



# Penjadwalan Tenaga Kebidanan Menggunakan Algoritma Memetika

A. Y. Erwin Dodu<sup>a,\*</sup>, Deny Wiria Nugraha<sup>b</sup>, Subkhan Dinda Putra<sup>c</sup>

<sup>abc</sup> Program Studi Teknik Informatika Jurusan Teknologi Informasi Fakultas Teknik  
Universitas Tadulako

*Naskah Diterima : 14 Februari 2018; Diterima Publikasi : 28 April 2018*

*DOI : 10.21456/vol8iss1pp99-106*

---

## Abstract

The problem of midwife scheduling is one of the most frequent problems in hospitals. Midwife should be available 24 hours a day for a full week to meet the needs of the patient. Therefore, good or bad midwife scheduling result will have an impact on the quality of care on the patient and the health of the midwife on duty. The midwife scheduling process requires a lot of time, effort and good cooperation between some parties to solve this problem that is often faced by the Regional Public Hospital Undata Palu Central Sulawesi Province. This research aimed to apply Memetics algorithm to make scheduling system of midwifery staff at Regional Public Hospital Undata Palu Central Sulawesi Province that can facilitate the process of midwifery scheduling as well as to produce optimal schedule. The scheduling system created will follow the rules and policies applicable in the hospital and will also pay attention to the midwife's preferences on how to schedule them according to their habits and needs. Memetics algorithm is an optimization algorithm that combines Evolution Algorithm and Local Search method. Evolution Algorithm in Memetics Algorithm generally refers to Genetic Algorithm so that the characteristics of Memetics Algorithm are identical with Genetic Algorithm characteristics with the addition of Local Search methods. Local Search in Memetic Algorithm aims to improve the quality of an individual so it is expected to accelerate the time to get a solution.

**Keywords:** Midwife Scheduling; Memetics Algorithm; Evolutionary Algorithm; Local Search

## Abstrak

Permasalahan penjadwalan bidan merupakan salah satu permasalahan yang sering terjadi di rumah sakit. Tenaga kebidanan harus tersedia 24 jam sehari selama satu minggu penuh untuk memenuhi kebutuhan pasien. Karena itu, baik buruknya penjadwalan bidan akan berdampak pada kualitas pelayanan terhadap pasien dan kesehatan bidan yang bertugas. Proses Penjadwalan bidan membutuhkan banyak waktu, tenaga dan kerja sama yang baik antara beberapa pihak untuk menyelesaikannya masalah inilah yang sering dihadapi oleh Rumah Sakit Umum Daerah (RSUD) Undata Palu Provinsi Sulawesi Tengah. Penelitian ini bertujuan mengimplementasikan Algoritma Memetika untuk membangun sistem penjadwalan tenaga kebidanan di RSUD Undata Palu Provinsi Sulawesi Tengah yang dapat memudahkan proses penjadwalan tenaga kebidanan sekaligus dapat menghasilkan jadwal yang optimal. Sistem penjadwalan yang dibuat akan mengikuti aturan serta kebijakan yang berlaku di rumah sakit dan juga akan memperhatikan preferensi (keinginan) dari bidan tentang bagaimana penyusunan jadwal yang sesuai kebiasaan dan kebutuhan mereka. Algoritma Memetika merupakan algoritma optimasi yang menggabungkan Algoritma Evolusi dan metode Pencarian Lokal. Algoritma Evolusi pada Algoritma Memetika pada umumnya merujuk pada Algoritma Genetika sehingga karakteristik Algoritma Memetika identik dengan karakteristik Algoritma Genetika dengan penambahan metode Pencarian Lokal. Pencarian Lokal pada Algoritma Memetika bertujuan untuk meningkatkan kualitas suatu individu sehingga diharapkan dapat mempercepat waktu ditemukannya suatu solusi.

**Kata Kunci:** Penjadwalan Bidan; Algoritma Memetika; Algoritma Evolusi; Pencarian Lokal

---

## 1. Pendahuluan

Tenaga kebidanan harus tersedia 24 (dua puluh empat) jam sehari selama satu minggu penuh (7 hari) untuk memenuhi kebutuhan pasien, hal ini berdampak pada kualitas pelayanan dan kesehatan setiap bidan yang bertugas, untuk itu pelayanan dibagi dalam sistem *shift* atau bisa disebut periode waktu dalam bekerja. Penjadwalan tenaga kebidanan pada RSUD

Undata Palu Provinsi Sulawesi Tengah disusun menggunakan sistem *shift* yang dibagi menjadi 3 (tiga) jenis *shift*, yaitu *shift* pagi (08.00 – 14.00), *shift* siang (14.00 – 20.00) dan *shift* malam (20.00 – 08.00). Pada setiap periode penjadwalan, setiap bidan akan mendapatkan jenis *shift* yang beragam. Pemberian *shift* pada proses penjadwalan tersebut harus memperhatikan beberapa faktor antara lain, kompetensi bidan, pembagian *shift* yang adil untuk

---

\*) Penulis korespondensi: [ayerwin.dodu@gmail.com](mailto:ayerwin.dodu@gmail.com)

setiap bidan, maksimal jam kerja, hari libur setelah shift tertentu, dan *shift* periode selanjutnya harus berhubungan dengan *shift* periode sebelumnya. Jadwal harus memenuhi persyaratan dan ketentuan instansi sementara juga harus memperhitungkan keinginan dan kebutuhan setiap bidan sebaik mungkin. Beberapa faktor tersebut berdampak pada kompleksitas penjadwalan bidan sehingga penyusunan jadwal secara manual cukup sulit untuk dilakukan dan memakan waktu yang cukup lama menyelesaikannya.

Untuk mendapatkan hasil penjadwalan bidan yang terbaik yaitu penjadwalan yang paling sedikit melanggar batasan yang ada, dibutuhkan suatu algoritma optimasi yang dapat meminimalkan pelanggaran batasan dengan baik. Algoritma Memetika (AM) merupakan algoritma optimasi berbasis metode metaheuristik yang menggabungkan Algoritma Evolusi (AE) dan Pencarian Lokal (PL) untuk mendapatkan hasil yang lebih baik dari AE konvensional.

Albar (2013), untuk menyelesaikan masalah penjadwalan kuliah menggunakan Algoritma Genetika Tabu Search dan AM. Pada penelitian tersebut dilakukan perbandingan performa Algoritma Genetik Tabu Search dan AM untuk menyelesaikan masalah penjadwalan kuliah, hasilnya Algoritma Genetik Tabu Search dua hingga tiga kali lebih lambat dibandingkan dengan AM.

Gonsalves dan Kuwata (2015), untuk menyelesaikan masalah *Nurse Scheduling Problem* menggunakan AM. Penelitian tersebut menggunakan AM dengan memakai data yang berasal dari Queen's Medical Centre (QMC), Nottingham, UK. Dengan menggunakan 6 *hard constraint* dan 5 *soft constraint*, solusi yang didapatkan mencapai hasil yang optimal dengan dapat meminimalkan *soft constraint* dan *hard constraint* dengan baik.

Hakim (2016), untuk menyelesaikan masalah penentuan komposisi pakan ayam petelur menggunakan AM Tujuan dari penelitian ini adalah untuk menghasilkan sistem yang dapat dengan optimal menentukan komposisi asupan makanan yang tepat dikonsumsi untuk memenuhi kebutuhan nutrisi ayam petelur agar menghasilkan telur dengan kualitas yang baik. AM pada penelitian ini telah berhasil cukup efektif untuk menyelesaikan kasus penentuan komposisi asupan makanan ayam. Persamaan dengan penelitian yang dilakukan penulis yaitu AM digunakan sebagai pendekatan untuk melakukan optimasi.

Altintas *et al.* (2014), untuk menyelesaikan masalah penjadwalan ujian pada universitas Suleyman Demirel University (SDU) Turki, menggunakan AM. Pada penelitian ini AM dimodifikasi menjadi *Self-Generating Multimeme Algorithm* (SGMA) yang mengelola 6 mutasi, 2 crossover, dan 2 hill climbing. Pemilihan operator saat proses evolusi akan ditentukan menurut skor yang diperoleh melalui

operasi roulette wheel. Hasil SGMA pada penelitian ini cukup baik untuk menyelesaikan masalah penjadwalan ujian.

Berdasarkan latar belakang diatas maka dapat dirumuskan permasalahannya adalah bagaimana mengimplementasikan AM untuk membangun sistem penjadwalan tenaga kebidanan.

Tujuan penelitian ini adalah untuk membangun sistem penjadwalan tenaga kebidanan dengan menggunakan AM yang dapat memudahkan dan menghasilkan jadwal tenaga kebidanan yang optimal.

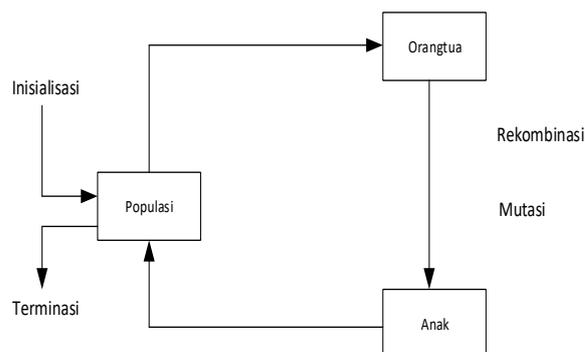
## 2. Kerangka Teori

### 2.1. Batasan pada Penjadwalan

Batasan pada penjadwalan merupakan suatu aturan-aturan yang jika tidak dipenuhi maka akan menurunkan kualitas suatu hasil penjadwalan. Adapun batasan yang terdapat pada penjadwalan biasanya terbagi menjadi dua jenis yaitu batasan keras dan batasan lunak. batasan keras merupakan batasan yang harus dipenuhi sedangkan batasan lunak merupakan batasan tambahan yang masih bisa ditolerir artinya jika batasan lunak terpenuhi maka akan menambah kualitas penjadwalan dan jika saja tidak terpenuhi maka jadwal masih bisa dianggap layak.

### 2.2. Algoritma Evolusi (AE)

Menurut Suyanto (2014), AE merupakan algoritma-algoritma optimasi yang berbasis "teori genetika" dari "teori evolusi" yang disampaikan oleh Charles Darwin. AE merepresentasikan suatu kandidat solusi sebagai satu individu dan sekumpulan kandidat solusi sebagai populasi. AE bekerja dengan cara membangkitkan, menguji, dan berusaha memperbaiki sekumpulan kandidat solusi sampai ditemukannya satu solusi yang bisa diterima. Rekombinasi dan mutasi, yang bersifat stokastik, meningkatkan keberagaman individu. Sebaliknya, seleksi menurunkan keberagaman individu dan berperan sebagai suatu kekuatan yang memaksa individu-individu untuk meningkatkan kualitasnya. Adapun skema AE dapat dilihat pada Gambar 1.

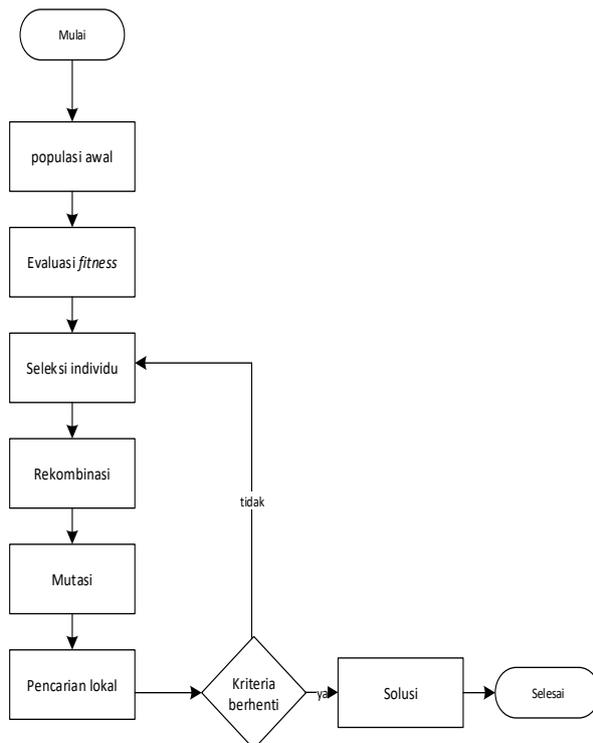


Gambar 1. Skema Umum AE  
(Sumber: Suyanto, 2014)

2.3. Algoritma Memetika (AM)

Menurut Moscato dalam Albar (2013), AM adalah suatu metode pencarian *heuristic* yang memiliki karakteristik yang sama dengan Algoritma Genetika dikombinasikan dengan metode PL yang secara bersama-sama dapat meningkatkan kualitas pencarian solusi. Metode PL bertujuan untuk melakukan perbaikan lokal yang dapat diterapkan sebelum atau sesudah proses seleksi, rekombinasi (*crossover*), dan mutasi. Metode PL juga dapat berguna untuk mengontrol besarnya ruang pencarian solusi.

AM meningkatkan efisiensi pencarian solusi dengan memperbaiki dan meningkatkan kemampuan individu didalam populasi menggunakan metode PL, bertujuan untuk mengeksplorasi ruang pencarian global dan demi mendapatkan solusi lebih baik dibandingkan AE konvensional. Keunggulan AM adalah kemampuan untuk mendapatkan solusi terbaik dengan lebih efektif dan efisien dikarenakan solusi yang buruk atau tidak layak dapat meningkat dan berubah menjadi solusi yang layak dan baik melalui proses peningkatan melalui metode PL. AE pada AM pada umumnya merujuk pada Algoritma Genetika sehingga karakteristik AM identik dengan karakteristik dalam Algoritma Genetika dengan penambahan metode PL. Skema AM dapat dilihat pada Gambar 2.



Gambar 2. Skema AM

1. Populasi awal

Langkah pertama dalam AM adalah membentuk populasi awal yang berisikan sekumpulan individu

(kromosom). Individu tersebut dilakukan penkodean menggunakan teknik *character encoding* dimana *value encoding* setiap jenis *shift* diwakili oleh karakter.

2. Evaluasi *fitness*

Setelah populasi awal telah terbentuk, maka selanjutnya adalah mengevaluasi setiap individu menggunakan fungsi *fitness* yang dimana nilai ini menyatakan baik tidaknya suatu individu. Individu yang melanggar batasan keras diberi nilai penalti 5, dan yang melanggar batasan lunak diberi nilai penalti 1. Menurut Suyanto (2005), jika solusi yang dicari adalah memaksimalkan sebuah fungsi *h* (masalah maksimasi) maka nilai *fitness* yang digunakan adalah nilai dari fungsi *h* tersebut, yaitu:

$$f = h \tag{1}$$

keterangan:

*f*: Nilai *fitness*

*h*: Maksimasi

Sedangkan jika yang dicari adalah meminimalkan sebuah fungsi (masalah minimasi) maka fungsi *h* tidak bisa langsung digunakan dan harus disesuaikan karna berlaku aturan bahwa individu yang lebih baik merupakan individu yang memiliki nilai *fitness* yang lebih baik. Oleh karena itu nilai *fitness* yang biasa digunakan adalah semakin kecil nilai *h* maka semakin besar nilai *f*, yaitu:

$$f = \frac{1}{(h+a)} \tag{2}$$

keterangan:

*f*: Nilai *fitness*

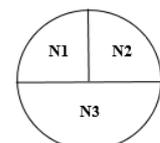
*h*: Minimasi

*a*: Bilangan sangat kecil bervariasi sesuai dengan masalah yang ada

3. Seleksi individu

Melakukan pemilihan pasangan individu sebagai orangtua. Metode yang digunakan adalah *roulette-wheel*, metode ini meniru permainan *roulette-wheel* dimana masing-masing individu seperti menempati potongan lingkaran pada roda roulette secara proposional sesuai dengan nilai *fitness*-nya. Ilustrasi pemilihan individu menggunakan metode *roulette-wheel* dapat dilihat pada Gambar 3.

Individu	Nilai <i>fitness</i>
1	0,5
2	0,5
3	2
Total <i>fitness</i>	3



Gambar 3. Pemilihan individu menggunakan metode *roulette-wheel*

4. Rekombinasi

Setelah orangtua terpilih maka selanjutnya proses rekombinasi dilakukan untuk menghasilkan individu anak (*offspring*). Dua individu anak akan tercipta dari

satu pasangan orangtua. Rekombinasi bisa berakibat nilai-nilai pada individu menyebar ke individu-individu lainnya lebih dini yang mengakibatkan konvergensi *premature*. Untuk itu operator rekombinasi digunakan yaitu probabilitas rekombinasi ( $P_r$ ). Rekombinasi hanya bisa dilakukan hanya jika bilangan yang acak yang dibangkitkan dalam rentan  $[0,1]$  lebih kecil dari probabilitas  $P_r$ . Seperti pada seleksi alam, kemungkinan individu untuk melakukan rekombinasi lumayan besar maka  $P_r$  biasanya diatur mendekati 1, misalnya 0.9.

#### 5. Mutasi

Setelah rekombinasi telah selesai selanjutnya proses mutasi dilakukan. Proses mutasi dilakukan hanya terhadap individu yang terpilih. Mutasi hanya bisa dilakukan hanya jika bilangan acak yang dibangkitkan dalam rentan  $[0,1]$  lebih kecil dari probabilitas  $P_m$ . Kemungkinan individu untuk mengalami mutasi sangat kecil, sehingga probabilitas mutasi biasanya diatur sangat kecil juga, misalnya 0.1.

#### 6. Pencarian Lokal (PL)

Setelah proses rekombinasi dan mutasi telah selesai proses PL dilakukan. PL biasa disebut individu *improvement* yaitu proses peningkatan nilai *fitness* individu yang bertujuan untuk memperbaiki individu buruk bahkan tidak layak menjadi lebih baik. Setelah proses ini juga individu terbaik akan dipilih kemudian disimpan untuk menghindari kerusakan individu terbaik pada generasi selanjutnya.

#### 2.5. Pencarian Lokal (PL)

Menurut Hakim *et al.* (2016), PL merupakan metode heuristik yang bermula dari sebuah solusi awal yang dibentuk secara acak. Kemudian dibentuk sebuah lingkungan (*neighbourhood*) dari solusi awal tersebut dengan tujuan untuk mencari solusi yang lebih baik di dalam lingkungan tersebut. Setelah solusi terbaik yang ada di dalam lingkungan tersebut didapat, maka langkah selanjutnya adalah membuat lingkungan yang baru dari solusi tersebut. Langkah ini kemudian terus diulangi sampai tidak ditemukan lagi solusi yang lebih baik dari suatu lingkungan.

Metode PL yang umum digunakan pada AM adalah algoritma *Hill Climbing* (HC). HC menerima solusi awal dan kemudian solusinya ditingkatkan secara berulang-ulang hingga mencapai pada suatu kondisi dimana solusi tersebut sudah tidak dapat ditingkatkan atau mencapai maksimal. Menurut Dangkuwa *et al.* (2017), proses algoritma *Simple Hill Climbing* adalah sebagai berikut:

1. Evaluasi state awal, jika state awal sama dengan tujuan, maka proses berhenti. Jika tidak sama dengan tujuan maka lanjutkan proses dengan membuat state awal sebagai state sekarang.
2. Mengerjakan langkah berikut sampai solusi ditemukan atau sampai tidak ada lagi operator baru yang dapat digunakan dalam state sekarang:

- a. Mencari sebuah operator yang belum pernah digunakan dalam state sekarang dan gunakan operator tersebut untuk membentuk state baru.
- b. Evaluasi state baru.
  - i. Jika state baru adalah tujuan, maka proses berhenti.
  - ii. Jika state baru tersebut bukan tujuan tetapi state baru lebih baik daripada state sekarang, maka buat state baru menjadi state sekarang.
  - iii. Jika state baru ini tidak lebih baik daripada state sekarang, maka kembali dan lanjutkan ke langkah ke 2.

### 3. Metode

#### 3.1. Pengumpulan Data

Penelitian tentang pembuatan sistem penjadwalan tenaga kebidanan ini dilakukan pada ruang Matahari Kebidanan di RSUD Undata Palu Provinsi Sulawesi Tengah.

Langkah-langkah dalam membangun sistem penjadwalan tenaga kebidanan dapat diuraikan seperti berikut:

Tahap pengumpulan data adalah melakukan pengambilan data secara langsung ke lapangan dengan melakukan wawancara secara langsung pada ruangan Matahari kebidanan RSUD Undata Palu serta meminta kelengkapan data berupa dokumen-dokumen. Adapun data yang diperoleh berupa data bidan yang bertugas pada ruangan tersebut, data jadwal bidan manual, dan data yang berkaitan dengan aturan-aturan penjadwalan. Adapun data bidan yang didapatkan berupa nama, nip, dan jabatan. Pada tahap ini berdasarkan data yang diperoleh batasan-batasan penjadwalan tenaga kebidanan telah ditentukan, adalah seperti berikut:

1. Batasan keras
  - a. Jumlah bidan yang ditugaskan per-hari pada setiap jenis shift harus memenuhi minimal jumlah dan komposisi jenis keterampilan.
  - b. Setiap bidan pada penjadwalan harus memenuhi minimal total jam kerja (120 jam) dan tidak boleh melebihi maksimal total jam kerja (180 jam).
  - c. Pemberian *shift* untuk setiap bidan harus memperhatikan dan mematuhi aturan pola *shift* penjadwalan.
2. Batasan lunak
  - a. Tidak boleh ada pola *shift* yang sama (pola shift siang/malam) dalam setiap minggu
  - b. Tidak boleh ada lebih dari 3 pola *shift* malam dalam 1 periode penjadwalan

#### 3.2. Perancangan Perangkat Lunak

Tahap perancangan perangkat lunak adalah melakukan perancangan kebutuhan perangkat lunak dan tahapan-tahapan proses operasi aplikasi mencakup proses AM.

3.3. Implementasi Perangkat Lunak

Tahap pembuatan perangkat lunak adalah melakukan implementasi pada tahap perancangan perangkat lunak dengan melakukan persiapan *software* yang digunakan seperti, membuat desain *interface*, *database*, dan penulisan program.

3.4. Pengujian Perangkat Lunak

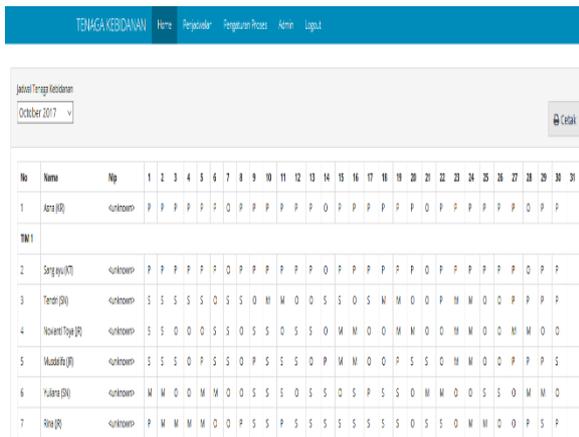
Pada tahap ini dilakukan pengujian perangkat lunak yang meliputi kemampuan sistem dalam menyelesaikan masalah penjadwalan dengan baik. Pengujian akan terbagi menjadi 3 jenis uji, pertama yaitu pengujian penjadwalan dengan periode penjadwalan tetap, kedua yaitu pengujian dengan periode penjadwalan yang berlanjut, terakhir yaitu pengujian terhadap metode PL.

4. Hasil dan Pembahasan

4.1. Sistem Penjadwalan Tenaga Kebidanan

Penelitian ini menghasilkan sistem penjadwalan tenaga kebidanan berbasis web aplikasi. Pada aplikasi ini terdapat halaman *user* dan halaman *admin*. Halaman *user* berfungsi untuk melihat hasil jadwal, sedangkan halaman *admin* berfungsi untuk memproses pembuatan jadwal dan melakukan operasi CRUD (*Create, Read, Update, Delete*) data bidan yang bertugas serta.

Pada halaman *user* terdapat pilihan *dropdown* untuk memilih jadwal pada periode tertentu untuk ditampilkan, selain itu terdapat tombol cetak untuk melakukan pencetakan jadwal. Halaman *user* dapat dilihat seperti pada Gambar 4.



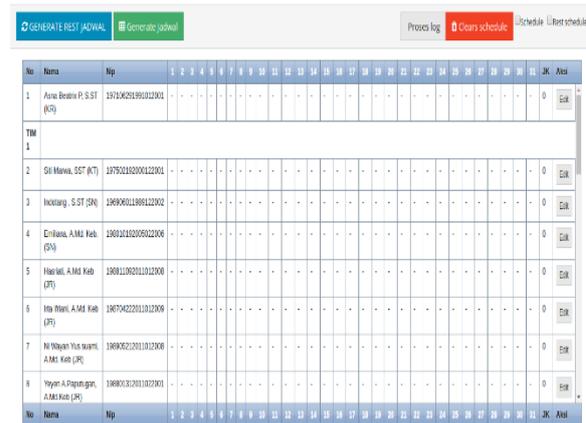
Gambar 4. Halaman user

Halaman *admin* terdiri dari beberapa halaman yaitu halaman pembuatan periode jadwal, halaman proses penjadwalan, dan halaman operasi CRUD data bidan. Halaman pembuatan periode jadwal dapat dilihat seperti pada Gambar 5.



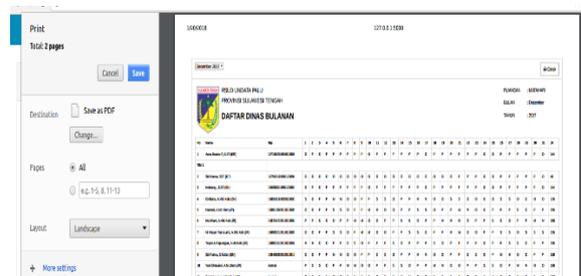
Gambar 5. Halaman pembuatan periode jadwal

Pada tahap selanjutnya, ketika *admin* sudah memilih periode penjadwalan kemudian melakukan proses penjadwalan dengan cara klik pada tombol proses penjadwalan, secara otomatis halaman akan berpindah pada halaman proses penjadwalan. Halaman proses penjadwalan dapat dilihat seperti pada Gambar 6.



Gambar 6. Halaman proses penjadwalan

Tahap selanjutnya pada halaman proses penjadwalan pertama yang harus dilakukan adalah mendapatkan sisa potongan jadwal dengan cara klik tombol generate rest jadwal selanjutnya, klik tombol generate jadwal untuk memulai proses pembuatan jadwal. hasil proses penjadwalan dapat dilihat pada Gambar 7.



Gambar 7. Hasil proses penjadwalan

Halaman operasi CRUD data bidan berfungsi untuk manajemen data bidan yang akan dijadwalkan. Halaman operasi CRUD data bidan dapat dilihat seperti pada Gambar 8.

Gambar 8. Halaman operasi CRUD data bidan

4.2. Implementasi Algoritma Memetika

Pembangkitan populasi awal dilakukan dengan membuat individu yang terdiri dari baris dan kolom. Setiap baris (Bi) pada individu mewakili bidan yang bertugas dan setiap kolom (Hi) mewakili hari tugas shift. Shift yang diberi huruf tebal merupakan sisa potongan pola *shift* generasi sebelumnya. Contoh populasi awal yang terdiri dari empat individu dapat dilihat pada Gambar 9.

INDIVIDU	B/H	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10
Individu 1	B1	M	M	O	O	S	S	O	P	S	M
	B2	<b>O</b>	<b>O</b>	M	S	S	O	S	P	M	P
	B3	<b>O</b>	P	M	M	O	D	P	S	M	P
	B4	<b>M</b>	<b>O</b>	<b>O</b>	P	S	S	P	P	P	M
	B5	S	S	S	O	P	M	M	P	P	P
Individu 2	B1	M	M	P	P	M	M	S	S	S	S
	B2	<b>O</b>	<b>O</b>	M	S	P	S	P	P	S	S
	B3	<b>O</b>	M	M	S	S	P	P	P	S	M
	B4	<b>M</b>	<b>O</b>	<b>O</b>	M	M	M	P	P	P	M
	B5	S	S	P	M	M	P	P	S	S	M
Individu 3	B1	S	P	P	M	M	S	S	P	P	M
	B2	<b>O</b>	<b>O</b>	P	P	S	P	M	P	S	M
	B3	<b>O</b>	M	M	M	M	P	P	P	P	P
	B4	<b>M</b>	<b>O</b>	<b>O</b>	M	M	M	P	P	P	M
	B5	P	M	M	M	M	M	S	S	S	S

Gambar 9. Contoh populasi awal

Tahap selanjutnya, nilai *fitness* dari setiap individu dalam populasi dihitung menggunakan persamaan (2) sebagai berikut:

$$F = \frac{1}{(\sum PMB + \sum PPOLS + \sum PWH + \sum PMM + \sum PRS) + 1}$$

Keterangan :

*PMB* : Pelanggaran minimal komposisi dan keterampilan bidan

*PPOLS* : Pelanggaran pola *shift*

*PWH* : Pelanggaran minimal atau maksimal total jam kerja

*PMM* : Pelanggaran maksimal pola *shift* malam

*PRS* : Pelanggaran pola *shift* yang sama dalam rentan 7 hari kerja

Untuk selanjutnya, seleksi orang tua dilakukan menggunakan metode *roulette-wheel*. Individu yang terpilih pada seleksi orang tua kemudian akan melalui proses rekombinasi menggunakan satu titik potong. Ilustrasi rekombinasi individu 2 dan individu 3 dapat dilihat pada Gambar 10.

Individu	N/H	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10
Individu 2	B1	M	M	P	P	M	M	M	S	S	S
	B2	<b>O</b>	<b>O</b>	M	S	P	S	P	P	S	S
	B3	<b>O</b>	M	M	S	S	P	P	P	S	M
	B4	<b>M</b>	<b>O</b>	<b>O</b>	M	M	M	P	P	P	M
	B5	S	S	P	M	M	P	P	S	S	M
Individu 3	B1	S	P	P	M	M	S	S	P	P	M
	B2	<b>O</b>	<b>O</b>	P	P	S	P	M	P	S	M
	B3	<b>O</b>	M	M	M	M	P	P	P	P	P
	B4	<b>M</b>	<b>O</b>	<b>O</b>	P	P	P	P	P	P	P
	B5	P	M	M	M	M	M	M	S	S	S

Individu	N/H	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10
anak 1	B1	M	M	P	P	M	M	S	P	P	M
	B2	<b>O</b>	<b>O</b>	M	S	P	S	M	P	S	M
	B3	<b>O</b>	M	M	S	S	P	P	P	P	P
	B4	<b>M</b>	<b>O</b>	<b>O</b>	M	M	M	P	P	P	P
	B5	S	S	P	M	M	P	M	S	S	S
anak 2	B1	S	P	P	M	M	S	M	S	S	S
	B2	<b>O</b>	<b>O</b>	P	P	S	P	P	P	S	S
	B3	<b>O</b>	M	M	M	M	P	P	P	S	M
	B4	<b>M</b>	<b>O</b>	<b>O</b>	P	P	P	P	P	P	M
	B5	P	M	M	M	M	M	P	S	S	M

Gambar 10. Contoh populasi awal

Tahap selanjutnya, operasi mutasi dilakukan pada individu yang terpilih dengan cara memilih 2 baris secara acak kemudian membangkitkan susunan *shift* yang baru untuk 2 baris tersebut. Ilustrasi mutasi dapat dilihat pada Gambar 11.

Individu	N/H	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10
Anak 1	B1	M	M	P	P	M	M	S	P	P	M
	B2	<b>O</b>	<b>O</b>	S	S	S	S	S	S	M	P
	B3	<b>O</b>	M	M	S	S	P	P	P	P	P
	B4	<b>M</b>	<b>O</b>	<b>O</b>	M	M	M	P	P	P	P
	B5	M	M	P	M	M	P	M	M	M	M

Gambar 11. Ilustrasi mutasi

Tahap selanjutnya, operasi pencarian lokal dilakukan menggunakan 4 *move* untuk melakukan upaya perbaikan pelanggaran, dapat dilihat pada Tabel 1.

Tabel 1. Operator PL

No.	Pencarian Lokal	Perbaikan pelanggaran
1.	Move pairshift overflow	PMM, PRS
2.	Move working hours	PWH
3.	Move min bidan	PMB
4.	Move pola shift	PPOLS

Terakhir, proses iterasi pada AM akan berhenti jika solusi yang dicari telah ditemukan, terjadi konvergensi *premature*, atau maksimal iterasi terpenuhi.

4.3. Pengujian Sistem

Pengujian dilakukan dengan melakukan tiga jenis pengujian, adalah sebagai berikut:

a. Pengujian penjadwalan dengan periode penjadwalan tetap

Pengujian akan mencoba menjadwalkan jadwal tenaga kebidanan periode Januari 2018 dilakukan sebanyak 20 kali percobaan. Hasil pengujian dapat dilihat pada Tabel 2.

Tabel 2. Tabel pengujian penjadwalan dengan periode penjadwalan tetap

No.	Generasi	Fitness Terbaik	Waktu Eksekusi	Pelanggaran
1.	9	1	19 Detik	0
2.	8	1	16 Detik	0
3.	18	1	36 Detik	0
4.	14	1	28 Detik	0
5.	10	1	20 Detik	0
6.	14	1	29 Detik	0
7.	18	1	20 Detik	0
8.	15	1	30 Detik	0
9.	14	1	29 Detik	0
10.	12	1	24 Detik	0
11.	8	1	16 Detik	0
12.	8	1	16 Detik	0
13.	11	1	22 Detik	0
14.	14	1	27 Detik	0
15.	9	1	18 Detik	0
16.	7	1	14 Detik	0
17.	13	1	27 Detik	0
18.	18	1	36 Detik	0
19.	7	1	14 Detik	0
20.	13	1	26 Detik	0
Rata -rata	12	1	23 Detik	0

b. Pengujian penjadwalan dengan periode penjadwalan berlanjut

Pengujian ini akan menjadwalkan jadwal tenaga kebidanan dari periode Januari 2018 hingga Maret 2018. Dilakukan 5 percobaan penjadwalan untuk masing-masing periode jadwal. Hasil pengujian dengan periode penjadwalan berlanjut dapat dilihat pada Tabel 3.

Tabel 3. Tabel pengujian penjadwalan dengan periode penjadwalan berlanjut

No.	Acuan Jadwal	Periode jadwal	Gene-rasi	Waktu Eksekusi	Pelanggaran
1.	Des. 2017	Jan. 2018	7	14 Detik	0
2.	Des. 2017	Jan. 2018	10	20 Detik	0
3.	Des. 2017	Jan. 2018	6	12 Detik	0
4.	Des. 2017	Jan. 2018	16	32 Detik	0
5.	Des. 2017	Jan. 2018	10	20 Detik	0
6.	Jan. 2018	Feb. 2018	9	17 Detik	0
7.	Jan. 2018	Feb. 2018	10	18 Detik	0
8.	Jan. 2018	Feb. 2018	9	19 Detik	0
9.	Jan. 2018	Feb. 2018	8	15 Detik	0
10.	Jan. 2018	Feb. 2018	10	19 Detik	0
11.	Feb. 2018	Mar. 2018	9	19 Detik	0
12.	Feb. 2018	Mar. 2018	9	19 Detik	0
13.	Feb. 2018	Mar. 2018	8	17 Detik	0
14.	Feb. 2018	Mar. 2018	7	15 Detik	0
15.	Feb. 2018	Mar. 2018	10	19 Detik	0
Rata – rata			9	18 Detik	0

c. Pengujian operator Pencarian Lokal (PL)

Pengujian akan mencoba menjadwalkan jadwal tenaga kebidanan periode Januari 2018. Pengujian ini akan mencoba kombinasi serangkain operator PL. Operator PL yang ada dapat dilihat pada tabel 1 dan Hasil pengujian PL dapat dilihat pada Tabel 4.

Tabel 4. Tabel pengujian operator PL

No	Operator Pencarian Lokal	Gene-rasi	Fitness Terbaik	Waktu Eksekusi	Pelanggaran
1.	Lengkap	8	1	16 Detik	0
2.	Lengkap	49	1	8 Detik	0
3.	Lengkap	15	1	30 Detik	0
4.	Lengkap	11	1	24 Detik	0
Rata-rata		21	1	19 Detik	0
5.	1, 2, 4	2000	0.062	1 Jam	3
6.	1, 2, 4	2000	0.032	1 Jam	6
7.	1, 2, 4	2000	0.062	1 Jam	3
8.	1, 2, 4	2000	0.047	1 Jam	4
Rata-rata		2000	0,050	1 Jam	4
13.	-	2000	0.007	37 Menit	32
14.	-	2000	0.007	37 Menit	29
15.	-	2000	0.006	37 Menit	34
16.	-	2000	0.006	36 Menit	34
Rata-rata		2000	0.007	37 Menit	32

Berdasarkan hasil pengujian dari tiga jenis pengujian yang telah dilakukan, AM dapat secara efektif dan efisien dalam menghasilkan jadwal yang optimal dengan waktu eksekusi kurang dari satu menit tanpa ada batasan yang terlanggar. Dapat dikatakan menghasilkan jadwal yang optimal dikarenakan tidak adanya batasan keras yang terlanggar dan mampu meminimalkan sebaik mungkin terjadinya pelanggaran batasan lunak.

Pelanggaran terjadi hanya jika penggunaan operator PL pada AM diubah menjadi tidak sebagaimana mestinya seperti yang dapat dilihat pada jenis uji pengujian operator PL No 5 sampai 16 pada tabel 4, ketika beberapa operator PL dihilangkan maka pada akhir generasi (2000 generasi) terjadi beberapa pelanggaran. Dari hasil pengujian tersebut dapat diketahui bahwa penggunaan operator PL yang tepat akan sangat berpengaruh pada total generasi dan waktu eksekusi yang dibutuhkan untuk mendapatkan hasil jadwal yang optimal.

## 5. Kesimpulan

Kesimpulan dari hasil penelitian ini adalah telah terbentuknya suatu sistem penjadwalan tenaga kebidanan menggunakan Algoritma Memetika. Hasil dari sistem ini telah melewati proses pengujian menggunakan pengujian, validasi, dan verifikasi sistem, dengan hasil menunjukkan bahwa sistem tenaga kebidanan menggunakan Algoritma Memetika mampu menghasilkan jadwal tenaga kebidanan yang optimal.

Dari hasil ini dapat ditunjukkan kemampuan menyelesaikan masalah penjadwalan tanpa adanya batasan keras dan batasan lunak yang terlanggar dengan waktu eksekusi kurang dari satu menit. Sistem ini dapat menghemat banyak waktu dan tenaga dikarenakan selain dapat menghasilkan jadwal yang optimal, memudahkan proses pembuatan jadwal, dan juga dapat menyelesaikannya dengan waktu yang relatif cepat.

## Daftar Pustaka

- Altintas, C., Asta, S., Ozcan, E., Yigit, T., 2014. A self-generating memetic algorithm for examination timetabling, 10th International Conference of the Practice and Theory of Automated Timetabling PATAT 2014, UK, 26-29 Agustus 2014.
- Albar, A.M., 2013. Algoritma genetik tabu search dan memetika pada permasalahan penjadwalan kuliah. Seminar Nasional Teknologi Informasi dan Multimedia 2013 STMIK AMIKOM, Yogyakarta, 19 Januari 2013.
- Dangkua, E.V., Gunawan, V., dan Adi, K., 2015. Penerapan metode hill climbing pada sistem informasi geografis untuk mencari lintasan terpendek. *Jurnal Sistem Informasi Bisnis* 5 (1).
- Gonsalves, T., Kuwata, K., 2015. Memetic algorithm for the nurse scheduling problem. *International Journal of Artificial Intelligence & Applications (IJAAI)* 6 (4).
- Hakim, L., Setya B., Qodriyah, N., 2016. Penerapan AM pada penentuan komposisi pakan ayam petelur. Skripsi, Universitas Muhammadiyah, Jember.
- IBI, 2016. Definisi Bidan. Website: [www.ibi.or.id/id/article\\_view/a20150112004/definisi.html](http://www.ibi.or.id/id/article_view/a20150112004/definisi.html), diakses tanggal 18 April 2017.
- Suyanto, 2014. *Algoritma Optimasi*, Graha Ilmu, Yogyakarta.
- Suyanto. 2005. *Algoritma Genetika dalam Matlab, Andi Offset*, Yogyakarta.