

**OPTIMASI PENJADWALAN PERSONALIA RUMAH SAKIT BERBASIS
AGENT MENGGUNAKAN KOMBINASI ALGORITMA GENETIKA DAN
*TABU SEARCH***

SKRIPSI

Oleh:

**ERMA DIANANTA SARI
NIM. 08650060**



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI (UIN)
MAULANA MALIK IBRAHIM MALANG
2013**

**OPTIMASI PENJADWALAN PERSONALIA RUMAH SAKIT BERBASIS
AGENT MENGGUNAKAN KOMBINASI ALGORITMA GENETIKA DAN *TABU*
SEARCH**

SKRIPSI

**Diajukan Kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

Oleh:

**ERMA DIANANTA SARI
NIM. 08650060**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2013**

**OPTIMASI PENJADWALAN PERSONALIA RUMAH SAKIT BERBASIS
AGENT MENGGUNAKAN ALGORITMA GENETIKA**

SKRIPSI

**Diajukan Kepada:
Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim (UIN Maliki) Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:
KARTIKA MEKAR KUSUMANINGRUM
NIM. 08650062**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI (UIN)
MAULANA MALIK IBRAHIM MALANG
2012**

**OPTIMASI PENJADWALAN PERSONALIA RUMAH SAKIT BERBASIS
AGENT MENGGUNAKAN KOMBINASI ALGORITMA GENETIKA DAN *TABU*
*SEARCH***

SKRIPSI

Oleh:

**ERMA DIANANTA SARI
NIM. 08650060**

Telah disetujui oleh :

Dosen Pembimbing I

**Zainal Abidin, M.Kom
NIP. 197606132005011004**

Dosen Pembimbing II

**Dr.Ahmad Barizi, M.A
NIP. 197312121998031001**

Malang, Januari 2013

Mengetahui,

Ketua Jurusan Teknik Informatika

**Ririen Kusumawati, M.Kom
NIP. 197203092005012002**

LEMBAR PENGESAHAN

**OPTIMASI PENJADWALAN PERSONALIA RUMAH SAKIT BERBASIS
AGENT MENGGUNAKAN KOMBINASI ALGORITMA GENETIKA DAN *TABU*
*SEARCH***

SKRIPSI

Oleh

ERMA DIANANTA SARI
NIM. 08650060

Telah Dipertahankan di Depan Dewan Penguji Skripsi dan Dinyatakan Diterima
Sebagai Salah Satu Persyaratan Untuk
Memperoleh Gelar Sarjana Komputer (S.Kom)
Januari 2013

Susunan Dewan Penguji:	Tanda Tangan
1. Penguji Utama : Totok Chamidy, M.Kom NIP. 196912222006041001	()
2. Ketua Penguji : Suhartono, M.Kom NIP. 196805192003121001	()
3. Sekretaris : Zainal Abidin, M.Kom NIP. 197606132005011004	()
4. Anggota : Dr.Ahmad Barizi, M.A NIP. 197312121998031001	()

Mengetahui dan Mengesahkan,
Ketua Jurusan Teknik Informatika

Ririen Kusumawati, M.Kom
NIP. 197203092005012002

HALAMAN PERSEMBAHAN

Skripsi ini ku persembahkan untuk

Kedua orangtuaku, Bapak Kusniti dan Ibu Sumari yang tanpa henti mendo'akanku, menasehatiku dan menyiapkanku untuk menghadapi masa depan.....

Suamiku tercinta, Herinanda Fahmy Fahlevie yang dengan penuh kesabaran dan kasih sayangnya menuntunku untuk menggapai impian, yang selalu ada untukku dalam suka dan duka, yang memberiku inspirasi untuk menjadi lebih baik. Bertemu denganmu disini adalah anugerah terindah bagiku.....

Muhammad Surya Al-Muzakki, kehadiranmu telah melengkapi kebahagiaanku, memberikan semangat yang luar biasa untuk menggapai mimpi itu.....

Nur Ana, Kartika Mekar, Anisa Amalia, R.Sawitri, Evana, terimakasih atas motivasi, nasehat dan bantuan yang tiada ternilai harganya..., saat-saat yang telah kita lalui bersama di kampus ini tak akan pernah terlupakan..., kalian lebih dari sekedar sahabat....

Mbak Siti, terimakasih atas 3 tahun yang penuh makna...

MOTTO

Lebih cepat, Lebih baik.....



PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama : Erma Diananta Sari

NIM : 08650060

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Judul Penelitian : **OPTIMASI PENJADWALAN PERSONALIA
RUMAH SAKIT BERBASIS *AGENT*
MENGUNAKAN KOMBINASI ALGORITMA
GENETIKA DAN *TABU SEARCH***

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 23 Januari 2013

Yang membuat pernyataan,

Erma Diananta Sari

NIM. 08650060

KATA PENGANTAR

Segala puji bagi Allah SWT karena atas rahmat, taufiq dan hidayah-Nya, penulis dapat menyelesaikan penulisan tugas akhir sebagai salah satu syarat untuk memmperoleh gelar Sarjana Komputer (S.Kom). Penulis menyadari bahwa banyak pihak yang telah membantu dalam menyelesaikan penulisan tugas akhir ini.

Untuk itu, iringan doa dan ucapan terima kasih yang sebesar-besarnya penulis sampaikan kepada:

1. Zainal Abidin, M.Kom dan Dr.Ahmad Barizi, M.A selaku dosen pembimbing skripsi, karena atas bimbingan, pengarahan, dan kesabarannya penulisan tugas akhir ini dapat terselesaikan.
2. Ir. Mokhammad Amin Hariyadi, M.T, selaku dosen wali penulis selama studi.
3. Prof. Dr. H. Imam Suprayogo, Rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
4. Prof. Drs. Sutiman B. Sumitro, SU., DSc., Dekan Fakultas Sains dan Teknologi Maulana Malik Ibrahim Malang.
5. Ririen Kusumawati, M.Kom, Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.
6. Seluruh Dosen Teknik Informatika UIN Maulana Malik Ibrahim Malang, yang telah memberikan ilmu dan pengetahuan sebagai bekal dalam menyelesaikan tugas akhir.

Atas segala kekurangan dalam laporan ini, penulis memohon maaf dan mengharap kritik serta saran dari pembaca. Semoga tugas akhir ini dapat bermanfaat dan menambah khasanah ilmu pengetahuan.

Malang, Januari 2013

Erma Diananta Sari

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
HALAMAN PERSEMBAHAN	iv
HALAMAN MOTTO	v
PERNYATAAN KEASLIAN TULISAN	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x

DAFTAR TABEL	xi
ABSTRAK.....	xii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	5
1.3. Tujuan Penelitian	5
1.4. Manfaat Penelitian	5
1.5. Batasan Masalah	6
1.6. Metodologi	6
1.7. Sistematika Penyusunan.....	7
BAB II TINJAUAN PUSTAKA	9
2.1. Sistem Penjadwalan Rumah Sakit.....	9
2.2. Optimasi	13
2.3. <i>Tabu Search</i>	14
2.4. Algoritma Genetika.....	17
2.5. <i>Agent Cerdas</i>	21
BAB III METODE PENELITIAN	23
3.1. Tahap Penelitian.....	23
3.1.1 Alat dan Bahan yang Diperlukan	25
3.1.2 Observasi Aplikasi Saat Ini	25
3.1.3 Analisis Data	26
3.2. Tahapan Implementasi	31
3.3. Perancangan Perangkat Lunak	30
3.3.1 Desain <i>Input</i>	33
3.3.2 Desain <i>Output</i>	33
3.3.3 Desain Proses	34
3.3.4 Desain <i>Database</i>	38

3.3.5 Desain *Interface*41

BAB IV HASIL DAN PEMBAHASAN47

4.1. Sumber Data.....47

4.2. Implementasi Aplikasi Penjadwalan Personalia48

4.3. Penjadwalan Personalia dalam Pandangan Islam58

BAB V PENUTUP.....60

5.1 Kesimpulan60

5.2 Saran.....60

DAFTAR PUSTAKA

DAFTAR GAMBAR

Gambar 3.1 *Activity* diagram penjadwalan27

Gambar 3.2 Representasi obyek dalam biner.....28

Gambar 3.3 *Use Case* sistem penjadwalan personalia rumah sakit34

Gambar 3.4 *Activity diagram* manajemen data pegawai35

Gambar 3.5 *Activity diagram* pengajuan cuti.....36

Gambar 3.6 *Activity diagram* melihat jadwal37

Gambar 3.7 *Activity diagram* menggantikan tugas dinas37

Gambar 3.8 Tampilan awal aplikasi41

Gambar 3.9 Tampilan login admin42

Gambar 3.10 Tampilan halaman administrator	43
Gambar 3.11 Tampilan login kepala unit	43
Gambar 3.12 Halaman kepala unit	44
Gambar 3.13 Tampilan <i>request</i> cuti	45
Gambar 3.14 Tampilan hasil proses cuti	46
Gambar 4.1 Hasil perhitungan Algoritma Genetika dan <i>Tabu Search</i>	50
Gambar 4.2 Potongan <i>source code decoding</i>	51
Gambar 4.3 Potongan <i>source code</i> mengacak kromosom	51
Gambar 4.4 Potongan <i>source code Tabu List generasi pertama</i>	52
Gambar 4.5 Potongan <i>source code Tabu List generasi ke 2 dst</i>	52
Gambar 4.6 Potongan <i>source code crossover</i>	53
Gambar 4.7 Potongan <i>source code</i> mutasi	53
Gambar 4.8 Potongan <i>source code</i> fungsi <i>fitness</i>	53
Gambar 4.9 Potongan <i>source code</i> koneksi	53
Gambar 4.10 Potongan <i>source code</i> broker <i>agent</i>	54
Gambar 4.11 Potongan <i>source code request</i> cuti	55
Gambar 4.12 Jadwal Rumah Sakit Wawa Husada	57

DAFTAR TABEL

Tabel 3.1 Hasil random kromosom	31
Tabel 3.2 Hasil Perhitungan nilai <i>fitness</i>	33
Tabel 3.3 Tabel pegawai	38
Tabel 3.4 Tabel <i>shift</i>	39
Tabel 3.5 Tabel unit kerja	39
Tabel 3.6 Tabel tukar dinas	40
Tabel 3.7 Tabel jadwal	41
Tabel 4.1 Inisialisasi kromosom	49
Tabel 4.2 Tabel hasil uji coba	57

Tabel 4.3 Hasil generate jadwal dengan aplikasi.....58



ABSTRAK

Sari, Erma Diananta. 2013. 08650060. *Optimasi Penjadwalan Personalia Rumah Sakit Berbasis Agent Menggunakan Kombinasi Algoritma Genetika dan Tabu Search*. Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing (I) Zainal Abidin, M.Kom, (II) Dr.Ahmad Barizi, M.A

Kata kunci : penjadwalan personalia rumah sakit, Algoritma Genetika dan *Tabu Search*, *agent*

Penjadwalan personalia merupakan penjadwalan yang mengatur pembagian jam kerja di bagian personalia. Salah satunya adalah penjadwalan personalia di rumah sakit. Dalam penjadwalan personalia rumah sakit ini terdiri dari 3 *shift* utama, yaitu *shift* pagi, *shift* siang dan *shift* malam. Sistem penjadwalan yang baik sangat diperlukan, karena dengan penerapan suatu jadwal yang baik akan berdampak pada peningkatan kualitas pelayanan yang diberikan oleh pegawai.

Hasil uji coba sistem yang menerapkan kombinasi Algoritma Genetika dan *Tabu Search*, untuk menghasilkan 6 pola jadwal optimal membutuhkan jumlah iterasi 1396 dan waktu 35 menit 14 detik. Sedangkan sistem yang menggunakan Algoritma Genetika saja dengan jumlah iterasi 2674 memerlukan waktu selama 57 menit 24 detik untuk menghasilkan 6 pola jadwal optimal. Hal tersebut, menunjukkan bahwa sistem dengan kombinasi Algoritma Genetika dan *Tabu Search* jauh lebih efisien jika dibandingkan dengan sistem yang hanya menggunakan Algoritma Genetika.

ABSTRACT

Sari, Erma Diananta. 2013. 08650060. *Schedule Optimization of Hospital Personnel Agent Bases Using Combination Genetic Algorithn and Tabu Search*. The State of Islamic University Maulana Malik Ibrahim Malang. Promotor (I) Zainal Abidin, M.Kom, (II) Dr.Ahmad Barizi, M.A

Key words : scheduling of hospital personnel, *Genetic Algorithm, Tabu Search, agent*

Scheduling is a schedule personnel who manage the distribution of working hours in personnel. One is the scheduling of hospital personnel. In the scheduling of hospital personnel is comprised of 3 major shift, the morning shift, afternoon shift and night shift. Good scheduling system is needed, because due to the introduction of a good schedule will have an impact on improving the quality of service provided by employees.

The result od testing the system which implements a combination of Genetic Algorithm and Tabu Search, to generate optimal schedules 6 optimal schedule pattern with the number of iterations 1396 and 35 minutes and 14 seconds. While the system using only Genetic Algorithm the needed number of iterations 2674 and 35 minutes and 14 seconds to generate 6 schedule pattern. It demonstrates that the system with the combination of Genetic Algorithm and Tabu Search is more efficient tha a system that only uses Genetic Algorithm.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengertian jadwal adalah pembagian waktu berdasarkan rencana pengaturan urutan kerja, daftar atau tabel kegiatan dengan pembagian waktu pelaksanaan yang terperinci. Sedangkan pengertian penjadwalan adalah proses, cara, perbuatan menjadwalkan atau memasukkan ke dalam jadwal(KBBI:1989). Pada penerapannya, penyusunan jadwal memiliki aturan tertentu sesuai dengan kondisi obyeknya.

Penjadwalan kegiatan personalia merupakan penjadwalan yang mengatur pembagian jam kerja di bagian personalia. Salah satunya adalah penjadwalan personalia di rumah sakit. Yang termasuk dalam penjadwalan personalia rumah sakit adalah jadwal dokter, jadwal jaga perawat, dan karyawan. Namun dalam skripsi ini yang akan dijadwalkan hanya perawat saja, karena jadwal kerja karyawan dan dokter tidak mengalami perubahan. Penjadwalan kegiatan personalia di rumah sakit adalah suatu hal yang rumit dan sering kali mengalami kesulitan pada prosesnya. Hal ini di sebabkan oleh beberapa faktor yang saling berkaitan yang harus dipertimbangkan, antara lain adalah banyaknya jumlah pegawai yang ada dalam lingkup personalia suatu rumah sakit, dalam satu waktu harus diperhitungkan jumlah perbandingan antara pasien dengan jumlah perawat dan pegawai yang bertugas, pengaturan jadwal harus adil, misalnya pembagian jumlah jadwal *shift* pagi, siang dan *shift* malam, izin untuk cuti serta penyusunan jadwal yang harus mempertimbangkan kesehatan pegawai dalam urutan *shift* yang

diperoleh. Misalnya saja seorang perawat yang mendapatkan jadwal tiga hari *shift* malam, pada hari berikutnya akan mendapatkan libur dan apabila perawat tersebut mendapat tugas untuk menggantikan rekannya yang mengajukan cuti maka perawat itu tidak boleh mendapatkan jadwal pengganti pada *shift* pagi. Hal ini dimaksudkan agar perawat tersebut tidak merasa terlalu lelah dan tidak terjadi penurunan kualitas dalam bekerja.

Penyusunan jadwal personalia di rumah sakit hingga saat ini masih menggunakan metode manual dan hanya berlaku untuk satu bulan, biasanya penyusunan jadwal membutuhkan waktu selama dua hari atau bahkan lebih untuk mendapatkan susunan jadwal yang optimal. Hal ini dikarenakan penyusun (kepala bagian personalia) harus melakukan pengecekan terhadap data perawat yang telah mengajukan izin cuti atau perawat yang dikategorikan mungkin untuk menggantikan jadwal perawat yang akan cuti, menghitung keseimbangan pembagian *shift* kerja dan pembagian libur perawat yang semuanya masih menggunakan cara manual.

Didalam pandangan Islam, waktu adalah karunia Allah SWT yang tak ternilai harganya sehingga sebagai umat yang beriman wajib untuk memanfaatkan waktu sebaik-baiknya. Dalam Alqur'an surat Al-Luqman ayat 29 menjelaskan mengenai keteraturan datangnya siang dan malam yang sudah diatur oleh Allah SWT.



Kandungan dari ayat ini sesuai dengan tujuan penjadwalan, yaitu mengatur waktu sehingga waktu yang ada dapat dimaksimalkan keberadaannya.

Semua kegiatan di dalam kehidupan harus terdapat pengaturan waktu yang baik agar tidak terjadi bentrok antara kegiatan yang satu dengan yang lain. Pengaturan waktu tersebut sangat dibutuhkan agar setiap kegiatan berjalan dengan baik dan dapat mencapai tujuan yang diinginkan.

Kegiatan di rumah sakit sangat padat dan harus siap siaga selama 24 jam setiap harinya. Oleh karena itu perawat yang bekerja harus selalu siap berjaga pada *shift* yang telah ditentukan. Rumah sakit menggunakan sistem tiga *shift* kerja yaitu, *shift* pagi, *shift* sore dan *shift* malam. Untuk mengatur penjadwalan yang baik, diperlukan persiapan yang matang dalam pengaturan jadwal untuk perawat. Hal ini dimaksudkan agar terdapat keseimbangan waktu bekerja dengan porsi libur yang diperoleh oleh masing-masing perawat, dan tidak terjadi kelelahan secara fisik dan psikologis pada perawat. Karena dengan pengaturan jadwal yang dipraktikkan secara baik diharapkan akan berdampak pada peningkatan kualitas pelayanan yang diberikan oleh perawat kepada pasiennya. Hal ini senada dengan hasil penelitian yang pernah dilakukan oleh Anisa Ulya dari ITS, bahwa dalam menyusun jadwal pegawai yang ada di rumah sakit harus memperhitungkan jumlah libur dan pembagian *shift-shiftnya*(Anisa Ulya:2010).

Berdasarkan deskripsi latar belakang yang ada, penulis terinspirasi untuk membuat suatu aplikasi yang dapat memudahkan dan memberi hasil yang lebih optimal pada penyusunan jadwal personalia rumah sakit. Metode yang digunakan pada aplikasi ini adalah kombinasi algoritma Genetika dan *Tabu Search*.

Penggunaan metode ini didasarkan pada kesesuaian dengan permasalahan, yaitu kasus penjadwalan, selain itu mengacu pada penelitian sebelumnya yang dilakukan oleh Komang Setemen dan Mauridhi H.Purnomo dari ITS Surabaya yang berjudul Kombinasi Algoritma Genetika dan *Tabu Search* dalam Pembuatan Tabel Jadwal Mata Kuliah.

Peran dari *Intelligent Agent* disini adalah menjalankan algoritma Genetika dan *Tabu Search* agar berjalan secara otomatis dan memproses izin cuti pegawai. Dengan komposisi kombinasi Algoritma Genetika dan *Tabu Search* serta *Intelligent Agent* yang diterapkan pada aplikasi ini, diharapkan penyusunan jadwal personalia rumah sakit akan lebih optimal dalam output hasil maupun prosesnya.

1.2 Rumusan Masalah

Apakah penerapan kombinasi Algoritma Genetika dan *Tabu Search* yang berbasis *Agent* pada sistem penjadwalan personalia rumah sakit lebih optimal jika dibandingkan dengan sistem yang hanya menerapkan Algoritma Genetika?

1.3 Tujuan Penelitian

Membuktikan bahwa kombinasi algoritma Genetika dan *Tabu Search* berbasis *Agent* dapat diterapkan pada sistem penjadwalan personalia rumah sakit dan hasilnya dapat lebih baik jika dibandingkan dengan sistem yang hanya menerapkan Algoritma Genetika.

1.4 Manfaat Penelitian

- a. Membantu pihak rumah sakit dalam menyelesaikan penjadwalan personalia.
- b. Mempercepat proses penjadwalan personalia.

1.5 Batasan Masalah

- a. *Tabu Search* hanya digunakan untuk memfilter kromosom yang mengalami *crossover*.
- b. Sistem penjadwalan pada penelitian ini hanya menjadwalkan perawat saja.
- c. Pegawai tidak boleh mengambil cuti sekaligus selama 7 hari, kecuali jika ada masalah kesehatan dan cuti melahirkan.
- d. Masing-masing *shift* dijalankan sebanyak tiga kali.
- e. Setelah pegawai mendapatkan jadwal *shift* malam, pegawai diberikan libur selama dua hari.
- f. Setelah pegawai mendapatkan jadwal *shift* sore, maka pegawai diberikan libur satu hari.
- g. Penjadwalan hanya untuk 3 ruang rawat inap.
- h. Terpusat pada bagian personalia di ruang rawat inap rumah sakit.
- i. Tidak termasuk ruang poli, bagian keamanan, dan apotek.

1.6 Metodologi

- a. Menyimpan data pegawai, data *shift*, tanggal, data ruang beserta aturan pembagian izin dinas luar dan cuti ke dalam *database*.
- b. Melakukan pengkodean sesuai aturan algoritma Genetika pada data penjadwalan, dimana data-data tersebut nantinya menjadi kumpulan kromosom.

- c. Menghitung nilai *fitness* sebagai acuan evaluasi, semakin besar nilai *fitness* maka semakin baik kualitas dari jadwal tersebut. Nilai *fitness* maksimal adalah 9.
- d. Melakukan proses reproduksi, yaitu dengan cara *crossover* dan mutasi. Dalam hal ini yang mengalami *crossover* dan mutasi hanya kode *shift*.
- e. Memasukkan dan membandingkan kromosom yang dihasilkan Algoritma Genetika ke dalam *Tabu List*.
- f. Melakukan uji coba, dengan parameter metode yang digunakan lalu membandingkan hasilnya. Dalam skripsi ini yang akan dibandingkan adalah penjadwalan menggunakan Algoritma Genetika saja dan penjadwalan yang menggunakan Kombinasi Algoritma Genetika dan *Tabu Search*.
- g. Melakukan dokumentasi pada hasil uji coba.

1.7 Sistematika Penyusunan

BAB I PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, batasan masalah, manfaat penelitian, tujuan penelitian, metodologi, dan sistematika penyusunan laporan skripsi.

BAB II KAJIAN PUSTAKA

Bab ini berisi tentang materi yang mendukung optimasi penjadwalan personalia rumah sakit, aturan penjadwalan

dari RS.Wawa Husada, Algoritma Genetika, *Tabu Search* dan *Agent*.

BAB III PERANCANGAN SISTEM

Bab ini membahas tentang analisa dan perancangan sistem Penjadwalan Personalia Rumah Sakit Berbasis *Agent* Menggunakan Kombinasi Algoritma Genetika dan *Tabu Search*, antara lain tentang UML, desain *database*, desain *interface* dan penerapan kombinasi Algoritma Genetika dan *Tabu Search* pada kasus penjadwalan personalia.

BAB IV IMPLEMENTASI DAN UJI COBA

Bab ini membahas tentang implementasi dari kombinasi Algoritma Genetika dan *Tabu Search* pada penjadwalan personalia dan hasil uji coba dengan mencantumkan tabel hasil dari pengujian.

BAB V PENUTUP

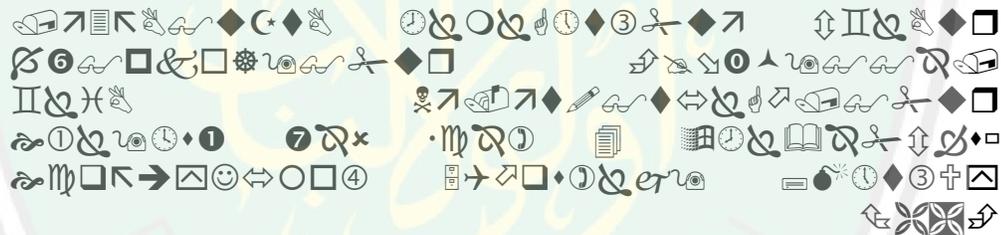
Bab ini berisi kesimpulan dari penelitian yang telah dilakukan dan saran yang diharapkan dapat membantu penelitian selanjutnya.

BAB II
TINJAUAN PUSTAKA

2.1 Sistem Penjadwalan Rumah Sakit

Penjadwalan merupakan pengaturan dalam rentang waktu tertentu untuk menyelesaikan sekumpulan tugas. Penjadwalan sangat penting untuk merancang dan memajemen suatu sistem yang terdiri dari sejumlah kegiatan.

Dalam pandangan Islam, waktu merupakan amal shaleh, amal shaleh disini diartikan menjadi tujuan utama dan bukan materi. Orang yang tidak memanfaatkan waktu dengan baik di dalam Al-Qur'an disebut sebagai orang yang merugi.



Artinya :

Dan di antara tanda-tanda kekuasaan-Nya ialah tidurnu di waktu malam dan siang hari dan usahamu mencari sebagian dari karuniaNya. Sesungguhnya pada yang demikian itu benar-benar terdapat tanda-tanda bagi kaum yang mendengarkan(QS.Ar-Ruum[30]:23).

Seperti yang tercantum dalam Al-Qur'an surat Ar-Ruum ayat 23 diatas menjelaskan bahwa Allah telah mengatur segala kejadian di bumi ini, seperti siang dan malam, musim dingin dan musim panas, terbit dan terbenamnya matahari. Dari ayat ini, sesungguhnya sudah dapat dilihat bahwa sistem penjadwalan sangat diperlukan untuk mengatur segala aktifitas yang ada di dunia. Begitu juga dengan sistem penjadwalan yang diterapkan di rumah sakit, hal ini

tentu telah mengamalkan apa yang ada dalam Al-Qur'an khususnya surat Ar-Ruum ayat 23. Sebagai contoh, perawat memiliki keteraturan antara hari libur yang didapatkannya dengan hari kerja, pengaturan *shift* yang memenuhi standart pola seperti setelah perawat mendapatkan *shift* malam maka pada hari berikutnya akan mendapat libur.

Penjadwalan sangat erat kaitannya dengan disiplin waktu, karena dengan sarana penjadwalan maka suatu disiplin waktu akan dapat terwujud. Misalnya saja disiplin waktu bekerja di rumah sakit, para karyawan yang telah dijadwalkan pada *shift* tertentu hendaknya mematuhi jadwal yang telah ditetapkan, baik disiplin masuk tepat pada jam yang ditentukan, hari yang telah ditentukan dan susunan kegiatan yang telah diprogramkan. Dalam Islam, disiplin waktu dibahas pada beberapa surat, yang pertama adalah surat An-Nisaa' ayat 103.



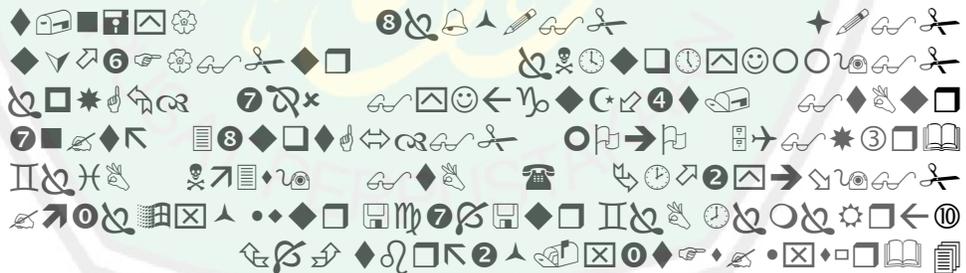
Artinya :

Maka apabila kamu telah menyelesaikan shalat(mu), ingatlah Allah di waktu berdiri, di waktu duduk dan di waktu berbaring. kemudian apabila kamu telah merasa aman, Maka dirikanlah shalat itu (sebagaimana biasa). Sesungguhnya shalat itu adalah fardhu yang ditentukan waktunya atas orang-orang yang beriman.(QS.An-Nisaa'[4]:103)

Pada surat An-Nisaa' ayat 103 diatas dijelaskan mengenai pentingnya disiplin waktu dalam Islam yaitu mengenai waktu sholat fardhu yang telah ditentukan waktunya. Dan waktu-waktu sholat yang telah ditentukan Allah SWT pasti

mempunyai arti dan telah disesuaikan dengan keadaan hambanya. Sebagai contoh, waktu sholat subuh yang jatuh pada saat fajar shiddiq hingga sebelum matahari terbit, hal ini dapat ditafsirkan bahwa umat Islam mengawali harinya disaat sebelum matahari terbit agar dapat memulai aktifitasnya lebih dini. Oleh karena itu, waktu sholat yang telah ditentukan wajib untuk dipatuhi, dan dari sinilah disiplin waktu dapat diawali. Sebagaimana diaturnya waktu sholat fardhu, waktu untuk bekerja hendaknya juga diatur untuk menciptakan disiplin waktu dan keteraturan dalam bekerja, sehingga hasil yang diperoleh akan maksimal. Jadwal yang sudah terbentuk dan ditetapkan hendaknya dipatuhi oleh seluruh pegawai, karena apabila jadwal tersebut dilanggar akan mengakibatkan ketidak teraturan dan tidak maksimalnya hasil yang diharapkan, oleh kerena itu disiplin waktu sangat penting untuk diterapkan.

Selain surat An-Nisaa' ayat 103, di dalam Al-Qur'an surat As-Sajdah ayat 4 juga menjelaskan mengenai keteraturan waktu.



Artinya :

Allah lah yang menciptakan langit dan bumi dan apa yang ada di antara keduanya dalam enam masa, kemudian Dia bersemayam di atas 'Arsy. tidak ada bagi kamu selain dari padanya seorang penolongpun dan tidak (pula) seorang pemberi syafa'at. Maka Apakah kamu tidak memperhatikan?(QS. As-Sajdah[32]: 4)

Pada ayat di atas dijelaskan bahwa Allah SWT menciptakan langit dan bumi dalam enam masa. Dalam enam masa itu bertahap alam dan seisinya terbentuk

sempurna. Keteraturan waktu pada penciptaan alam dan seisinya dalam waktu 6 masa menggambarkan bahwa Allah telah mengatur tahap demi tahap dari penciptaan itu sehingga terbentuklah alam dan isinya yang maha sempurna serta menempatkan tahap penciptaan sesuai dengan waktu yang tepat. Dari penjelasan ayat diatas, keteraturan juga dapat dipraktikkan dalam manajemen waktu bekerja. Misalnya menyesuaikan jadwal pegawai dengan porsi yang diperlukan oleh rumah sakit, sebagai contoh apabila jumlah pasien lebih banyak, seharusnya diimbangi juga dengan jumlah pegawai yang bekerja.

Peran *agent* dalam sistem penjadwalan sangat diperlukan, karena dengan adanya *agent*, proses perhitungan dengan algoritma tidak perlu dijalankan kembali pada saat ada perawat yang mengajukan cuti, dan akhirnya dapat menghemat waktu dari suatu pekerjaan. Hal ini tentunya sangat sesuai dan sangat dianjurkan dalam Islam, bahwa waktu yang telah diberikan oleh Allah SWT kepada hambanya harus dimanfaatkan sebaik mungkin. Pengoptimalan *agent* yang diterapkan pada sistem penjadwalan diharapkan bisa menjadi titik awal untuk menciptakan suatu inovasi teknologi yang handal di masa mendatang agar Islam dapat bersaing dalam kemajuan teknologi.

Sistem penjadwalan di rumah sakit memiliki beberapa aturan yang dipakai sebagai acuan penyusunan jadwal. Dalam sehari terdapat tiga *shift* jaga, yaitu *shift* pagi, *shift* siang, dan *shift* malam. *Shift* pagi dimulai dari pukul 08.00 pagi hingga pukul 15.00 sore, *shift* siang dimulai pada pukul 15.00 hingga pukul 21.00 malam, kemudian untuk *shift* malam dimulai dari pukul 21.00 hingga pukul 08.00 pagi.

Selain itu, terdapat beberapa aturan baku dari pihak rumah sakit yang harus dipenuhi, diantaranya adalah :

- a. Pegawai yang mendapat jadwal tiga hari *shift* malam, pada hari berikutnya tidak boleh berada pada *shift* pagi.
- b. Pegawai yang mendapat jadwal tiga hari *shift* malam, mendapat jatah libur satu hari dan satu hari ekstra.
- c. Pegawai yang mendapat jadwal tiga hari *shift* sore, mendapat jatah libur satu hari.
- d. Setiap pegawai mendapat hak cuti selama 12 hari dalam satu tahun, namun tidak boleh di ambil sekaligus dalam satu bulan.

2.2 Optimasi

Optimasi berarti menjadikan paling baik, menjadikan paling tinggi. Secara umum, pengertian optimasi adalah pencarian nilai “terbaik dari yang tersedia” dari beberapa fungsi yang diberikan pada suatu konteks. (KBBI:1991,626) Oleh karena itu, optimasi dapat diartikan sebagai upaya untuk meningkatkan kinerja sehingga mempunyai kualitas yang baik dan hasil kerja yang tinggi.

Teknik optimasi secara umum dapat dibagi menjadi dua bagian, yang pertama adalah *Mathematical Programming*, yang kedua adalah *Combinatorial Optimatimization*. Teknik optimasi yang digunakan pada penelitian ini adalah *Combinatorial Optimatimization*, yaitu penyelesaian menggunakan algoritma optimasi.

Algoritma optimasi dapat didefinisikan sebagai algoritma atau metode numerik untuk menemukan nilai x sedemikian hingga menghasilkan $f(x)$ yang

bernilai sekecil (atau sebesar) mungkin untuk suatu fungsi f yang diberikan, yang mungkin disertai dengan beberapa batasan pada x . Disini x berupa skalar atau vektor dari nilai-nilai kontinu atau diskrit (Suyanto,2010).

Berdasarkan metode operasinya, algoritma optimasi dibagi menjadi dua, yaitu algoritma deterministik dan probabilistik. Pada algoritma deterministik, pada setiap langkah eksekusi terdapat satu jalan untuk diproses. Jika tidak ada jalan, berarti algoritma sudah selesai. Algoritma deterministik sering digunakan untuk masalah yang memiliki relasi yang jelas antara karakteristik calon solusi dengan utilitasnya. Untuk permasalahan dengan ruang pencarian yang sangat besar, biasanya para praktisi lebih sering menggunakan algoritma probabilistik. Hampir semua algoritma probabilistik menggunakan konsep dasar dari metode *Monte Carlo*. Metode *Monte Carlo* bersandar pada proses pengambilan sampel secara acak yang berulang-ulang untuk menghasilkan solusi. Algoritma-algoritma probabilistik berusaha menemukan solusi yang “bagus” tanpa melebihi batasan waktu yang disediakan. Solusi yang “bagus” belum tentu yang paling optimum tetapi sudah dapat diterima oleh user(Suyanto,2010).

2.3 *Tabu Search*

Tabu Search (TS) adalah suatu metode optimasi yang menerapkan sistem pemanfaatan memory. Pada metode *Tabu Search*, solusi yang sudah pernah ditemui akan dilarang untuk di eksekusi kembali pada iterasi berikutnya.

Tabu Search diperkenalkan pertama kali oleh Glover pada tahun 1970-an. Banyak eksperimen yang menunjukkan bahwa *Tabu Search* saat ini telah menjadi suatu teknik optimasi yang dapat dikombinasikan dengan teknik optimasi lainnya.

Diantaranya adalah penelitian yang berjudul Kombinasi Algoritma Genetika dan *Tabu Search* dalam Pembuatan Tabel Jadwal Mata Kuliah yang dilakukan oleh Komang Setemen dan Mauridhi Herry Purnomo dari ITS Surabaya. Hasil dari penelitian yang dilakukan dengan menggunakan data mata kuliah dari Fakultas Teknik dan Kejuruan di Universitas Pendidikan Ganesha Singaraja menyatakan bahwa kombinasi Algoritma Genetika dan *Tabu Search* lebih efisien dalam iterasi dan waktu dengan data uji coba yang sama jika dibandingkan dengan sistem penjadwalan yang hanya menggunakan Algoritma Genetika saja.

Algoritma *Tabu Search* bisa menerima solusi yang lebih buruk dari pada solusi saat ini. Untuk menjaga agar solusi yang terbaik tidak hilang, *Tabu Search* menyimpan solusi terbaik tersebut dan terus mencari berdasarkan solusi terakhir yang di dapat pada saat iterasi. Selain itu, *Tabu Search* menyimpan solusi yang pernah ditemui dan melarang solusi tersebut digunakan kembali, hal ini bertujuan agar tidak terjadi perulangan berkali-kali yang sia-sia dan memakan waktu yang cukup lama.

Tabu Search menggunakan struktur *memory* yang disebut *Tabu List*. *Tabu List* digunakan untuk menyimpan solusi yang pernah ditemui selama iterasi berjalan agar proses pencarian tidak berulang-ulang pada daerah solusi yang sama, selain itu untuk menuntun proses pencarian menelusuri solusi-solusi yang belum pernah dikunjungi sebelumnya.

Tabu Search bekerja secara iteratif menggunakan algoritma *Local search* pada setiap iterasi untuk mencari solusi terbaik di antara sebagian tetangga dari solusi terbaik saat ini. Pada setiap iterasi, algoritma *Local search* memilih solusi

tetangga yang memberikan peningkatan kualitas tertinggi. Tetapi, jika semua solusi tetangga tidak memberikan peningkatan kualitas, maka *Local search* akan memilih solusi yang penurunan kualitasnya paling rendah. Kualitas di sini bergantung pada masalah yang dihadapi. Terdapat tiga strategi utama yang digunakan dalam *Tabu Search*, yaitu :

- a. Strategi pelarangan (*the firbidding strategy*) untuk mengontrol apa saja yang boleh masuk ke *Tabu List*
- b. Strategi pembebasan (*the freeing strategy*) untuk memutuskan apa saja yang boleh dikeluarkan dari *Tabu List* dan kapan pengeluaran dilakukan
- c. Strategi jangka pendek (*the short-term strategy*) yang mengatur interaksi antara strategi pelarangan dan strategi pembebasan untuk membangkitkan dan menyeleksi solusi-solusi percobaan(Suyanto,2010).

Tabu Search memiliki lima parameter utama yang harus ditentukan secara hati-hati, yaitu : prosedur *local search*, struktur ketetanggaan, kondisi tabu, kondisi aspirasi, dan kriteria penghentian. Algoritma *Tabu Search* bisa dihentikan berdasarkan kriteria tertentu, misalnya sejumlah iterasi yang ditentukan *user*, sejumlah waktu CPU tertentu, atau sejumlah iterasi berurutan tanpa peningkatan nilai fungsi objektif terbaik, dan sebagainya. *Tabu Search* juga memiliki lima unsur dasar, yaitu :

- a. Langkah utama untuk memanfaatkan *memory* di dalam *Tabu Search* adalah mengklasifikasi suatu sub himpunan langkah di dalam suatu ketetanggaan sebagai larangan atau *tabu*.

- b. Suatu ketetangaan dibangun untuk mengidentifikasi solusi-solusi tetangga yang dapat dicapai dari solusi saat ini.
- c. Klasifikasi bergantung pada sejarah pencarian, dan khususnya pada kebaruan atau *frekuensi* bahwa langkah atau komponen solusi tertentu, yang disebut atribut, telah berpartisipasi pada pembangkitan solusi-solusi sebelumnya.
- d. Suatu *Tabu List* mencatat langkah-langkah terlarang atau *tabu moves*.
- e. Batasan-batasan tabu bisa diberikan pengecualian. Ketika suatu langkah *tabu* memberikan suatu solusi yang lebih baik dibandingkan semua langkah terbaik sebelumnya, maka status *tabu* dari langkah tersebut bisa diabaikan (artinya: statusnya diubah dari *tabu* menjadi tidak *tabu*). Kondisi atau kriteria pengabaian status *tabu* ini disebut kondisi aspirasi.

Tabu Search dapat diatur dengan baik selama proses pencarian ketetangaan menggunakan *intensification* dan *diversification*. Untuk mengimplementasikan keduanya, kita membutuhkan suatu tambahan pada fungsi objektif. *Intensification* bisa dilakukan dengan memberikan penalti (denda) terhadap solusi yang jauh lebih buruk dari solusi saat ini. Sedangkan *diversification* dilakukan dengan memberikan penalti pada solusi yang kualitasnya mendekati solusi saat ini. Tujuan dari *intensification* adalah agar proses pencarian menjadi lebih sistematis dan fokus pada tetangga-tetangga yang lebih menjajnikan peningkatan kualitas. *Diversification* membuat proses pencarian menjadi lebih meluas pada ruang solusi yang ada (Suyanto,2010).

2.4 Algoritma Genetika

Algoritma genetika adalah algoritma pencarian heuristik yang didasarkan atas mekanisme evolusi biologis.

Pada dasarnya ada 4 kondisi yang sangat memengaruhi proses evaluasi, yakni sebagai berikut :

- a. Kemampuan organisme untuk melakukan reproduksi.
- b. Keberadaan populasi organisme yang bisa melakukan reproduksi.
- c. Keberagaman organisme dalam suatu populasi.
- d. Perbedaan kemampuan untuk *survive*.

Individu yang lebih kuat (*fit*) akan memiliki tingkat *survival* dan tingkat reproduksi yang lebih tinggi jika dibandingkan dengan individu yang kurang *fit*. Pada kurun waktu tertentu (sering dikenal dengan istilah generasi), populasi secara keseluruhan akan lebih banyak memuat organisme yang *fit* (Sri Kusumadewi dan Hari Purnomo,2005).

John Holland dari Universitas Michigan adalah orang yang pertama kali mengembangkan konsep Algoritma Genetika pada awal tahun 1970-an. John Holland mengatakan bahwa setiap masalah yang berbentuk adaptasi dapat diformulasikan dalam terminologi genetika. Algoritma genetika adalah algoritma pencarian yang berdasarkan evolusi biologis, dimana terdapat individu atau kandidat yang kuat maka individu itulah yang akan bertahan melalui proses seleksi yang terjadi. Individu yang terdapat dalam satu populasi disebut dengan istilah kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol, karena dalam Algoritma Genetika suatu permasalahan yang akan diselesaikan harus terlebih dulu dilakukan pengkodean dengan mengubah obyek

ke dalam bentuk biner. Populasi awal dibangun secara acak, sedangkan populasi berikutnya merupakan hasil dari evolusi kromosom-kromosom yang diperoleh melalui proses iterasi mulai dari seleksi, penghitungan nilai *fitness* untuk mengevaluasi kualitas kromosom hingga proses rekombinasi yang disebut dengan istilah generasi. Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut dengan fungsi *fitness*. Nilai *fitness* dari suatu kromosom akan menunjukkan kualitas dari setiap kromosom atau solusi yang muncul. Generasi berikutnya dikenal dengan istilah anak (*off spring*) terbentuk dari gabungan 2 kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyilangan (*crossover*). Selain operator penyilangan, suatu kromosom dapat juga dimodifikasi dengan menggunakan operator mutasi. Setelah melalui beberapa proses generasi, maka algoritma Genetika akan berangsur-angsur menemukan solusi yang terbaik dari permasalahan yang diselesaikan.

2.4.1 Komponen utama Algoritma Genetika

a. Teknik Penyandian

Teknik penyandian disini meliputi penyandian gen dan kromosom. Gen merupakan bagian dari kromosom. Satu gen biasanya akan mewakili satu variabel. Gen dapat direpresentasikan dalam bentuk string bit, pohon, *array* bilangan real, daftar aturan, elemen permutasi, elemen program, atau representasi lainnya yang dapat diimplementasikan untuk operator genetika.

b. Prosedur Inisialisasi

Ukuran populasi tergantung pada masalah yang akan dipecahkan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, kemudian harus dilakukan inisialisasi terhadap kromosom yang terdapat pada populasi tersebut. Inisialisasi kromosom dilakukan secara acak, namun demikian harus tetap memperhatikan domain solusi dan kendala permasalahan yang ada.

c. Fungsi evaluasi

Fungsi evaluasi sangat besar peranannya dalam Algoritma Genetika, karena dengan adanya fungsi evaluasi inilah suatu solusi yang muncul dapat diketahui kualitasnya. Dalam mengevaluasi kromosom atau solusi menggunakan konversi fungsi objektif ke dalam fungsi *fitness*. Untuk menghasilkan fungsi *fitness* yang baik, perlu dipahami mengenai permasalahan yang akan diselesaikan. Hal ini bertujuan agar suatu fungsi evaluasi dapat benar-benar menangkap kualitas dari suatu kromosom.

d. Seleksi

Seleksi bertujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling *fit*. *Fit* yang dimaksud adalah kualitas kromosom, apabila suatu kromosom memiliki nilai *fitness* yang termasuk dalam kategori baik jika dibandingkan dengan kromosom yang lainnya, maka kromosom tersebutlah yang dikatakan lebih *fit* dan lebih memiliki kesempatan untuk berkembangbiak lagi.

e. Rekombinasi atau *Crossover*

Crossover berarti memilih posisi acak pada rangkaian (misalnya setelah dua digit) dan menukar segmennya ke kanan atau ke kiri dari titik ini dengan bagian rangkaian lain yang serupa untuk menghasilkan dua anak baru. Jadi, fungsi utama *crossover* adalah menghasilkan *string* baru yang serupa dengan menggabungkan bagian dari kedua string ciri yang utama. Lebih lanjut, *crossover* merupakan perkawinan dua orang tua utama dan disebut pula proses rekombinasi.

f. Mutasi

Mutasi adalah perubahan situasi yang terus-menerus. Kadang-kadang mutasi digunakan untuk mencegah algoritma berhenti. Prosedur ini mengubah satu menjadi nol atau mengubah nol menjadi satu, dan bukan menduplikasinya. Perubahan ini muncul dalam probabilitas yang sangat kecil (Ririen Kusumawati, 2007).

2.5 Agen cerdas

Agen cerdas adalah suatu program yang membantu user dalam melakukan aktifitasnya. Agen cerdas dibekali dengan pengetahuan dan aturan untuk menjalankan tugasnya.

Dewasa ini, dengan berkembangnya teknologi jaringan komputer, termasuk internet di dalamnya, kebutuhan paradigma *software* dan program yang bisa menjalankan tugas yang didelegasikan kepadanya secara mandiri, memiliki intelegensi, dan kemampuan bergerak dalam lingkungan jaringan komputer, sudah sangat mendesak. Disinilah peran

agen cerdas menjadi sesuatu yang harus ada untuk mengatasi beberapa masalah-masalah yang timbul seperti disebut diatas.

Istilah agen diturunkan dari konsep agensi (*agency*), maksudnya memekerjakan seseorang untuk bertindak atas kepentingan anda. Suatu agen manusia melambangkan seseorang dan interaksi dengan lainnya untuk menyelesaikan tugas yang telah ditentukan sebelumnya (Ririen Kusumawati,2007).

Menurut Wooldridge (2002) dalam Turban (2005:916) menyatakan bahwa kecerdasan adalah fitur utama yang berhubungan dengan pendefinisian agen cerdas. Wooldridge menyatakan bahwa kecerdasan dalam pengertian ini memiliki fitur berikut :

- a. Reaktivitas. Agen cerdas mampu merasakan lingkungannya dan merespons perubahan yang muncul di dalamnya secara tepat waktu untuk memenuhi tujuan perancangannya.
- b. Proaktivitas. Agen cerdas mampu menunjukkan perilaku yang mengarah ke tujuan dengan mengambil inisiatif untuk memenuhi tujuan perancangannya.
- c. Kemampuan sosial. Agen cerdas mampu berinteraksi dengan agen lain (dan mungkin dengan manusia) untuk memenuhi tujuan perancangannya.

BAB III

METODE PENELITIAN

3.1 Tahap Penelitian

Penelitian ini dilakukan untuk mempermudah proses penyusunan jadwal personalia di rumah sakit. Karena sistem penyusunan jadwal yang berjalan pada saat ini masih dilakukan secara manual. Obyek yang menjadi bahan penelitian adalah jadwal yang diperoleh dari RS.Wawa Husada, Kepanjen, Kab.Malang. Data hasil observasi dari rumah sakit berupa jadwal jaga perawat, pedoman ketenaga kerjaan rumah sakit, dan pedoman penjadwalan. Untuk membuat suatu jadwal diperlukan beberapa pertimbangan, diantaranya adalah keseimbangan jumlah libur, jumlah dan jenis *shift* yang di dapat serta pola jadwal yang harus sesuai dengan aturan yang telah ditetapkan. Suatu jadwal dikatakan sebagai pola jadwal yang benar apabila telah memenuhi aturan penjadwalan yang telah ditetapkan, dimana setelah perawat mendapatkan jadwal *shift* malam pada hari berikutnya akan mendapatkan libur dua hari.

Berdasarkan data hasil observasi yang diperoleh, dengan memperhatikan output yang diharapkan, bentuk dari jadwal, maka Algoritma Genetika adalah

algoritma yang sesuai untuk diterapkan pada masalah optimasi penjadwalan ini. Pernyataan ini didasarkan pada pengertian Algoritma Genetika yang merupakan suatu algoritma pencarian berdasarkan evolusi biologis dan masalah yang akan diselesaikan harus direpresentasikan dalam bentuk biner. Yang akan direpresentasikan dalam bentuk biner disini adalah pola jadwal. Semua jenis *shift* akan diubah ke dalam bentuk biner untuk selanjutnya diproses dengan tahapan yang ada pada Algoritma Genetika. *Shift-shift* yang telah diubah ke dalam biner berperan sebagai gen, sedangkan *shift* yang telah terkumpul menjadi satu pola jadwal akan dikatakan sebagai kromosom.

Dalam proses penjadwalan seorang penyusun jadwal akan mencari suatu susunan jadwal yang dianggap paling optimal. Hal ini dilakukan dengan cara menghitung, memperhatikan aturan penjadwalan dan membandingkan jadwal yang diperoleh perawat satu dengan yang lainnya. Pada tahap ini, evolusi biologis yang menjadi dasar dari Algoritma Genetika sangat sesuai diterapkan untuk mengatasi masalah penjadwalan, jika Algoritma Genetika diterapkan dalam proses penjadwalan maka kegiatan yang dilakukan secara manual untuk menyusun jadwal dapat diambil alih oleh Algoritma Genetika. Sistem evolusi biologis diterapkan dalam proses seleksi, yang mana yang kuat, dialah yang akan *survive*. Kuat disini diartikan kromosom yang mempunyai nilai *fitness* tinggi. Fungsi *fitness* yang ada pada Algoritma Genetika digunakan sebagai tolok ukur kualitas dari jadwal yang dihasilkan, misalkan suatu jadwal belum memenuhi batas *fitness* yang telah dipatok, maka Algoritma Genetika akan terus memroses sampai ditemukan jadwal yang ideal.

3.1.1 Alat dan bahan yang diperlukan

a. Hardware

Sebuah laptop untuk pembuatan dan pengujian aplikasi dengan spesifikasi sebagai berikut :

- Prosesor Intel core I5
- Memori 4GB

b. Software

- NetBeans 6.1
- My Sql sebagai database
- Sistem operasi windows vista basic

c. Data

Data-data yang diperlukan antara lain adalah data pegawai, data *shift* dan ketentuannya, serta data unit kerja di rumah sakit.

3.1.2 Observasi aplikasi saat ini

Penyusunan jadwal di Rumah Sakit Wawa Husada dilakukan secara manual oleh kepala unit. Setiap sampai pada pertengahan bulan, kepala unit menyusun jadwal baru untuk pegawai. Dalam penyusunan jadwal ini yang harus diperhatikan antara lain adalah ketersediaan pegawai dibandingkan dengan hari kerja, mempertimbangkan jumlah pegawai yang cuti dan pegawai yang akan menggantikannya.

Untuk permohonan ijin cuti, pegawai terlebih dulu melapor ijin cutinya kepada kepala unit sebelum jadwal yang baru dikeluarkan. Setelah itu, pegawai menunggu konfirmasi dari kepala unit mengenai persetujuan ijin cutinya. Setelah jadwal selesai disusun, jadwal tersebut akan diperiksa oleh pihak manajemen pegawai.

a. Komponen penjadwalan personalia

Komponen utama yang digunakan dalam penyusunan jadwal yaitu data pegawai, data *shift*, data tanggal setiap bulannya, dan unit kerja pegawai. Di rumah sakit ini, setiap *shift*nya diberikan waktu 8 jam.

Jenis *shift* diterapkan sesuai dengan unit kerja masing-masing. Misalnya saja pada bagian personalia, jenis *shift* yang digunakan hanya KT, PS, TG dan TP. Penentuan *shift* yang sesuai dengan unit kerja ditentukan diluar penyusunan penjadwalan ini.

b. Aturan penjadwalan

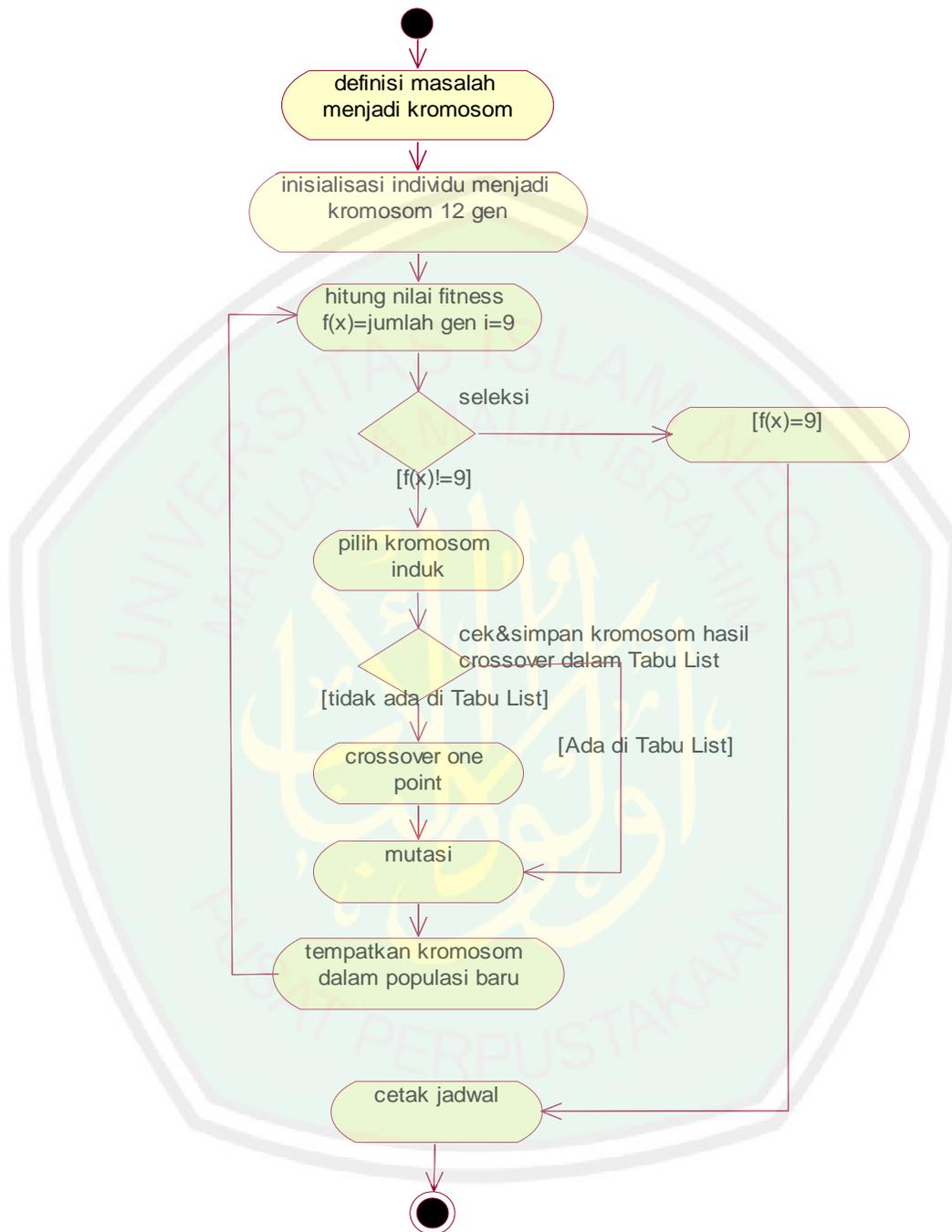
1. Setelah *shift* malam, pegawai mendapatkan libur selama dua hari
2. Jumlah pegawai pada setiap *shift* harus seimbang
3. Libur maksimal lima hari dalam satu bulan
4. Jatah cuti tidak boleh diambil selama 7 hari berturut-turut dalam satu bulan, kecuali cuti melahirkan.

3.1.3 Analisis data

Dari analisa data hasil observasi, diperoleh beberapa permasalahan yang perlu untuk di optimasi menggunakan algoritma dan *agent*. Permasalahan tersebut di antaranya adalah :

- a. Masing- masing pegawai mendapat jadwal maksimal 1 *shift* dalam sehari
- b. Dalam 1 hari terdapat 3 *shift*, dan tidak boleh ada *shift* yang kosong
- c. Tidak boleh ada jadwal *shift* yang bentrok antar pegawai





Gambar 3.1 Activity diagram penjadwalan

Activity diagram proses penghitungan dengan kombinasi Algoritma Genetika dan Tabu Search pada gambar 3.1 dijabarkan sebagai berikut :

a. Definisi masalah menjadi kromosom

Masalah yang dihadapi dalam penjadwalan di Rumah Sakit Wawa Husada adalah banyaknya *shift* yang terdapat dalam satu ruangan. Selain itu, satu pegawai mempunyai kemungkinan dalam satu hari menjalankan 2 *shift* yang berbeda. Hal ini dinilai tidak efektif dari segi kesehatan pegawainya. Oleh karena itu, disusun kromosom untuk menyelesaikan masalah pembagian *shift*. *Shift* dibagi menjadi 3, yakni *shift* pagi, *shift* siang dan *shift* malam. Ketiga *shift* ini diinisialisasikan menjadi 12 digit angka biner.

1	1	1	0	0	1	1	1	0	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---

Gambar 3.2 Representasi obyek dalam biner

1110 : *Shift* Siang

111 : *Shift* Pagi

11100 : *Shift* Malam

b. Inisialisasi individu menjadi 12 kromosom

Individu yang diinisialisasikan menjadi 12 kromosom, diacak nilainya sehingga menghasilkan urutan angka 0 dan 1 yang berbeda. Kromosom ini di acak sampai menghasilkan 12 kromosom`baru. Hasil dari pengacakan kromosom merupakan populasi awal yang akan diproses dengan tahap Algoritma Genetika.

c. Hitung nilai *fitness*

Nilai *fitness* ditentukan sesuai dengan tujuannya, disini nilai *fitness* berperan sebagai acuan untuk proses seleksi dan reproduksi. Nilai *fitness* dihitung berdasarkan total angka 1 pada tiap kromosom. Total angka 1 untuk *fitness* ideal dalam masalah penjadwalan personalia rumah sakit adalah 9.

Nilai *fitness* dihitung menggunakan rumus :

$$\Sigma \text{Gen } i = 9$$

d. Seleksi

Setelah masing-masing kromosom diketahui nilai *fitness*-nya, kromosom tersebut diseleksi, dimana kromosom yang bernilai *fitness* 9 adalah kromosom yang dinyatakan sebagai kromosom yang ideal. Setelah suatu kromosom dinyatakan ideal, maka langkah selanjutnya adalah mencocokkan kromosom tersebut dengan pola jadwal yang bernilai optimal yang kemudian dicetak menjadi jadwal. Namun, jika kromosom belum dinyatakan memiliki nilai *fitness* yang ideal, kromosom tersebut akan diseleksi untuk menjadi kromosom induk yang nantinya akan mengalami tahapan yang ada pada Algoritma Genetika dan di filter oleh *Tabu List*.

e. Pilih kromosom induk

Kromosom yang dipilih menjadi kromosom induk ditentukan oleh tahap seleksi menggunakan *roulette wheel*. Dari kromosom tersebut nantinya akan diperoleh individu baru.

f. Cek dan simpan dalam *Tabu List*

Kromosom induk yang telah ditentukan untuk di *crossover* melalui proses seleksi, pada saat iterasi yang pertama, kromosom tersebut langsung disimpan dalam *Tabu List*, kemudian dilakukan *crossover*. Untuk iterasi berikutnya, kromosom yang mengalami *crossover* di cek terlebih dahulu pada *Tabu List*, jika kromosom tersebut sudah ada pada *Tabu List*, maka kromosom itu akan langsung di mutasi,

namun jika kromosom itu belum ada pada *Tabu List*, maka kromosom itu akan disimpan dalam *Tabu List* terlebih dahulu dan untuk selanjutnya di *crossover*.

g. *Crossover*

Crossover yang digunakan untuk masalah penjadwalan ini adalah *crossover one point*, yaitu *crossover* yang memotong pada satu titik kromosomnya.

110111011101 : Kromosom Induk 1
111111100011 : Kromosom Induk 2
110111100011 : Kromosom Anak 1
111111011101 : Kromosom Anak 2

h. Mutasi

Untuk masalah penjadwalan ini, mutasi yang digunakan adalah mutasi secara acak, caranya adalah dengan mengubah salah satu gen yang ada pada kromosom hasil *crossover*. Tahap mutasi ini dimaksudkan untuk mempertahankan kemiripannya dengan kromosom induk.

i. Populasi baru

Setelah melewati tahap mutasi, hasil dari proses tersebut ditempatkan pada populasi yang baru atau generasi baru. Populasi baru ini selanjutnya akan diproses kembali pada tahap perhitungan nilai *fitness* dan tahapan-tahapan selanjutnya hingga kromosom itu mencapai *fitness* optimal

3.2 Perancangan Perangkat Lunak

Pada tahap ini, akan dijabarkan mengenai perancangan aplikasi yang akan dibangun. Terdapat 3 komponen yang digunakan dalam proses penjadwalan, yaitu id pegawai, id *shift* dan tanggal. Dalam suatu penjadwalan, satu id *shift* dan tanggal dapat memiliki beberapa id pegawai, selain itu terdapat pula id pegawai

dan id *shift* yang sama pada tanggal yang berbeda.hal ini terjadi karena dalam 1 hari terdapat 3 *shift*, dan masing-masing *shift* itu di isi oleh beberapa pegawai.

Implementasi perhitungan Algoritma Genetika dimisalkan dengan individu yang berisi 12 kromosom, dan masing-masing kromosom berisi 12 gen. Kromosom-kromosom ini merupakan populasi awal atau generasi pertama yang akan diproses menjadi jadwal melalui tahapan yang ada pada kombinasi Algoritma Genetika dan *Tabu Search*. Implementasi *Tabu Search* adalah sebagai filtering kromosom-kromosom yang mengalami *crossover*, dimana kromosom yang sudah pernah di *crossover* akan disimpan pada *Tabu List*, agar selanjutnya kromosom tersebut tidak mengalami *crossover* ulang yang sia-sia. Hal ini bertujuan untuk mengurangi kemungkinan iterasi yang terjadi. Populasi awal dapat dilihat pada **tabel 3.1**.

Tabel 3.1 Hasil Random Kromosom

Kromosom 1	100011111011
Kromosom 2	011100101101
Kromosom 3	000111001011
Kromosom 4	110010100000
Kromosom 5	011100101101
Kromosom 6	001101111111
Kromosom 7	100100100000
Kromosom 8	101011111011
Kromosom 9	001110010000
Kromosom 10	000111001011
Kromosom 11	100011111011
Kromosom 12	001011100011

Jika kromosom telah ditentukan, maka tahap selanjutnya adalah menghitung nilai *fitness*nya. Dari nilai *fitness* inilah yang nantinya digunakan sebagai acuan untuk menentukan apakah suatu kromosom sudah dikatakan ideal atau belum.

Angka 9 diperoleh dari jumlah angka biner yang bernilai 1 dari setiap kromosom 11111001110, biner ini adalah representasi kromosom yang bernilai optimum. Kromosom tersebut bermakna *shift* dan hari libur untuk sebuah komponen penjadwalan. 3 digit dari kiri (111) merupakan inisialisasi untuk *shift* pagi, 3 digit berikutnya adalah inisialisasi untuk *shift* malam yang diikuti dengan libur 2 hari, yaitu (11100). Hari libur pertama adalah wajib untuk pegawai, sedangkan libur yang kedua adalah libur cadangan. Jika sewaktu-waktu terdapat pegawai lain yang cuti, maka pegawai yang sedang libur cadangan berkewajiban untuk menggantikan pegawai yang sedang cuti. Hal ini dimaksudkan agar semua pegawai mendapatkan jatah cuti yang sama. Untuk 3 digit terakhir yang diikuti angka 0 (1110), adalah inisialisasi untuk *shift* siang dengan libur satu hari. Dapat disimpulkan dari penjabaran ini, jika angka biner yang bernilai 1 jumlahnya belum mencapai 9 atau mungkin lebih dari 9, maka nilai *fitness*nya dianggap belum optimal, karena belum sesuai dengan fungsi tujuan.

Tahap selanjutnya adalah menghitung nilai *fitness* pada kromosom, yang hasilnya dapat digunakan sebagai acuan untuk menilai kromosom tersebut ideal atau belum dan sejauh mana kualitasnya. Prinsip dasar dari Algoritma Genetika adalah, bahwa individu yang memiliki kualitas lebih unggul adalah individu yang berhak bertahan. Pada masalah ini dapat diimplementasikan pada nilai *fitness* yang muncul pada kromosom, semakin mendekati optimal nilai *fitness*nya maka semakin baik kualitas dari kromosom itu. Dan kromosom yang memiliki kualitas baik, akan lebih mempunyai kesempatan untuk berkembang biak. Hasil nilai *fitness* dapat dilihat pada **tabel 3.2**

Tabel 3.2 Hasil Perhitungan Nilai *Fitness*

Kromosom 1	000111001011	6
Kromosom 2	011100101101	7
Kromosom 3	000111001011	6
Kromosom 4	110010100000	4
Kromosom 5	100111001111	8
Kromosom 6	100011111011	8
Kromosom 7	100100100000	3
Kromosom 8	110010100000	4
Kromosom 9	001110010000	4
Kromosom 10	100010110110	6
Kromosom 11	000100110110	5
Kromosom 12	000100110110	5

Sesuai dengan prinsip Algoritma Genetika, individu yang memiliki kualitas buruk berarti tidak layak untuk melakukan *crossover*. Pada masalah ini, kromosom-kromosom yang telah diketahui nilai *fitness*nya akan diurutkan berdasarkan besar dan kecilnya nilai *fitness*.

3.3.1 Desain Input

Desain input untuk aplikasi ini adalah unit yang akan diproses. Jadi dengan menginputkan unit mana yang akan diproses, maka sistem hanya akan memproses penjadwalan pada unit tersebut.

3.3.2 Desain Output

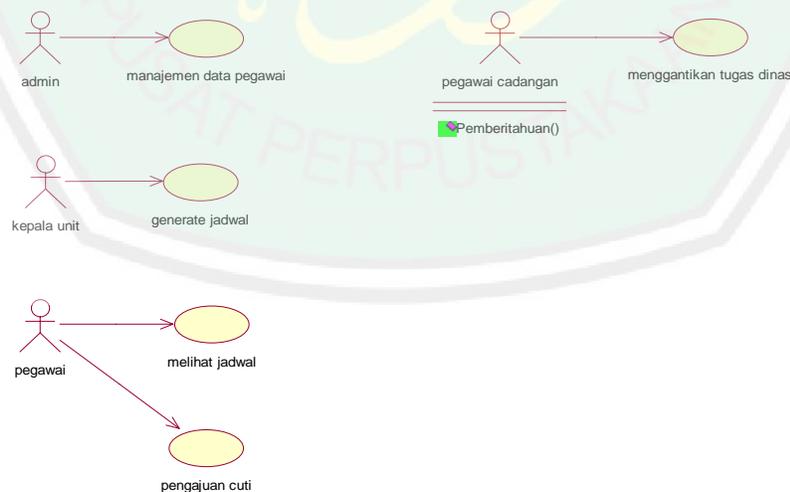
Output dari aplikasi ini berupa jadwal yang telah sesuai dengan aturan jadwal yang optimal. Jadwal ini berisi nama pegawai, tanggal dan *shift*. Jadwal tersebut akan ditampilkan melalui monitor dan dievaluasi oleh admin sistem. Apabila jadwal yang ada sudah disetujui oleh admin, maka jadwal tersebut di *print out* dan disebarakan ke pegawai.

3.3.3 Desain Proses

Algoritma Genetika dan *Tabu Search* memiliki beberapa tahapan dalam menyusun jadwal. Dimulai dari inialisasi masalah, dalam bentuk kromosom, penghitungan nilai *fitness*, seleksi, setelah *crossover*, Algoritma *Tabu Search* akan menyeleksi kromosom apakah sudah ada di dalam *Tabu List* atau belum, jika belum ada maka kromosom tersebut akan disimpan, tetapi jika sudah ada maka akan dilakukan inialisasi kromosom baru, kemudian diolah lagi menggunakan Algoritma Genetika. Setelah kromosom disimpan dalam *Tabu List*, kromosom tersebut akan dimutasi kembali dengan menggunakan Algoritma Genetika.

Dalam pembuatan jadwal ini, tidak hanya alhoritma genetika dan *Tabu Search* saja yang berperan, akan tetapi terdapat tahapan lain yang harus dilakukan. Tahapan-tahapan tersebut akan dijelaskan menggunakan pemodelan UML pada gambar 3.4 :

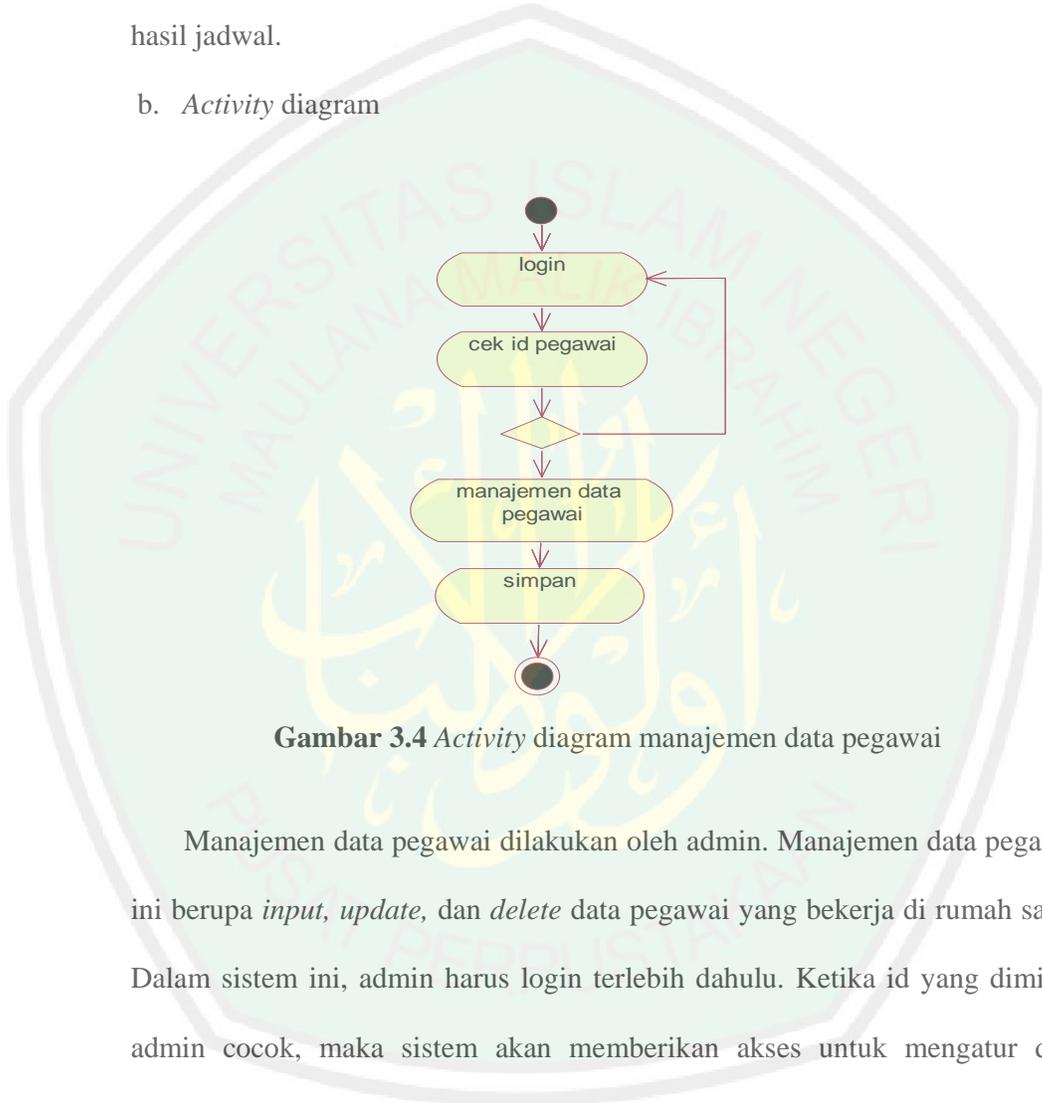
a. Use Case Diagram



Gambar 3.3 Use Case sistem penjadwalan personalia rumah sakit

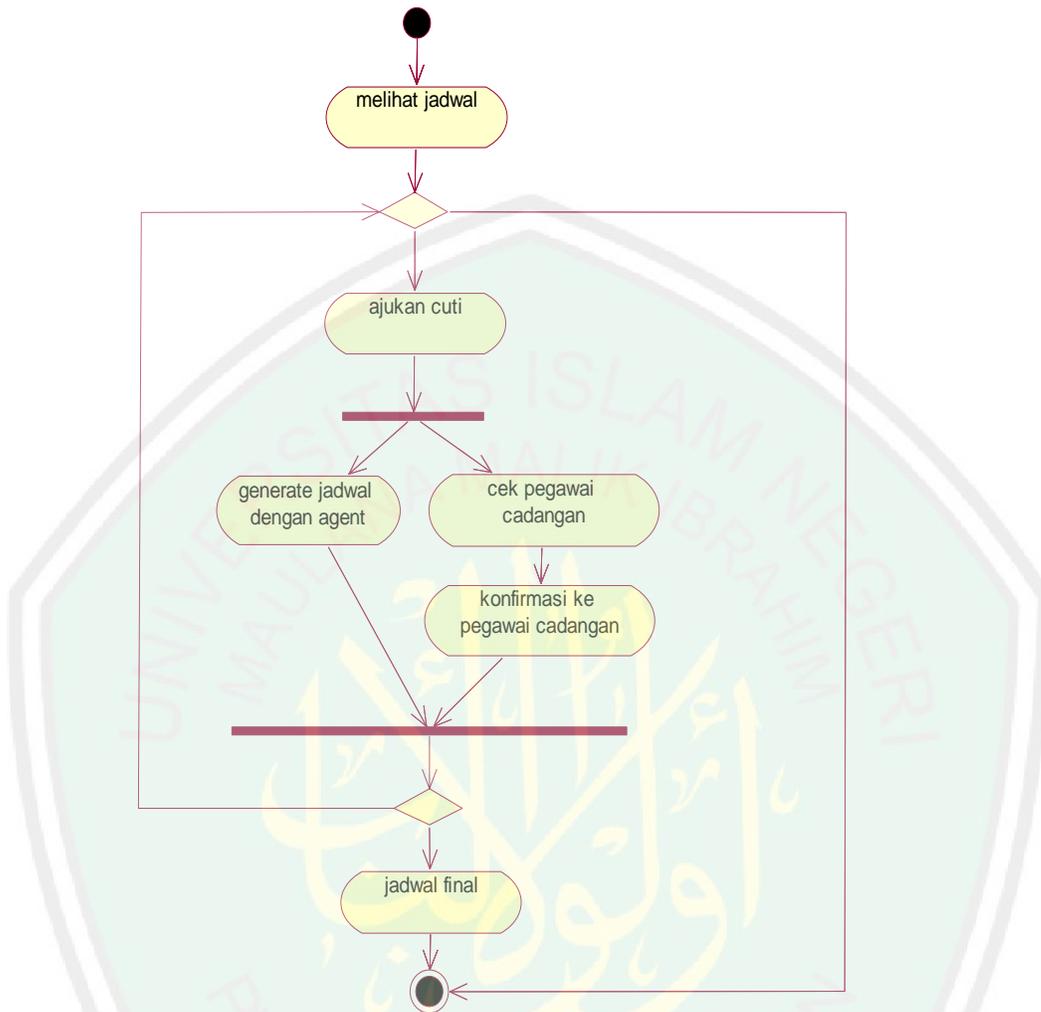
Pada *use case* diagram di atas, aktor yang berperan adalah pegawai dan admin. Admin berperan menjalankan sistem penjadwalan dan mengentry data komponen, sedangkan pegawai berperan ketika menginputkan cuti dan melihat hasil jadwal.

b. *Activity* diagram



Gambar 3.4 *Activity* diagram manajemen data pegawai

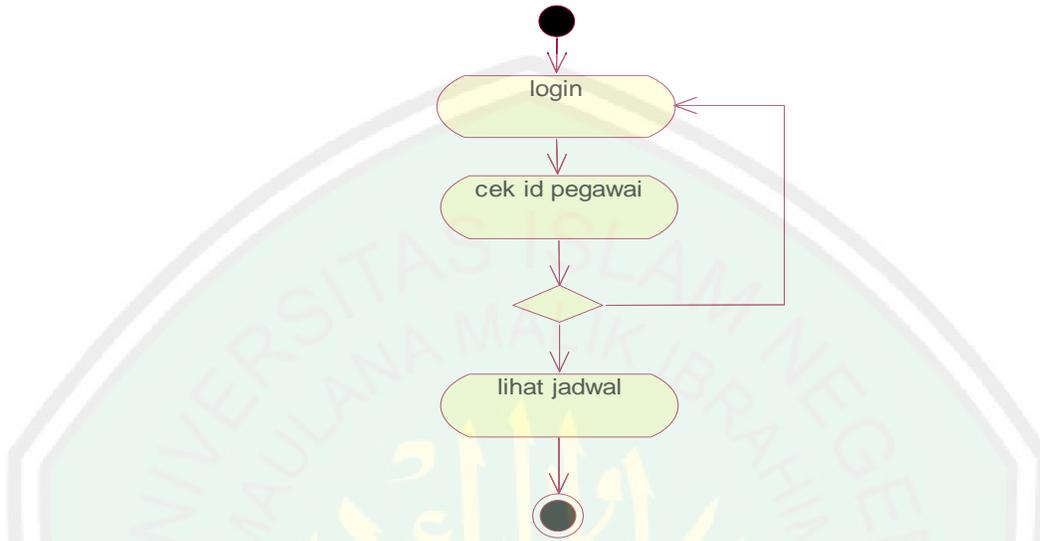
Manajemen data pegawai dilakukan oleh admin. Manajemen data pegawai ini berupa *input*, *update*, dan *delete* data pegawai yang bekerja di rumah sakit. Dalam sistem ini, admin harus login terlebih dahulu. Ketika id yang dimiliki admin cocok, maka sistem akan memberikan akses untuk mengatur data pegawai rumah sakit.



Gambar 3.5 Activity diagram pengajuan cuti

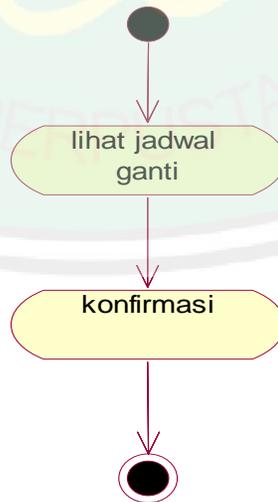
Cuti merupakan hak setiap pegawai dalam rumah sakit. Jika pegawai ingin mengajukan cuti, para pegawai harus melihat jadwal terlebih dahulu. Ketika pegawai telah mengajukan cuti, sistem akan memroses dengan mengecek pegawai, kemudian mengkonfirmasi kepada pegawai cadangan serta mencetak jadwal sebelumnya dengan *agent*. Setelah sistem berjalan dan memenuhi ketentuan, yakni pegawai cadangan bersedia menggantikan jadwal masuk pegawai

yang sedang cuti dan jadwal telah di-generate oleh *agent*, maka jadwal baru akan ditampilkan sebagai jadwal akhir.



Gambar 3.6 Activity diagram melihat jadwal

Aktifitas yang dilakukan oleh pegawai jika ingin melihat jadwal adalah login terlebih dahulu. ketika id yang dimiliki pegawai cocok, maka sistem akan menampilkan jadwal kerja terbaru pegawai.



Gambar 3.7 Activity diagram menggantikan tugas dinas

Semua pegawai memiliki cuti serta memiliki kewajiban untuk menggantikan tugas dinas untuk menggantikan pegawai lain yang sedang melakukan cuti. Oleh karena itu, pegawai cadangan ketika melihat jadwal ganti memiliki kewajiban untuk mengkonfirmasi pergantian tugas dinas.

3.3.4 Desain Database

Desain database berfungsi sebagai tempat rancangan data dan tabel yang akan digunakan dalam pembuatan aplikasi ini. Berikut adalah beberapa tabel yang akan dimasukkan dalam database :

1. Tabel pegawai

Tabel pegawai berisi tentang identitas seluruh pegawai yang bekerja. Dari tabel pegawai inilah yang nanti akan digunakan untuk mengambil id pegawai.

Tabel 3.3 Tabel pegawai

No	Nama Field	Tipe Data
1	Id_pegawai	Integer
2	Nama_pegawai	Varchar
3	Alamat	Varchar
4	No_Telpon	Varchar
5	Tempat_lahir	Varchar
6	Tanggal_lahir	Date
7	Id_unit	Integer

2. Tabel *shift*

Tabel *shift* berisi daftar *shift* beserta waktu yang berlaku untuk setiap *shift*nya.

Dari tabel ini id *shift* di ambil sebagai komponen penjadwalan.

Tabel 3.4 Tabel *shift*

No.	Nama Field	Tipe Data
1	Id_ <i>shift</i>	Integer
2	<i>Shift</i>	Varchar
3	Jam_masuk	Varchar
4	Jam_keluar	Varchar

3. Tabel unit kerja

Tabel unit kerja berisi daftar unit kerja yang ada di rumah sakit.

Tabel 3.5 Tabel unit kerja

No.	Nama Field	Tipe Data
1	Id_unit	Integer
2	Unit_kerja	Varchar

4. Tabel tukar dinas

Tabel tukar dinas berisi data pegawai yang melakukan tukar dinas. Dari tabel inilah jumlah tukar dinas yang telah dilakukan pegawai akan diketahui.

Tabel 3.6 Tabel tukar dinas

No.	Nama Field	Tipe Data
1	Id_tukar	Integer
2	Tanggal	Date
3.	Id_peg_tukar	Integer
4.	Freq_tukar	Integer
5.	Id_peg_ganti	Integer
6.	Freq_ganti	Integer
7.	Tgl_awal	Date
8.	Tgl_akhir	Date
9.	Keterangan	Date

5. Tabel Jadwal

Tabel jadwal berisi hasil dari proses penjadwalan.

Tabel 3.7 Tabel Jadwal

No.	Nama Field	Tipe Data
1.	Id_jadwal	Integer
2.	Id_pegawai	Integer
3.	Id_shift	Integer
4.	Id_unit	Integer
5.	tanggal	Date

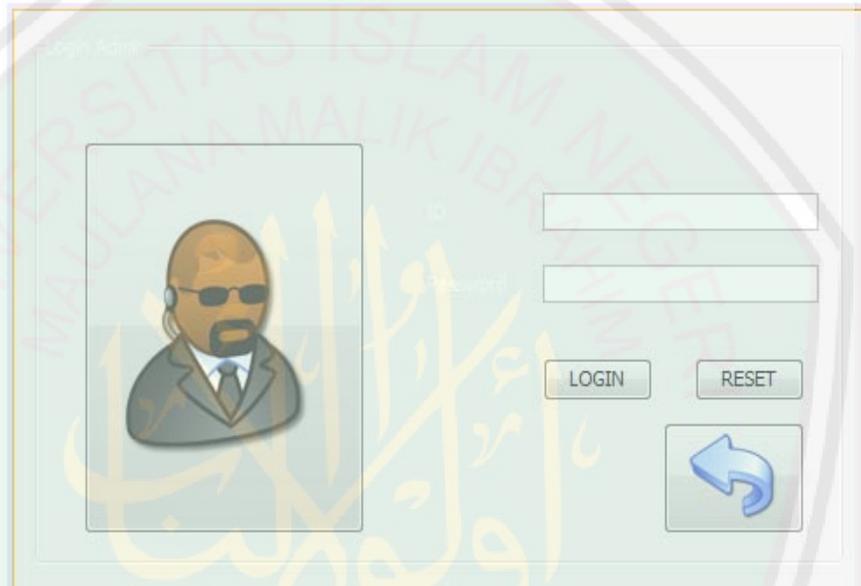
3.3.5 Desain Interface

Desain *interface* dibuat dengan *software* netbeans 7.1, pada desain *interface* ini akan dijelaskan bagaimana aplikasi akan dijalankan nantinya.



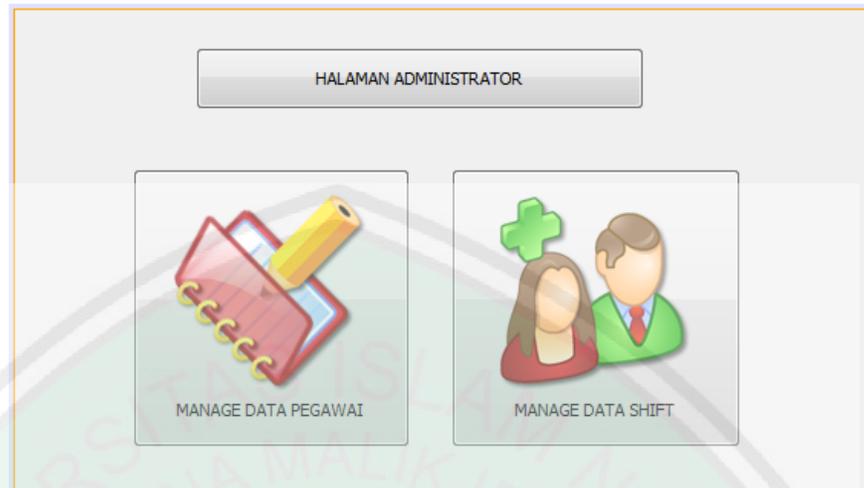
Gambar 3.8 Tampilan awal aplikasi

Gambar 3.9 merupakan desain *interface* untuk tampilan awal aplikasi yang akan digunakan. Pada tampilan awal ini terdapat 2 fungsi. Fungsi yang pertama adalah untuk digunakan admin sebagai penghubung dengan form loginnya, dan fungsi ke dua adalah sebagai penghubung kepala unit dengan form loginnya.



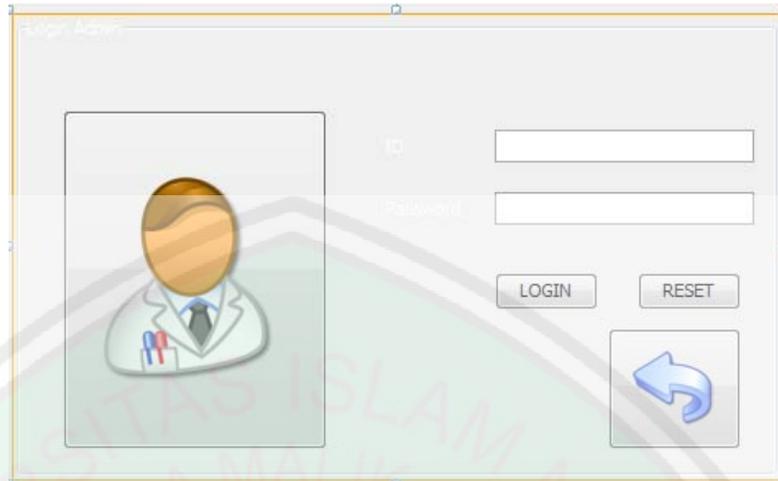
Gambar 3.9 Tampilan login admin

Gambar 3.10 adalah tampilan login yang dapat digunakan oleh admin untuk login ke dalam aplikasi. Jika id dan *password* yang dimiliki admin sesuai maka admin dapat masuk ke halaman selanjutnya.



Gambar 3.10 Tampilan halaman administrator

Pada tampilan administrator, admin diberikan dua pilihan. Yang pertama adalah masuk ke dalam manage data pegawai yang nantinya akan terhubung dengan kumpulan data pegawai yang telah tersimpan dalam *database*. Disana admin dapat melakukan *entry*, *edit* dan *delete* data pegawai. Sedangkan pada manage data *shift*, admin dapat menginput jenis *shift* yang ada beserta jam berlakunya *shift* tersebut.



Gambar 3.11 Tampilan login kepala unit

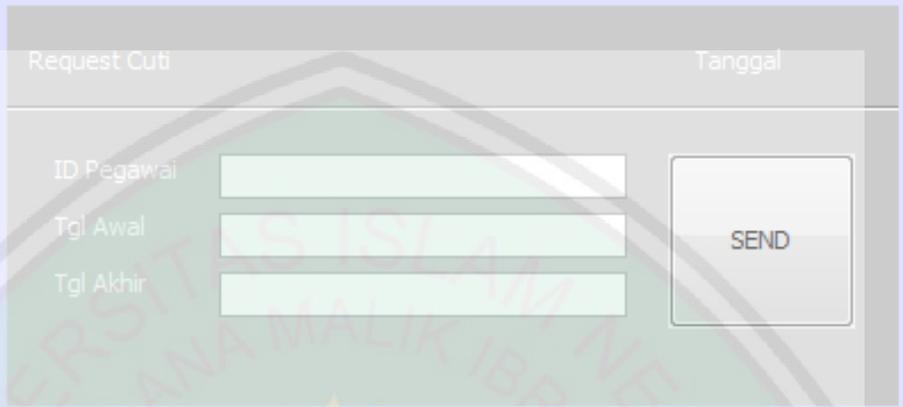
Pada halaman ini kepala unit harus memasukkan id dan passwordnya jika ingin masuk ke halaman berikutnya.



Gambar 3.12 Halaman kepala unit

Pada halaman ini kepala unit dapat memilih halaman manakah yang akan dituju. Buat jadwal adalah halaman yang akan digunakan untuk memroses jadwal baru, input cuti digunakan apabila ada perawat yang

mengajukan cuti, sedangkan view jadwal akan menampilkan jadwal yang telah selesai diproses.



Request Cuti	Tanggal
ID Pegawai	<input type="text"/>
Tgl Awal	<input type="text"/>
Tgl Akhir	<input type="text"/>
<input type="button" value="SEND"/>	

Gambar 3.13 Tampilan request cuti

Pada halaman ini kepala unit harus menginputkan id perawat beserta tanggal mulai cuti dan tanggal akhir. Setelah itu klik send untuk selanjutnya diproses oleh *agent*.

MESSAGE CUTI

CARI PENGGANTI

Tanggal Awal

Tanggal Akhir

Cari

ID	Tanggal Libur

Bersedia

Gambar 3.14 Tampilan hasil proses cuti

Pada halaman ini akan nantinya akan ditampilkan hasil proses cuti yang dilakukan oleh *agent* cuti. *Agent* cuti akan mencari pengganti untuk mengisi *shift* dari perawat yang sedang cuti.

BAB IV

HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil uji coba sistem penjadwalan personalia yang telah dilakukan. Hal ini bertujuan untuk mengetahui apakah sistem sudah sesuai dengan yang diharapkan atau belum, yaitu *output* berupa jadwal kerja perawat yang memenuhi pola penjadwalan yang telah ditetapkan sebelumnya. Disamping itu, uji coba pada sistem penjadwalan yang menerapkan kombinasi Algoritma Genetika dan *Tabu Search* akan dibandingkan dengan sistem penjadwalan yang hanya menerapkan Algoritma Genetika saja. Sehingga dari hasil uji coba yang didapatkan akan diketahui metode manakah yang paling optimal untuk digunakan untuk menyelesaikan masalah penjadwalan personalia rumah sakit.

4.1. Sumber Data

Sumber data untuk aplikasi penjadwalan personalia rumah sakit diperoleh dari hasil wawancara dengan pihak Rumah Sakit Wawa Husada di Kepanjen Kabupaten Malang. Data-data yang dikumpulkan meliputi jenis *shift* pada Rumah Sakit Wawa Husada, aturan penjadwalan pada perawat dengan sistem *shift*, dan jumlah unit pada Rumah Sakit.

4.2. Implementasi Aplikasi Penjadwalan Personalia

Berdasarkan pada perancangan dan analisis pada bab sebelumnya, pada bab ini akan dijelaskan mengenai hasil implementasi yang telah dilakukan. Mulai dari pengimplementasian jadwal pada Algoritma Genetika, *filter* kromosom yang dilakukan oleh *Tabu Search* hingga output yang dihasilkan. Terdapat beberapa tahap untuk memroses jadwal dengan kombinasi Algoritma Genetika dan *Tabu Search*, yang dimulai dari inialisasi state awal atau *decoding*. Yakni menyusun kromosom berisi 12 gen yang menggunakan angka biner, tahap penghitungan *fitness*, filterisasi dengan *Tabu list*, *crossover*, mutasi sampai tahap *encoding*.

Individu awal diinisialisasikan dengan 12 kromosom, dimana setiap kromosom mewakili pegawai yang sedang bekerja maupun libur pada 12 hari. Inialisasi untuk kromosom dari penjadwalan dapat dilihat pada **tabel 4.1**

Tabel 4.1 Inisialisasi kromosom

MASUK=1 ; LIBUR=0											
1	1	1	0	0	1	1	1	0	1	1	1
1	1	0	0	1	1	1	0	1	1	1	1
1	0	0	1	1	1	0	1	1	1	1	1
0	0	1	1	1	0	1	1	1	1	1	1
0	1	1	1	0	1	1	1	1	1	1	0
1	1	1	0	1	1	1	1	1	1	0	0
1	1	0	1	1	1	1	1	1	0	0	1
1	0	1	1	1	1	1	1	0	0	1	1
0	1	1	1	1	1	1	0	0	1	1	1
1	1	1	1	1	1	0	0	1	1	1	0
1	1	1	1	1	0	0	1	1	1	0	1
1	1	1	1	0	0	1	1	1	0	1	1

Kromosom pada **tabel 4.1** memiliki 3 warna yang berbeda. yang pertama adalah kotak yang berwarna ungu, terdiri dari 3 digit angka 1 yang diikuti dengan 2 digit angka 0 merupakan inisialisasi untuk *shift* malam. Angka biner yang berwarna hijau adalah inisialisasi dari *shift* siang yang terdiri dari 3 digit angka 1 dan 1 digit angka 0. Yang terakhir adalah *shift* pagi, kotak biner yang berwarna kuning berisi 3 digit angka 1. pemberian hari libur yang diwakili dengan angka 0 telah dipertimbangkan dengan tujuan untuk memperhatikan kesehatan fisik maupun psikis dari pegawai.

Tahap *decoding* disini adalah menginisialisasikan masalah ke dalam bentuk kromosom, tahap selanjutnya adalah memasukkan kromosom ke dalam

Tabu List, hal ini bertujuan untuk memberi angka awal pada *Tabu List* yang pada generasi berikutnya digunakan sebagai pembandingan, setelah semua kromosom dimasukkan dalam *Tabu List* kemudian baru dilakukan proses *crossover*. yaitu menyilangkan antara biner yang pertama dengan biner yang kedua, setelah itu baru dilakukan mutasi. Pada generasi kedua dan selanjutnya terdapat perbedaan pada tahap *crossover*, dimana pada generasi pertama semua kromosom *dicrossover*, namun pada generasi kedua, setiap kromosom terlebih dahulu di cek pada *Tabu List*, jika ada kromosom yang kembar maka kromosom tersebut langsung dimutasi tanpa melalui proses *crossover* lagi.

```

Output - trans (run-single)  acak3.java  array.java  Koneksi.java
hasil mutasi= [1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1]
360 generasi ke 34 [0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1] [0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1] hasil cros= [0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1]
hasil mutasi= [1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1]
361 generasi ke 34 [1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1] [0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0] hasil cros= [1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0]
hasil mutasi= [1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0]
362 generasi ke 34 [0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0] [0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1] hasil cros= [0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1]
hasil mutasi= [1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1]
list ke 363 [1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1] << biner 1 ada pada tabulist ke- 535
dimutasi >> [1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1]
364 generasi ke 34 [0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0] [1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0] hasil cros= [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0]
hasil mutasi= [1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0]
list ke 365 [0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1] << biner 1 ada pada tabulist ke- 628
dimutasi >> [1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1]
list ke 366 [1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0] << biner 1 ada pada tabulist ke- 58
dimutasi >> [1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0]
367 generasi ke 34 [1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0] [1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1] hasil cros= [1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1]
hasil mutasi= [1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1]
368 generasi ke 34 [0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0] [0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1] hasil cros= [0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1]
hasil mutasi= [1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1]

```

Gambar 4.1 Hasil perhitungan Algoritma Genetika dan *Tabu Search*

Gambar 4.1 merupakan hasil dari proses penerapan kombinasi algoritma Genetika dan *Tabu Search*, hasil *crossover*, mutasi hingga hasil kerja *Tabu List* dapat dilihat pada potongan output tersebut. *Crossover* diperoleh dari penyilangan biner 1 dan biner 2 yang dipotong pada titik tengahnya, lalu potongan dari

masing-masing biner dijadikan 1 kromosom, dan kromosom itulah yang merupakan hasil *crossover*. Sedangkan mutasi diperoleh dengan cara mengganti salah satu nilai 0 yang ada pada hasil *crossover* dengan nilai 1. Untuk hasil kerja *Tabu List* diperoleh dengan cara membandingkan kromosom dengan kromosom yang telah tersimpan dalam *Tabu List*. Dari **gambar 4.1**, kromosom yang diberi keterangan sudah ada dalam *Tabu List* langsung dimutasi tanpa mengalami *crossover* terlebih dahulu, namun untuk kromosom yang belum ada pada *Tabu List* terlebih dahulu *dicrossover* sebelum dimutasi.

Adapun *source code* yang digunakan untuk menjalankan menjalankan tahap Algoritma Genetika dan *Tabu Search* adalah sebagai berikut :

```
Vector tb = new Vector();
    Vector biner = new Vector();
    Vector biner2 = new Vector();
    Collections.addAll(biner, "0", "0", "1", "1", "1", "1", "1", "1",
"1", "1", "1", "1");
    Collections.addAll(biner2, "1", "1", "1", "1", "1", "1", "0", "0",
"1", "1", "1", "0");
```

Gambar 4.2 Potongan source code *decoding*

```
for (int b = 0; b < 12; b++) {
    biner.add(random());
    biner2.add(random());
```

Gambar 4.3 Potongan *source code* mengacak kromosom

```

if (jmlBilangan==0) {

    System.out.println("GENERASI ke : "+ pop);

    bilangan.put(jmlBilangan, stringbiner);

    bilangan.put(jmlBilangan + 1, stringbiner2);

    String crossover = stringbiner.substring(0,
tengah).concat(stringbiner2.substring(tengah));

    System.out.println(i + " generasi ke " +pop+" "+
stringbiner + " " + stringbiner2 + " hasil cross= " + crossover);

    mutasi = crossover.replaceFirst("0", "1");

    System.out.println(" hasil mutasi= " + mutasi);
}

```

Gambar 4.4 Potongan Source Code *Tabu List* Generasi Pertama

```

for (int l = jmlBilangan - 1; l >= 0; l--) {

    if (((bilangan.get(l)).equals(stringbiner)) ||
((bilangan.get(l)).equals(stringbiner2))) {

        if ((bilangan.get(l)).equals(stringbiner)) {

            mutasi = stringbiner.replaceFirst("0", "1");

            System.out.println("list ke "+i+ "
"+stringbiner+" << biner 1 ada pada tabulist ke- "+l);

            count1++;

            System.out.println(" dimutasi >> "+mutasi);

        } else if ((bilangan.get(l)).equals(stringbiner2))

        {

            System.out.println("list ke "+i+ "
"+stringbiner2+ " << biner 2 tidak masuk tabu lagi "+l);

            mutasi = stringbiner2.replaceFirst("0", "1");

            count2++;

            System.out.println(" dimutasi >> "+mutasi);
}
}

```

Gambar 4.5 Potongan source code *Tabu List* generasi ke 2 dan selanjutnya

```
String crossover = stringbiner.substring(0,
tengah).concat(stringbiner2.substring(tengah));

        System.out.println(i + " generasi ke " +pop+ " " +
stringbiner + " " + stringbiner2 + " hasil cross= " + crossover);
```

Gambar 4.6 Potongan *source code* crossover

```
mutasi = crossover.replaceFirst("0", "1");

        System.out.println(" hasil mutasi= " + mutasi);
```

Gambar 4.7 Potongan *source code* mutasi

```
public String fitness(String str2) {
    String fix = null;
    StringTokenizer token = new StringTokenizer(str2, " 0[,]");
    if (token.hasMoreTokens()) {
        int kata2 = token.countTokens();
```

Gambar 4.8 Potongan *source code* fungsi *fitness*

```
public class Koneksi {
    String url="jdbc:mysql://localhost/jadwal";
    String user="root";
    String pass="toor";
    String driver="com.mysql.jdbc.Driver";
    Connection con;
    public Connection logon(){
        try{
            Class.forName(driver);
            con=DriverManager.getConnection(url,user,pass);
            System.out.println("Koneksi Berhasil!!!");
        }
        catch(Exception ex){
            ex.printStackTrace();
            System.out.println("Koneksi Gagal!" +ex);
        }
        return con;
    }
}
```

Gambar 4.9 Potongan *source code* koneksi

```

public class Broker extends Agent {
    protected void setup() {
        addBehaviour (new brobekaviour(this,1000));
    }
}
class brobekaviour extends TickerBehaviour{
    public brobekaviour(Agent a, int i) {
        super(a,i);
    }
    public void onTick () {
        MessageTemplate dari_broker =
        MessageTemplate.and(MessageTemplate.MatchSender(new AID("AgentPegawai",
        AID.ISLOCALNAME)), MessageTemplate.MatchPerformative(ACLMessage.INFORM));
        ACLMessage msg = myAgent.receive(dari_broker);
        if (msg!=null) {
            String title1 = msg.getContent();
            ACLMessage msg2 = new ACLMessage(ACLMessage.INFORM);
            msg2.setContent(title1);
            msg2.addReceiver(new AID("AgentCuti",AID.ISLOCALNAME));
            myAgent.send(msg2);
        }
        MessageTemplate dari_broker1 =
        MessageTemplate.and(MessageTemplate.MatchSender(new AID("AgentCuti",
        AID.ISLOCALNAME)), MessageTemplate.MatchPerformative(ACLMessage.INFORM));
        ACLMessage msg1 = myAgent.receive(dari_broker1);
        if (msg1!=null) {
            String title2 = msg1.getContent();
            ACLMessage msg3 = new ACLMessage(ACLMessage.INFORM);
            msg3.setContent(title2);
            msg3.addReceiver(new AID("AgentPegawai",AID.ISLOCALNAME));
            myAgent.send(msg3);
        }
    }
}

```

Gambar 4.10 Potongan source code broker agent

```

public class PegCuti extends Agent {
    private ReqCuti request;
    protected void setup() {
        addBehaviour(new cuti(this, 2000));
        request = new ReqCuti();
        request.show();
    }

    class cuti extends TickerBehaviour {
        cuti(Agent g, int i) {
            super(g, i);
        }

        @Override
        protected void onTick() {
            String l = request.idl;
            String k = request.tglk;
            String m = request.tglw;
            ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
            msg.setContent("Id Pegawai\t\t:" + l + "\nTanggal Awal Cuti\t:" +
m + "\nTanggal Akhir Cuti\t:" + k);
            msg.addReceiver(new AID("broker", AID.ISLOCALNAME));
            myAgent.send(msg);

            MessageTemplate dari_server =
MessageTemplate.and(MessageTemplate.MatchSender(new AID("broker",
AID.ISLOCALNAME)), MessageTemplate.MatchPerformative(ACLMessage.INFORM));
            ACLMessage msg1 = myAgent.receive(dari_server);
            if (msg1 != null) {
                String title = msg1.getContent();
                ReqCuti.teksPesan.append("\nKonfirmasi\t:\n" + title);
            }
        }
    }
}

```

Gambar 4.11 Potongan *source code request cuti*

Uji coba dalam penelitian ini dilakukan dengan cara membuat dua program yang berbeda algoritma penyelesaian namun diterapkan dalam data yang sama. Program yang pertama di uji adalah program yang hanya menerapkan Algoritma Genetika, selama proses *running* berlangsung, hal yang perlu di dokumentasikan adalah jumlah iterasi dan waktu yang diperlukan pada setiap pola jadwal yang dihasilkan. Setelah program berhenti, dilakukan pengecekan terhadap *output* yang dihasilkan. Apabila terdapat ketidakcocokan antara aturan penjadwalan dengan pola jadwal yang telah dihasilkan, maka uji coba dihentikan terlebih dahulu dan selanjutnya dilakukan pembenahan pada *source code*

pengaturan pola jadwal. Namun apabila tidak terdapat kesalahan pada output jadwal yang dihasilkan, maka program kedua yaitu program yang di dalamnya terdapat penerapan kombinasi Algoritma Genetika dan *Tabu Search* dijalankan dan dilakukan dokumentasi seperti pada langkah program pertama. Setelah itu, hasil dari dokumentasi kedua program dibandingkan dalam sebuah tabel untuk diketahui metode manakah yang lebih optimal dalam segi iterasi dan waktu yang diperlukan untuk menghasilkan suatu jadwal optimal.

Uji coba selanjutnya adalah uji coba yang dilakukan pada bagian cuti pegawai. Disini yang berperan mencari pegawai pengganti adalah *agent* cuti. *Agent* cuti dibekali pengetahuan berupa aturan-aturan yang digunakan untuk memilih kandidat pegawai pengganti. Aturan tersebut diantaranya adalah :

1. Pegawai pengganti harus berada dalam posisi libur.
2. Pegawai pengganti pernah melakukan cuti, dan apabila terdapat beberapa pegawai yang pernah cuti, maka yang diambil adalah pegawai yang paling banyak melakukan cuti, sortir data dilakukan mulai id pegawai paling atas.

Pada sistem ini, yang berperan sebagai aktor adalah kepala unit. Untuk memroses cuti, kepala unit terlebih dahulu *login* ke dalam sistem kemudian masuk pada halaman *input* cuti. Selanjutnya melakukan pengisian data yang diperlukan seperti id pegawai yang mengajukan cuti, tanggal mulai cuti dan tanggal selesai cuti. Apabila data yang diperlukan sudah terisi lengkap, langkah selanjutnya adalah mengirim data tersebut pada *agent* cuti untuk selanjutnya diproses. Jika proses telah selesai, *agent* cuti akan menampilkan hasilnya pada halaman hasil proses cuti. Pada halaman tersebut akan ditampilkan id pegawai pengganti beserta

keterangan shift yang di dapat pada tanggal permintaan cuti. Untuk memastikan kinerja *agent* cuti, dilakukan pengecekan pada *database* jadwal. Apabila pegawai pengganti memang berada dalam posisi libur, maka tidak terjadi kesalahan pada sistem cuti pegawai. Namun apabila pegawai pengganti itu berada dalam posisi *shift* kerja (*shift* pagi, *shift* siang dan *shift* malam) maka program yang mengatur pengetahuan *agent* cuti harus dilakukan pembenahan.

Tabel 4.2 Tabel hasil uji coba

No	Algoritma Genetika & <i>Tabu Search</i>		Algoritma Genetika	
	Jumlah Generasi	Jadwal Optimal	Jumlah Generasi	Jadwal Optimal
1	97	1	298	1
2	163	2	733	2
3	244	3	845	3
4	543	4	1750	4
5	831	5	2241	5
6	1396	6	2674	6

Hasil uji coba yang ada pada **tabel 4.2** menggambarkan bahwa kombinasi Algoritma Genetika dan *Tabu Search* lebih efisien. Hal ini ditunjukkan dengan jumlah generasi yang ke 97 pada kombinasi Algoritma Genetika dan *Tabu Search* sudah dapat menemukan 1 kromosom jadwal yang optimal, sedangkan pada sistem yang menggunakan Algoritma Genetika saja baru mendapat 1 kromosom jadwal yang optimal pada generasi ke 289.

01	02	03	04	05	06	07	08	09	10	11	12
PG	PG	PG	PG	PG	■	PG	PG	PG	■	PG	PG
■	SG	SG	PG	■	ML	ML	■	■	SG	SG	PG
ML	■	■	SG	SG	PG	PG	ML	ML	■	■	SG
SG	PG	PG	ML	ML	■	■	SG	SG	PG	PG	ML
PG	ML	ML	■	■	■	SG	PG	PG	ML	ML	■
■	SG	PG	PG	PG	PG	ML	■	■	SG	SG	PG

Gambar 4.12 Jadwal rumah sakit Wawa husada

Tabel 4.3 Hasil generate jadwal dengan aplikasi

1	2	3	4	5	6	7	8	9	10	11	12
S	S	L	P	P	P	M	M	M	L	E	S
S	L	P	P	P	M	M	M	L	E	S	S
L	P	P	P	M	M	M	L	E	S	S	S
P	P	M	M	M	L	E	S	S	S	L	P
P	M	M	M	L	E	S	S	S	L	P	P
M	M	M	L	E	S	S	S	L	P	P	P

Berdasarkan data uji coba pada **gambar 4.9** yang diambil dari Rumah Sakit Wawa Husada, terdapat beberapa jadwal pegawai yang tidak sesuai dengan aturan yang telah dijelaskan pada bab sebelumnya. Aplikasi penjadwalan personalia rumah sakit telah berhasil mencetak jadwal baru yang sesuai dengan pola penjadwalan yang optimal dengan pembagian *shift* dan hari libur yang merata. Pada **tabel 4.3** dapat dilihat bahwa seluruh pegawai telah mendapatkan jumlah hari kerja dan libur yang sama.

4.3 Penjadwalan dalam Pandangan Islam

Penjadwalan merupakan pengalokasian sumber daya dalam rentang waktu tertentu untuk menyelesaikan sekumpulan tugas. Penjadwalan sangat penting untuk merancang dan memanajemen suatu sistem yang terdiri dari sejumlah kegiatan dengan keterbatasan sumber daya.

Dalam pandangan Islam, waktu merupakan amal shaleh, amal shaleh disini diartikan menjadi tujuan utama dan bukan materi. Orang yang tidak memanfaatkan waktu dengan baik di dalam Al-Qur'an disebut sebagai orang yang merugi.



Artinya :

Supaya jangan ada orang yang mengatakan: "Amat besar penyesalanku atas kelalaianku dalam (menunaikan kewajiban) terhadap Allah, sedang aku Sesungguhnya Termasuk orang-orang yang memperolok-olokkan (agama Allah)(QS.Az-Zumar [39]:56)

Orang yang merugi adalah orang yang tidak dapat memanfaatkan waktu yang di dapatkannya untuk beramal shaleh dan untuk kehidupan yang abadi. Surat Az-Zumar ayat 56 menjelaskan bahwa waktu yang telah diberikan oleh Allah SWT wajib untuk dikelola sebaik-baiknya, agar nantinya tidak ada penyesalan karena telah melalaikan waktu yang diberikan. Untuk mewujudkan hal itu, diperlukan sebuah management waktu yang baik. Seperti menjadwalkan rutinitas dengan baik, khususnya pada penelitian ini menjadwalkan waktu kerja bagi perawat yang bekerja di rumah sakit. Dengan menjadwalkan kegiatan kerja dengan baik dan mentaatinya maka disiplin waktu akan terwujud, sehingga waktu yang telah diberikan Allah SWT tidak akan terbuang sia-sia dan kita tidak akan termasuk ke dalam orang yang merugi seperti yang dijelaskan pada surat Az-Zumar ayat 56 diatas.

BAB V

PENUTUP

5.1 Kesimpulan

Kombinasi Algoritma Genetika dan *Tabu Search* dapat diterapkan pada masalah penjadwalan personalia rumah sakit. Berdasarkan hasil uji coba yang dilakukan pada sistem, yaitu dengan membandingkan sistem yang menerapkan kombinasi Algoritma Genetika dan *Tabu Search* dengan sistem yang hanya menerapkan Algoritma Genetika, menyatakan bahwa sistem yang menerapkan kombinasi Algoritma Genetika dan *Tabu Search* lebih efisien. Hal ini dikarenakan sistem yang menerapkan kombinasi Algoritma Genetika dan *Tabu Search* jauh lebih sedikit melakukan iterasi yaitu sebanyak 1396 kali selama 35 menit 14 detik untuk mendapatkan 6 pola jadwal yang optimal, sedangkan sistem yang menerapkan Algoritma Genetika saja memerlukan iterasi sebanyak 2674 kali selama 57 menit 24 detik untuk mendapatkan pola jumlah pola jadwal yang sama.

5.2 Saran

Berdasarkan hasil uji coba sistem yang telah dilakukan, adapun saran yang dapat diberikan untuk kelanjutan penelitian ini adalah dengan mengujicoba Kombinasi Algoritma Genetika dan *Tabu Search* pada permasalahan yang berbeda.

DAFTAR PUSTAKA

- Berlianty, Intan & Arifin, Miftahol. 2010. *Teknik-Teknik Optimasi Heuristik*. Yogyakarta: Graha Ilmu
- Suyanto. 2010. *Algoritma Optimasi Deterministik atau Probabilistik*. Yogyakarta: Graha Ilmu
- Kusuma Dewi, Sri & Purnomo, Hari. 2005. *Penyelesaian Masalah Optimasi Dengan Teknik-Teknik Heuristik*. Yogyakarta: Graha Ilmu
- Kusumawati, Ririen. 2007. *Artificial Intelligence menyamai kecerdasan buatan Ilahi?*. Malang: Uin Press
- Setemen, Komang & Purnomo, Mauridhi Herry. 2008. *Kombinasi Algoritma Genetika dan Tabu Search dalam Pembuatan Tabel Jadwal Mata Kuliah*. ITS Surabaya
- Suyanto. 2007. *Artificial Intelligence*. Bandung: Informatika
- Ulya, Anisa. 2010. *Penggunaan Algoritma Genetik Dengan Pemodelan Dua Tingkat Dalam Permasalahan Penjadwalan Perawat Pada Unit Gawat Darurat Rumah Sakit XYZ Surabaya*. ITS Surabaya