 6 jenis shift baik shift kerja maupun shift libur, yaitu shift pagi, shift sore shift malam, shift lepas malam, shift middle, dan shift middle 2. Batasan yang dirumuskan berdasarkan batasan sebagai berikut:

**Hard Constraint 1**

Jumlah staff di shift pagi tiap harinya lebih dari sama dengan satu atau kurang dari sama dengan dua. Dirumuskan: Untuk setiap nilai  $d$ .

$$1 \leq \sum_{i=1}^n X_{id1} \leq 2 \quad (4.10)$$

**Hard Constraint 2**

Jumlah staff di shift siang setiap hari harus lebih dari sama dengan dua dan kurang dari sama dengan empat. Dirumsukan: Untuk setiap nilai d.

$$2 \leq \sum_{i=1}^n X_{id1} \leq 4 \quad (4.11)$$

### Hard Constraint 3

Jumlah staff di shift malam setiap hari harus berjumlah satu atau dua. Dirumsukan: Untuk setiap nilai d.

$$1 \leq \sum_{i=1}^n X_{id8} \leq 2 \quad (4.12)$$

#### 5. Pemodelan Unit Kamar Operasi

Unit OK di RSIA Kendangsari Merr Surabaya ini terdiri atas 6 staff. Dalam penjadwalan instalasi OK terdapat 4 jenis shift baik shift kerja maupun shift libur, yaitu shift pagi dari pukul, shift sore dan shift malam sesuai panggilan (on call). Batasan yang dirumuskan berdasarkan batasan sebagai berikut:

### Hard Constraint 1

Jumlah staff di shift pagi setiap hari harus lebih dari sama dengan dua dan kurang dari sama dengan empat. Dirumsukan: Untuk setiap nilai d.

$$2 \leq \sum_{i=1}^n X_{id1} \leq 4 \quad (4.13)$$

### Hard Constraint 2

Jumlah staff di shift siang setiap hari harus lebih dari sama dengan dua atau kurang dari sama dengan tiga. Dirumsukan: Untuk setiap nilai d.

$$2 \leq \sum_{i=1}^n X_{id5} \leq 3 \quad (4.14)$$

### Hard Constraint 3

Jumlah staff di shift pagi pada hari Minggu harus terdiri dari empat. Dirumuskan: Untuk setiap nilai d, jika d adalah hari Minggu

$$\sum_{i=1}^n X_{id1} = 4 \quad (4.15)$$

#### 6. Pemodelan Unit SIM & RM

Unit SIM & RM di RSIA Kendangsari Merr Surabaya ini terdiri atas dari 6 staff. Dalam penjadwalan instalasi SIM & RM terdapat 5 jenis shift baik shift kerja maupun shift libur, yaitu shift pagi, shift sore shift malam, shift lepas malam dan shift middle. Batasan yang dirumuskan berdasarkan batasan sebagai berikut:

### Hard Constraint 1

Jumlah staff di shift pagi tiap harinya lebih dari sama dengan 1 atau kurang dari sama dengan 2. Dirumsukan: Untuk setiap nilai d.

$$1 \leq \sum_{i=1}^n X_{id1} \leq 2 \quad (4.16)$$

### Hard Constraint 2

Jumlah staff di shift siang tiap harinya lebih dari sama dengan 1 atau kurang dari sama dengan 2. Dirumsukan: Untuk setiap nilai d.

$$1 \leq \sum_{i=1}^n X_{id5} \leq 2 \quad (4.17)$$

### Hard Constraint 3

Jumlah staff di shift malam tiap harinya harus sama dengan 1. Dirumuskan: Untuk setiap nilai d.

$$\sum_{i=1}^n X_{id8} = 1 \quad (4.18)$$

#### 4.3.5 Pemodelan Soft Constraint

##### Soft Constraint 1

Ukuran optimality dari Jain Fairness Index untuk setiap unit pada RSIA Kendangsari.

$$JFI = \frac{(\sum_{i=1}^n f_i)^2}{(n * \sum_{i=1}^n (f_i)^2)} \quad (4.19)$$

$$f_i = \sum_{d=1}^k \sum_{t=1}^z |X_{idt} - 1| (4M_d + 2S_d + W_d) \quad (4.20)$$

$$M_d \begin{cases} 1, \text{jika } d \text{ adalah hari Minggu} \\ 0, \text{jika tidak} \end{cases}$$

$$S_d \begin{cases} 1, \text{jika } d \text{ adalah hari Sabtu} \\ 0, \text{jika tidak} \end{cases}$$

$$W_d \begin{cases} 1, \text{jika } d \text{ adalah hari Senin sampai Jumat} \\ 0, \text{jika tidak} \end{cases}$$

##### Soft Constraint 2

Jumlah libur tiap staf dalam 1 bulan sama dengan jumlah hari minggu pada 1 bulan

$$\sum_{i=1}^n X_{id10} = C \quad (4.21)$$

$$C = \sum \text{hari Minggu dalam 1 bulan}$$

$$X_{id10} \begin{cases} 1, \text{ jika staff } i \text{ pada hari ke } - j \text{ mendapat shift "L"} \\ 0, \text{ jika tidak} \end{cases}$$

#### 4.3.6 Fungsi Tujuan

Tujuan dari pemodelan permasalahan ini adalah untuk memaksimalkan hasil perhitungan *Jain's Fairness Index (JFI)*. Pemodelan fungsi tujuannya dapat dilihat pada persamaan dibawah:

$$MAX JFI = \frac{(\sum_{i=1}^n f_i)^2}{(n * \sum_{i=1}^n (f_i)^2)} \quad (4.22)$$

$$f_i = \sum_{d=1}^k \sum_{t=1}^z |X_{idt} - 1| (4M_d + 2S_d + W_d) \quad (4.23)$$

$$M_d \begin{cases} 1, \text{ jika } d \text{ adalah hari Minggu} \\ 0, \text{ jika tidak} \end{cases}$$

$$S_d \begin{cases} 1, \text{ jika } d \text{ adalah hari Sabtu} \\ 0, \text{ jika tidak} \end{cases}$$

$$W_d \begin{cases} 1, \text{ jika } d \text{ adalah hari Senin sampai Jumat} \\ 0, \text{ jika tidak} \end{cases}$$

#### 4.4 Pemodelan Algoritma Reinforcement Learning

Seperti yang sudah dijelaskan sebelumnya, algoritma reinforcement learning digunakan untuk menentukan low-level heuristic mana yang akan dipilih. Tujuannya adalah untuk mendapatkan solusi baru yang lebih baik dari solusi yang sebelumnya sudah dihasilkan. Secara garis besar, ide dari algoritma ini adalah tiap *low-level heuristic* akan diberikan *score* yang sama. Lalu jika heuristic terpilih dan memberikan hasil pada solusi yang diterima, maka *score* akan bertambah 1. Sebaliknya, jika heuristic terpilih tetapi memberikan hasil pada solusi yang ditolak, maka *score* akan berkurang 1 sampai batas

bawah *score* tercapai. Jika ada lebih dari satu *low-level heuristics* dengan *score* tertinggi, salah satunya akan dipilih secara acak.

```

1: procedure SOLVE(ProblemDomain P)
2:   Set stopping condition  $S_t$ 
3:   Set  $M \leftarrow P.getNumberOfHeuristics()$ 
4:   // R is an array to store reward given to each low-
level heuristic.
5:   Set  $R \leftarrow \text{new int}[M]$ 
6:   initiateReward(R)
7:   P.initiateSolution(0)
8:    $S_{best} \leftarrow P.getFunctionValue(0)$ 
9:    $S_{curr} \leftarrow P.getFunctionValue(0)$ 
10:   $B \leftarrow P.getFunctionValue(0)$ 
11:  Set decay rate  $\alpha$ 
12:  while  $S_t$  is not met do
13:     $l \leftarrow \text{selectLLH}(R)$ 
14:    newFunctionVal  $\leftarrow P.applyLLH(1,0,1)$ 
15:    if (newFunctionVal)  $\leq S_{curr}$  OR
newFunctionVal  $\leq B$ ) then
16:      //the new solution is accepted
17:      P.copySolution(1,0)
18:       $S_{curr} \leftarrow \text{newFunctionVal}$ 
19:      upgradeReward(R[l])
20:      if newFunctionVal  $< S_{best}$  then
21:        Set  $S_{best} \leftarrow$ 
newFunctionVal
22:      end if
23:    else
24:      downgradeReward(R[l])
25:    end if
26:     $B \leftarrow B \cdot \alpha$ 
27:  end while
28:  return  $S_{best}$ 
29: end procedure

```

4.1 Pseudocode Algoritma Reinforcement Learning Extended Great Deluge base Hyper-heuristics. A. Muklason, "Hyper-Heuristics And Fairness In Examination Timetabling Problems," 2017

**Table 4.20** Daftar variabel yang digunakan. A. Muklason, "Hyper-Heuristics And Fairness In Examination Timetabling Problems," 2017

Variabel	Deskripsi
P	Menguji instance permasalahan jadwal
St	Kondisi berhenti
M	Nomor low-level heuristics
R	Array untuk reward yang diberikan untuk tiap low-level heuristics
Sbest	Nilai fungsi tujuan dari solusi terbaik sejauh ini
Scurr	Nilai fungsi tujuan dari solusi saat ini
B	Level boundary
A	Nilai decay (kerusakan)
l	Low-level heuristics yang terpilih
newFunctionVal	Nilai fungsi tujuan dari solusi yang baru
Ic	Idle Counter
Rp	Reheating point

#### 4.5 Pemodelan Algoritma Hill Climbing

*Hill Climbing Algorithm (HCA)* adalah teknik untuk optimasi masalah berbasis matematika yang bersifat iteratif, memberikan sebuah solusi untuk masalah yang awalnya ditentukan dengan cara memberikan solusi berubah-ubah dan kemudian berusaha untuk menemukan solusi yang lebih baik dengan memodifikasi elemen secara bertahap. Pada metode ini iterasi akan terus dilakukan hingga mencapai titik teratas yang dinamakan best solution. Awalnya, perlu dideklarasikan variabel untuk current solution dan best solution. Kemudian, pemilihan low-level heuristic dilakukan secara random. Low-level yang dipakai pada Tugas Akhir ini adalah move dan swap. Kemudian setelah salah satu dieksekusi, hasilnya akan diberi label sebagai current solution. Jika current solution lebih besar dari best solution, maka current solution tersebut akan dijadikan sebagai beest

solution untuk iterasi selanjutnya. Jika tidak, maka kembali menjadi solusi awal.

```
1: Function Hill-Climbing (problem) returns a state that is a
   local maximum
2:   Inputs: problem
3:   Local variables: current node(c), Neighbouring
   node(n)
4:   c = makenode (initial[problem])
5:   do while
6:     {
7:     n = highest-valued successor of c (looks at all
   successors)
8:     If value[n] < value[c] then return the state[current]
9:     c = n
10:    }
```

**Gambar 4.2 Pseudocode Algoritma Hill Climbing**

## **BAB V IMPLEMENTASI**

Bab ini akan membahas mengenai implementasi dari model yang telah dibuat sebelumnya

### **5.1 Lingkungan Uji Coba**

Uji coba otomatisasi dan optimasi dengan menggunakan algoritma Reinforcement Learning untuk kasus penjadwalan staf IGD di Rumah Sakit Ibnu Sina Makassar ini dilakukan menggunakan bantuan Netbeans dan bahasa pemrograman Java. Komputer yang digunakan adalah sebuah laptop Intel Core i5 dengan kapasitas RAM sebesar 8 GB dan kapasitas Hard disk sebesar 750GB. Perangkat lunak yang digunakan dalam pengerjaan penelitian ini adalah sistem operasi Windows 8, dengan aplikasi komputasi Netbeans IDE 8.2, dan media penyimpanan data Microsoft Excel. Pada tabel 5.1 menampilkan perangkat keras dan lunak yang dipergunakan dalam pengerjaan penelitian tugas akhir ini.

**Tabel 5.1 Perangkat Keras dan Lunak yang digunakan**

<b>Perangkat Keras</b>	
Komputer	Laptop
Processor	Intel® Core i5
RAM	8 GB
Hard disk	750 GB
<b>Perangkat Lunak</b>	
Microsoft Windows 8	Sistem Operasi
Netbeans	Aplikasi pengembangan perangkat lunak
Microsoft Excel 2013	Editor media penyimpanan data masukan dan keluaran

### **5.2 Tahapan Otomasi Penjadwalan**

Setelah menentukan batasan-batasan yang akan digunakan sebelumnya, tahap berikutnya adalah membuat jadwal dengan

suatu pola tertentu secara manual. Harapannya, jadwal manual ini dapat menjadi patokan dalam pembuatan jadwal yang terotomasi dengan menggunakan Java. Melihat data jadwal staff yang sangat tidak beraturan, jadi model yang ada di literatur tidak bisa dipakai disini sehingga ada beberapa pola yang diasumsikan sehingga nantinya dapat memudahkan dalam pembuatan otomasi jadwalnya. Asumsi tersebut tentunya dibuat dengan terlebih dahulu memahami model matematis yang telah dibuat, dan juga tentu tidak mengabaikan batasan-batasan dan regulasi dari tiap unitnya. Berikut merupakan tahapan pembuatan jadwal secara manual untuk masing-masing unit:

### 5.2.1 Unit Farmasi

1. Kepala Unit dan Petugas Gudang selalu memiliki shift pagi dari hari Senin-Sabtu dan libur setiap hari Minggu.
2. Untuk staf nomor 2 hanya memiliki pola Shift Siang,
3. Lihat pola M-M-LM-L yang belum lengkap pada bulan sebelumnya
4. Isikan pola M-M-LM-L melanjutkan pola bulan sebelumnya untuk Staff nomor 3 sampai 7
5. Setiap hari harus memiliki paling tidak 1 orang shift malam, sehingga jika salah satu staf sudah shift LM maka harus ada staf yang lain shift M
6. Untuk staf nomor 3 sampai 7, pola sehabis shift libur adalah P-P-S-S
7. Untuk staf no 3 sampai 7 sebelum mendapatkan shift malam harus mendapatkan shift siang
8. Dalam satu hari shift pagi harus sebanyak 3, shift siang sebanyak 2, dan 1 shift malam
9. Khusus hari minggu shift pagi sama dengan 1, siang sama dengan 2, malam sama dengan 1, dan tidak ada middle shift
10. Alokasikan request Libur dan Cuti
11. Jika jumlah Libur, Cuti, dan Lepas malam kurang dari sama dengan satu maka shift sisa merupakan shift middle

### 5.2.2 Unit Bayi/NICU

1. Isikan Pola M-M-LM-L pada setiap anggotanya, kecuali pada jadwal kepala anggota, karena kepala anggota yang

hanya memiliki pola P dan L dalam satu minggunya, setiap minggunya kepala anggota libur pada hari minggu dan memiliki shift pagi dari hari senin sampai sabtu

2. Selanjutnya untuk anggota kedua sampai anggota ke tiga belas isikan pola P-P-S-S-Ms-Ms setelah mengisi pola M-M-LM-L
3. Isi juga secara menurun geser dua kotak kekanan secara berulang

### 5.2.3 Unit Gizi

1. Ahli Gizi selalu memiliki shift pagi dari hari Senin-Sabtu dan libur setiap hari Minggu.
2. Untuk Staf nomor 2 sampai 5 (Chef), isikan pola P1-P-S-P1-P-L-S-S-MS-MS dengan mengisi jarak 10 kotak dengan starting point +2
3. Untuk Staf nomor 7-9 (Helper), isikan pola P2-P-S-S-L-P-S-P2-P-MS dengan mengisi jarak 10 kotak dengan starting point +2
4. Untuk staf nomer 6 isikan pola L-P-P-S-S-S-P-P dengan jarak antar pola 7
5. Staff nomor 10 isikan pola S-S-P-P-MD-MD-L dengan jarak 7
6. Lihat pola M-M-LM-L yang belum lengkap pada bulan sebelumnya untuk staff nomor 11-15
7. Isikan pola M-M-LM-L melanjutkan pola bulan sebelumnya untuk staff nomor 11-15
8. Setiap hari harus memiliki paling tidak 1 orang shift malam sehingga jika salah satu staf sudah shift LM maka harus ada staf yang lain shift M
9. Untuk staf nomor 11-15 memiliki pola sehabis shift libur adalah P-P-S-S
10. Untuk staf nomor 11-15 saat ada sel yang masih kosong dan jika pada kolom tersebut terdapat MD maka isikan shift P pada sel tersebut, dan jika tidak ada MD maka isikan dengan MD
11. Alokasikan request Libur dan Cuti

12. Untuk penyaji, jika jumlah Libur, Cuti, dan Lepas malam kurang dari sama dengan satu maka shift sisa merupakan shift middle
13. Untuk staff 16 dan 19 memiliki pola PS-S-S-P-P-P-PS-L-P-P-S-S-S-L
14. Untuk staff 17 dan 18 memiliki pola L-P-P-S-S-S-L-PS-S-S-P-P-P-PS

#### 5.2.4 Unit IGD

1. Masukkan pola M-M-LM-L berdasarkan pola bulan sebelumnya
2. Setiap minggu tiap staf dapat 1 kali shift pagi, 3 kali shift sore, 1 Libur, sisanya middle shift
3. Total setiap bulan untuk tiap shift adalah 4 kali shift pagi, 9 kali shift sore, 4 libur, malam 4 kali (dari mmlm l) sisanya MD atau MS (acak)
4. Masukkan pola M-M-LM-L-P-P-S-S-S-S-L-MS-MD-S-S-S

#### 5.2.5 Unit Kamar Operasi

1. Isikan Pola P-P-S-P-S-L-P-P-S-(-1) untuk setiap stafi dengan roling turun satu kotak (gunakan perulangan kolom didalam baris)
2. Mengisikan kolom kosong dengan spesifikasi siang sama dengan 2, dan p kurang dari sama dengan 3 dengan langkah menjumlahkan isi kolom, kemudian membuat kondisi yang sesuai
3. Setiap hari minggu shift P harus berjumlah 4 dan L berjumlah 2 (isi Kolom L-L-P-P-P-P secara berulang ke bawah turun 2 kotak )

#### 5.2.6 Unit SIM & RM

1. Isikan pola M-M-LM-L mengikuti pola sebelumnya pada setiap anggota.
2. Pengisian pola M-M-LM-L mengikuti aturan setiap shift malam hanya berisi 1 staff. Bila hari tersebut sudah ada orang yang mendapat shift malam, maka staff lain tidak boleh mendapat shift malam.

3. Pengisian pola M-M-LM-L diikuti dengan pola P-P-P-L-S-S-S(-1) di bagian belakangnya.
4. Mengisi pola yang kosong dengan ketentuan apabila dalam satu hari jumlah P, S, dan M-nya kurang dari sama dengan 3 maka diisi dengan “M”. Apabila lebih dari 3 maka diisi P

### 5.3 Algoritma Otomasi dan Cek Constraint Penjadwalan

Hal pertama yang perlu diperhatikan disini adalah program didesain untuk membuat jadwal secara otomatis untuk bulan selanjutnya dengan melihat jadwal manual bulan sebelumnya, khususnya bulan November 2017 yang didapatkan dari rumah sakit. Pada kasus ini, algoritma yang dibuat akan menghasilkan jadwal untuk bulan Desember 2017. Untuk menyimpan jadwal bulan November pada Netbeans, digunakan array 2 dimensi yang berisi staff yang ada pada suatu unit dan jumlah hari yang ada pada bulan tersebut. Jadwal yang ingin ditampilkan harus terlebih dahulu disimpan dalam format .csv dan baru akan bisa di import kedalam netbeans. Kemudian untuk menghasilkan jadwal bulan selanjutnya, dilakukan perintah *arraycopy* sehingga jumlah hari untuk bulan selanjutnya dapat bertambah sesuai jumlah hari pada bulan tersebut. Untuk inisiasi, ada beberapa variabel yang perlu dideklarasikan. Variabel-variabel tersebut dapat dilihat seperti gambar dibawah:

```

1. public class Parent {
2.     int tahun = 2017;
3.     int LengthOfMonth[] = {31,28,31,30,31,30,31,31,30,31,30,31};
4.     String Days[] = {"Sn", "S1", "R", "K", "J", "Sb", "M"};
5.     int NumberOfEmployee;
6.     int NumberOfDays = 30;
7.     int FirstDay = 3;
8.     int newMonths = 11;
9.     //int StartingPoint=0;
10.    String roster[][];
11.    String csvFile;
12.
13.        Random rand = new Random();
14.
15.    public Parent(int NumberOfEmployee, String csvFile) {
16.        if(tahun%400 == 0 || tahun%4==0){
17.            LengthOfMonth[1] = 29;
18.        }
19.        this.NumberOfEmployee = NumberOfEmployee;
20.        this.csvFile = csvFile;
21.        this.roster= new String[NumberOfEmployee][NumberOfDays];
22.    }

```

**Kode Program 5.1 Deklarasi Variabel Pada Parent Class**

Variabel tahun untuk menandakan bahwa tahun penjadwalan adalah 2017. Kemudian array *LengthOfMonth* yang berisikan jumlah hari yang ada pada bulan Januari sampai Desember 2017. Disini yang berbeda antar unit adalah variabel *NumberOfEmployee*, karena variabel tersebut digunakan untuk mendeklarasikan jumlah staf pada unit tertentu. Variabel *NumberOfDays* digunakan untuk mendelkarsikan jumlah hari pada bulan November 2017 dan *FirstDay* yang bernilai 3 dengan asumsi hari Rabu memiliki nilai 3 dan Rabu merupakan hari pertama pada bulan November. Dan terakhir ada array *Days* yang berisikan inisial dari nama-nama hari. Setelah semua dideklarasikan, maka tahap selanjutnya adalah menyusun algoritma penjadwalan tiap unitnya dengan melihat dari step-step penjadwalan manual yang telah dibuat.

### 5.3.1 Otomasi Unit Farmasi

Dalam membuat jadwal otomatis, hard constraint perlu diperhatikan agar tidak dilanggar. unit farmasi, algoritma yang dibuat adalah sebagai berikut:

```

1. public class GenerateFarmasi extends Parent{
2.
3.     public GenerateFarmasi(int NumberOfEmployee, String csvFile) {
4.         super(NumberOfEmployee, csvFile);
5.     }
6.
7.     int StartingPoint=0;
8.     int NumbE= 5;
9.     public String [][] initiateFarmasi(){
10.        //STEP 1 bikin baris pertama P
11.        for (int i = NumberOfDays; i <NumberOfDays+LengthOfMonth[newMonths] ; i++)
12.        {
13.            roster[0][i] = "P";
14.            roster[NumberOfEmployee-1][i] = "P";
15.            if((i+FirstDay)%7 == 0){ //minggu
16.                roster[0][i]= "L"; //maka libur
17.                roster[NumberOfEmployee-1][i] = "L";
18.            }
19.        }
20.        //STEP 2 bikin staff no2 spesialis siang
21.        for (int i = NumberOfDays; i <NumberOfDays+LengthOfMonth[newMonths] ; i++)
22.        {
23.            roster[1][i] = "S";
24.            if((i+FirstDay)%7 == 0){
25.                roster[1][i]= "L";
26.            }
27.        }

```

```

27.
28. //STEP 3 memasukkan pola M M LM L P P S S untuk employee no 3
29. String Pattern [] = {"M", "M", "LM", "L", "P", "P", "S", "S", "-1", "-1"};
30. for (int i = 0; i < NumbE; i++) {
31.     for (int j = NumberOfDays; j < NumberOfDays+LengthOfMonth[newMonths]; j
++) {
32.         roster[i+2][j]= Pattern[(j-StartingPoint)%10];
33.     }
34.     StartingPoint += 2;
35. }
36.
37. // STEP 4 : menentukan midle shift
38. String NamaShift[]={ "P", "S", "M", "LM", "L", "CT", "MS"};
39. for (int j = NumberOfDays; j <NumberOfDays+LengthOfMonth[newMonths]; j+
+) {
40.     int JmlShift [] ={0,0,0,0,0,0,0};
41.     for (int i = 0; i <NumberOfEmployee; i++) {
42.         for (int k = 0; k < JmlShift.length; k++) {
43.             if(roster[i][j].equalsIgnoreCase>NamaShift[k]) {
44.                 JmlShift[k] ++;
45.             }
46.         }
47.         if(roster[i][j] == "-1"){
48.             if((JmlShift[3]+JmlShift[4]+JmlShift[5])<= 1){
49.                 roster[i][j]="MS";
50.             }else{
51.                 roster[i][j]="S";
52.             }
53.         }
54.     }
55. }
56. return this.roster;
57. }

```

### Kode Program 5.2 Penjadwalan otomatis unit farmasi

Penjelasan Algoritma :

1. Step pertama ditujukan untuk apoteker farmasi dan petugas obat yang memiliki jadwal selalu pagi dan libur di hari minggu
2. Step kedua ditujukan untuk staf senior yang memiliki jadwal selalu pagi dan libur di hari minggu
3. Step ketiga untuk anggota yang memiliki pola M-M-LM-L-P-P-S-S dengan starting point +2
4. Step terakhir adalah untuk menentukan shift yang akan diisikan di sisa 2 sel yang kosong. Untuk mengisinya, dilihat jumlah tiap shift per harinya. Jika dalam satu hari, jumlah LM L CT  $\leq$  1, maka sel yang kosong diisikan MS. Jika tidak diisi S

```

1. public boolean cekConstraint(String [][] roster){
2.     boolean value1= false;
3.     boolean value2= false;
4.     for (int j = NumberOfDays; j <NumberOfDays+LengthOfMonth[newMonths]; j++) {
5.         String [] Shift ={"P","S","M"};
6.         int [] Jml_Shift = {0,0,0};
7.         for (int i = 0; i < NumberOfEmployee; i++) {
8.             if((j+FirstDay)%7 == 0){
9.                 if(roster[i][j].equalsIgnoreCase("P")){
10.                    Jml_Shift[0]++;
11.                } else if(roster[i][j].equalsIgnoreCase("S")){
12.                    Jml_Shift[1]++;
13.                } else if(roster[i][j].equalsIgnoreCase("M")){
14.                    Jml_Shift[2]++;
15.                }
16.            }
17.            if ((Jml_Shift[0] == 1 && Jml_Shift[1]== 2 )&& Jml_Shift[2]== 1
18.        ){
19.                value1 = true;
20.            }else{
21.                value1 = false;
22.            }
23.        } else{
24.            if(roster[i][j].equalsIgnoreCase("P")){
25.                Jml_Shift[0]++;
26.            } else if(roster[i][j].equalsIgnoreCase("S")){
27.                Jml_Shift[1]++;
28.            } else if(roster[i][j].equalsIgnoreCase("M")){
29.                Jml_Shift[2]++;
30.            }
31.            if (Jml_Shift[0]== 3 && Jml_Shift[1]== 2 && Jml_Shift[2]== 1){
32.                value2 = true;
33.            } else{
34.                value2 = false;
35.            }
36.        }
37.    }
38.
39.    System.out.println("Cek Constraint");
40.    System.out.println("Constraint 1 : " + value1);
41.    System.out.println("Constraint 2 : " + value2);
42.
43.    if (value1=value2) {
44.        value1=true;
45.    } else {
46.        value1=false;
47.    }
48.    return value1;
49. }

```

### Kode Program 5.3 Cek constraints unit farmasi

Sebelum Kode Program 5.2 dijalankan, ada satu method lagi yang perlu dibuat yaitu CekConstraint. Method ini bertujuan untuk mengecek apakah constraint ada pada unit farmasi terpenuhi atau tidak. Variabel value1 digunakan untuk mengecek constraint pada hari Minggu yaitu jumlah shift pagi sama dengan 1, shift siang sama dengan 2, dan shift malam sama dengan 1. Untuk value2, digunakan untuk mengecek jumlah shift pagi tiap harinya harus sama dengan 3, jumlah shift siang tiap harinya harus sama dengan 2, dan jumlah shift malam tiap harinya harus sama dengan 1.

### 5.3.2 Otomasi Unit Bayi/NICU

Untuk membuat jadwal otomatis unit NICU/Bayi, algoritma yang dibuat adalah sebagai berikut:

```

1. public class GenerateBayinicc extends Parent{
2.
3.     public GenerateBayinicc(int NumberOfEmployee, String csvFile) {
4.         super(NumberOfEmployee, csvFile);
5.     }
6.
7.     int StartingPoint=0;
8.     int NumbE= 12;
9.     public String [][] initiateBayinicc(){
10.        //STEP 1 bikin baris pertama P
11.        for (int i = NumberOfDays; i <NumberOfDays+LengthOfMonth[newMonths] ; i++)
12.        {
13.            roster[0][i] = "P";
14.            if((i+FirstDay)%7 == 0){
15.                roster[0][i]= "L";
16.            }
17.        }
18.        //STEP 2 memasukkan pola M M LM L P P S S untuk employee no 2 sampai 13
19.        int StartingPoint=0;
20.        String Pattern []={"M", "M", "LM", "L", "P", "P", "S", "S", "MS", "MS"};
21.        for (int i = 0; i < NumbE; i++) {
22.            for (int j = NumberOfDays; j < NumberOfDays+LengthOfMonth[newMonths]; j
23.            ++) {
24.                roster[i+1][j]= Pattern[(j-StartingPoint)%10];
25.            }
26.            StartingPoint +=2;
27.        }
28.        return this.roster;
29.    }

```

#### Kode Program 5.4 Penjadwalan Otomatis Unit Bayi/NICU

Penjelasan algoritma:

1. Step pertama adalah dengan membuat staf nomor 1 hanya memiliki shift pagi dan shift libur di hari minggu
2. Step selanjutnya adalah mengaplikasikan pola P-P-S-S-Ms-Ms setelah mengisi pola M-M-LM-L dengan starting point +2 untuk anggota staf kedua sampai anggota ke tiga belas

```

1. public boolean cekConstraint(String [][] roster){
2.     boolean value1 = false;
3.     boolean value2 = false;
4.     for (int j = NumberOfDays; j <NumberOfDays+LengthOfMonth[newMonths]; j++) {
5.         String [] Shift ={"P", "S", "M"};
6.         int [] Jml_Shift = {0,0,0};
7.         for (int i = 0; i < NumberOfEmployee; i++) {
8.             if(roster[i][j].equalsIgnoreCase("P")){
9.                 Jml_Shift[0]++;
10.            } else if(roster[i][j].equalsIgnoreCase("S")){
11.                Jml_Shift[1]++;
12.            } else if(roster[i][j].equalsIgnoreCase("M")){
13.                Jml_Shift[2]++;
14.            }
15.            if (((Jml_Shift[0] >= 2) || (Jml_Shift[0] <= 4)) &&
16.                ((Jml_Shift[1] >= 1) || (Jml_Shift[1] <= 2))
17.                &&((Jml_Shift[2] >= 2) || (Jml_Shift[2] <= 3))){
18.                value1 = true;
19.            }else{
20.                value1 = false;
21.            }
22.        }
23.    }
24.    }
25.    System.out.println("Cek Constraint");
26.    System.out.println("Constraint 1 : " + value1);
27.    return value1;
28. }

```

### Kode Program 5.5 Cek constraint unit Bayi/NICU

Sebelum Kode Program 5.4 dijalankan, ada satu method lagi yang perlu dibuat yaitu CekConstraint. Method ini bertujuan untuk mengecek apakah batasan-batasan yang ada pada unit bayi/NICU terpenuhi atau tidak. Variabel value1 digunakan untuk mengecek constraint di setiap harinya yaitu jumlah shift pagi setiap hari harus lebih dari sama dengan 2 dan kurang dari sama dengan 4, jumlah shift siang setiap hari harus lebih dari sama dengan 1 dan kurang dari sama dengan 2, dan jumlah shift malam setiap hari harus lebih dari sama dengan 2 dan kurang dari sama dengan 3.

#### 5.3.3 Otomasi Penjadwalan Gizi

Untuk membuat jadwal otomatis unit gizi, algoritma yang dibuat adalah sebagai berikut:

```

1. public class Gizi extends New{
2.
3.     public Gizi(int NumberOfEmployee, String csvFile) {
4.         super(NumberOfEmployee, csvFile);
5.     }
6.
7.     public String [][] GenerateGizi(){
8.         //STEP 1 bikin employee pertama P
9.         for (int i = NumberOfDays; i <NumberOfDays+LengthOfMonth[newMonths] ; i++)
10.        {
11.            roster[0][i] = "P";
12.            roster[NumberOfEmployee-1][i] = "P";
13.            if((i+FirstDay)%7 == 0){ //minggu
14.                roster[0][i]= "L"; //maka libur
15.                roster[NumberOfEmployee-1][i] = "L";
16.            }
17.        }
18.
19.        //STEP 2 memasukkan pola P1,P,S,P1,P,L,S,S,MS,MS untuk employee no 2-5
20.        int StartingPointChef=0;
21.        int NumbChef= 4;
22.        String PatternChef [] = {"P1","P","S","P1","P","L","S","S","MD","MD"};
23.        for (int i = 0; i < NumbChef; i++) {
24.            for (int j = NumberOfDays; j < NumberOfDays+LengthOfMonth[newMonths]; j
25.            ++){
26.                roster[i+1][j]= PatternChef[(j-StartingPointChef)%10];
27.            }
28.            StartingPointChef += 2;
29.        }
30.
31.        //STEP 3 memasukkan pola L P P S S S P P untuk employee no 6 (Riza Johan)
32.        String Pattern7 [] = {"L","P1","P1","S","S","S","P1","P1"};
33.        for (int j = NumberOfDays; j < NumberOfDays+LengthOfMonth[newMonths]; j
34.        ++){
35.            roster[5][j]= Pattern7[(j+5)%8];
36.        }
37.
38.        //STEP 4 memasukkan pola untuk employee no 7-9
39.        int StartingPointHelper=0;
40.        int NumbHelper = 3;
41.        String PatternHelper [] = {"P2","P","S","S","L","P","S","P2","P","MS"};
42.        for (int i = 0; i < NumbHelper; i++) {
43.            for (int j = NumberOfDays; j < NumberOfDays+LengthOfMonth[newMonths]; j
44.            ++){
45.                roster[i+6][j]= PatternHelper[(j-StartingPointHelper)%10];
46.            }
47.            StartingPointHelper += 2;
48.        }
49.
50.        //STEP 5 memasukkan pola L P P S S S P P untuk employee no 10
51.        String Pattern10 [] = {"S","S","P","P","MD","MD","L"};
52.        for (int j = NumberOfDays; j < NumberOfDays+LengthOfMonth[newMonths]; j
53.        ++){
54.            roster[9][j]= Pattern10[(j+5)%7];
55.        }
56.
57.        //STEP 6 memasukkan pola M M LM L P P S S untuk employee no 11-15
58.        int StartingPointPenyaji=0;
59.        int NumbE= 5;
60.        String Pattern [] = {"M","M","LM","L","P","P","S","S","-1","-1"};
61.        for (int i = 0; i < NumbE; i++) {
62.            for (int j = NumberOfDays; j < NumberOfDays+LengthOfMonth[newMonths]; j
63.            ++){
64.                roster[i+10][j]= Pattern[(j-StartingPointPenyaji)%10];
65.            }
66.            StartingPointPenyaji += 2;

```

```

63.
64. //STEP 7 Membuat jadwal untuk staff nomor 16 dan 19
65. for (int i = NumberOfDays; i <NumberOfDays+LengthOfMonth[newMonths] ; i++) {
66.     String Pattern1619 [] = {"PS", "S", "S", "P", "P", "P", "PS", "L", "P", "P", "S", "S",
        "S", "L"};
67.     for (int j = NumberOfDays; j < NumberOfDays+LengthOfMonth[newMonths]; j++)
        {
68.         roster[15][j]= Pattern1619[(j+9)%14];
69.         roster[18][j]= Pattern1619[(j+9)%14];
70.     }
71. }
72.
73. //STEP 8 Membuat jadwal untuk staff nomor 17 dan 18
74. for (int i = NumberOfDays; i <NumberOfDays+LengthOfMonth[newMonths] ; i++) {
75.     String Pattern1718 [] = {"L", "P", "P", "S", "S", "S", "L", "PS", "S", "S", "P", "P", "
        P", "PS"};
76.     for (int j = NumberOfDays; j < NumberOfDays+LengthOfMonth[newMonths]; j++)
        {
77.         roster[16][j]= Pattern1718[(j+9)%14];
78.         roster[17][j]= Pattern1718[(j+9)%14];
79.     }
80. }
81.
82. // STEP 9 : menentukan midle shift
83. String NamaShift[]={ "P", "S", "M", "LM", "L", "CT", "MD"};
84. for (int j = NumberOfDays; j <NumberOfDays+LengthOfMonth[newMonths]; j++) {
85.
86.     int JmlShift [] ={0,0,0,0,0,0,0};
87.     for (int i = 0 ; i <NumberOfEmployee; i++) {
88.         for (int k = 0; k < JmlShift.length; k++) {
89.             if(roster[i][j].equalsIgnoreCase(NamaShift[k])) {
90.                 JmlShift[k] ++;
91.             }
92.             if(roster[i][j] == "-1"){
93.                 if(JmlShift[6] == 0){
94.                     roster[i][j]="MD";
95.                 }else{
96.                     roster[i][j]="P";
97.                 }
98.             }
99.         }
100.     }

```

### Kode Program 5.6 Penjadwalan Otomatis Unit Gizi

Penjelasan algoritma:

1. Step pertama untuk Ahli Gizi selalu memiliki shift pagi dan libur setiap hari Minggu.
2. Step kedua adalah mengisi pola P1-P-S-P1-P-L-S-S-MS-MS dengan mengisi jarak 10 kotak dengan starting point +2 untuk staf nomor 2 sampai 5 (Chef),
3. Untuk staf nomer 6 isikan pola L-P-P-S-S-S-P-P dengan jarak antar pola 7

4. Untuk Staf nomor 7-9 (Helper), isikan pola P2-P-S-S-L-P-S-P2-P-MS dengan mengisi jarak 10 kotak dengan starting point +2
5. Staff nomor 10 isikan pola S-S-P-P-MD-MD-L dengan jarak 7
6. Isikan pola M-M-LM-L-P-P-S-S melanjutkan pola bulan sebelumnya untuk staff nomor 11-15
7. Untuk staff 16 dan 19 memiliki pola PS-S-S-P-P-P-PS-L-P-P-S-S-S-L
8. Untuk staff 17 dan 18 memiliki pola L-P-P-S-S-S-L-PS-S-S-P-P-P-PS
9. Untuk staf 10 sampai 15 jika sel masih kosong, dan jika pada kolom terdapat MD, maka isikan P pada sel yang kosong. Jika tidak ada MD maka isi MD

```

1. public boolean cekConstraint(String [][] roster){
2.     boolean value1= false;
3.     boolean value2= false;
4.     for (int j = NumberOfDays; j <NumberOfDays+LengthOfMonth[newMonths]; j++) {
5.         String [] Shift ={"PS","M"};
6.         int [] Jml_Shift = {0,0,0};
7.         for (int i = 0; i < NumberOfEmployee; i++) {
8.             if((j+FirstDay)%7 <= 1){
9.                 if(roster[i][j].equalsIgnoreCase("PS")){
10.                    Jml_Shift[0]++;
11.                }
12.                if (Jml_Shift[0] == 2){
13.                    value1 = true;
14.                }else{
15.                    value1 = false;
16.                }
17.            }else{
18.                if(roster[i][j].equalsIgnoreCase("PS")){
19.                    Jml_Shift[0]++;
20.                } else if(roster[i][j].equalsIgnoreCase("M")){
21.                    Jml_Shift[1]++;
22.                }
23.                if (Jml_Shift[1]== 1){
24.                    value2 = true;
25.                } else{
26.                    value2 = false;
27.                }
28.            }
29.        }
30.    }
31.
32.    System.out.println("Cek Constraint");
33.    System.out.println("Constraint 1 : " + value1);
34.    System.out.println("Constraint 2 : " + value2);
35.
36.    if (value1=value2) {
37.        value1=true;
38.    } else {
39.        value1=false;
40.    }
41.    return value1;

```

**Kode Program 5.7 Cek constraint unit Gizi**

Sebelum Kode Program 5.6 dijalankan, ada satu method lagi yang perlu dibuat yaitu CekConstraint. Method ini bertujuan untuk mengecek apakah batasan-batasan yang ada pada unit gizi terpenuhi atau tidak. Variabel value1 digunakan untuk mengecek untuk staf cafe dan driver, jumlah shift pagi sore (PS) harus sama dengan 2 di hari Minggu dan Senin. Value2 digunakan untuk penyaji yaitu jumlah shift malam selalu sama dengan 1 tiap harinya

### 5.3.4 Otomasi Unit IGD

Untuk membuat jadwal otomatis unit IGD, algoritma yang dibuat adalah sebagai berikut:

```

1. public class GenerateIGD extends Parent{
2.
3.     public GenerateIGD (int NumberOfEmployee, String csvFile) {
4.         super(NumberOfEmployee, csvFile);
5.     }
6.
7.     int StartingPoint=0;
8.     public String [][] initiateIGD(){
9.
10.        //STEP 1 memasukkan pola M M LM L P P S S
11.        int StartingPoint=0;
12.        //int NumbE= 8;
13.        String Pattern []={"M", "M", "LM", "L", "P", "P", "S", "S", "S", "S", "S", "L", "MD",
14.        "MS", "S", "S", "S"};
15.        for (int i = 0; i < NumberOfEmployee; i++) {
16.            for (int j = NumberOfDays; j < NumberOfDays+LengthOfMonth[newMonths]; j
17.            ++) {
18.                roster[i][j]= Pattern[(j-StartingPoint)%15];
19.            }
20.            StartingPoint +=2;
21.        }
22.        return this.roster;
23.    }

```

#### Kode Program 5.8 Penjadwalan Otomatis Unit IGD

Penjelasan algoritma:

1. Pada unit IGD, hanya ada satu step karena semua staf memiliki pola yang sama yaitu M-M-LM-L-P-P-S-S-S-S-L-MS-MD-S-S-S dengan starting point +2

```

1. public boolean cekConstraint(String [][] roster){
2.     boolean value1 = false;
3.     boolean value2 = false;
4.     for (int j = NumberOfDays; j <NumberOfDays+LengthOfMonth[newMonths]; j++) {
5.         String [] Shift ={"P", "S", "M"};
6.         int [] Jml_Shift = {0,0,0};
7.         for (int i = 0; i < NumberOfEmployee; i++) {
8.             if(roster[i][j].equalsIgnoreCase("P")){
9.                 Jml_Shift[0]++;
10.            } else if(roster[i][j].equalsIgnoreCase("S")){
11.                Jml_Shift[1]++;
12.            } else if(roster[i][j].equalsIgnoreCase("M")){
13.                Jml_Shift[2]++;
14.            }
15.        }
16.
17.        if (((Jml_Shift[0] >= 1)|| (Jml_Shift[0] <= 2)) &&
18.            ((Jml_Shift[1] >= 2) || (Jml_Shift[1] <= 4)) &&
19.            ((Jml_Shift[2] >= 1)|| (Jml_Shift[2] <= 2))){
20.            value1 = true;
21.        }else{
22.            value1 = false;
23.        }
24.    }
25.
26. }
27. System.out.println("Cek Constraint");
28. System.out.println("Constraint 1 : " + value1);
29. return value1;

```

### Kode Program 5.9 Cek constraint unit IGD

Sebelum Kode Program 5.8 dijalankan, ada satu method lagi yang perlu dibuat yaitu CekConstraint. Method ini bertujuan untuk mengecek apakah batasan-batasan yang ada pada unit IGD terpenuhi atau tidak. Variabel value1 digunakan untuk mengecek batasan jumlah shift pagi tiap harinya lebih dari sama dengan 1 atau kurang dari sama dengan 2, jumlah shift siang setiap hari harus lebih dari sama dengan 2 dan kurang dari sama dengan 4, dan jumlah shift malam setiap hari harus berjumlah 1 atau 2.

#### 5.3.5 Otomasi Unit Kamar Operasi

Untuk membuat jadwal otomatis unit Kamar Operasi, algoritma yang dibuat adalah sebagai berikut:

```

1. public class GenerateOperasi extends Parent{
2.
3.     public GenerateOperasi(int NumberOfEmployee, String csvFile) {
4.         super(NumberOfEmployee, csvFile);
5.     }
6.
7.     int StartingPoint=0;
8.     public String [][] initiateOperasi(){
9.
10.        //STEP 1 mengisi Pola berdderet
11.        int StartingPoint=0;
12.        String Pattern [{"P","P","S","P","S","L","P","S","-1"}];
13.        for (int i = 0; i < NumberOfEmployee; i++) {
14.            for (int j = NumberOfDays; j < NumberOfDays+LengthOfMonth[newMonths]; j
15.        ++) {
16.                roster[i][j]= Pattern[(j-StartingPoint)%10];
17.                StartingPoint ++;
18.            }
19.
20.            //Step 2 Mengisi Kolom kosong
21.            String NamaShift[]="S","P","L","CT";
22.            for (int j = NumberOfDays; j <NumberOfDays+LengthOfMonth[newMonths]; j+
23.        +) {
24.                int JmlShift [] ={0,0,0,0};
25.                for (int i = 0; i <NumberOfEmployee; i++) {
26.                    for (int k = 0; k < JmlShift.Length; k++) {
27.                        if(roster[i][j].equalsIgnoreCase>NamaShift[k]) {
28.                            JmlShift[k] ++;
29.                        }
30.                    }
31.                    if(roster[i][j]== "-1"){
32.                        if(JmlShift[0]<1){
33.                            roster[i][j]= "S";
34.                        }else{
35.                            roster[i][j]="P";
36.                        }
37.                    }
38.                }
39.                //STEP 3 HARI MINGGU P WAJIB 4 DAN L HARUS 2
40.                int StartingPointLibur=0;
41.                String pola2 [{"L","L","P","P","P","P"}];
42.                for (int j = NumberOfDays; j < NumberOfDays+LengthOfMonth[newMonths]; j++
43.        ) {
44.                    for (int k =0; k <pola2.length; k++) {
45.                        if((j+FirstDay)%7 == 0 ){
46.                            roster[k][j]=pola2[(k+StartingPointLibur+2)%6];
47.                        }
48.                    }StartingPointLibur +=2;
49.                }
50.            }
51.        }
52.        return this.roster;
53.    }
54. }

```

### Kode Program 5.10 Penjadwalan Otomatis Unit Kamar Operasi

Penjelasan algoritma:

1. Step pertama adalah mengisi Pola P-P-S-P-S-L-P-P-S-(-1) untuk setiap staf dengan roling turun satu kotak (menggunakan perulangan kolom didalam baris)
2. Selanjutnya adalah mengisi kolom kosong dengan spesifikasi siang sama dengan 2, dan p kurang dari sama dengan 3 dengan langkah menjumlahkan isi kolom, kemudian membuat kondisi yang sesuai

- Setiap hari minggu shift P harus berjumlah 4 dan L berjumlah 2 (isi Kolom L-L-P-P-P-P secara berulang ke bawah turun 2 kotak )

```

1. public boolean cekConstraint(String [][] roster){
2.     boolean value1 = false;
3.     boolean value2 = false;
4.
5.     for (int j = NumberOfDays; j <NumberOfDays+LengthOfMonth[newMonths]; j++) {
6.         String [] Shift ={"P", "S"};
7.         int [] Jml_Shift = {0,0,0};
8.         for (int i = 0; i < NumberOfEmployee; i++) {
9.             if((j+FirstDay)%7 == 0){
10.                if(roster[i][j].equalsIgnoreCase("P")){
11.                    Jml_Shift[0]++;
12.                } else if(roster[i][j].equalsIgnoreCase("S")){
13.                    Jml_Shift[1]++;
14.                }
15.
16.                if (Jml_Shift[0] == 4 && Jml_Shift[1] == 0){
17.                    value1 = true;
18.                } else{
19.                    value1 = false;
20.                }
21.
22.            } else{
23.                if(roster[i][j].equalsIgnoreCase("P")){
24.                    Jml_Shift[0]++;
25.                } else if(roster[i][j].equalsIgnoreCase("S")){
26.                    Jml_Shift[1]++;
27.                }
28.
29.                if (((Jml_Shift[0] >= 2) || (Jml_Shift[0] <= 4)) &&
30.                    ((Jml_Shift[1] >= 2) || (Jml_Shift[1] <= 3))){
31.                    value2 = true;
32.                }else{
33.                    value2 = false;
34.                }
35.            }
36.        }
37.    }
38.    System.out.println("Cek Constraint");
39.    System.out.println("Constraint 1 : " + value1);
40.    System.out.println("Constraint 2 : " + value2);
41.
42.    if (value1=value2) {
43.        value1=true;
44.    } else {
45.        value1=false;
46.    }
47.    return value1;

```

### Kode Program 5.11 Cek constraint unit Kamar Operasi

Sebelum Kode Program 5.10 dijalankan, ada satu method lagi yang perlu dibuat yaitu CekConstraint. Method ini bertujuan untuk mengecek apakah batasan-batasan yang ada pada unit Kamar Operasi terpenuhi atau tidak. Variabel value1 digunakan untuk mengecek batasan yaitu jumlah shift pagi pada hari Minggu harus terdiri dari empat. Value2 digunakan untuk batasan jumlah shift pagi setiap hari harus lebih dari sama dengan 2 dan kurang dari sama dengan 4 dan jumlah shift siang

setiap hari harus lebih dari sama dengan 2 atau kurang dari sama dengan 3.

### 5.3.6 Otomasi Unit SIM & RM

Untuk membuat jadwal otomatis unit SIM & RM, algoritma yang dibuat adalah sebagai berikut:

```

1. public class GenerateRMRegis extends Parent{
2.
3.     public GenerateRMRegis (int NumberOfEmployee, String csvFile) {
4.         super(NumberOfEmployee, csvFile);
5.     }
6.
7.     int StartingPoint=0;
8.     //int Numbe= 8;
9.     public String [][] initiateRMRegis(){
10.
11.         //STEP 1 memasukkan pola M M LM L P P P L S S S untuk semua employee
12.         int StartingPoint=0;
13.         String Pattern []={ "M", "M", "LM", "L", "P", "P", "P", "L", "S", "S", "S", "-1" };
14.         for (int i = 0; i < NumberOfEmployee; i++) {
15.             for (int j = NumberOfDays; j < NumberOfDays+LengthOfMonth[newMonths]; j
16.             ++) {
17.                 roster[i][j]= Pattern[(j-StartingPoint)%12];
18.             }
19.             StartingPoint +=2;
20.         }
21.
22.         // STEP 2 : menentukan middle shift
23.         String NamaShift[]={ "P", "S", "M", "LM", "L", "CT", "MS" };
24.         for (int j = NumberOfDays; j <NumberOfDays+LengthOfMonth[newMonths]; j+
25.         +) {
26.             int JmlShift [] = {0,0,0,0,0,0};
27.             for (int i = 0; i < NumberOfEmployee; i++) {
28.                 for (int k = 0; k < JmlShift.length; k++) {
29.                     if(roster[i][j].equalsIgnoreCase(NamaShift[k])) {
30.                         JmlShift[k] ++;
31.                     }
32.                 }
33.                 if(roster[i][j]== "-1"){
34.                     if((JmlShift[0]+JmlShift[1]+JmlShift[2]) <= 3){
35.                         roster[i][j]="MS";
36.                     }else {
37.                         roster[i][j]="P";
38.                         // roster[i][j]="S";
39.                     }
40.                 }
41.             }
42.         }
43.         return this.roster;
44.     }
45. }

```

#### Kode Program 5.12 Penjadwalan Otomatis Unit SIM & RM

Penjelasan algoritma:

1. Pada unit SIM & RM juga hanya memiliki pola M-M-LM-L diikuti dengan pola P-P-P-L-S-S-S(-1) di bagian belakangnya untuk semua staff
2. Kemudian adalah mengisi pola yang kosong dengan ketentuan apabila dalam satu hari jumlah P, S, dan M-

nya kurang dari sama dengan 3 maka diisi dengan “M”. Apabila lebih dari 3 maka diisi P

```

1. public boolean cekConstraint(String [][] roster){
2.     boolean value1 = false;
3.     boolean value2 = false;
4.
5.     for (int j = NumberOfDays; j <NumberOfDays+LengthOfMonth[newMonths]; j++)
6.     ) {
7.         String [] Shift ={"P","S","M"};
8.         int [] Jml_Shift = {0,0,0};
9.         for (int i = 0; i < NumberOfEmployee; i++) {
10.            if(roster[i][j].equalsIgnoreCase("P")){
11.                Jml_Shift[0]++;
12.            } else if(roster[i][j].equalsIgnoreCase("S")){
13.                Jml_Shift[1]++;
14.            }else if(roster[i][j].equalsIgnoreCase("M")){
15.                Jml_Shift[2]++;
16.            }
17.
18.            if (((Jml_Shift[0] >= 1) || (Jml_Shift[0] <= 2)) &&
19.                ((Jml_Shift[1] >= 1) || (Jml_Shift[1] <= 2))
20.                && (Jml_Shift[2] == 1)){
21.                value1 = true;
22.            }else{
23.                value1 = false;
24.            }
25.        }
26.        System.out.println("Cek Constraint");
27.        System.out.println("Constraint 1 : " + value1);
28.
29.    return value1;

```

### Kode Program 5.13 Cek constraint unit SIM &RM

Sebelum Kode Program 5.12 dijalankan, ada satu method lagi yang perlu dibuat yaitu CekConstraint. Method ini bertujuan untuk mengecek apakah batasan-batasan yang ada pada unit SIM & RM terpenuhi atau tidak. Variabel value1 digunakan untuk mengecek batasan jumlah shift pagi tiap harinya lebih dari sama dengan 1 atau kurang dari sama dengan 2, jumlah shift siang tiap harinya lebih dari sama dengan 1 atau kurang dari sama dengan 2, dan jumlah shift malam tiap harinya harus sama dengan 1.

## 5.4 Algoritma Optimasi Penjadwalan

Optimasi dilakukan dengan menerapkan algoritma Reinforcement Learning. Algoritma tersebut diterapkan dengan tujuan untuk memilih *low-level heuristic* yang memiliki solusi paling baik. Solusi tersebut merupakan nilai dari *Jain Fairness Index (JFI)*. Semakin nilai JFI mendekati 1, maka solusi yang dihasilkan juga berarti semakin bagus. Solusi baru tersebut

nantinya akan dibandingkan dengan solusi awal sebelum dilakukan optimasi.

Disini, kami menggunakan dua metode *low-level heuristic*, yaitu *Move* dan *Swap*. Pada algoritma Reinforcement Learning, algoritma yang dibuat bertujuan untuk memilih *low-level heuristic* yang terbaik. Caranya adalah dengan memberikan score untuk masing-masing *move* dan *swap*. Kemudian jika salah satu dieksekusi dan menghasilkan solusi yang lebih baik dari solusi sebelumnya, maka score dari *low-level heuristic* tersebut akan bertambah 1. Begitu seterusnya sampai pada iterasi maksimal mendapatkan solusi terbaik.

#### 5.4.1 Jain Fairness Index

Untuk mengetahui tingkat optimasi jadwal, perlu dihitung dengan menggunakan Jain Fairness Index, dengan kode program sebagai berikut:

```

1. public double JFI(String[][] roster){
2.     String libur[]={ "Hari Minggu", "Hari Sabtu", "Hari Kerja"};
3.     double jfi = 0;
4.     double JmlhStafKuadrat = 0;
5.     double JmlSUMStaf = 0;
6.     double JmlSUMStafKuadrat = 0;
7.     for (int i = 0 ; i <NumberOfEmployee; i++) {
8.         int a=0;
9.         int JmlLibur[]= {0,0,0};
10.        int Bobot[]={4,2,1};
11.        int liburXbobot = 0;
12.        int Jml_liburXbobot =0;
13.        double StafKuadrat = 0;
14.        for (int j = NumberOfDays; j <NumberOfDays+LengthOfMonth[newMonths]; j
++ ) {
15.            if(roster[i][j].equalsIgnoreCase("L")){
16.                if((j+FirstDay)%7 == 0){
17.                    JmlLibur[0]++;
18.                }else if ((j+FirstDay)%7 == 6){
19.                    JmlLibur[1]++;
20.                }else{
21.                    JmlLibur[2]++;
22.                }
23.            }
24.        }
25.        for (int k = 0; k < JmlLibur.length; k++) {
26.            liburXbobot = JmlLibur[k]*Bobot[k];
27.            //System.out.printf("%15s:%5d",Libur[k],liburXbobot);
28.            //jumlah dengan pembobotan
29.            Jml_liburXbobot +=liburXbobot; // hitung per staf
30.        }
31.        //System.out.printf(" Sum = "+ Jml_liburXbobot);
32.        StafKuadrat = Math.pow(Jml_liburXbobot, 2); //Total SUM kuadrat ma
sing-masing staf (bawah)
33.        JmlhStafKuadrat += StafKuadrat; //Menghitung total staff kuadrat se
mua staf
34.        JmlSUMStaf += Jml_liburXbobot; //Menghitung total SUM semua staf
35.        //System.out.printf("");
36.        //System.out.println(" ");
37.    }
38.    JmlSUMStafKuadrat = Math.pow(JmlSUMStaf, 2); //Total SUM semua staf dikuadr
atkan (atas)
39.    jfi=(JmlSUMStafKuadrat /(NumberOfEmployee*JmlhStafKuadrat));
40.    //System.out.println("JFI: " + jfi );
41.    return jfi;
42. }

```

**Kode Program 5.14 Perhitungan Jain Fairness Index**

*Jain Fairness Index* (JFI) merupakan sebuah metode yang digunakan untuk menangani masalah keadilan dengan menggunakan pendekatan kuantitatif. Pada algoritma diatas, perhitungan dilakukan dengan memberikan bobot pada tiap kategori hari libur dengan deklarasi dari variabel array Libur dan variabel array bobot. Jika staf mendapatkan libur di hari minggu, maka akan mendapatkan bobot 4. Jika libur di hari sabtu mendapatkan bobot 2, dan jika libur di hari kerja akan mendapatkan bobot 1. Kemudian dilakukan looping dengan melihat sel yang berisikan L dan kapan staf tersebut mendapatkan L. Lalu barulah akan dilakukan perhitungan. Secara sederhana, JFI hanya terdiri dari 2 operasi hitung yaitu SUM dan kuadrat. Pertama-tama, hitung hari libur dengan dikalikan dengan bobotnya untuk masing-masing staff. Kemudian hitung total SUM kuadrat masing-masing staf (StafKuadrat), total staff kuadrat semua staf (JmlhStafkuadrat), total SUM semua staf (JmlSUMStaf), dan total SUM semua staf dikuadratkan (JmlSUMStafKuadrat). Terakhir, baru hitung sesuai rumus JFI.

#### 5.4.2 Low Level Heuristic

*Hyper heuristic* dapat dikatakan sebagai *heuristic* tingkat tinggi untuk memilih atau menghasilkan *low-level heuristic*. *Hyper-heuristic* yang digunakan disini adalah *reinforcement learning*. Sedangkan *low-level heuristic* sendiri merupakan beberapa aturan sederhana untuk memilih komponen solusi atau fungsi lingkungan untuk memodifikasi solusi. Karena pada tugas akhir ini ditujukan untuk menyelesaikan masalah penjadwalan, maka ada beberapa *low-level heuristic* yang sebenarnya bisa digunakan seperti *insertion*, *inversion*, *swap*, *move*, dan lain-lain. Disini penulis hanya menggunakan dua *low-level heuristic* yaitu *move* dan *swap*.

##### 5.4.2.1 Move

*Move* dilakukan dengan cara melihat jadwal per staf. Jika jumlah libur staf tersebut belum sama dengan jumlah hari Minggu pada bulan itu, dan jika staf tersebut memiliki shift MS,

maka shift MS akan diganti menjadi L. Kode programnya dapat dilihat sebagai berikut:

```

1. public void Move(){
2.     int JmlLibur;
3.     int selisih;
4.     int JumlahLibur_seharusnya = (LengthOfMonth[newMonths]/7);
5.     Random rand = new Random();
6.     if(((NumberOfDays + (FirstDay - 1)) % 7 ) >= 4){
7.         JumlahLibur_seharusnya ++;
8.     }
9.     for (int i = 0 ; i <NumberOfEmployee; i++) {
10.        JmlLibur=0;
11.        selisih=0;
12.        List<Integer> posisiMS = new ArrayList<>();
13.        for (int j = NumberOfDays; j <NumberOfDays+LengthOfMonth[newMonths]
; j++) {
14.            if(roster[i][j].equalsIgnoreCase("L")){
15.                JmlLibur++;
16.            }
17.            if (roster [i][j].equalsIgnoreCase("MS")){
18.                posisiMS.add(new Integer(j));
19.            }
20.        }
21.        selisih = JumlahLibur_seharusnya-JmlLibur;
22.        if (selisih >0 && posisiMS.size()>0){
23.            for (int k = 0; k < selisih; k++) {
24.                int IndexMSygDiganti = rand.nextInt(posisiMS.size());
25.                roster[i][posisiMS.get(IndexMSygDiganti)]= "L";
26.            }
27.        }
28.    }
29. }

```

**Kode Program 5.15 Low Level Heuristic Move**

#### 5.4.2.2 Swap

*Swap* awalnya dilakukan dengan mencari sel yang mengandung shift L pada roster[ ][ ]. Kemudian menyimpan posisiLHari dan posisiLEmployee sel tersebut. Selanjutnya adalah mencari secara random staf lain pada hari yang sama yang akan menggantikan shift L staf pertama. Terakhir baru lah akan ditukar shift L staf pertama dengan shift yang ada pada staf kedua.

```

1. public void Swap(){
2.     Random rand = new Random();
3.     List<Integer> posisiHari = new ArrayList<>();
4.     List<Integer> posisiEmployee = new ArrayList<>();
5.     for (int j = NumberOfDays; j <NumberOfDays+LengthOfMonth[newMonths]; j++) {
6.         for (int i = 0 ; i <NumberOfEmployee; i++) {
7.             if(roster[i][j].equalsIgnoreCase("L")){
8.                 posisiHari.add(new Integer(j));
9.                 posisiEmployee.add(new Integer(i));
10.            }
11.        }
12.    }
13.    int memilihHari = rand.nextInt(posisiHari.size());
14.    String temp = roster[posisiEmployee.get(memilihHari)][posisiHari.get(memilihHari)];
15.    int memilihEmployee;
16.
17.    do {
18.        memilihEmployee = rand.nextInt(NumberOfEmployee);
19.    }while(memilihEmployee == posisiEmployee.get(memilihHari));
20.    roster[posisiEmployee.get(memilihHari)][posisiHari.get(memilihHari)]
21.        = roster[posisiEmployee.get(memilihEmployee)][posisiHari.get(memilihHari)];
22.    roster[posisiEmployee.get(memilihEmployee)][posisiHari.get(memilihHari)] =
23.    temp;
    }

```

### Kode Program 5.16 Low Level Heuristic Swap

#### 5.4.3 Reinforcement Learning

Pada algoritma Reinforcement Learning, awalnya ditentukan terlebih dahulu *stopping condition* dari solusi baru yang akan dihasilkan. *Stopping condition* disini berupa jumlah maksimal iterasi yang nantinya akan dilakukan. Langkah selanjutnya adalah menentukan *low-level heuristic*, yaitu *move dan swap*. Untuk menyimpan score yang nantinya akan ditambahkan, maka perlu dibuatkan sebuah array. Setelah itu, perlu diinisiasikan solusi awal yang sudah dihasilkan untuk nantinya akan dibandingkan dengan solusi setelah dilakukan *move* atau *swap*. Kemudian disini juga harus ditentukan score awal dari *move dan swap*, batas bawah dari masing-masing kategori tersebut, dan juga batas atasnya. Setelah itu barulah akan dilakukan iterasi. Karena keduanya memiliki score awal 10, maka untuk iterasi pertama akan dipilih secara random. Misalkan yang pertama terpilih adalah *move*. Jika solusi yang dihasilkan lebih baik dari pada solusi awal, maka score dari *move* akan bertambah 1 menjadi 11. Kemudian lanjut ke iterasi kedua. Untuk setiap iterasi, cara menentukan apakah *move* atau *swap* yang akan dipilih adalah dengan melihat score terbaru dari keduanya. Karena sebelumnya *move* sudah bernilai 11,

maka untuk iterasi selanjutnya move kembali dipilih dan dilihat kembali apakah solusi yang dihasilkan lebih baik atau lebih buruk dari solusi awal. Seandainya solusi yang dihasilkan sekarang lebih buruk, maka score dari move akan dikurangi 1 dan kembali menjadi 10. Tetapi jika ada yang sudah mencapai batas bawah yang sudah ditentukan sebelumnya yaitu 0, maka score sudah tidak dapat berkurang. Begitu juga sebaliknya, jika ada yang sudah mencapai batas atas yang sudah ditentukan sebelumnya yaitu 10, maka score sudah tidak dapat bertambah. Begitu seterusnya sampai iterasi maksimal tercapai. Saat iterasi maksimal sudah tercapai, pilih mana yang akan dipilih dengan melihat score yang paling tinggi. Untuk kode program dari algoritma Reinforcement Learning dapat dilihat sebagai berikut:

```

1. int Score[] = {10,10};
2.     int ScoreMove = Score[0];
3.     int ScoreSwap = Score[1];
4.     public void Reinforcement(int NumIt){
5.         String [][] bestSolution = new String [NumberOfEmployee][LengthOfMonth[newMonth
s]];
6.         bestJFICost = curJFICost = JFI(roster);
7.         //curScoreSwap = ScoreSwap = curScoreMove = ScoreMove;
8.         for (int i = 0; i < NumIt; i++) {
9.             System.out.printf("%2d, %5f, %5f \n", i,curJFICost, bestJFICost );
10.            String [][] Solpertama = roster.clone();
11.                if (ScoreSwap==ScoreMove) {
12.                    if (Math.random() <0.6) {
13.                        Move();
14.                        curJFICost = JFI(roster);
15.                        if (curJFICost > bestJFICost){
16.                            if(ScoreMove<20){
17.                                ScoreMove++;
18.                            } else {
19.                                ScoreMove=20;
20.                            }
21.                        } else{
22.                            if(ScoreMove>0){
23.                                ScoreMove--;
24.                            } else {
25.                                ScoreMove=0;
26.                            }
27.                        }
28.                    }
29.                }else{
30.                    Swap();
31.                    curJFICost = JFI(roster);
32.                    if (curJFICost > bestJFICost){
33.                        if(ScoreSwap<20) {
34.                            ScoreSwap++;
35.                        } else {
36.                            ScoreSwap=20;
37.                        }

```

```

38.         }else{
39.             if(ScoreSwap>0) {
40.                 ScoreSwap--;
41.             } else {
42.                 ScoreSwap=0;
43.             }
44.         }
45.     }
46. }
47.
48.     else if (ScoreSwap > ScoreMove) {
49.         Swap();
50.         curJFIcost = JFI(roster);
51.         if (curJFIcost > bestJFIcost){
52.             if(ScoreSwap<20){
53.                 ScoreSwap++;
54.             } else {
55.                 ScoreSwap=20;
56.             }
57.         }else{
58.             ScoreSwap--;
59.         }
60.     }else {
61.         Move();
62.         curJFIcost= JFI(roster);
63.         if (curJFIcost > bestJFIcost){
64.
65.             if(ScoreMove<20){
66.                 ScoreMove++;
67.             } else {
68.                 ScoreMove=20;
69.             }
70.         } else{
71.             if(ScoreMove>0){
72.                 ScoreMove--;
73.             } else {
74.                 ScoreMove=0;
75.             }
76.         }
77.     }
78.
79.     //System.out.println("Nilai Move : "+ScoreMove+", Nilai Swap : "+ScoreSwap);
80.     curJFIcost= JFI(roster);
81.     if (curJFIcost > bestJFIcost) {
82.         bestJFIcost = curJFIcost;
83.     } else{
84.         roster = Solpertama.clone();
85.     }
86. }
87. System.out.println("BestJFI : " + bestJFIcost );
88. }
89. }

```

### Kode Program 5.17 Algoritma Reinforcement Learning

#### 5.4.4 Hill Climbing

Algoritma *Hill Climbing* yang diterapkan disini cukup sederhana. Awalnya adalah mendeklarasikan variabel yang

nantinya akan digunakan untuk menyimpan JFI yaitu `curJFICost` dan `bestJFICost`. Sebelum melakukan iterasi, nilai `bestJFICost` dianggap sama dengan `curJFICost`. Kemudian akan dilakukan pemilihan *low-level heuristic* yang akan dieksekusi secara random menggunakan probabilitas. Jika nilai random yang dikeluarkan bernilai dibawah 0,5 maka *low-level heuristic* yang akan dieksekusi adalah *move*. Sebaliknya jika nilai yang dikeluarkan bernilai lebih dari 0,5 maka *low-heuristic* yang akan dieksekusi adalah *swap*. Setelah dieksekusi, akan dihitung nilai JFI dari jadwal baru (`curJFICost`) dan dibandingkan dengan `bestJFICost` yang sebelumnya sudah didapatkan. Jika `curJFICost` lebih kecil dari `bestJFICost`, maka solusi tersebut bisa diabaikan. Jika `curJFICost` lebih besar dari `bestJFICost`, maka solusi tersebut akan menjadi `bestJFICost` yang baru untuk iterasi selanjutnya. Detail kode program algoritma hill climbing dapat dilihat seperti dibawah ini:

```

1. double bestJFICost;
2. double curJFICost;
3.
4. public void HillClimbing(int NumbIt){
5.     String [][] bestSolution = new String [NumberOfEmployee][LengthOfMonth[newMonths]];
6.     bestJFICost = curJFICost = JFI(roster);
7.     for (int i = 0; i < NumbIt; i++) {
8.         System.out.printf("%2d, %5f, %5f \n", i,curJFICost, bestJFICost );
9.         String [][] Solpertama = roster.clone(); //this is the best Solution So Far
10.        if (Math.random() <0.5) {
11.            Move();
12.        } else {
13.            Swap();
14.        }
15.        curJFICost= JFI(roster);
16.        if (curJFICost > bestJFICost) {
17.            bestJFICost = curJFICost;
18.        } else{
19.            roster = Solpertama.clone();
20.        }
21.    }
22.    System.out.println("BestJFI : " + bestJFICost );
23. }

```

**Kode Program 5.18 Algoritma Hill Climbing**

## BAB VI HASIL DAN PEMBAHASAN

Pada bagian ini akan dijelaskan hasil implementasi dari rancangan model yang telah dibuat sebelumnya, baik dari otomasi penjadwalan, Jain Fairness Index, dan optimasinya. Bagian ini akan menjelaskan perbandingan hasil antara JFI dari jadwal manual rumah sakit, JFI jadwal hasil otomasi, dan JFI jadwal hasil otomasi dengan menggunakan metode *Reinforcement Learning Hyper-Heuristic*.

### 6.1 Jain Fairness Index Jadwal Eksisting

Untuk menghitung *Jain Fairness Index*, jadwal eksisting rumah sakit yang digunakan adalah jadwal keenam unit pada bulan November 2017. Perhitungan ini nantinya akan dibandingkan dengan jadwal hasil otomasi dan jadwal hasil optimasi. Hasil perhitungan *Jain Fairness Index* untuk jadwal eksisting dapat dilihat sebagai berikut:

#### 1. Unit Farmasi

Jadwal eksisting bulan November 2017 yang sudah di cetak pada Netbeans dan nilai JFI dari unit farmasi dapat dilihat seperti dibawah ini:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
R	K	J	Sb	M	Sn	S1	R	K	J	Sb	M	Sn	S1	R	K
P	P	P	P	L	P	P	P	P	P	P	L	P	P	P	P
S	MS	S	S	MS	L	MS	MS	MS	S	S	L	S	S	S	S
M	M	LM	L	P	S	P	P	M	M	LM	L	CT	P	S	MS
P	P	L	MS	M	M	LM	L	P	P	CT	S	S	S	M	M
LM	L	S	S	S	P	M	M	LM	L	P	MS	P	CT	P	S
L	S	M	M	LM	L	S	S	S	CT	S	P	M	M	LM	L
S	S	P	P	L	S	S	S	S	S	M	M	LM	L	MS	P
P	P	P	P	L	P	P	P	P	P	P	L	P	P	P	P
17	18	19	20	21	22	23	24	25	26	27	28	29	30		
J	Sb	M	Sn	S1	R	K	J	Sb	M	Sn	S1	R	K		
P	P	L	P	P	P	P	P	P	L	P	P	P	P		
S	L	MS	S	MS	S	S	MS	L	P	S	MS	S	MS		
MS	P	CT	MS	M	M	LM	L	P	S	S	S	L	P		
LM	L	P	P	S	S	M	M	LM	L	P	P	S	S		
M	M	LM	L	P	P	S	S	M	M	LM	L	P	S		
S	S	S	S	S	CT	P	S	S	P	M	M	LM	L		
P	S	M	M	LM	L	MS	P	S	S	L	S	M	M		
P	P	L	P	P	P	P	P	P	L	P	P	P	P		

**Gambar 6.1 Jadwal eksisting unit farmasi bulan November**

Dari jadwal eksisting tersebut, ada beberapa pola shift yang dapat diasumsikan untuk menjadi dasar pembuatan otomasi jadwal seperti yang sudah dijelaskan pada bab 5.2.1 Setelah dilakukan perhitungan, didapatkan nilai JFI sebesar 0,80798005.

2. Unit Bayi/NICU

Jadwal eksisting bulan November 2017 yang sudah di cetak pada Netbeans dan nilai JFI dari unit Bayi/NICU dapat dilihat seperti dibawah ini:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K
P	P	P	P	L	P	P	P	P	P	P	L	P	P	P	P
L	P	P	P	M	M	LM	L	P	S	P	P	P	L	P	P
LM	L	L	S	P	P	M	M	LM	L	S	S	S	M	M	LM
P	CT	M	M	LM	L	P	P	Mid	L	S	M	M	LM	L	P
S	S	S	M	M	LM	L	S	S	P	M	M	LM	L	L	M
M	M	LM	L	S	S	S	P	M	M	LM	L	Mid	P	S	L
LM	L	S	S	P	L	L	CT	M	M	LM	L	CT	S	P	S
M	M	LM	L	L	P	M	M	LM	L	Mid	S	S	S	M	M
S	Mid	M	M	LM	L	S	Mid	CT	M	M	LM	L	P	S	Mid
Mid	P	P	L	Mid	S	Mid	L	P	S	L	P	M	M	LM	L
L	S	Mid	P	M	M	LM	S	S	Mid	M	M	LM	L	Mid	S
M	LM	L	Mid	S	M	M	LM	L	P	P	Mid	M	M	LM	L
P	M	M	LM	L	Mid	P	M	M	LM	L	L	P	L	M	M
17	18	19	20	21	22	23	24	25	26	27	28	29	30		
J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K		
P	P	L	P	P	P	P	P	P	L	P	P	P	P		
P	L	P	P	P	L	P	P	M	M	LM	L	CT	P		
L	P	P	CT	S	P	M	M	LM	L	S	S	P	S		
P	S	S	P	L	M	M	LM	L	P	M	M	LM	L		
M	LM	L	S	S	P	L	S	P	L	P	CT	M	M		
S	P	S	M	M	LM	L	LM	S	S	L	L	CT	CT		
Mid	M	M	LM	L	S	S	Mid	M	M	LM	L	S	S		
LM	L	CT	Mid	M	M	LM	L	S	P	LM	P	M	M		
L	Mid	M	M	LM	L	S	P	Mid	S	M	M	LM	L		
M	M	LM	L	P	Mid	Mid	M	M	LM	L	S	P	P		
S	S	M	M	LM	L	M	M	LM	L	S	M	M	LM		
M	M	LM	L	Mid	S	P	S	L	Mid	P	P	S	M		
LM	L	Mid	S	M	M	LM	L	P	M	M	LM	LM	Mid		

Gambar 6.2 Jadwal eksisting unit Bayi/NICU bulan November

Dari jadwal eksisting tersebut, ada beberapa pola shift yang dapat diasumsikan untuk menjadi dasar pembuatan otomasi jadwal seperti yang sudah dijelaskan pada bab 5.2.2 Setelah dilakukan perhitungan, didapatkan nilai JFI sebesar 0,667.

### 3. Unit Gizi

Jadwal eksisting bulan November 2017 yang sudah di cetak pada Netbeans dan nilai JFI dari unit Gizi dapat dilihat seperti dibawah ini:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
R	K	J	Sb	M	Sn	S1	R	K	J	Sb	M	Sn	S1	R	K
P	P	P	P	L	P	P	P	P	P	P	L	P	P	P	P
P	L	S	P1	P	P1	S	S	P	CT	L	L	P1	P1	S	S
L	MD	MD	MD	P1	MD	CT	MD	S	P1	P1	L	L	S	MD	S
MD	P2	P1	L	S	S	P	CT	CT	L	MD	S	S	MD	P1	MD
S	S	CT	S	MD	L	MD	P	S	S	S	P1	MD	L	CT	P
P1	P1	S	CT	L	S	P1	P1	P1	MD	S	MD	L	S	S	P1
P2	S	S	S	S	L	S	S	S	S	S	L	S	P	P	P2
P2	L	P2	CT	L	P2	P2	P2	L	P	P	S	P2	P2	P2	L
S	S	P	P	P2	L	S	S	S	P2	P2	P2	L	L	CT	S
P	P	MD	MD	P	L	S	P	P	P	MD	MD	L	S	P	MD
L	CT	S	S	MD	M	M	M	LM	L	P	P	M	M	LM	L
S	S	M	M	M	LM	L	MD	S	S	S	L	CT	CT	M	M
MD	MD	P	P	L	P	P	S	M	M	LM	L	S	S	P	P
M	M	LM	CT	S	MD	S	P	P	P	M	M	LM	CT	S	S
P	P	MD	MD	P	S	L	M	M	LM	L	S	P	P	MD	L
P	P	P	S	L	PS	P	S	S	P	L	PS	S	P	P	P
S	S	S	P	PS	L	S	P	P	P	S	PS	L	P	S	S
S	S	S	P	PS	L	P	S	S	P	P	L	PS	S	S	P
CT	CT	CT	S	L	PS	S	CT	CT	S	S	PS	L	CT	CT	S
17	18	19	20	21	22	23	24	25	26	27	28	29	30		
J	Sb	M	Sn	S1	R	K	J	Sb	M	Sn	S1	R	K		
P	P	MOD	P	P	P	P	P	L	L	P	P	P	P		
S	P	P1	S	L	P	P2	S	S	P1	P2	P	P1	S		
P	MD	MD	P1	MD	MD	S	CT	L	S	MD	MD	MD	P		
L	S	S	MD	S	S	CT	P	P1	MD	P1	P1	L	CT		
MD	L	CT	CT	P1	CT	MD	MD	MD	L	S	S	S	MD		
P1	P1	S	S	L	P1	P1	P1	S	CT	S	L	CT	P1		
P2	P2	P	S	S	S	S	P2	P	P	L	S	P	S		
S	S	S	P	P	CT	CT	S	P2	P2	S	S	P2	P2		
S	S	P2	P2	P2	P2	S	L	S	S	CT	P2	S	S		
S	S	P	L	S	P	P	P	CT	L	CT	CT	CT	CT		
CT	MD	MD	P	M	M	LM	L	S	S	MD	S	P	P		
LM	L	S	S	P	P	CT	CT	M	M	LM	L	S	S		
M	M	LM	L	S	S	M	M	LM	L	S	P	MD	MD		
P	P	L	MD	MD	CT	CT	L	L	P	M	M	LM	L		
MD	MD	M	M	LM	L	S	S	P	P	P	MD	M	M		
P	S	PS	L	CT	CT	CT	CT	PS	PS	L	S	P	P		
S	P	L	PS	S	P	P	P	CT	L	PS	P	S	S		
P	L	S	PS	S	P	P	P	S	PS	L	P	S	S		
S	PS	P	L	P	S	S	S	P	L	PS	S	P	P		

**Gambar 6.3 Jadwal eksisting unit gizi bulan November**

Dari jadwal eksisting tersebut, ada beberapa pola shift yang dapat diasumsikan untuk menjadi dasar pembuatan otomasi jadwal seperti yang sudah dijelaskan pada bab 5.2.3 Setelah dilakukan perhitungan, didapatkan nilai JFI sebesar 0,901226.

4. Unit IGD

Jadwal eksisting bulan November 2017 yang sudah di cetak pada Netbeans dan nilai JFI dari unit IGD dapat dilihat seperti dibawah ini:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
R	K	J	Sb	M	Sn	S1	R	K	J	Sb	M	Sn	S1	R	K
P	MS	S	MS	L	S	P	MS	L	CT	M	M	LM	S	S	P
MS	S	P	L	M	M	LM	MS	S	MS	L	L	S	MS	MS	S
S	P	S	P	L	P	MS	P	MS	CT	S	S	MS	P	L	CT
L	S	MS	MS	S	CT	S	S	M	M	LM	L	P	S	S	MS
S	S	M	M	LM	L	MS	S	S	S	MS	L	S	S	P	MS
M	M	LM	L	P	S	S	CT	MS	S	P	L	M	M	LM	L
LM	S	L	P	L	S	S	S	P	P	MS	L	S	S	M	M
S	MS	S	S	L	MS	M	M	LM	S	P	P	P	L	MS	S
17	18	19	20	21	22	23	24	25	26	27	28	29	30		
J	Sb	M	Sn	S1	R	K	J	Sb	M	Sn	S1	R	K		
MS	L	S	S	S	M	L	P	P	CT	S	P	MS	M		
S	P	P	CT	S	MS	S	S	P	L	P	MS	CT	MS		
M	M	LM	S		S	MS	L	S	P	M	M	LM	L		
CT	S		S	M	L	P	MS	MS	S	S	S	M	L		
L	MS	M	M	LM	P	MS	S	L	CT	MS	S	S	P		
P	MS	L	S	S	S	M	M	LM	L	S	MS	S	S		
LM	L	L	MS	P	MS	S	S	MS	L	CT	S	P	MS		
S	P	L	P	MS	S	S	S	M	M	LM	L	MS	S		

Gambar 6.4 Jadwal eksisting unit IGD bulan November

Dari jadwal eksisting tersebut, ada beberapa pola shift yang dapat diasumsikan untuk menjadi dasar pembuatan otomatisasi jadwal seperti yang sudah dijelaskan pada bab 5.2.4. Setelah dilakukan perhitungan, didapatkan nilai JFI sebesar 0,918402778.

5. Unit Kamar Operasi

Jadwal eksisting bulan November 2017 yang sudah di cetak pada Netbeans dan nilai JFI dari unit Kamar Operasi dapat dilihat seperti dibawah ini:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
R	K	J	Sb	M	Sn	S1	R	K	J	Sb	M	Sn	S1	R	K
P	P	P	P	P	CT	CT	S	S	P	P	L	P	S	P	S
P	S	S	S	L	P	P	P	P	L	S	P	S	P	S	P
P	S	P	L	P	S	S	P	P	S	P	P	L	P	P	S
S	P	P	P	L	S	S	P	P	P	CT	L	S	P	P	P
L	P	S	S	P	P	P	L	P	S	S	P	P	S	L	P
P	L	P	P	P	P	L	S	S	P	P	P	P	CT	S	P
17	18	19	20	21	22	23	24	25	26	27	28	29	30		
J	Sb	M	Sn	S1	R	K	J	Sb	M	Sn	S1	R	K		
S	L	P	S	S	S	P	L	P	P	P	S	S	P		
L	P	P	P	P	P	L	P	S	P	S	S	L	S		
P	S	L	S	P	P	S	P	CT	L	S	P	P	P		
P	P	P	L	S	S	S	P	P	L	P	S	S	S		
S	S	P	P	P	L	P	S	P	P	P	P	P	CT		
P	P	L	P	P	P	P	S	S	P	L	P	P	P		

Gambar 6.5 Jadwal eksisting unit Kamar Operasi bulan November

Dari jadwal eksisting tersebut, ada beberapa pola shift yang dapat diasumsikan untuk menjadi dasar pembuatan otomasi jadwal seperti yang sudah dijelaskan pada bab 5.2.5. Setelah dilakukan perhitungan, didapatkan nilai JFI sebesar 0,806983

## 6. Unit SIM & RM

Jadwal eksisting bulan November 2017 yang sudah di cetak pada Netbeans dan nilai JFI dari unit SIM & RM dapat dilihat seperti dibawah ini:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
R	K	J	Sb	M	Sn	S1	R	K	J	Sb	M	Sn	S1	R	K
P	CT	P	P	L	MS	S	S	P	P	P	L	S	P	S	M
L	S	S	S	P	M	M	LM	L	S	S	MS	MS	M	M	LM
S	P	P	MS	S	L	P	P	M	M	LM	L	CT	S	MS	S
S	M	M	LM	L	S	S	M	LM	L	S	P	P	MS	L	S
P	MS	S	M	M	LM	L	S	S	P	P	S	M	LM	L	P
M	LM	L	L	MS	P	P	P	MS	S	M	M	LM	L	P	P
17	18	19	20	21	22	23	24	25	26	27	28	29	30		
J	Sb	M	Sn	S1	R	K	J	Sb	M	Sn	S1	R	K		
M	LM	L	MS	S	M	M	M	LM	L	S	S	S	P		
L	P	P	S	S	P	P	L	S	P	P	P	M	LM		
S	S	L	M	M	LM	L	P	S	L	S	P	P	M		
P	P	S	P	P	CT	P	S	M	M	LM	L	P	MS		
P	S	MS	L	P	MS	S	P	P	S	M	M	LM	L		
S	M	M	LM	L	S	S	S	P	MS	P	S	S	S		

**Gambar 6.6 Jadwal eksisting unit SIM & RM bulan November**

Dari jadwal eksisting tersebut, ada beberapa pola shift yang dapat diasumsikan untuk menjadi dasar pembuatan otomasi jadwal seperti yang sudah dijelaskan pada bab 5.2.6. Setelah dilakukan perhitungan, didapatkan nilai JFI sebesar 0,746714.

## 6.2 Hasil Otomasi Jadwal dan Jain Fairness Index

Berikut merupakan bentuk jadwal staf yang diotomasi dengan menggunakan bahasa pemrograman Java.

### 6.2.1 Hasil Otomasi Unit Farmasi

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
J	Sb	M	Sn	S1	R	K	J	Sb	M	Sn	S1	R	K	J	Sb
P	P	L	P	P	P	P	P	P	L	P	P	P	P	P	P
S	S	L	S	S	S	S	S	S	L	S	S	S	S	S	S
M	M	LM	L	P	P	S	S	MS	S	M	M	LM	L	P	P
MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L
S	S	S	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M
P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS
LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S
P	P	L	P	P	P	P	P	P	L	P	P	P	P	P	P
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
M	Sn	S1	R	K	J	Sb	M	Sn	S1	R	K	J	Sb	M	
L	P	P	P	P	P	P	L	P	P	P	P	P	P	L	
L	S	S	S	S	S	S	L	S	S	S	S	S	S	L	
S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	
P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	S	
LM	L	P	P	S	S	MS	S	M	M	LM	L	P	P	S	
M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	
S	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	
L	P	P	P	P	P	P	L	P	P	P	P	P	P	L	

Gambar 6.7 Otomasi jadwal staf unit farmasi

Hari Minggu:	20	Hari Sabtu:	0	Hari Kerja:	0	Sum :	20
Hari Minggu:	20	Hari Sabtu:	0	Hari Kerja:	0	Sum :	20
Hari Minggu:	4	Hari Sabtu:	0	Hari Kerja:	2	Sum :	6
Hari Minggu:	0	Hari Sabtu:	2	Hari Kerja:	2	Sum :	4
Hari Minggu:	0	Hari Sabtu:	0	Hari Kerja:	3	Sum :	3
Hari Minggu:	4	Hari Sabtu:	2	Hari Kerja:	1	Sum :	7
Hari Minggu:	0	Hari Sabtu:	2	Hari Kerja:	2	Sum :	4
Hari Minggu:	20	Hari Sabtu:	0	Hari Kerja:	0	Sum :	20

JFI: 0.665158371040724

Gambar 6.8 Otomasi jadwal dan Jain Fairness Index hasil otomasi unit Farmasi

Dari jadwal hasil otomasi diatas, dengan mengikuti pola yang sudah diasumsikan didapatkan nilai JFI sebesar 0,6651 yang artinya nilai tersebut turun sebesar 0,142 jika dibandingkan dengan nilai JFI jadwal eksisting rumah sakit. Untuk itu perlu dilakukan optimasi agar nilai tersebut bisa meningkat.

### 6.2.2 Hasil Otomasi Unit Bayi/NICU

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
J	Sb	M	Sn	S1	R	K	J	Sb	M	Sn	S1	R	K	J	Sb
P	P	L	P	P	P	P	P	P	L	P	P	P	P	P	P
M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P
MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L
S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M
P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS
LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S
M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P
MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L
S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M
P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS
LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S
M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P
MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L

17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
M	Sn	S1	R	K	J	Sb	M	Sn	S1	R	K	J	Sb	M
L	P	P	P	P	P	P	L	P	P	P	P	P	P	L
S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M
P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS
LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S
M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P
MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM
S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M
P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS
LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S
M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P
MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM
S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M
P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS

**Gambar 6.9 Otomasi jadwal staf unit Bayi/Nicu**

Hari Minggu:	20	Hari Sabtu:	0	Hari Kerja:	0	Sum :	20
Hari Minggu:	4	Hari Sabtu:	0	Hari Kerja:	2	Sum :	6
Hari Minggu:	0	Hari Sabtu:	2	Hari Kerja:	2	Sum :	4
Hari Minggu:	0	Hari Sabtu:	0	Hari Kerja:	3	Sum :	3
Hari Minggu:	4	Hari Sabtu:	2	Hari Kerja:	1	Sum :	7
Hari Minggu:	0	Hari Sabtu:	2	Hari Kerja:	2	Sum :	4
Hari Minggu:	4	Hari Sabtu:	0	Hari Kerja:	2	Sum :	6
Hari Minggu:	0	Hari Sabtu:	2	Hari Kerja:	2	Sum :	4
Hari Minggu:	0	Hari Sabtu:	0	Hari Kerja:	3	Sum :	3
Hari Minggu:	4	Hari Sabtu:	2	Hari Kerja:	1	Sum :	7
Hari Minggu:	0	Hari Sabtu:	2	Hari Kerja:	2	Sum :	4
Hari Minggu:	4	Hari Sabtu:	0	Hari Kerja:	2	Sum :	6
Hari Minggu:	0	Hari Sabtu:	2	Hari Kerja:	2	Sum :	4

JFI: 0.6647727272727273

**Gambar 6.10 Jain Fairness Index hasil otomasi unit Bayi/Nicu**

Dari jadwal hasil otomasi diatas, dengan mengikuti pola yang sudah diasumsikan didapatkan nilai JFI sebesar 0,6647 yang artinya nilai tersebut hanya turun sebesar 0,003 jika dibandingkan dengan nilai JFI jadwal eksisting rumah sakit. Untuk itu perlu dilakukan optimasi agar nilai tersebut bisa meningkat.

### 6.2.3 Hasil Otomasi Unit Gizi

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
J	Sb	M	Sn	S1	R	K	J	Sb	M	Sn	S1	R	K	J	Sb
P	P	L	P	P	P	P	P	P	L	P	P	P	P	P	P
P1	P	S	P1	P	L	S	S	MD	MD	P1	P	S	P1	P	L
MD	MD	P1	P	S	P1	P	L	S	S	MD	MD	P1	P	S	P1
S	S	MD	MD	P1	P	S	P1	P	L	S	S	MD	MD	P1	P
P	L	S	S	MD	MD	P1	P	S	P1	P	L	S	S	MD	MD
S	S	S	P1	P1	L	P1	P1	S	S	P1	P1	L	L	P1	P1
F2	P	S	S	L	P	S	F2	P	MS	F2	P	S	S	L	P
P	MS	F2	P	S	S	L	P	S	F2	P	MS	F2	P	S	S
S	F2	P	MS	F2	P	S	S	L	P	S	F2	P	MS	F2	P
S	S	P	P	MD	MD	L	S	S	P	P	MD	MD	L	S	S
M	M	LM	L	P	P	S	S	P	P	M	M	LM	L	P	P
P	P	M	M	LM	L	P	P	S	S	P	P	M	M	LM	L
S	S	P	P	M	M	LM	L	P	P	S	S	P	P	M	M
P	P	S	S	P	P	M	M	LM	L	P	P	S	S	P	P
LM	L	P	P	S	S	MD	MD	M	M	LM	L	P	P	S	S
S	S	L	PS	S	S	P	P	PS	L	P	P	S	S	P	P
P	P	PS	L	P	P	S	S	L	PS	S	S	P	P	P	P
P	P	PS	L	P	P	S	S	L	PS	S	S	P	P	P	P
S	S	L	PS	S	S	P	P	PS	L	P	P	S	S	P	P
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
M	Sn	S1	R	K	J	Sb	M	Sn	S1	R	K	J	Sb	M	
L	P	P	P	P	P	P	L	P	P	P	P	P	P	L	
S	S	MD	MD	P1	P	S	P1	P	L	S	S	MD	MD	P1	
P	L	S	S	MD	MD	P1	P	S	P1	P	L	S	S	MD	
S	P1	P	L	S	S	MD	MD	P1	P	S	P1	P	L	S	
P1	P	S	P1	P	L	S	S	MD	MD	P1	P	S	P1	P	
S	S	S	P1	P1	L	P1	P1	S	S	S	P1	P1	L	P1	
S	P2	P	MS	F2	P	S	S	L	P	S	F2	P	MS	F2	
L	P	S	F2	P	MS	F2	P	S	S	L	P	S	F2	P	
S	S	L	P	S	F2	P	MS	F2	P	S	S	L	P	S	
P	P	MD	MD	L	S	S	P	P	MD	MD	L	S	S	P	
S	S	P	P	M	M	LM	L	P	P	S	S	P	P	M	
P	P	S	S	P	P	M	M	LM	L	P	P	S	S	P	
LM	L	P	P	S	S	P	P	M	M	LM	L	P	P	S	
M	M	LM	L	P	P	S	S	P	P	M	M	LM	L	P	
MD	MD	M	M	LM	L	P	P	S	S	P	MD	M	M	LM	
L	PS	S	S	P	P	PS	L	P	P	S	S	S	S	L	
PS	L	P	P	S	S	S	L	PS	S	S	P	P	P	PS	
PS	L	P	P	S	S	S	L	PS	S	S	P	P	P	PS	
L	PS	S	S	P	P	P	PS	L	P	P	S	S	S	L	

Gambar 6.11 Otomasi jadwal staf unit gizi

Hari Minggu:	20	Hari Sabtu:	0	Hari Kerja:	0	Sum :	20
Hari Minggu:	0	Hari Sabtu:	2	Hari Kerja:	2	Sum :	4
Hari Minggu:	0	Hari Sabtu:	0	Hari Kerja:	3	Sum :	3
Hari Minggu:	4	Hari Sabtu:	2	Hari Kerja:	1	Sum :	7
Hari Minggu:	0	Hari Sabtu:	2	Hari Kerja:	2	Sum :	4
Hari Minggu:	0	Hari Sabtu:	2	Hari Kerja:	3	Sum :	5
Hari Minggu:	0	Hari Sabtu:	0	Hari Kerja:	3	Sum :	3
Hari Minggu:	4	Hari Sabtu:	0	Hari Kerja:	2	Sum :	6
Hari Minggu:	0	Hari Sabtu:	2	Hari Kerja:	2	Sum :	4
Hari Minggu:	0	Hari Sabtu:	0	Hari Kerja:	4	Sum :	4
Hari Minggu:	4	Hari Sabtu:	0	Hari Kerja:	2	Sum :	6
Hari Minggu:	0	Hari Sabtu:	2	Hari Kerja:	2	Sum :	4
Hari Minggu:	0	Hari Sabtu:	0	Hari Kerja:	3	Sum :	3
Hari Minggu:	4	Hari Sabtu:	2	Hari Kerja:	1	Sum :	7
Hari Minggu:	0	Hari Sabtu:	2	Hari Kerja:	2	Sum :	4
Hari Minggu:	12	Hari Sabtu:	0	Hari Kerja:	2	Sum :	14
Hari Minggu:	8	Hari Sabtu:	0	Hari Kerja:	2	Sum :	10
Hari Minggu:	8	Hari Sabtu:	0	Hari Kerja:	2	Sum :	10
Hari Minggu:	12	Hari Sabtu:	0	Hari Kerja:	2	Sum :	14

JFI: 0.7000401767778224

Gambar 6.12 Jain Fairness Index hasil otomasi unit gizi

Dari jadwal hasil otomasi diatas, dengan mengikuti pola yang sudah diasumsikan didapatkan nilai JFI sebesar 0,700 yang artinya nilai tersebut turun sebesar 0,201 jika dibandingkan dengan nilai JFI jadwal eksisting rumah sakit. Untuk itu perlu dilakukan optimasi agar nilai tersebut bisa meningkat.

## 6.2.4 Hasil Otomasi Unit IGD

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
J	Sb	M	Sn	S1	R	K	J	Sb	M	Sn	S1	R	K	J	Sb
M	M	LM	L	P	P	S	S	S	S	S	L	MD	MS	S	M
MS	S	M	M	LM	L	P	P	S	S	S	S	S	L	MD	MS
L	MD	MS	S	M	M	LM	L	P	P	S	S	S	S	S	L
S	S	L	MD	MS	S	M	M	LM	L	P	P	S	S	S	S
S	S	S	S	L	MD	MS	S	M	M	LM	L	P	P	S	S
P	S	S	S	S	S	L	MD	MS	S	M	M	LM	L	P	P
L	P	P	S	S	S	S	L	MD	MS	S	M	M	LM	L	P
M	LM	L	P	P	S	S	S	S	S	L	MD	MS	S	M	M
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
M	Sn	S1	R	K	J	Sb	M	Sn	S1	R	K	J	Sb	M	
M	LM	L	P	P	S	S	S	S	S	L	MD	MS	S	M	
S	M	M	LM	L	P	P	S	S	S	S	S	L	MD	MS	
MD	MS	S	M	M	LM	L	P	P	S	S	S	S	S	L	
S	L	MD	MS	S	M	M	LM	L	P	P	S	S	S	S	
S	S	S	L	MD	MS	S	M	M	LM	L	P	P	S	S	
S	S	S	S	S	L	MD	MS	S	M	M	LM	L	P	P	
P	P	S	S	S	S	S	L	MD	MS	S	M	M	LM	L	
LM	L	P	P	S	S	S	S	S	L	MD	MS	S	M	M	

**Gambar 6.13 Otomasi jadwal staf unit IGD**

Hari Minggu:	0	Hari Sabtu:	0	Hari Kerja:	4	Sum :	4
Hari Minggu:	0	Hari Sabtu:	0	Hari Kerja:	4	Sum :	4
Hari Minggu:	4	Hari Sabtu:	4	Hari Kerja:	2	Sum :	10
Hari Minggu:	8	Hari Sabtu:	0	Hari Kerja:	2	Sum :	10
Hari Minggu:	0	Hari Sabtu:	0	Hari Kerja:	4	Sum :	4
Hari Minggu:	0	Hari Sabtu:	0	Hari Kerja:	4	Sum :	4
Hari Minggu:	8	Hari Sabtu:	4	Hari Kerja:	1	Sum :	13
Hari Minggu:	4	Hari Sabtu:	0	Hari Kerja:	3	Sum :	7

JFI: 0.8132780082987552

**Gambar 6.14 Jain Fairness Index hasil otomasi unit IGD**

Dari jadwal hasil otomasi diatas, dengan mengikuti pola yang sudah diasumsikan didapatkan nilai JFI sebesar 0,8132 yang artinya nilai tersebut turun sebesar 0,105 jika dibandingkan dengan nilai JFI jadwal eksisting rumah sakit. Untuk itu perlu dilakukan optimasi agar nilai tersebut bisa meningkat.

### 6.2.5 Hasil Otomasi Unit Kamar Operasi

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
J	Sb	M	Sn	S1	R	K	J	Sb	M	Sn	S1	R	K	J	Sb
P	P	L	P	S	L	P	P	S	P	S	P	S	P	S	L
S	P	L	S	P	S	L	P	P	P	S	P	P	S	P	S
S	S	P	P	S	P	S	L	P	P	S	S	P	P	P	S
P	S	P	P	P	S	P	S	L	P	P	S	P	P	P	S
P	P	P	P	P	P	S	P	S	L	P	P	P	S	P	P
L	P	P	S	P	P	P	S	P	L	L	P	P	S	P	P
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
M	Sn	S1	R	K	J	Sb	M	Sn	S1	R	K	J	Sb	M	
P	P	S	S	P	P	S	L	S	L	P	P	S	S	P	
P	P	P	S	S	P	P	L	P	S	L	P	P	S	P	
L	L	P	P	S	S	P	P	S	P	S	L	P	P	P	
L	S	L	P	P	S	P	P	P	S	P	S	L	P	P	
P	P	S	L	P	P	S	P	P	S	P	S	L	P	L	
P	S	P	S	L	P	P	P	P	P	P	S	P	S	L	

Gambar 6.15 Otomasi jadwal staf unit OK

Hari Minggu:	8	Hari Sabtu:	2	Hari Kerja:	2	Sum :	12
Hari Minggu:	8	Hari Sabtu:	0	Hari Kerja:	2	Sum :	10
Hari Minggu:	4	Hari Sabtu:	0	Hari Kerja:	3	Sum :	7
Hari Minggu:	4	Hari Sabtu:	2	Hari Kerja:	2	Sum :	8
Hari Minggu:	8	Hari Sabtu:	2	Hari Kerja:	1	Sum :	11
Hari Minggu:	8	Hari Sabtu:	0	Hari Kerja:	3	Sum :	11

JFI: 0.9685587089593768

Gambar 6.16 Jain Fairness Index hasil otomasi unit OK

Dari jadwal hasil otomasi diatas, dengan mengikuti pola yang sudah diasumsikan didapatkan nilai JFI sebesar 0,9685 yang artinya nilai tersebut meningkat sebesar 0,162 jika dibandingkan dengan nilai JFI jadwal eksisting rumah sakit. Nilai tersebut sudah sangat baik, tetapi optimasi tetap perlu dilakukan untuk menguji apakah masih terdapat solusi yang lebih baik atau tidak.

### 6.2.6 Hasil Otomasi Unit SIM & RM

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
J	Sb	M	Sn	S1	R	K	J	Sb	M	Sn	S1	R	K	J	Sb
P	L	S	S	S	MS	M	M	LM	L	P	P	P	L	S	S
P	P	P	L	S	S	S	MS	M	M	LM	L	P	P	P	L
LM	L	P	P	P	L	S	S	S	MS	M	M	LM	L	P	P
M	M	LM	L	P	P	P	L	S	S	S	MS	M	M	LM	L
S	MS	M	M	LM	L	P	P	P	L	S	S	S	MS	M	M
S	S	S	MS	M	M	LM	L	P	P	L	S	S	S	S	MS
17	18	19	20	21	22	23	24	25	26	27	28	29	30		
M	Sn	S1	R	K	J	Sb	M	Sn	S1	R	K	J	Sb	M	
S	MS	M	M	LM	L	P	P	P	L	S	S	S	MS	M	
S	S	S	MS	M	M	LM	L	P	P	L	S	S	S	S	
P	L	S	S	S	MS	M	M	LM	L	P	P	P	L	S	
P	P	P	L	S	S	S	MS	M	M	LM	L	P	P	P	
LM	L	P	P	P	L	S	S	S	MS	M	M	LM	L	P	
M	M	LM	L	P	P	P	L	S	S	S	S	MS	M	M	

Gambar 6.17 Otomasi jadwal staf unit SIM & RM

Hari Minggu:	4	Hari Sabtu:	2	Hari Kerja:	3	Sum :	9
Hari Minggu:	4	Hari Sabtu:	2	Hari Kerja:	3	Sum :	9
Hari Minggu:	0	Hari Sabtu:	4	Hari Kerja:	4	Sum :	8
Hari Minggu:	0	Hari Sabtu:	2	Hari Kerja:	4	Sum :	6
Hari Minggu:	4	Hari Sabtu:	2	Hari Kerja:	3	Sum :	9
Hari Minggu:	4	Hari Sabtu:	0	Hari Kerja:	3	Sum :	7

JFI: 0.9795918367346939

**Gambar 6.18 Jain Fairness Index hasil otomasi unit SIM & RM**

Dari jadwal hasil otomasi diatas, dengan mengikuti pola yang sudah diasumsikan didapatkan nilai JFI sebesar 0,9795 yang artinya nilai tersebut meningkat sebesar 0,2328 jika dibandingkan dengan nilai JFI jadwal eksisting rumah sakit. Nilai tersebut sudah sangat baik, tetapi optimasi tetap perlu dilakukan untuk menguji apakah masih terdapat solusi yang lebih baik atau tidak.

### 6.3 Hasil Optimasi Jadwal

Setelah dilakukan otomasi dan didapatkan nilai Jain Fairness Index masing-masing unit, selanjutnya adalah mengoptimalkan jadwal tersebut agar nilai JFI mendekati 1. Optimasi dilakukan dengan algoritma reinforcement learning hyper-heuristic dan dua low-level heuristic yaitu move dan swap. Hasil jadwal tiap unit dapat dilihat sebagai berikut:

#### 1. Unit Farmasi

**Tabel 6.1 Jadwal hasil optimasi unit farmasi**

No	Pekerjaan	Kode Staff	Tanggal																														
			J	Sb	M	Sn	Sl	Rb	K	J	Sb	M	Sn	Sl	Rb	K	J	Sb	M	Sn	Sl	Rb	K	J	Sb	M	Sn	Sl	Rb	K	J	Sb	M
1	Apoteker	101	P	P	L	P	P	P	P	P	P	P	P	P	P	P	P	L	P	P	P	P	P	P	P	L	P	L	P	P	P	P	L
2	Staff Senior	102	S	S	L	L	S	L	S	L	S	L	S	S	S	S	S	P	S	S	S	S	S	S	S	S	S	S	S	S	S	S	L
3	Anggota	103	M	M	LM	S	P	L	S	L	L	S	M	M	LM	L	P	P	L	S	S	L	M	M	LM	L	P	P	S	S	MS	MS	M
4	Anggota	104	MS	L	M	M	LM	P	P	P	S	S	MS	MS	M	M	LM	L	L	L	L	S	L	L	M	M	LM	S	P	P	S	S	S
5	Anggota	105	S	S	S	L	M	M	LM	S	P	L	S	S	L	L	M	M	LM	P	P	P	S	S	L	S	M	M	LM	L	P	P	S
6	Anggota	106	P	P	S	S	L	S	M	M	LM	L	P	P	S	S	MS	L	M	M	LM	M	P	P	S	L	MS	P	M	M	LM	L	P
7	Anggota	107	LM	L	P	P	S	S	L	S	M	M	LM	L	P	P	S	S	S	L	M	L	LM	MS	P	P	S	L	L	P	M	M	LM
8	Petugas Gudang	108	P	P	L	P	P	P	P	P	P	L	P	P	P	P	P	P	S	P	P	P	P	P	P	L	P	P	P	L	P	L	

## 2. Unit Bayi/NICU

Tabel 6.2 Jadwal hasil optimasi unit bayi/NICU

No	Pekerjaan	Kode Staf	Tanggal																														
			J	Sb	M	Sn	Sl	Rb	K	J	Sb	M	Sn	Sl	Rb	K	J	Sb	M	Sn	Sl	Rb	K	J	Sb	M	Sn	Sl	Rb	K	J	Sb	M
			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	26	27	28	29	30	31
1	Kepala Unit	201	P	P	L	P	P	P	P	P	P	L	P	P	P	P	P	S	P	P	P	P	P	P	P	P	P	P	P	P	P	L	
2	Anggota	202	M	M	L	L	P	L	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	L	M	M	LM	L	P	P	S	S	L	MS	M
3	Anggota	203	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	L	M	M	LM	L	P	P	S	S	L
4	Anggota	204	S	S	LM	L	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	S	L	M	M	LM	L	P	P	S
5	Anggota	205	P	P	S	S	MS	L	M	M	LM	L	P	P	S	S	L	MS	M	M	LM	L	P	P	L	S	MS	MS	M	M	LM	L	P
6	Anggota	206	LM	L	P	P	S	S	MS	L	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	L	MS	M	M	LM
7	Anggota	207	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	L	S	L	MS	M	M	LM	L	P	P	S	S	L	MS	M
8	Anggota	208	MS	MS	M	M	LM	P	P	P	S	S	L	MS	M	M	LM	L	P	P	S	S	L	MS	M	M	LM	L	P	P	S	S	L
9	Anggota	209	S	S	MS	MS	M	M	LM	L	P	P	S	S	L	MS	M	M	LM	L	P	P	S	S	MS	L	M	M	LM	L	P	P	S
10	Anggota	210	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	L	M	M	LM	L	P	P	S	S	MS	L	M	M	LM	L	P
11	Anggota	211	LM	L	P	P	S	S	L	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	L	S	S	MS	L	M	M	LM
12	Anggota	212	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	L	L	M	M	LM	L	P	P	S	S	MS	MS	M
13	Anggota	213	L	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	L

3. Unit Gizi

Tabel 6.3 Jadwal hasil optimasi unit gizi

No	Pekerjaan	Kode Staf	Tanggal																																
			J	Sb	M	Sn	Sl	Rb	K	J	Sb	M	Sn	Sl	Rb	K	J	Sb	M	Sn	Sl	Rb	K	J	Sb	M	Sn	Sl	Rb	K	J	Sb	M		
1	Ahli gizi	401	P	P	L	P	P	P	L	P	P	L	P	P	P	P	P	S	L	P	P	P	P	P	P	S	L	P	P	P	P	P	P	P	S
2	chef	402	P1	P	S	P1	P	P	S	S	MD	MD	P1	P	S	P1	P	P1	S	L	MD	MD	P1	P	S	P1	P	L	S	S	MD	MD	P1		
3	chef	403	MD	MD	L	P	S	P1	P	L	S	S	MD	MD	P1	P	S	L	P	L	S	S	MD	L	P1	P	L	P1	P	L	S	S	MD		
4	chef	404	S	S	MD	MD	P1	P	S	P1	P	L	S	S	MD	MD	P1	P	S	P1	P	L	S	S	MD	MD	P1	P	S	P1	P	P2	S		
5	chef	405	P	L	S	S	MD	MD	P1	P	S	P1	P	L	S	L	MD	MD	P1	P	S	P1	P	MD	S	S	MD	MD	P1	P	S	P1	P		
6	Helper	406	S	S	S	P1	P1	L	P1	P1	S	S	S	P1	P1	S	P1	P1	S	S	P1	P1	L	P1	P1	S	P	S	P1	P1	L	P1			
7	Helper	407	P2	P	S	S	L	L	S	P2	P	L	P2	P	S	S	L	P	S	P2	P	L	P2	P	S	S	P	S	P2	P	L	P2			
8	Helper	408	P	L	P2	P	S	S	P	P	S	P2	P	MS	P2	P	S	S	L	P	S	P2	P	L	P2	P	S	S	L	P	S	L	P		
9	Helper	409	S	P2	P	MS	P2	P	S	S	L	P	S	L	P	P	P2	P	S	S	L	P	S	P2	P	L	P2	P	S	S	L	P	S		
10	Penyaji	410	S	S	P	P	MD	MD	L	S	S	P	P	MD	MD	L	S	S	P	L	MD	MD	L	S	S	P	P	MD	MD	L	S	S	P		
11	Penyaji	411	M	M	LM	L	P	P	S	S	P	P	M	M	LM	L	P	P	L	S	P	P	M	M	LM	L	L	L	S	S	P	P	L		
12	Penyaji	412	P	P	M	M	LM	L	P	P	S	S	P	P	M	M	LM	L	P	P	S	S	P	P	M	M	L	S	P	P	S	P			
13	Penyaji	413	S	S	P	P	M	M	LM	L	P	P	S	S	P	P	M	M	LM	L	P	P	S	S	P	P	M	M	LM	L	P	P	L		
14	Penyaji	414	P	P	S	P	P	M	M	LM	L	P	P	S	S	P	P	M	M	LM	L	P	P	S	S	P	P	M	M	LM	L	P	P		
15	Penyaji	415	LM	L	P	P	S	S	MD	MD	M	M	LM	P2	P	L	S	S	MD	MD	M	M	LM	S	P	P	S	S	P	MD	M	M	LM		
16	Cafe	416	S	S	L	PS	S	S	P	P	P	PS	L	P	P	S	S	S	L	PS	S	S	P	P	P	PS	LM	P	P	S	S	S	M		
17	Cafe	417	P	P	PS	L	P	P	S	S	S	L	PS	S	S	P	P	P	L	P	P	P	S	L	S	L	PS	S	S	P	P	P	PS		
18	Driver	418	P	P	PS	L	P	P	S	S	S	L	PS	S	S	P	P	P	PS	P	P	P	S	S	S	L	PS	S	S	P	P	P	PS		
19	Driver	419	S	S	P1	PS	S	S	P	P	P	PS	L	P	P	S	S	S	PS	PS	S	S	P	P	P	PS	P	P	P	S	S	S	L		

4. Unit IGD

Tabel 6.4 Jadwal hasil optimasi unit IGD

No	Pekerjaan	Kode Staf	Tanggal																															
			J	Sb	M	Sn	Sl	Rb	K	J	Sb	M	Sn	Sl	Rb	K	J	Sb	M	Sn	Sl	Rb	K	J	Sb	M	Sn	Sl	Rb	K	J	Sb	M	
1	Anggota	501	M	M	LM	L	P	P	S	S	S	S	S	L	MD	MS	S	M	M	LM	L	P	L	S	S	S	S	S	L	MD	L	S	M	
2	Anggota	502	MS	S	M	M	LM	L	P	P	S	S	S	S	L	MD	S	S	M	M	LM	P	P	P	S	S	S	S	S	L	MD	L	S	M
3	Anggota	503	L	MD	MS	S	M	M	LM	L	P	P	S	S	S	S	L	MD	MS	S	M	M	LM	L	P	P	S	S	S	S	S	S	L	
4	Anggota	504	S	S	L	MD	MS	S	M	M	LM	L	P	P	S	S	S	S	L	MD	L	S	M	M	LM	L	P	P	S	S	S	S		
5	Anggota	505	L	S	S	S	L	MD	L	S	M	M	LM	L	P	P	S	L	S	S	S	L	MD	MS	S	M	M	L	L	P	P	S	S	
6	Anggota	506	P	S	S	S	S	S	L	MD	MS	S	M	M	LM	L	P	P	S	S	S	S	S	L	MD	L	S	M	M	LM	L	P	P	
7	Anggota	507	S	P	P	S	S	S	S	S	L	MD	MS	S	M	M	LM	L	P	P	S	S	S	S	L	MD	LM	S	M	M	LM	L		
8	Anggota	508	M	LM	L	P	P	S	S	S	S	L	MD	L	S	M	M	LM	L	P	P	S	S	S	S	S	L	MD	MS	S	M	M		

5. Unit Kamar Operasi

**Tabel 6.5 Jadwal hasil optimasi unit Kamar Operasi**

No	Pekerjaan	Kode Staf	Tanggal																														
			J	Sb	M	Sn	Sl	Rb	K	J	Sb	M	Sn	Sl	Rb	K	J	Sb	M	Sn	Sl	Rb	K	J	Sb	M	Sn	Sl	Rb	K	J	Sb	M
			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	26	27	28	29	30	31
1	Staf	601	P	P	L	P	S	L	P	P	S	P	P	P	S	P	S	L	P	P	S	S	P	P	S	L	S	L	P	P	S	S	P
2	Staf	602	S	P	L	S	P	S	L	P	P	P	S	P	P	S	P	S	P	P	P	S	S	P	P	L	P	S	L	P	P	S	P
3	Staf	603	S	S	P	P	S	P	S	L	P	P	S	S	P	P	S	P	L	L	P	P	S	S	P	P	S	P	S	L	P	P	P
4	Staf	606	P	S	P	P	P	S	P	S	L	P	P	S	P	P	P	S	L	S	L	P	P	S	P	P	P	S	P	S	L	P	P
5	Staf	604	P	P	P	P	P	P	S	P	S	L	P	P	S	P	P	P	P	S	L	P	P	S	P	P	P	S	P	S	L	L	L
6	Staf	605	L	P	P	S	P	P	S	P	L	L	P	P	S	P	P	S	P	S	L	P	P	P	P	P	P	P	S	P	S	L	L

## 6. Unit SIM &amp; RM

Tabel 6.6 Jadwal hasil optimasi unit SIM &amp; RM

No	Pekerjaan	Kode Staf	Tanggal																																
			J	Sb	M	Sn	Sl	Rb	K	J	Sb	M	Sn	Sl	Rb	K	J	Sb	M	Sn	Sl	Rb	K	J	Sb	M	Sn	Sl	Rb	K	J	Sb	M		
			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	26	27	28	29	30	31		
1	Staf	301	P	L	S	S	S	MS	M	M	LM	L	P	P	P	L	S	S	S	MS	M	M	LM	S	P	P	P	L	S	S	S	MS	M		
2	Staf	302	P	P	P	L	S	S	S	MS	M	M	LM	L	P	P	P	L	S	S	S	MS	M	M	LM	L	P	P	P	L	S	S	S		
3	Staf	303	LM	L	P	P	P	L	S	S	S	MS	M	M	LM	L	P	P	P	L	S	S	S	MS	M	M	LM	L	P	P	P	L	S		
4	Staf	304	M	M	LM	L	P	P	P	L	S	S	S	MS	M	M	LM	L	P	P	P	L	S	L	S	MS	M	M	LM	L	P	P	P		
5	Staf	305	S	MS	M	M	LM	L	P	P	P	L	S	S	S	MS	M	M	LM	L	P	P	P	L	S	S	S	MS	M	M	LM	L	P		
6	Staf	306	S	S	S	L	M	M	LM	L	P	P	P	L	S	S	S	S	MS	M	M	LM	L	P	P	P	L	S	S	S	S	MS	M	M	LM

### 6.3.1 Analisis Hasil Optimasi

#### 1. Unit Farmasi

Berikut adalah analisis jadwal hasil optimasi unit farmasi.

**Tabel 6.7 Jumlah shift tiap staf hasil optimasi unit farmasi**

<b>Kode Staff</b>	<b>P</b>	<b>S</b>	<b>M</b>	<b>MS</b>	<b>LM</b>	<b>L</b>	<b>CT</b>
101	26	0	0	0	0	5	0
102	1	24	0	0	0	6	0
103	5	7	7	2	3	7	0
104	5	7	6	3	3	7	0
105	6	10	6	0	3	6	0
106	8	6	7	2	3	5	0
107	7	7	5	1	4	7	0
108	25	1	0	0	0	5	0

**Tabel 6.8 Jumlah hari libur tiap staf hasil optimasi unit farmasi**

	<b>M</b>	<b>Sb</b>	<b>Sn</b>	<b>Sl</b>	<b>Rb</b>	<b>K</b>	<b>J</b>
<b>Bobot</b>	<b>4</b>	<b>2</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
<b>Kode Staff</b>	<b>Libur</b>						
101	4	0	0	1	0	0	0
102	3	0	1	0	1	0	1
103	2	1	0	0	2	1	1
104	1	2	1	1	0	1	1
105	1	1	1	0	1	2	0
106	2	2	0	1	0	0	0
107	0	1	1	2	2	1	0
108	4	0	0	0	0	1	0

Dengan melakukan optimasi, jumlah libur dari staf unit farmasi berkisar antara 5 sampai 7 dalam satu bulan. Berdasarkan jadwal hasil optimasi yang dapat dilihat pada bab 6.3 dan alokasi bobot tiap hari libur seperti tabel diatas, didapatkan nilai JFI dengan menggunakan algoritma reinforcement learning adalah sebesar 0,951 untuk unit farmasi.

## 2. Unit Bayi/NICU

Berikut adalah analisis jadwal hasil optimasi unit bayi/NICU.

**Tabel 6.9 Jumlah shift tiap staf hasil optimasi unit bayi/NICU**

<b>Kode Staf</b>	<b>P</b>	<b>S</b>	<b>M</b>	<b>MS</b>	<b>LM</b>	<b>L</b>	<b>CT</b>
201	27	1	0	0	0	3	0
202	5	6	7	4	2	7	0
203	6	6	6	5	3	5	0
204	6	8	6	2	4	5	0
205	7	5	6	4	3	6	0
206	6	6	6	4	4	5	0
207	6	5	7	4	3	6	0
208	7	6	6	4	3	5	0
209	6	7	6	4	3	5	0
210	7	6	6	4	3	5	0
211	5	6	6	4	4	6	0
212	6	6	7	4	3	5	0
213	6	6	6	5	3	5	0

**Tabel 6.10 Jumlah hari libur tiap staf hasil optimasi unit bayi/NICU**

	<b>M</b>	<b>Sb</b>	<b>Sn</b>	<b>Sl</b>	<b>Rb</b>	<b>K</b>	<b>J</b>
<b>Bobot</b>	<b>4</b>	<b>2</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
<b>Staff</b>	<b>Libur</b>						
201	3	0	0	0	0	0	0
202	2	0	1	0	2	1	1
203	1	1	0	1	1	0	1
204	1	0	2	0	0	1	1
205	1	2	0	0	2	0	1
206	0	1	0	1	1	0	2
207	2	0	1	1	0	1	1
208	1	1	1	1	0	1	0
209	1	0	1	0	1	1	1
210	1	2	0	1	1	0	0
211	1	1	0	1	0	2	1
212	1	0	1	1	1	1	0
213	1	1	0	1	1	0	1

Dengan melakukan optimasi, jumlah libur dari staf unit farmasi berkisar antara 3 sampai 7 dalam satu bulan. Berdasarkan jadwal hasil optimasi yang dapat dilihat pada bab 6.3 dan alokasi bobot tiap hari libur seperti tabel diatas, didapatkan nilai JFI dengan menggunakan algoritma reinforcement learning adalah sebesar 0,962 untuk unit bayi/NICU

### 3. Unit Gizi

Berikut adalah analisis jadwal hasil optimasi unit gizi.

**Tabel 6.11 Jumlah shift tiap staf hasil optimasi unit gizi**

Kode Staf	P	P1	P2	S	M	MD	LM	L	PS
401	24	0	0	2	0	0	0	5	0
402	7	8	0	9	0	6	0	1	0
403	6	4	0	8	0	6	0	7	0
404	6	6	1	10	0	6	0	2	0
405	7	6	0	8	0	7	0	3	0
406	1	15	0	12	0	0	0	3	0
407	8	0	7	10	0	0	0	6	0
408	11	0	5	9	0	0	0	5	0
409	10	0	5	10	0	0	0	5	0
410	8	0	0	10	0	8	0	5	0
411	10	0	0	5	6	0	3	7	0
412	13	0	0	7	6	0	2	3	0
413	12	0	0	6	6	0	3	4	0
414	13	0	0	6	6	0	3	3	0
415	6	0	1	7	6	5	4	2	0
416	10	0	0	12	1	0	1	3	4
417	13	0	0	9	0	0	0	5	4
418	13	0	0	10	0	0	0	3	5
419	11	1	0	12	0	0	0	2	5

**Tabel 6.12 Jumlah hari libur tiap staf hasil optimasi unit gizi**

	M	Sb	Sn	Sl	Rb	K	J
<b>Bobot</b>	<b>4</b>	<b>2</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
<b>Staff</b>	<b>Libur</b>						
1	3	0	1	0	0	1	0

	M	Sb	Sn	Sl	Rb	K	J
<b>Bobot</b>	<b>4</b>	<b>2</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
<b>Staff</b>	<b>Libur</b>						
2	0	0	0	1	0	0	0
3	1	1	2	0	0	1	2
4	1	0	0	0	1	0	0
5	0	1	0	1	0	1	0
6	0	1	0	0	1	0	1
7	1	1	0	1	2	0	1
8	1	2	0	0	1	0	1
9	1	1	0	2	0	0	1
10	0	0	1	0	0	4	0
11	3	0	2	1	0	1	0
12	0	1	1	0	1	0	0
13	1	0	1	0	0	1	1
14	1	1	0	0	1	0	0
15	0	1	0	0	0	1	0
16	2	0	1	0	0	0	0
17	3	0	1	0	0	0	1
18	2	0	1	0	0	0	0
19	1	0	1	0	0	0	0

Dengan melakukan optimasi, jumlah libur dari staf unit farmasi berkisar antara 1 sampai 7 dalam satu bulan. Perbedaan yang cukup signifikan tersebut berakibat terhadap nilai JFI yang dihasilkan menjadi kurang optimal. Berdasarkan jadwal hasil optimasi yang dapat dilihat pada bab 6.3 dan alokasi bobot tiap hari libur seperti tabel diatas, didapatkan nilai JFI dengan menggunakan

algoritma reinforcement learning adalah sebesar 0,788 untuk unit gizi

#### 4. Unit IGD

Berikut adalah analisis jadwal hasil optimasi unit IGD.

**Tabel 6.13 Jumlah shift tiap staf hasil optimasi unit IGD**

Kode Staf	P	S	M	MS	LM	L	CT
501	3	12	5	1	2	6	0
502	5	13	4	1	2	4	0
503	4	12	4	2	2	5	0
504	4	13	4	1	2	5	0
505	4	11	4	1	1	8	0
506	5	12	4	1	2	5	0
507	4	13	4	1	3	4	0
508	4	12	5	1	2	5	0

**Tabel 6.14 Jumlah hari libur tiap staf hasil optimasi unit IGD**

	M	Sb	Sn	Sl	Rb	K	J
<b>Bobot</b>	<b>4</b>	<b>2</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
<b>Staff</b>	<b>Libur</b>						
501	0	0	1	2	1	1	1
502	1	0	0	0	1	1	1
503	1	2	0	0	0	0	2
504	2	0	2	0	1	0	0
505	0	1	0	3	2	1	1
506	1	0	0	0	0	2	2
507	2	2	0	0	0	0	0
508	1	0	2	1	1	0	0

Dengan melakukan optimasi, jumlah libur dari staf unit farmasi berkisar antara 4 sampai 8 dalam satu bulan. Berdasarkan jadwal hasil optimasi yang dapat dilihat pada bab 6.3 dan alokasi bobot tiap hari libur seperti tabel diatas, didapatkan nilai JFI dengan menggunakan algoritma reinforcement learning adalah sebesar 0,956 untuk unit IGD.

#### 5. Unit Kamar Operasi

Berikut adalah analisis jadwal hasil optimasi unit Kamar Operasi.

**Tabel 6.15 Jumlah shift tiap staf hasil optimasi unit Kamar Operasi**

<b>Kode Staff</b>	<b>P</b>	<b>S</b>	<b>M</b>	<b>MS</b>	<b>LM</b>	<b>L</b>
601	16	10	0	0	0	5
602	17	10	0	0	0	4
603	16	11	0	0	0	4
606	18	9	0	0	0	4
604	20	7	0	0	0	4
605	19	7	0	0	0	5

**Tabel 6.16 Jumlah hari libur tiap staf hasil optimasi unit Kamar Operasi**

	<b>M</b>	<b>Sb</b>	<b>Sn</b>	<b>Sl</b>	<b>Rb</b>	<b>K</b>	<b>J</b>
<b>Bobot</b>	<b>4</b>	<b>2</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
<b>Staff</b>	<b>Libur</b>						
601	2	1	0	1	1	0	0
602	2	0	0	0	1	1	0
603	1	0	1	0	0	1	1
606	1	1	0	1	0	0	1

	M	Sb	Sn	Sl	Rb	K	J
<b>Bobot</b>	4	2	1	1	1	1	1
<b>Staff</b>	<b>Libur</b>						
604	2	1	0	0	1	0	0
605	2	0	1	0	0	1	1

Dengan melakukan optimasi, jumlah libur dari staf unit farmasi berkisar antara 4 sampai 5 dalam satu bulan. Perbedaan yang terpaut sedikit tersebut berakibat terhadap nilai JFI yang dihasilkan menjadi sangat optimal. Berdasarkan jadwal hasil optimasi yang dapat dilihat pada bab 6.3 dan alokasi bobot tiap hari libur seperti tabel diatas, didapatkan nilai JFI dengan menggunakan algoritma reinforcement learning adalah sebesar 0,968 untuk unit Kamar Operasi.

#### 6. Unit SIM & RM

Berikut adalah analisis jadwal hasil optimasi unit SIM & RM.

**Tabel 6.17 Jumlah shift tiap staf hasil optimasi unit SIM & RM**

<b>Kode Staf</b>	<b>P</b>	<b>S</b>	<b>M</b>	<b>MS</b>	<b>LM</b>	<b>L</b>
301	7	10	5	3	2	4
302	9	9	4	2	2	5
303	9	7	4	2	3	6
304	9	5	6	2	3	6
305	7	7	6	3	3	5
306	6	9	6	2	3	5

**Tabel 6.18 Jumlah hari libur tiap staf hasil optimasi unit SIM & RM**

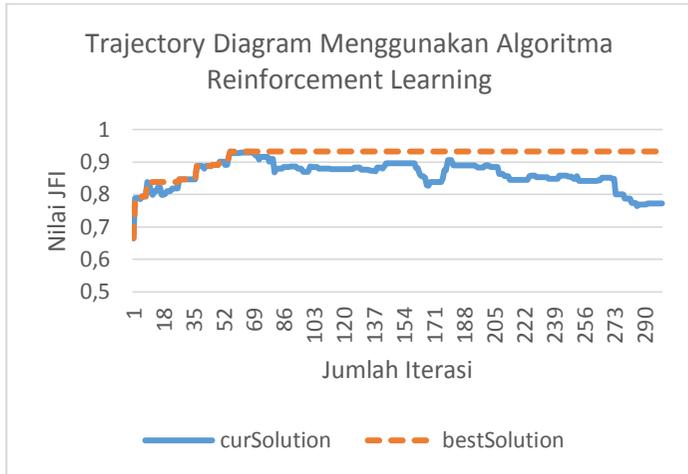
	M	Sb	Sn	Sl	R	K	J
<b>Bobot</b>	<b>4</b>	<b>2</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
<b>Staff</b>	<b>Libur</b>						
301	1	1	0	1	0	1	0
302	1	1	1	1	0	1	0
303	0	2	1	1	1	1	0
304	0	1	1	0	1	1	2
305	1	1	1	0	1	0	1
306	1	0	1	1	1	0	1

Dengan melakukan optimasi, jumlah libur dari staf unit farmasi berkisar antara 2 sampai 3 dalam satu bulan. Perbedaan yang terpaut sedikit tersebut berakibat terhadap nilai JFI yang dihasilkan menjadi sangat optimal. Berdasarkan jadwal hasil optimasi yang dapat dilihat pada bab 6.3 dan alokasi bobot tiap hari libur seperti tabel diatas, didapatkan nilai JFI dengan menggunakan algoritma reinforcement learning adalah sebesar 0,992 untuk unit SIM & RM.

### 6.3.2 Diagram Trajectory

Diagram trajectory digunakan untuk memudahkab dalam melihat pergerakan nilai current solution dan best solution. Berikut merupakan hasil dari diagram trajectory pada keenam unit rumah sakit.

## 1. Unit Farmasi



**Gambar 6.19** Diagram trajectory unit farmasi menggunakan algoritma reinforcement learning

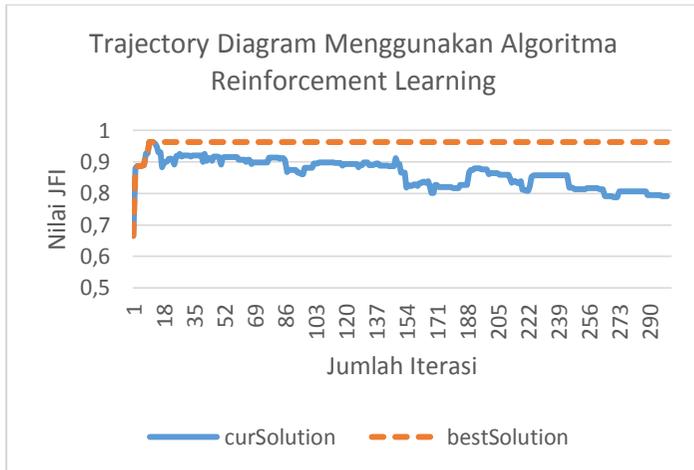
Dari gambar diatas dapat dilihat bahwa terjadi peningkatan yang cukup signifikan dari nilai JFI jadwal hasil otomasi dengan jadwal hasil optimasi. Nilai peningkatan yang terjadi adalah sekitar 0,3 dihitung dari JFI jadwal otomasi sebesar 0,6 sampai JFI jadwal optimasi yaitu 0,9. Best solution menggunakan algoritma reinforcement learning didapatkan pada iterasi sekitar 60 dan setelah itu nilai JFI cenderung stabil dan sedikit menurun mulai iterasi ke-200. Sehingga untuk mendapat solusi yang optimal dibutuhkan setidaknya 100 kali iterasi. Walaupun sebenarnya hasil ini didapatkan dari satu kali program dijalankan, sehingga masih terdapat kemungkinan solusi terbaik didapatkan pada iterasi diatas 100.



**Gambar 6.20 Diagram trajectory unit farmasi menggunakan algoritma hill climbing**

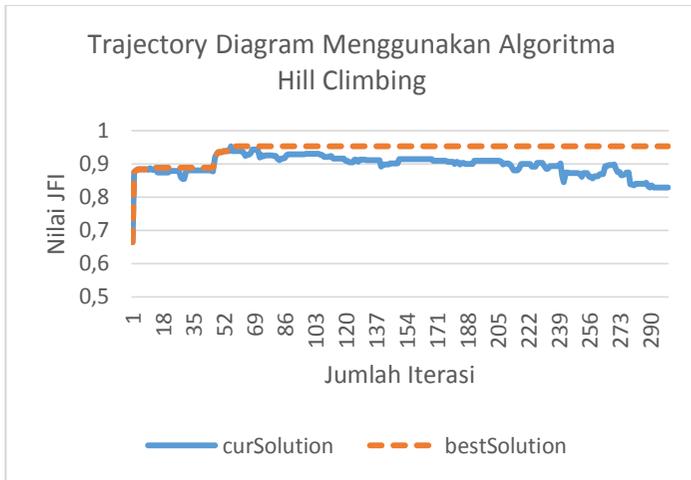
Dari gambar diatas dapat dilihat bahwa terjadi peningkatan yang cukup signifikan dari nilai JFI jadwal hasil otomasi dengan jadwal hasil optimasi. Nilai peningkatan yang terjadi adalah sekitar 0,3 dihitung dari JFI jadwal otomasi sebesar 0,6 sampai JFI jadwal optimasi yaitu 0,9. Best solution menggunakan algoritma hill climbing didapatkan pada iterasi sekitar 80 dan setelah itu nilai JFI cenderung menurun sampai iterasi ke-300. Sehingga untuk mendapat solusi yang optimal dibutuhkan setidaknya 100 kali iterasi. Walaupun sebenarnya hasil ini didapatkan dari satu kali program dijalankan, sehingga masih terdapat kemungkinan solusi terbaik didapatkan pada iterasi diatas 100.

## 2. Unit Bayi/NICU



**Gambar 6.21** Diagram trajectory unit Bayi/NICU menggunakan algoritma reinforcement learning

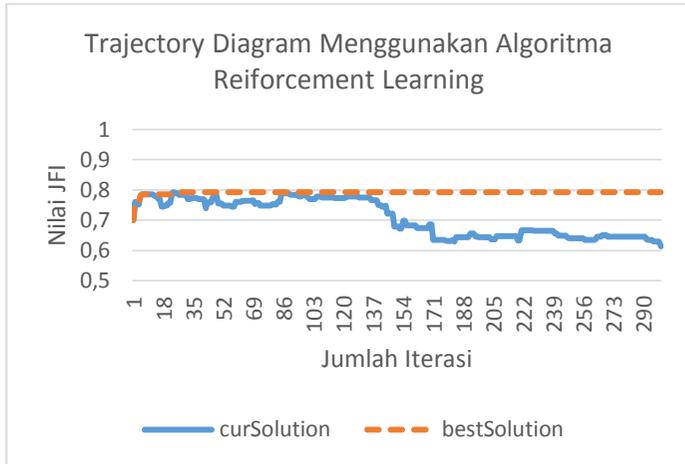
Dari gambar diatas dapat dilihat bahwa terjadi peningkatan yang cukup signifikan dari nilai JFI jadwal hasil otomasi dengan jadwal hasil optimasi. Nilai peningkatan yang terjadi adalah sekitar 0,3 dihitung dari JFI jadwal otomasi sebesar 0,6 sampai JFI jadwal optimasi yaitu 0,9. Best solution menggunakan algoritma reinforcement learning sudah bisa didapatkan pada iterasi sekitar 15 dan setelah itu nilai JFI cenderung stabil dan menurun mulai iterasi ke-90. Sehingga untuk mendapat solusi yang optimal dibutuhkan setidaknya 50 kali iterasi. Walaupun sebenarnya hasil ini didapatkan dari satu kali program dijalankan, sehingga masih terdapat kemungkinan solusi terbaik didapatkan pada iterasi diatas 50.



**6.22 Diagram trajectory unit Bayi/NICU menggunakan algoritma hill climbing**

Dari gambar diatas dapat dilihat bahwa terjadi peningkatan yang cukup signifikan dari nilai JFI jadwal hasil otomasi dengan jadwal hasil optimasi. Nilai peningkatan yang terjadi adalah sekitar 0,3 dihitung dari JFI jadwal otomasi sebesar 0,6 sampai JFI jadwal optimasi yaitu 0,9. Best solution menggunakan algoritma hill climbing sudah bisa didapatkan pada iterasi sekitar 60 dan setelah itu nilai JFI cenderung menurun hingga iterasi ke-300. Sehingga untuk mendapat solusi yang optimal dibutuhkan setidaknya 100 kali iterasi. Walaupun sebenarnya hasil ini didapatkan dari satu kali program dijalankan, sehingga masih terdapat kemungkinan solusi terbaik didapatkan pada iterasi diatas 100.

### 3. Unit Gizi



**Gambar 6.23 Diagram trajectory unit gizi menggunakan algoritma reinforcement learning**

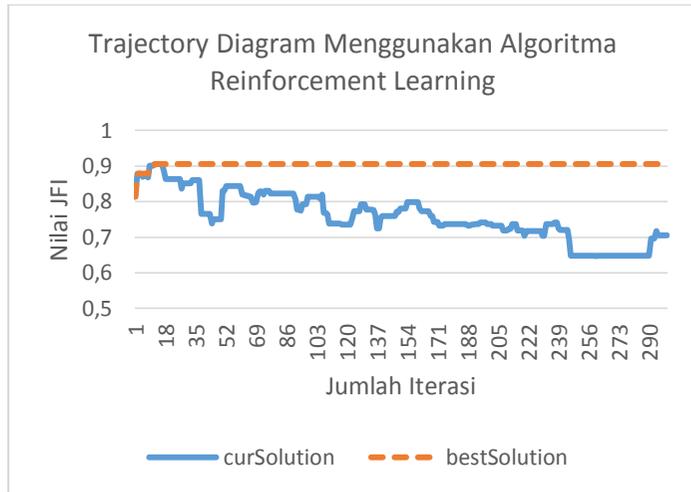
Dari gambar diatas dapat dilihat bahwa terjadi peningkatan dari nilai JFI jadwal hasil otomasi dengan jadwal hasil optimasi. Nilai peningkatan yang terjadi adalah sekitar 0,1 dihitung dari JFI jadwal otomasi sebesar 0,7 sampai JFI jadwal optimasi yaitu 0,8. Best solution menggunakan algoritma reinforcement learning sudah bisa didapatkan pada iterasi sekitar 90 dan setelah itu nilai JFI cenderung menurun hingga iterasi ke-300. Sehingga untuk mendapat solusi yang optimal dibutuhkan setidaknya 100 kali iterasi. Walaupun sebenarnya hasil ini didapatkan dari satu kali program dijalankan, sehingga masih terdapat kemungkinan solusi terbaik didapatkan pada iterasi diatas 100.



**Gambar 6.24 Diagram trajectory unit gizi menggunakan algoritma reinforcement learning**

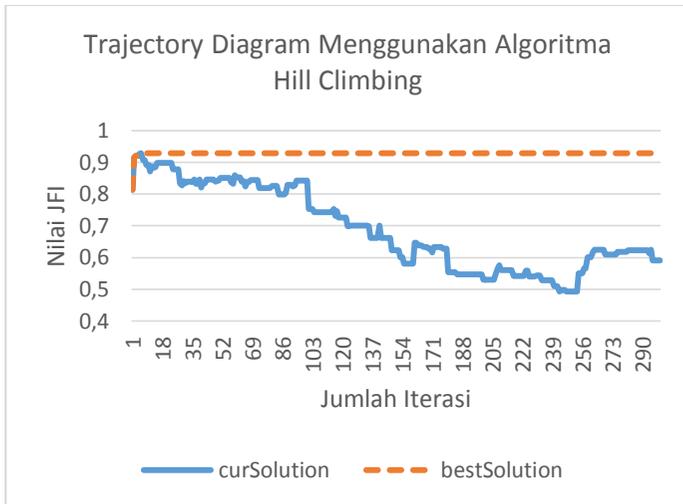
Dari gambar diatas dapat dilihat bahwa terjadi peningkatan dari nilai JFI jadwal hasil otomasi dengan jadwal hasil optimasi. Nilai peningkatan yang terjadi adalah sekitar 0,1 dihitung dari JFI jadwal otomasi sebesar 0,7 sampai JFI jadwal optimasi yaitu 0,8. Best solution menggunakan algoritma hill climbing sudah bisa didapatkan pada iterasi sekitar 90 dan setelah itu nilai JFI cenderung menurun hingga iterasi ke-300. Sehingga untuk mendapat solusi yang optimal dibutuhkan setidaknya 100 kali iterasi. Walaupun sebenarnya hasil ini didapatkan dari satu kali program dijalankan, sehingga masih terdapat kemungkinan solusi terbaik didapatkan pada iterasi diatas 100.

#### 4. Unit IGD



**Gambar 6.25** Diagram trajectory unit IGD menggunakan algoritma reinforcement learning

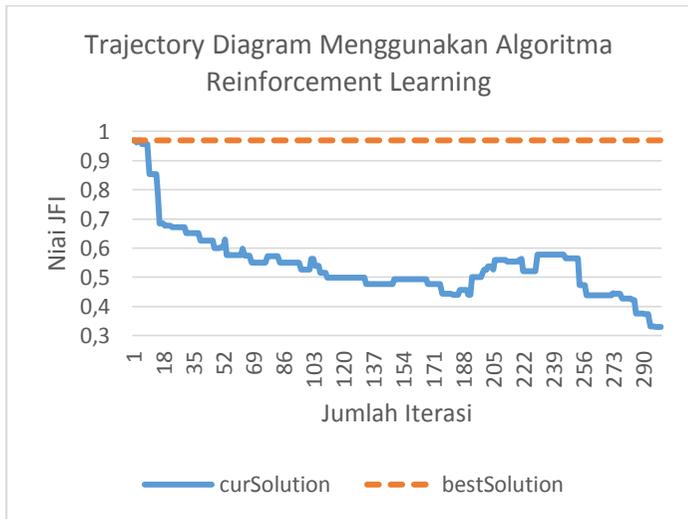
Dari gambar diatas dapat dilihat bahwa terjadi peningkatan dari nilai JFI jadwal hasil otomasi dengan jadwal hasil optimasi. Nilai peningkatan yang terjadi adalah sekitar 0,1 dihitung dari JFI jadwal otomasi sebesar 0,8 sampai JFI jadwal optimasi yaitu 0,9. Best solution menggunakan algoritma reinforcement learning sudah bisa didapatkan pada iterasi sekitar 15 dan setelah itu nilai JFI cenderung menurun hingga iterasi ke-300. Sehingga untuk mendapat solusi yang optimal dibutuhkan setidaknya 50 kali iterasi. Walaupun sebenarnya hasil ini didapatkan dari satu kali program dijalankan, sehingga masih terdapat kemungkinan solusi terbaik didapatkan pada iterasi diatas 50.



**Gambar 6.26 Diagram trajectory unit IGD menggunakan algoritma hill climbing**

Dari gambar diatas dapat dilihat bahwa terjadi peningkatan dari nilai JFI jadwal hasil otomasi dengan jadwal hasil optimasi. Nilai peningkatan yang terjadi adalah sekitar 0,1 dihitung dari JFI jadwal otomasi sebesar 0,8 sampai JFI jadwal optimasi yaitu 0,9. Best solution menggunakan algoritma hill climbing sudah bisa didapatkan pada iterasi sekitar 10 dan setelah itu nilai JFI cenderung menurun hingga iterasi ke-300. Sehingga untuk mendapat solusi yang optimal dibutuhkan setidaknya 50 kali iterasi. Walaupun sebenarnya hasil ini didapatkan dari satu kali program dijalankan, sehingga masih terdapat kemungkinan solusi terbaik didapatkan pada iterasi diatas 50.

## 5. Unit Kamar Operasi



**Gambar 6.27 Diagram trajectory unit Kamar Operasi menggunakan algoritma reinforcement learning**

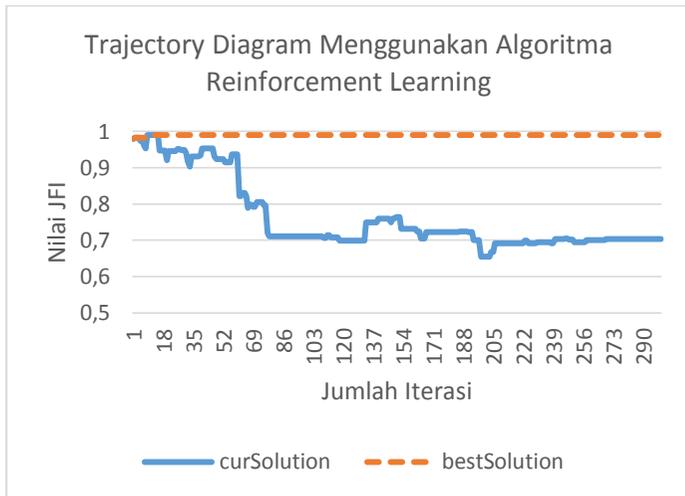
Dari gambar diatas dapat dilihat bahwa terjadi peningkatan yang sangat kecil dari nilai JFI jadwal hasil otomasi dengan jadwal hasil optimasi. Nilai peningkatan yang terjadi tidak dapat terlihat dengan jelas karena nilai peningkatannya yang sangat kecil. Hal tersebut dikarenakan solusi awal hasil otomasi bisa dikatakan sudah sangat baik, sehingga sulit untuk menghasilkan peningkatan solusi yang signifikan. Best solution menggunakan algoritma reinforcement learning sudah bisa didapatkan pada sekitar iterasi ke-5 dan setelah itu nilai JFI cenderung menurun hingga iterasi ke-300. Sehingga untuk mendapat solusi yang optimal hanya dibutuhkan setidaknya 10 kali iterasi. Walaupun sebenarnya hasil ini didapatkan dari satu kali program dijalankan, sehingga masih terdapat kemungkinan solusi terbaik didapatkan pada iterasi diatas 10.



**Gambar 6.28** Diagram trajectory unit Kamar Operasi menggunakan algoritma hill climbing

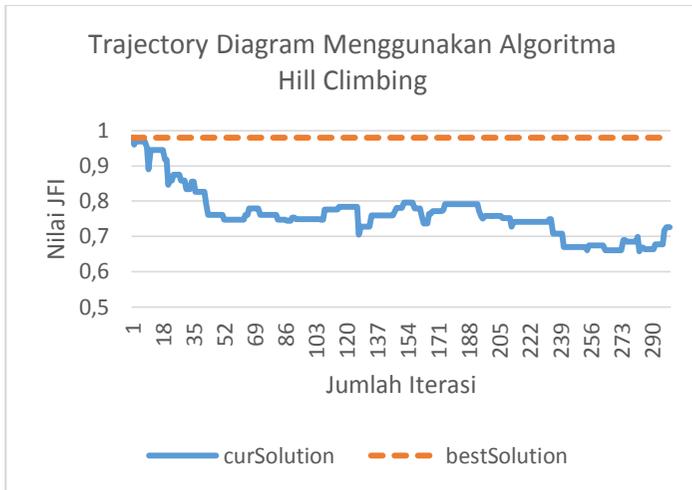
Dari gambar diatas dapat dilihat bahwa terjadi peningkatan yang sangat kecil dari nilai JFI jadwal hasil otomasi dengan jadwal hasil optimasi. Nilai peningkatan yang terjadi tidak dapat terlihat dengan jelas karena nilai peningkatannya yang sangat kecil. Hal tersebut dikarenakan solusi awal hasil otomasi bisa dikatakan sudah sangat baik, sehingga sulit untuk menghasilkan peningkatan solusi yang signifikan. Best solution menggunakan algoritma hill climbing bahkan sudah bisa didapatkan pada iterasi pertama dan setelah itu nilai JFI cenderung menurun hingga iterasi ke-300. Sehingga untuk mendapat solusi yang optimal hanya dibutuhkan setidaknya 10 kali iterasi. Walaupun sebenarnya hasil ini didapatkan dari satu kali program dijalankan, sehingga masih terdapat kemungkinan solusi terbaik didapatkan pada iterasi diatas 10.

## 6. Unit SIM & RM



**Gambar 6.29** Diagram trajectory unit SIM & RM menggunakan algoritma reinforcement learning

Dari gambar diatas dapat dilihat bahwa terjadi peningkatan yang sangat kecil dari nilai JFI jadwal hasil otomasi dengan jadwal hasil optimasi. Nilai peningkatan yang terjadi tidak dapat terlihat dengan jelas karena nilai peningkatannya yang sangat kecil. Hal tersebut dikarenakan solusi awal hasil otomasi bisa dikatakan sudah sangat baik, sehingga sulit untuk menghasilkan peningkatan solusi yang signifikan. Best solution menggunakan algoritma reinforcement learning sudah bisa didapatkan pada sekitar iterasi ke-5 dan setelah itu nilai JFI cenderung menurun hingga iterasi ke-300. Sehingga untuk mendapat solusi yang optimal hanya dibutuhkan setidaknya 10 kali iterasi. Walaupun sebenarnya hasil ini didapatkan dari satu kali program dijalankan, sehingga masih terdapat kemungkinan solusi terbaik didapatkan pada iterasi diatas 10.



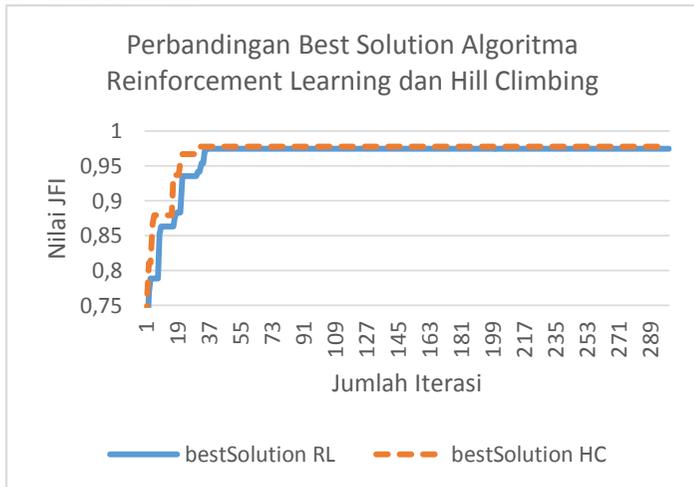
**Gambar 6.30 Diagram trajectory unit SIM & RM menggunakan algoritma hill climbing**

Dari gambar diatas dapat dilihat bahwa terjadi peningkatan yang sangat kecil dari nilai JFI jadwal hasil otomasi dengan jadwal hasil optimasi. Nilai peningkatan yang terjadi tidak dapat terlihat dengan jelas karena nilai peningkatannya yang sangat kecil. Hal tersebut dikarenakan solusi awal hasil otomasi bisa dikatakan sudah sangat baik, sehingga sulit untuk menghasilkan peningkatan solusi yang signifikan. Best solution menggunakan algoritma hill climbing bahkan sudah bisa didapatkan pada awal iterasi dan setelah itu nilai JFI cenderung menurun hingga iterasi ke-300. Sehingga untuk mendapat solusi yang optimal hanya dibutuhkan setidaknya 10 kali iterasi. Walaupun sebenarnya hasil ini didapatkan dari satu kali program dijalankan, sehingga masih terdapat kemungkinan solusi terbaik didapatkan pada iterasi diatas 10.

### 6.3.3 Perbandingan hasil algoritma reinforcement learning dan hill climbing

Perbandingan dilakukan untuk melihat algoritma mana yang memiliki best solution paling tinggi. Berikut merupakan diagram perbandingan solusi kedua algoritma pada keenam unit pada rumah sakit:

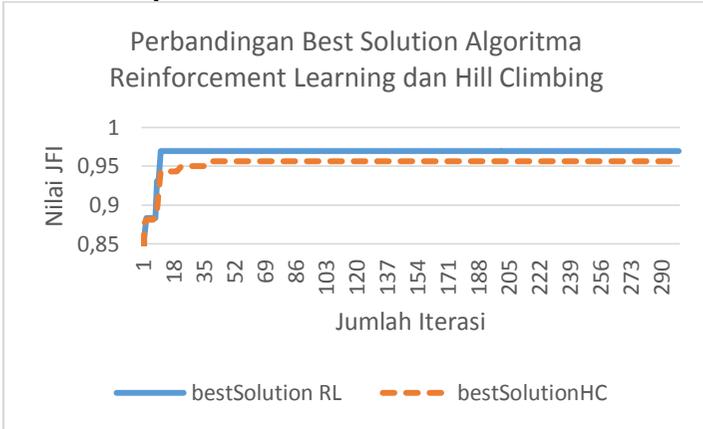
#### 1. Unit Farmasi



**Gambar 6.31 Perbandingan best solution algoritma reinforcement learning dan hill climbing unit farmasi**

Dari gambar diatas dapat terlihat bahwa algoritma hill climbing memiliki nilai best solution yang sedikit lebih baik dari pada algoritma reinforcement learning. Best solution dari kedua algoritma yang dipakai bernilai sekitar 0,98 dan best solution keduanya juga didapatkan pada iterasi sekitar ke-40.

2. Unit Bayi/NICU



**Gambar 6.32 Perbandingan best solution algoritma reinforcement learning dan hill climbing unit Bayi/NICU**

Terlihat bahwa algoritma reinforcement learning memiliki nilai best solution yang sedikit lebih baik dari pada algoritma hill climbing dengan nilai sekitar 0,97.

3. Unit Gizi



**Gambar 6.33 Perbandingan best solution algoritma reinforcement learning dan hill climbing unit gizi**

Terlihat bahwa algoritma reinforcement learning memiliki nilai best solution yang sedikit lebih baik dari pada algoritma hill climbing dengan nilai sekitar 0,85.

#### 4. Unit IGD



**Gambar 6.34 Perbandingan best solution algoritma reinforcement learning dan hill climbing unit IGD**

Terlihat bahwa algoritma hill climbing memiliki nilai best solution yang sedikit lebih baik dari pada algoritma reinforcement dengan nilai sekitar 0,98. best solution keduanya juga didapatkan pada iterasi sekitar ke-20.

## 5. Unit Kamar Operasi



**Gambar 6.35 Perbandingan best solution algoritma reinforcement learning dan hill climbing unit Kamar Operasi**

Terlihat bahwa algoritma hill climbing memiliki nilai best solution yang sedikit lebih baik dari pada algoritma reinforcement learning dengan nilai sekitar 0,99.

## 6. Unit SIM & RM



**Gambar 6.36 Perbandingan best solution algoritma reinforcement learning dan hill climbing unit SIM & RM**

Terlihat bahwa algoritma reinforcement learning memiliki nilai best solution yang sedikit lebih baik dan bisa dikatakan sempurna jika dibandingkan dengan algoritma hill climbing dengan nilai JIF maksimal yaitu 1.

#### 6.3.4 Statistik

Berikut merupakan hasil statistik nilai minimum, maximum, dan rata-rata dari total 21 kali program di jalankan:

##### 1. Unit Farmasi

**Tabel 6.19 Statistik Algoritma Reinforcement Learning**

Min	Max	Average
0,665158	0,974642	0,898775

**Tabel 6.20 Statistik Algoritma Hill Climbing**

Min	Max	Average
0,665158	0,977606	0,891378

Dari tabel diatas, terlihat bahwa nilai minimum JFI dari dua algoritma yang dipakai bernilai sama karena nilai tersebut berasal dari nilai JFI hasil otomasi. Dan dari total 21 kali program di jalankan, rata-rata nilai JFI optimasi menggunakan reinforcement learning sedikit lebih tinggi dari pada menggunakan hill climbing yaitu sebesar 0,898775.

##### 2. Unit Bayi/NICU

**Tabel 6.21 Statistik Algoritma Reinforcement Learning Unit Bayi/Nicu**

Min	Max	Average
0,664773	0,969958	0,920596

**Tabel 6.22 Statistik Algoritma Hill Climbing Unit Bayi/Nicu**

Min	Max	Average
0,664773	0,956274	0,914694

Dari tabel diatas, terlihat bahwa nilai minimum JFI dari dua algoritma yang dipakai bernilai sama karena nilai tersebut berasal dari nilai JFI hasil otomasi. Dan dari total 21 kali program di jalankan, rata-rata nilai JFI optimasi menggunakan reinforcement learning sedikit lebih tinggi dari pada menggunakan hill climbing yaitu sebesar 0,920596.

### 3. Unit Gizi

**Tabel 6.23 Statistik Algoritma Reinforcement Learning Unit Gizi**

Min	Max	Average
0,70004	0,848634	0,787794

**Tabel 6.24 Statistik Algoritma Hill Climbing Unit Gizi**

Min	Max	Average
0,70004	0,833895	0,780947

Dari tabel diatas, terlihat bahwa nilai minimum JFI dari dua algoritma yang dipakai bernilai sama karena nilai tersebut berasal dari nilai JFI hasil otomasi. Dan dari total 21 kali program di jalankan, rata-rata nilai JFI optimasi menggunakan reinforcement learning sedikit lebih tinggi dari pada menggunakan hill climbing yaitu sebesar 0,787794.

## 4. Unit IGD

**Tabel 6.25 Statistik Algoritma Reinforcement Learning Unit IGD**

Min	Max	Average
0,813278	0,96305	0,919193

**Tabel 6.26 Statistik Algoritma Hill Climbing Unit IGD**

Min	Max	Average
0,813278	0,979977	0,931647

Dari tabel diatas, terlihat bahwa nilai minimum JFI dari dua algoritma yang dipakai bernilai sama karena nilai tersebut berasal dari nilai JFI hasil otomasi. Dan dari total 21 kali program di jalankan, rata-rata nilai JFI optimasi menggunakan hill climbing sedikit lebih tinggi dari pada menggunakan reinforcement learning yaitu sebesar 0,931647.

## 5. Unit Kamar Operasi

**Tabel 6.27 Statistik Algoritma Reinforcement Learning Unit Kamar Operasi**

Min	Max	Average
0,813278	0,981669	0,968193

**Tabel 6.28 Statistik Algoritma Hill Climbing Unit Kamar Operasi**

Min	Max	Average
0,968559	0,988359	0,971596

Dari tabel diatas, terlihat bahwa nilai minimum JFI dari dua algoritma yang dipakai bernilai sama karena nilai tersebut berasal dari nilai JFI hasil otomasi. Dan dari total 21 kali program di jalankan,

rata-rata nilai JFI optimasi menggunakan hill climbing sedikit lebih tinggi dari pada menggunakan reinforcement learning yaitu sebesar 0,971596.

## 6. Unit SIM & RM

**Tabel 6.29 Statistik Algoritma Reinforcement Learning Unit SIM & RM**

Min	Max	Average
0,979592	1	0,988959

**Tabel 6.30 Statistik Algoritma Hill Climbing Unit SIM & RM**

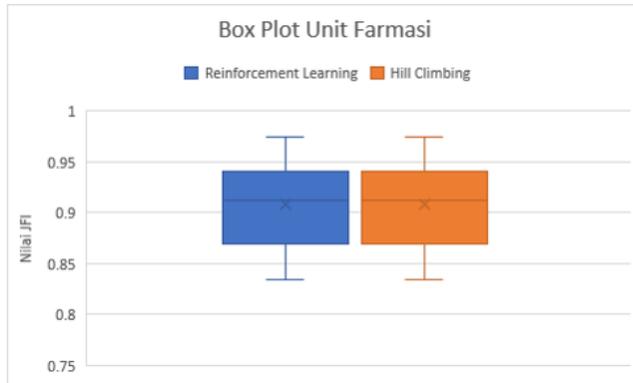
Min	Max	Average
0,979592	0,994819	0,985949

Dari tabel diatas, terlihat bahwa nilai minimum JFI dari dua algoritma yang dipakai bernilai sama karena nilai tersebut berasal dari nilai JFI hasil otomasi. Dan dari total 21 kali program di jalankan, rata-rata nilai JFI optimasi menggunakan reinforcement learning sedikit lebih tinggi dari pada menggunakan hill climbing yaitu sebesar 0,988959.

### 6.3.5 Box Plot

Boxplot adalah salah satu cara dalam statistik deskriptif untuk menggambarkan secara grafik dari data numeris melalui lima ukuran yaitu nilai observasi terkecil, kuartil terendah atau kuartil pertama, median (Q2) atau nilai pertengahan, kuartil tertinggi atau kuartil ketiga (Q3), dan nilai observasi terbesar. Berikut merupakan box plot dari keenam unit pada rumah sakit:

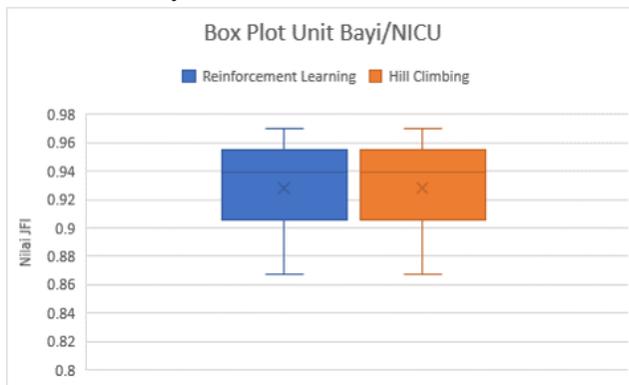
## 1. Unit Farmasi



**Gambar 6.37 Box Plot unit farmasi**

Dari statistik, untuk algoritma reinforcement learning memiliki nilai kuartil 1 sebesar 0,871649, median 0,912062, kuartil 3 0,931922, solusi minimum 0,665158, dan solusi maksimal 0,974642. Sedangkan untuk algoritma hill climbing memiliki nilai kuartil 1 sebesar 0,875773, median 0,900602, kuartil 3 0,94761, solusi minimum 0,66515, dan solusi maksimum 0,977606.

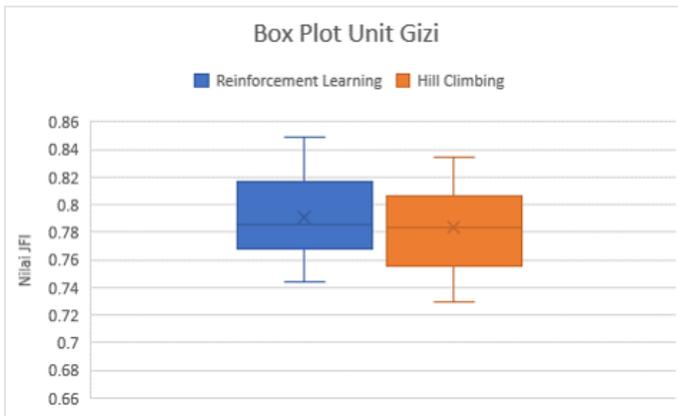
## 2. Unit Bayi/Nicu



**Gambar 6.38 Box Plot unit Bayi/NICU**

Dari statistik, untuk algoritma reinforcement learning memiliki nilai kuartil 1 sebesar 0,914003, median 0,939199, kuartil 3 0,949226, solusi minimum 0,664773, dan solusi maksimal 0,664773. Sedangkan untuk algoritma hill climbing memiliki nilai kuartil 1 sebesar 0,915124, median 0,940761, kuartil 3 0,957511, solusi minimum 0,664773, dan solusi maksimum 0,956274.

### 3. Unit Gizi



**Gambar 6.39 Box Plot unit gizi**

Dari statistik, untuk algoritma reinforcement learning memiliki nilai kuartil 1 sebesar 0,767918, median 0,785365, kuartil 3 0,810719, solusi minimum 0,70004, dan solusi maksimal 0,848634. Sedangkan untuk algoritma hill climbing memiliki nilai kuartil 1 sebesar 0,75845, median 0,783181, kuartil 3 0,803757, solusi minimum 0,70004, dan solusi maksimum 0,833895.

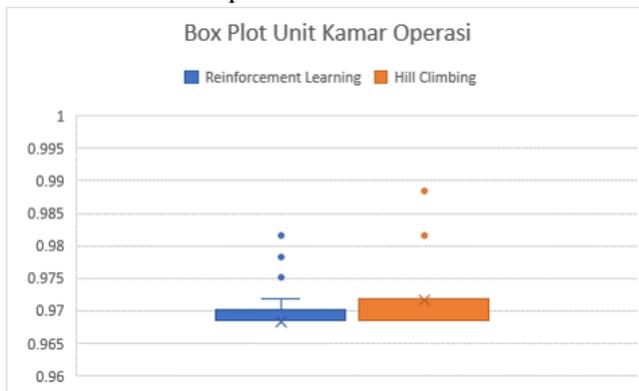
#### 4. Unit IGD



**Gambar 6.40** Box Plot unit IGD

Dari statistik, untuk algoritma reinforcement learning memiliki nilai kuartil 1 sebesar 0,905822, median 0,921388, kuartil 3 0,92803, solusi minimum 0,813278, dan solusi maksimal 0,96305. Sedangkan untuk algoritma hill climbing memiliki nilai kuartil 1 sebesar 0,914171, median 0,933765, kuartil 3 0,955257, solusi minimum 0,813278, dan solusi maksimum 0,979977.

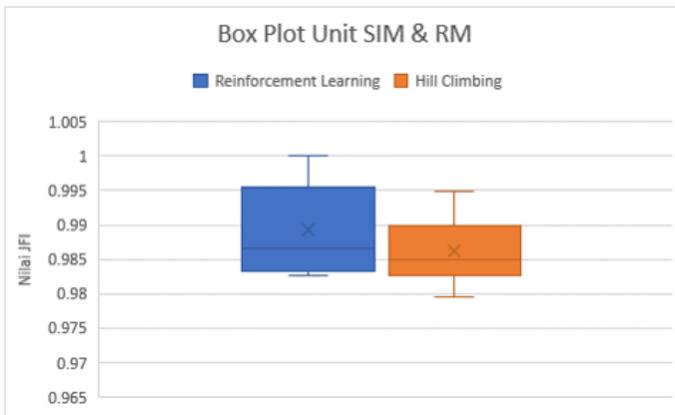
#### 5. Unit Kamar Operasi



**Gambar 6.41** Box Plot unit Kamar Operasi

Dari statistik, untuk algoritma reinforcement learning memiliki nilai kuartil 1 sebesar 0,968559, median 0,968559, kuartil 3 0,968559, solusi minimum 0,813278, dan solusi maksimal 0,981669. Sedangkan untuk algoritma hill climbing memiliki nilai kuartil 1 sebesar 0,968559, median 0,968559, kuartil 3 0,971803, solusi minimum 0,968559, dan solusi maksimum 0,988359.

## 6. Unit SIM & RM



**Gambar 6.42 Box Plot unit SIM & RM**

Dari statistik, untuk algoritma reinforcement learning memiliki nilai kuartil 1 sebesar 0,98321, median 0,986628, kuartil 3 0,993984, solusi minimum 0,979592, dan solusi maksimal 1. Sedangkan untuk algoritma hill climbing memiliki nilai kuartil 1 sebesar 0,982704, median 0,985003, kuartil 3 0,989945, solusi minimum 0,979592, dan solusi maksimum 0,994819.

## 6.4 Perbandingan Nilai Jain Fairness Index

Hasil Jain Fairness Index untuk setiap jadwal yang dihasilkan pada keenam unit rumah sakit dapat dilihat sebagai berikut:

### 1. Unit Farmasi

**Tabel 6.31 Perbandingan JFI unit farmasi**

<b>Jadwal</b>	<b>Nilai JFI</b>
Jadwal Eksisting	0,80798005
Jadwal Otomasi	0,665
Jadwal Optimasi (Reinforcement Learning)	0,974642
Jadwal Optimasi (Hill Climbing)	0,977606

Dari tabel diatas, dapat diketahui bahwa solusi terbaik untuk unit farmasi adalah optimasi dengan menggunakan algoritma hill climbing.

### 2. Unit Bayi/NICU

**Tabel 6.32 Perbandingan JFI unit Bayi/NICU**

<b>Jadwal</b>	<b>Nilai JFI</b>
Jadwal Eksisting	0,667
Jadwal Otomasi	0,664
Jadwal Optimasi (Reinforcement Learning)	0,969958
Jadwal Optimasi (Hill Climbing)	0,956274

Dari tabel diatas, dapat diketahui bahwa solusi terbaik untuk unit bayi/NICU adalah optimasi dengan menggunakan algoritma reinforcement learning.

## 3. Unit Gizi

**Tabel 6.33 Perbandingan JFI unit gizi**

<b>Jadwal</b>	<b>Nilai JFI</b>
Jadwal Eksisting	0,901226
Jadwal Otomasi	0,700
Jadwal Optimasi (Reinforcement Learning)	0,848634
Jadwal Optimasi (Hill Climbing)	0,833895

Dari tabel diatas, dapat diketahui bahwa solusi terbaik untuk unit farmasi adalah optimasi dengan menggunakan algoritma hill climbing.

## 4. Unit IGD

**Tabel 6.34 Perbandingan JFI unit IGD**

<b>Jadwal</b>	<b>Nilai JFI</b>
Jadwal Eksisting	0,918402778
Jadwal Otomasi	0,813
Jadwal Optimasi (Reinforcement Learning)	0,96305
Jadwal Optimasi (Hill Climbing)	0,979977

Dari tabel diatas, dapat diketahui bahwa solusi terbaik untuk unit farmasi adalah optimasi dengan menggunakan algoritma hill climbing.

## 5. Unit Kamar Operasi

**Tabel 6.35 Perbandingan JFI unit Kamar Operasi**

<b>Jadwal</b>	<b>Nilai JFI</b>
Jadwal Eksisting	0,806983
Jadwal Otomasi	0,968
Jadwal Optimasi (Reinforcement Learning)	0,981669
Jadwal Optimasi (Hill Climbing)	0,988359

Dari tabel diatas, dapat diketahui bahwa solusi terbaik untuk unit farmasi adalah optimasi dengan menggunakan algoritma hill climbing.

6. Unit SIM & RM

**Tabel 6.36 Perbandingan JFI unit SIM & RM**

<b>Jadwal</b>	<b>Nilai JFI</b>
Jadwal Eksisting	0,746714
Jadwal Otomasi	0,979
Jadwal Optimasi (Reinforcement Learning)	1
Jadwal Optimasi (Hill Climbing)	0,994819

Dari tabel diatas, dapat diketahui bahwa solusi terbaik untuk unit farmasi adalah optimasi dengan menggunakan algoritma reinforcement learning.

*(Halaman sengaja dikosongkan)*

## **BAB VII**

### **KESIMPULAN DAN SARAN**

Pada bab ini akan dibahas mengenai kesimpulan yang dapat diambil dari semua proses yang telah dilakukan dan saran yang dapat diberikan untuk pengembangan dari Tugas Akhir ini.

#### **7.1 Kesimpulan**

Kesimpulan dalam penelitian ini ialah berupa hasil analisis dari optimasi penjadwalan staf menggunakan algoritma reinforcement learning, yaitu antara lain:

1. Penulis dapat membandingkan tingkat keadilan/fairness antara staff rumah sakit dengan melihat hasil Jain fairness index yang di hasilkan oleh penjadwalan manual dan penjadwalan otomatis yang telah di optimasi menggunakan metode Reinforcement Learning Hyper-Heuristics.
2. Jadwal yang telah optimal tersebut dipilih dari hasil Jain Fairness Index yang paling besar dengan menggunakan Hard Constraint yang tidak boleh dilanggar dan Soft Constraint yang ada dilanggar dan ada yang terpenuhi.
3. Untuk unit farmasi, solusi paling optimal didapatkan dengan menggunakan algoritma hill climbing dengan nilai JFI 0,977606
4. Untuk unit bayi/NICU, solusi paling optimal didapatkan dengan menggunakan algoritma reinforcement learning dengan nilai JFI 0,969958
5. Untuk unit gizi, solusi paling optimal berasal dari jadwal eksisting rumah sakit dengan nilai JFI 0,901226
6. Untuk unit IGD, solusi paling optimal didapatkan dengan menggunakan algoritma hill climbing dengan nilai JFI 0,979977
7. Untuk unit Kamar Operasi, solusi paling optimal didapatkan dengan menggunakan algoritma hill climbing dengan nilai JFI 0,988359

8. Untuk unit SIM & RM, solusi paling optimal didapatkan dengan menggunakan algoritma reinforcement learning dengan nilai JFI sempurna yaitu 1.

## 7.2 Saran

Saran yang dapat diberikan berdasarkan proses uji coba, penarikan kesimpulan, dan batasan masalah dari tugas akhir ini adalah :

1. Pada penelitian ini, data Penjadwalan yang digunakan adalah data pada bulan November 2017 untuk menentukan model penjadwalan manual bulan Desember 2017. Sangat disarankan untuk penelitian selanjutnya menggunakan data 1 tahun sehingga hasil yang diberikan lebih akurat
2. Pada penelitian ini, terdapat beberapa batasan yang seharusnya menjadi hard constraint seperti pada unit gizi dimana ada shift yang mengharuskan shift tersebut dilakukan berdasarkan gender.
3. Khusus ntuk unit gizi yang memiliki banyak staf dan pekerjaan yang sangat variatif, sebaiknya perhitungan nilai JFI dibagi berdasarkan pekerjaannya sehingga dapat menghasilkan solusi yang lebih optimal.
4. Penelitian ini hanya menggunakan dua low-level heuristic. Penelitian selanjutnya sangat dianjurkan untuk menggunakan low-level heuristic yang lebih bervariasi seperti insertion, inversion, dan yang lainnya.
5. Penelitian ini hanya menggunakan algoritma reinforcement learning. Penelitian selanjutnya dapat mencoba menggunakan metode lain seperti Threshold Accepting dan late acceptance Hill Climbing dan melakukan perbandingan untuk menemukan metode terbaik dalam penjadwalan staf rumah sakit.
6. Penjadwalan staff rumah sakit ini dapat di kembangkan menjadi aplikasi menggunakan Bahasa pemrograman yang lain seperti PHP sesuai dengan kebutuhan rumah sakit.

## DAFTAR PUSTAKA

- [1] " Tentang Kami: RSIA Kendangsari," [Online]. Available: <http://rsia.kendangsari.com/tentang-kami/>. [Accessed 25 September 2017].
- [2] J. Junkeirmer, "The Nurse Scheduling Problem".
- [3] Ilmi, RR, Mahmudy, WF and Ratnawati, Optimasi Penjadwalan Staf Menggunakan Algoritma Genetika, 2015.
- [4] M. Hadwan, M. Ayo, N. R. Sabar and R. Qu, A harmony search algorithm for nurse rostering problems, 2013.
- [5] C.-C. Tsai and S. H. Li, A two-stage modeling with genetic algorithms for the nurse scheduling problem, 2008.
- [6] Cheang, B., Li, H., A. Lim, & Rodrigues and B, Nurse rostering problems - a bibliographic survey, 2003.
- [7] Burke, E. K. and e. al, Hyper-heuristics: A survey of the state of the art, 2013.
- [8] Burke, E., Kendall, G., Newall, J., Hart, E., Ross, P., & Schulenburg and S, Hyper-heuristics: An emerging direction in modern search technology, 2003.
- [9] A. Muklason, Hyper-heuristics and fairness in examination timetabling problems, 2017.
- [10] Burke, E. K., D. Causmaecker, P., Berghe, G. V., V. Landeghem and H, The state of the art of nurse rostering, 2004.
- [11] A. Muklason, Solver Penjadwal Ujian Otomatis Dengan Algoritma Maximal Clique dan Hyper-heuristics, 2017.
- [12] E. K. Burke, M. Hyde, G. Kendall, G. Ochoa and E. O. zcan, A classification of Hyper-heuristic Approaches, 2010.
- [13] S. Asta, E. Ozcan and T. Curtois, A tensor based hyper-heuristic for nurse rostering, 2016.

- [14] A. Muklason, A. J. Parkes, B. McCollum and E. Ozcan, Initial Results on Fairness in Examination Timetabling, 2013.
- [15] L. Maulida, N. Fauziah, A. Susanti and S. B. H, Kecerdasan Buatan Simple Hill Climbing, 2014.
- [16] M. Khari, Empirical Evaluation of Hill, 2017.

## BIODATA PENULIS



Penulis bernama lengkap Fata Hirzi Abi Karami, lahir di Bogor, 26 September 1996. Penulis merupakan anak perama dari pasangan suami istri Budi Prijanto dan Agustin Rusiana Sari. Riwayat Pendidikan penulis yaitu TK Islam PB Sudirman, SD Islam PB Sudirman, SMPN 103 Jakarta, SMAN 39 Jakarta, dan akhirnya penulis masuk menjadi mahasiswa Sistem Informasi angkatan

2014 melalui jalur Program Kemitraan dan Mandiri pada Institut Teknologi Sepuluh Nopember Surabaya. Selama perkuliahan, penulis aktif sebagai panitia kegiatan baik tingkat jurusan maupun fakultas dengan menjadi panitia Information System Expo (ISE), FTIf Festival, dan Advance Media Schooling FTIf. Penulis juga aktif berorganisasi di Badan Eksekutif Mahasiswa FTIf di mana penulis sempat menjabat sebagai kepala departemen Information Media. Di Jurusan Sistem Informasi, penulis mengambil bidang minat Rekayasa dan Intelegensi Bisnis. Penulis dapat dihubungi melalui email [fatahirzi@gmail.com](mailto:fatahirzi@gmail.com).

## LAMPIRAN

### Lampiran A: Interview Protocol

#### Informasi interview

Interviewer : Rizka Pordella, Indriarti Kusumanita, Fachrur Zaffrinda, Fata Hirzi, & Zuli Maulidati  
Narasumber : Ibu Silvy  
Hari, Tanggal : Senin, 28 Oktober 2017  
Pukul : 10.30-12.30  
Lokasi : RSIA Merr Kendangsari

#### Informasi narasumber

Nama : Bu Sylvy Medtasya Dzykrzyanka, S. Farm,  
M. Farm. Klin, APT, MARS  
Jabatan : Kepala Bagian  
Divisi : Sumber Daya Manusia (SDM) & Hukum  
Instansi : RSIA Merr Kendangsari  
Lama Bekerja : 3.5 tahun

#### Penjelasan interview

1. Interview ini bertujuan untuk salah satu sumber data untuk tugas akhir dengan judul “Optimasi Penjadwalan Perawat Dengan Menggunakan Algoritma Tabu Search, Late acceptance, Reinforcement Learning, Self adaptive learning, simulated annealing Based Hyper-Heuristics (Studi Kasus: Rumah Sakit Ibu Dan Anak Kendangsari)” yang dimaksudkan agar peneliti bisa mendapatkan gambaran mengenai permasalahan penjadwalan perawat yang ada di RSIA Kendangsari Surabaya.
2. Dengan melakukan interview ini diharapkan peneliti mendapatkan informasi mengenai penjadwalan perawat yang sudah diterapkan Rumah Sakit Ibu dan Anak kendangsari Surabaya

3. Untuk menjaga dan menjamin kerahasiaan, maka data – data yang bersifat pribadi akan dirahasiakan oleh peneliti.

### **Interview list**

#### **Senior management**

Berikut adalah daftar pertanyaan untuk Senior Manager sebagai penanggungjawab proses bisnis yang dijalankan pada RSIA kendangsari Surabaya:

1. Berapa jumlah Staff yang dimiliki oleh RS?
 

Answer: terdapat 124 Staff (Keseluruhan medis dan non medis)
2. Ada berapakan Tipe staff perawat dipekerjakan? (Skill Type)?
 

Answer: klasifikasi Perawat dilakukan berdasarkan Pelatihan keterampilan yang dimiliki masing masing perawat. Secara keseluruhan perawat yang dimiliki oleh RSIA hanya memiliki sampai PK2 (Pelatihan Keterampilan tingkat 2) sedangkan untuk perawat sebiior atau kepalaperawat memiliki PK3 (Pelatihan Keterampilan tingkat 3)
3. Berapa jumlah ward atau bangsal yang dimiliki oleh RS?
 

Answer : RSIA memiliki 14 unit atau bangsal, terdapat 6 unit RSIA yang memiliki pattern yang unik dalam menjadwalkan staff baik medis maupun non medis yaitu:

  - Kamar Operasi
  - Ruang bayi dan NICU
  - SIM dan RM
  - Instalasi Farmasi
  - Instalasi rawat jalan dan IGD
  - Instalasi Gizi
4. Bagaimana cara pembuatan jadwal perawat saat ini, apakah secara terpusat atau dibagi setiap ward atau bangsal?

Answer: pembuatan Jadwal dilakukan secara per unit yang dilakukan oleh masing masing PJ unit. PJ unit mengumpulkan jadwal kepada bagian SDM RSIA maksimal tanggal 25 pada setiap bulan untuk bulan berikutnya untuk diinputkan ke dalam sistem mereka.

5. Bagaimanakah aturan umum mengenai penjadwalan perawat?

Answer:

- Staff yang memperoleh shift malam sebanyak maka harus memperoleh libur dihari berikutnya
- Setiap staff dalam satu bulan harus mendapatkan bagian shift malam
- Staff memiliki maksimal jam kerja 7 jam per hari dalam 6 hari perminggu
- jam lembur dihitung minimal 1 jam

6. Siapa yang melakukan penjadwalan perawat?

Answer : Penjadwalan dilakukan oleh setiap pj unit atau bangsal

7. Berapa kali penjadwalan perawat dilakukan? Apakah setiap minggu, atau bulan?

Answer: Penjadwalan perawat dilakukan sekali dalam sat bulan

Apakah ada rotasi perpindahan perawat yang dilakukan?

Answer: Ada, akan tetapi perpindahan jadwal perawat terseut biasanya tanpa sepengetahuan pihak SDM. Kebanyakan perawat mengganti shift mereka sendiri tanpa melakukan konfirmasi terlebih dahulu kepada pihak SDM.

8. Berapa shift yang dijalankan setiap harinya, dan berapa lama alokasi waktu setiap shift?

Answer: secara umum terdapat 3 shift dengan pembagian 7 jam yaitu

- Pagi: 07.00-14.00
- Sore : 14.00-21.00
- Malam: 21.00-07.00

untuk mengatasi jam kerja yang sibuk pada jam-jam tertentu RSIA menerapkan shift middle yang berlangsung dari jam 10.00-17.00 atau jam 12.00-19.00

9. Berapa maksimum jam kerja yang didapatkan masing-masing perawat dalam satu minggu?

Answer: Tidak ada maksimum jam kerja setiap minggunya. Yang ada hanyalah batas minimal libur untuk setiap bulannya. Setiap bulannya, jatah libur untuk setiap perawat diusahakan harus sama. Apabila tidak sama, maka perawat yang mendapat libur lebih sedikit akan diganti hari liburnya di bulan depan.

10. Bagaimana penjadwalan perawat yang sudah dilakukan saat ini? Apakah terdapat keluhan dari perawat mengenai penjadwalan yang sudah ada?

Answer: Penjadwalan perawat RSIA saat ini dilakukan secara manual oleh masing-masing PJ bangsal atau unit RS. Hal tersebut mengakibatkan memakan banyak waktu.

11. Apa saja kekurangan penjadwalan yang telah diterapkan saat ini? apa yang perlu dioptimalkan misal di tingkat manajemen atau di tingkat perawat?

Answer: Pihak SDM kesulitan untuk mengontrol jalannya penjadwalan dikarenakan banyak dari perawat yang berganti shift kerja tanpa ada izin terlebih dahulu. Hal yang ingin dioptimalkan oleh RS adalah bagaimana cara mengontrol jalannya penjadwalan dengan menggunakan sistem yang langsung terintegrasi langsung dengan fingerprint yang ada sehingga kecurangan atau pelanggaran Perawat akan saling tukar menukar jam dapat diminimalisasi.

Masalah lain juga adanya jam-jam dimana rumah sakit sepi atau sangat ramai. Ketika rumah sakit keadaannya sangat ramai, masalah tersebut biasanya diselesaikan dengan cara mengoper perawat yang bekerja pada shift tersebut pada unit yang renggang ke unit yang padat pasien.

Namun apabila rumah sakit sepi, maka bagian SDM akan meliburkan perawatnya. Misal dokter yang bertugas di RS tersebut harus keluar negeri semuanya untuk mengikuti pekan ilmiah. Melihat hal ini maka SDM berinisiatif untuk meliburkan perawat yang berada pada shift dan unit tersebut.

12. Bagaimana skill yang dimiliki perawat apakah ada perawat khusus yang menangani kasus tertentu (Type skill masing-masing perawat)

Answer: bagian unit yang harus memiliki perawat skill khusus adalah:

- Ruang Operasi
- Rekam Medis

13. Bagaimana jadwal yang sudah diatur, tiba-tiba mengalami perubahan karena salah perawat meminta libur?

Answer: Jatah ambil cuti harus ditentukan masing masing perawat pada saat PJ unit melakukan penjadwalan. Dalam satu bulan perawat diberikan jatah cuti satu hari. Pengajuan cuti dilakukan perawat maksimal tanggal 25 untuk setiap bulannya. Untuk kasus berbeda, misalkan terdapat keluarga meninggal atau terkena musibah yang lain berarti dapat dikurangi dengan jatah cuti yang sudah dia tentukan pada awal penjadwalan.

14. Apakah ada perawat yang memiliki pengurangan jam kerja? (ex. perawat yang sedang hamil, perawat memiliki keterbatasan)

Answer: Tidak ada pengurangan jam kerja setiap shiftnya untuk perawat yang hamil. Namun terdapat pemberian cuti selama 3 bulan yaitu 1.5 bulan sebelum melahirkan dan 1,5 bulan sesudah melahirkan. Pengambilan cuti ini bersifat fleksibel.

## Lampiran B: Hasil Pengujian Algoritma

Unit Farmasi menggunakan reinforcement learning			
Iterasi	Average	Min	Max
1	0,946406	0,665158	0,963721
2	0,904825	0,665158	0,913591
3	0,911847	0,665158	0,928714
4	0,950683	0,665158	0,969677
5	0,881638	0,665158	0,890646
6	0,849061	0,665158	0,855215
7	0,856478	0,665158	0,868047
8	0,84092	0,665158	0,844233
9	0,942715	0,665158	0,948262
10	0,893651	0,665158	0,906166
11	0,859949	0,665158	0,863568
12	0,905482	0,665158	0,912062
13	0,8328	0,665158	0,834723
14	0,867569	0,665158	0,871649
15	0,878141	0,665158	0,885561
16	0,962759	0,665158	0,974642
17	0,952066	0,665158	0,97155
18	0,916475	0,665158	0,922424
19	0,913804	0,665158	0,918478
20	0,890124	0,665158	0,897235
21	0,916888	0,665158	0,931922

Unit Farmasi menggunakan hill climbing			
Iterasi	Average	Min	Max
1	0,880232	0,665158	0,884673
2	0,824101	0,665158	0,82672
3	0,837469	0,665158	0,841984
4	0,96981	0,665158	0,977606
5	0,868428	0,665158	0,873144
6	0,889656	0,665158	0,925643
7	0,917184	0,665158	0,933234
8	0,845912	0,665158	0,849246
9	0,941252	0,665158	0,954891
10	0,961336	0,665158	0,971278
11	0,89376	0,665158	0,902819
12	0,92029	0,665158	0,952402
13	0,8368	0,665158	0,842026
14	0,888294	0,665158	0,895661
15	0,922031	0,665158	0,92918
16	0,874859	0,665158	0,88366
17	0,931168	0,665158	0,953361
18	0,879877	0,665158	0,898384
19	0,91924	0,665158	0,926282
20	0,837126	0,665158	0,849719
21	0,880105	0,665158	0,897343

Unit bayi/NICU menggunakan reinforcement learning			
Iterasi	Average	Min	Max
1	0,943674	0,664773	0,948114
2	0,908283	0,664773	0,918884
3	0,912488	0,664773	0,918485
4	0,939023	0,664773	0,946691
5	0,904334	0,664773	0,925427
6	0,957064	0,664773	0,967026
7	0,912123	0,664773	0,914003
8	0,939833	0,664773	0,949226
9	0,875568	0,664773	0,884205
10	0,918759	0,664773	0,923822
11	0,878172	0,664773	0,879181
12	0,890037	0,664773	0,896526
13	0,966572	0,664773	0,969958
14	0,956229	0,664773	0,960272
15	0,925476	0,664773	0,943436
16	0,9332	0,664773	0,942323
17	0,91597	0,664773	0,939199
18	0,865787	0,664773	0,866739
19	0,96127	0,664773	0,963414
20	0,867575	0,664773	0,868353
21	0,96107	0,664773	0,963852

Unit bayi/NICU menggunakan hill climbing			
Iterasi	Average	Min	Max
1	0,904975	0,664773	0,909189
2	0,877582	0,664773	0,892869
3	0,904468	0,664773	0,919729
4	0,893317	0,664773	0,899239
5	0,927567	0,664773	0,930803
6	0,904254	0,664773	0,921219
7	0,910267	0,664773	0,921624
8	0,893163	0,664773	0,895304
9	0,925647	0,664773	0,935349
10	0,935854	0,664773	0,943477
11	0,934535	0,664773	0,94599
12	0,946188	0,664773	0,952202
13	0,949565	0,664773	0,955113
14	0,944605	0,664773	0,95231
15	0,898021	0,664773	0,899863
16	0,895639	0,664773	0,896955
17	0,892818	0,664773	0,896906
18	0,911252	0,664773	0,914814
19	0,952449	0,664773	0,956274
20	0,864841	0,664773	0,866739
21	0,941575	0,664773	0,953041

Unit gizi menggunakan reinforcement learning			
Iterasi	Average	Min	Max
1	0,828988	0,70004	0,834421
2	0,844087	0,70004	0,846785
3	0,76752	0,70004	0,767918
4	0,783192	0,70004	0,785365
5	0,762919	0,70004	0,767116
6	0,752604	0,70004	0,753158
7	0,756507	0,70004	0,756844
8	0,771948	0,70004	0,776848
9	0,771207	0,70004	0,772228
10	0,786233	0,70004	0,789702
11	0,791218	0,70004	0,793247
12	0,821075	0,70004	0,822503
13	0,806669	0,70004	0,810719
14	0,770967	0,70004	0,77512
15	0,743643	0,70004	0,743789
16	0,751724	0,70004	0,75194
17	0,772289	0,70004	0,773721
18	0,79	0,70004	0,799777
19	0,838323	0,70004	0,848634
20	0,840588	0,70004	0,846433
21	0,791967	0,70004	0,793247

Unit gizi menggunakan hill climbing			
Iterasi	Average	Min	Max
1	0,746563	0,70004	0,747167
2	0,819275	0,70004	0,822735
3	0,797119	0,70004	0,797942
4	0,750267	0,70004	0,750597
5	0,798784	0,70004	0,803757
6	0,750369	0,70004	0,751696
7	0,786051	0,70004	0,791113
8	0,820257	0,70004	0,823545
9	0,830983	0,70004	0,833895
10	0,782309	0,70004	0,790351
11	0,795258	0,70004	0,796929
12	0,8244	0,70004	0,829804
13	0,72959	0,70004	0,729892
14	0,758039	0,70004	0,75845
15	0,762504	0,70004	0,764484
16	0,771915	0,70004	0,773721
17	0,78032	0,70004	0,783181
18	0,807353	0,70004	0,808447
19	0,772855	0,70004	0,77402
20	0,749391	0,70004	0,749566
21	0,766296	0,70004	0,768569

Unit IGD menggunakan reinforcement learning			
Iterasi	Average	Min	Max
1	0,907059	0,813278	0,908805
2	0,91269	0,813278	0,914179
3	0,927827	0,813278	0,929707
4	0,923263	0,813278	0,927439
5	0,961215	0,813278	0,96305
6	0,92063	0,813278	0,921388
7	0,918839	0,813278	0,919822
8	0,894666	0,813278	0,895105
9	0,901736	0,813278	0,904053
10	0,923029	0,813278	0,923949
11	0,890068	0,813278	0,890599
12	0,938676	0,813278	0,942265
13	0,925642	0,813278	0,92803
14	0,905268	0,813278	0,905822
15	0,92911	0,813278	0,930761
16	0,901196	0,813278	0,90149
17	0,949554	0,813278	0,950658
18	0,925173	0,813278	0,926282
19	0,916373	0,813278	0,916988
20	0,926234	0,813278	0,927479
21	0,904812	0,813278	0,905822

Unit IGD menggunakan hill climbing			
Iterasi	Average	Min	Max
1	0,940376	0,813278	0,941399
2	0,912368	0,813278	0,914171
3	0,976609	0,813278	0,979977
4	0,932072	0,813278	0,933765
5	0,916803	0,813278	0,918478
6	0,898429	0,813278	0,9
7	0,964633	0,813278	0,967164
8	0,945486	0,813278	0,950415
9	0,883753	0,813278	0,884434
10	0,950694	0,813278	0,95756
11	0,896309	0,813278	0,896647
12	0,947469	0,813278	0,948887
13	0,901014	0,813278	0,90149
14	0,975923	0,813278	0,978435
15	0,913665	0,813278	0,914286
16	0,906213	0,813278	0,908765
17	0,948806	0,813278	0,955257
18	0,95736	0,813278	0,961424
19	0,926214	0,813278	0,928019
20	0,942851	0,813278	0,948121
21	0,927535	0,813278	0,928019

Unit Ikamar Operasi menggunakan reinforcement learning			
Iterasi	Average	Min	Max
1	0,971781	0,968559	0,971803
2	0,968559	0,968559	0,968559
3	0,968559	0,968559	0,968559
4	0,968559	0,968559	0,968559
5	0,968559	0,968559	0,968559
6	0,968559	0,968559	0,968559
7	0,968559	0,968559	0,968559
8	0,981101	0,968559	0,981669
9	0,968559	0,968559	0,968559
10	0,97176	0,968559	0,971803
11	0,968559	0,968559	0,968559
12	0,968559	0,968559	0,968559
13	0,968559	0,968559	0,968559
14	0,968559	0,968559	0,968559
15	0,968559	0,968559	0,968559
16	0,974831	0,968559	0,97507
17	0,968559	0,968559	0,968559
18	0,968559	0,968559	0,968559
19	0,977967	0,968559	0,978359
20	0,926234	0,813278	0,927479
21	0,968559	0,968559	0,968559

Unit Ikamar Operasi menggunakan hill climbing			
Iterasi	Average	Min	Max
1	0,968559	0,968559	0,968559
2	0,968559	0,968559	0,968559
3	0,968559	0,968559	0,968559
4	0,968559	0,968559	0,968559
5	0,968559	0,968559	0,968559
6	0,968559	0,968559	0,968559
7	0,968559	0,968559	0,968559
8	0,981582	0,968559	0,981669
9	0,971781	0,968559	0,971803
10	0,968559	0,968559	0,968559
11	0,968559	0,968559	0,968559
12	0,968559	0,968559	0,968559
13	0,981319	0,968559	0,981669
14	0,968559	0,968559	0,968559
15	0,968559	0,968559	0,968559
16	0,98794	0,968559	0,988359
17	0,968559	0,968559	0,968559
18	0,968559	0,968559	0,968559
19	0,971792	0,968559	0,971803
20	0,980718	0,968559	0,981669
21	0,968559	0,968559	0,968559

Unit SIM & RM menggunakan reinforcement learning			
Iterasi	Average	Min	Max
1	0,983198	0,979592	0,98321
2	0,984224	0,979592	0,984848
3	0,984278	0,979592	0,985507
4	0,993649	0,979592	0,993984
5	0,991567	0,979592	0,992063
6	0,9835	0,979592	0,983626
7	0,983198	0,979592	0,98321
8	0,983186	0,979592	0,98321
9	0,983198	0,979592	0,98321
10	0,982683	0,979592	0,982704
11	0,996055	0,979592	0,99705
12	0,991815	0,979592	0,992063
13	0,998099	0,979592	0,998223
14	0,983198	0,979592	0,98321
15	0,99118	0,979592	0,991837
16	0,999644	0,979592	1
17	0,983198	0,979592	0,98321
18	0,999701	0,979592	1
19	0,98633	0,979592	0,986628
20	0,996522	0,979592	0,99705
21	0,989709	0,979592	0,989899

Unit SIM & RM menggunakan hill climbing			
Iterasi	Average	Min	Max
1	0,989664	0,979592	0,989945
2	0,979592	0,979592	0,979592
3	0,983198	0,979592	0,98321
4	0,982683	0,979592	0,982704
5	0,984598	0,979592	0,984615
6	0,987419	0,979592	0,987472
7	0,982694	0,979592	0,982704
8	0,989811	0,979592	0,990504
9	0,993601	0,979592	0,993984
10	0,994751	0,979592	0,994819
11	0,982694	0,979592	0,982704
12	0,984818	0,979592	0,985003
13	0,979592	0,979592	0,979592
14	0,98729	0,979592	0,987362
15	0,987261	0,979592	0,987472
16	0,983198	0,979592	0,98321
17	0,987574	0,979592	0,987805
18	0,992685	0,979592	0,992969
19	0,983198	0,979592	0,98321
20	0,98901	0,979592	0,989945
21	0,979592	0,979592	0,979592