PROSIDING Seminar Nasional SISFOTEK ISSN 2597-3584 (Media Online)
Sistem Informasi dan Teknologi
website : www.seminar.iaii.or.id | email : seminar@iaii.or.id

Optimasi Penjadwalan Perkuliahan Pada Universitas Handayani Makassar Menggunakan Algoritma Particle Swarm Optimization

Dwiky Darmawansyah[1] , Najirah Umar[2], Nurfaedah[3]
[1,2]Teknik Informatika, Ilmu Komputer, Universitas Handayani Makassar
[3]Manajemen Informatika, Ilmu Komputer, Universitas Handayani Makassar
najirah@handayani.ac.id

Abstract

Lesson planning is a regular task that every educational institution does right from the beginning of the semester. However, creating an optimal study schedule is quite difficult because there are many interrelated variables that require careful management. Current class scheduling at Handayani University Makassar is still done manually, including searching for empty columns and placing class schedules in those columns. The director of BAAK UHM, who was responsible for preparing the conference program, felt overwhelmed. Because the planning done so takes a lot of time. Therefore, the established course program must be regularly revised. One method considered to provide a solution to scheduling problems is swarm optimization (PSO). PSO if translated means Particle Swarm Optimization. This algorithm can solve the problem by randomly forming particles in the initial population, evaluating appropriate values, and updating particle velocities and positions. This is intended to address the issue of proper function of each particle. The data includes 225 course data plus 1 speaker's request schedule data. This study was successful in achieving a course schedule consistent with the course conditions and policies at UHM. From the test results, the lesson planning application using the PSO algorithm can provide an optimal class schedule that is consistent with the teacher's teaching schedule preferences. The calculation time used is less than 5 seconds.

Keywords: Optimization, Lecture Scheduling, Constraint, Fitness, Velocity, Particle Swarm Optimization

Abstrak

Penjadwalan mata kuliah merupakan pekerjaan rutin yang selalu dilakukan setiap institusi pendidikan pada awal semester. Namun menciptakan suatu jadwal kuliah yang optimal cukup sulit dikarenakan banyak variabel yang saling terkait sehingga membutuhkan penanganan yang teliti. Penjadwalan kuliah di Universitas Handayani Makassar yang ada saat ini masih dibuat secara manual, yaitu dengan pencarian kolom-kolom mana saja yang masih kosong, kemudian menempatkan jadwal pada kolom tersebut. Kepala BAAK UHM selaku pihak yang bertanggung jawab menyusun jadwal perkuliahan merasa kewalahan. Karena jadwal yang dihasilkan dengan cara seperti ini, memerlukan waktu yang lama. Sehingga jadwal kuliah yang sudah dibuat seringkali perlu dilakukan perbaikan lagi. Salah satu metode yang dianggap dapat memberikan solusi atas permasalahan penjadwalan ialah Particle Swarm Optimization (PSO). PSO jika diterjemahkan berarti optimasi segerombolan partikel. Algoritma ini dapat memecahkan masalah dengan membentuk partikel-partikel pada populasi awal secara acak, mengevaluasi nilai fitness, dan meng-update velocity serta posisi dari partikel. Ini bertujuan memecahkan permasalahan yang ditinjau dari fungsi fitness setiap partikel. Data tersebut terdiri dari 225 data perkuliahan ditambah 1 data jadwal permintaan dosen. Penelitian ini berhasil mewujudkan penjadwalan perkuliahan yang sesuai dengan kondisi dan kebijakan perkuliahan di UHM. Dari hasil pengujian, aplikasi penjadwalan perkuliahan menggunakan algoritma PSO mampu menghasilkan jadwal perkuliahan yang optimal serta mampu mengakomodasi preferensi jadwal mengajar dosen. Jumlah waktu komputasi yang digunakan ialah kurang dari 5 detik.

Kata kunci: Optimasi, Penjadwalan Perkuliahan, Constraint, Fitness, Velocity, Particle Swarm Optimization

1. Pendahuluan

Penjadwalan mata kuliah merupakan pekerjaan rutin yang selalu dilakukan setiap institusi pendidikan pada awal semester. Namun menciptakan suatu jadwal kuliah yang optimal cukup sulit dikarenakan banyak variabel yang saling terkait sehingga membutuhkan suatu penanganan yang signifikan[1],[2],[3]. Penjadwalan kuliah merupakan pengaturan penempatan dosen, mata kuliah, dan kelas (kelompok mahasiswa), terhadap hari, jam/waktu dan ruangan yang tersedia.

Sistem Penjadwalan dikatakan baik apabila dapat memberikan solusi terhadap faktor-faktor tertentu, seperti ketersediaan ruangan yang sedikit, dosen tidak boleh mengajar banyak kelas dalam satu waktu, kelas tidak boleh mengikuti banyak mata kuliah dalam satu waktu[4],[5].

Dalam penjadwalan kuliah di Universitas Handayani Makassar (UHM) yang ada saat ini masih dibuat secara manual, yaitu dengan pencarian blok-blok atau kolom-kolom mana saja yang masih kosong, kemudian menempatkan jadwal pada blok atau kolom tersebut. Jadwal yang dihasilkan dengan cara seperti ini memerlukan waktu yang lama dan cukup membuat Kepala BAAK UHM (yang bertugas menyusun jadwal kuliah) kewalahan. Hal tersebut terjadi karena cara manual yang dilakukan saat ini, memungkinkan terjadinya kesalahan dengan intensitas yang cukup banyak. Sehingga jadwal kuliah yang sudah dibuat seringkali perlu dilakukan perbaikan lagi. Alasannya lagi-lagi karena menyusun jadwal kuliah yang optimal bukan pekerjaan yang mudah. Dibutuhkan ketelitian yang tinggi karena harus memperhatikan banyak variabel agar dapat menemukan kombinasi penempatan dosen, mata kuliah, dan kelas (kelompok mahasiswa), terhadap hari, jam/waktu dan ruangan yang tersedia[6].

Berdasarkan uraian di atas perlu dibuat sebuah aplikasi yang menggunakan algoritma yang tepat untuk membantu penyusunan jadwal perkuliahan yang optimal di Universitas Handayani Makassar., dengan menggunakan algoritma *Particle Swarm Optimization* (PSO). Karena optimasi penjadwalan perkuliahan menggunakan algoritma (PSO) adalah salah satu metode yang dikembangkan untuk memecahkan masalah optimasi kompleks dalam penjadwalan perkuliahan[7],[8],[5].

2. Metode Penelitian

Metode dalam PSO yang akan digunakan untuk penyelesaian persoalan penjadwalan mata kuliah pada UHM adalah :

2.1. Inisialisasi Posisi dan *Velocity* Awal Partikel

Dalam penjadwalan mata kuliah ini posisi dan kecepatan (*velocity*) partikel dimisalkan sebagai slot. Posisi dan kecepatan awal ditentukan secara acak dengan batasan minimum slot dan maksimum slot. Nilai posisi dan kecepatan awal partikel diinisialisasi sama, sehingga hanya melakukan satu kali random untuk mendapatkan nilai kecepatan dan posisi partikel[9],[10].

Pada penjadwalan ini posisi partikel diwakili oleh slot-slot, satu slot jadwal merupakan satu partikel. Setiap ruangan memiliki 42 slot dalam satu minggu perkuliahan, di mana setiap harinya terdapat 6 slot yang dapat dipergunakan dan setiap slot mewakili 2 jam perkuliahan[11],[12].

Pada tahap ini dilakukan proses pembangkitan posisi dan juga kecepatan partikel. Pembangkitan posisi dan kecepatan ini dilakukan secara acak dengan menggunakan persamaan (1) dan persamaan (2).

$$x_i^0 = x_{min} + rand(x_{max} - x_{min}) \quad (1)$$

$$v_i^0 = x_{min} + rand(x_{max} - x_{min}) \quad (2)$$

Dengan x_i^0(posisi awal), v_i^0 adalah *velocity*/kecepatan, x_{min} (batas bawah), x_{max} (batas atas) dan $rand$ adalah nilai acak antara 0 dan 1. Terhadap *records* pada tabel ruangan, tabel hari dan tabel jam sebanyak jumlah *records* yang terdapat pada tabel pengampu. Nilai x_{min} (batas bawah) dan x_{max} (batas atas) untuk setiap tabel adalah sebagai berikut:

Tabel 1 Batas Bawah dan Batas Atas Hari, Jam, dan Ruangan

Batas Bawah / Batas Atas	Tabel		
	Hari	Jam	Ruangan
xmin	1	1	1
xmax	7	6	11

Sedangkan tabel pengampu dan pesanan sendiri sudah mengandung *fields* mata kuliah, dosen, kelas, SKS, semester, jenis matkul, kategori kelas dan fakultas[13],[14].

Contoh perhitungan manual pada iterasi pertama:

Menentukan posisi untuk matkul A dosen 1, dimulai dengan menentukan posisi hari, jam kemudian ruangan.

Menentukan posisi hari
Berdasarkan persamaan (1) dan persamaan (2) didapatkan hasil:
x_1^1 = 1 + $rand$ (7-1)
= 1 + 0.02 (7-1) = 1.1

Menentukan posisi jam
Berdasarkan persamaan (1) dan persamaan (2) didapatkan hasil:
x_1^1 = 1 + $rand$ (6-1)
= 1 + 0.88 (6-1) = 5.4

Menentukan posisi ruangan
Berdasarkan persamaan (1) dan persamaan (2) didapatkan hasil:
x_1^1 = 1 + $rand$ (11-1)
= 1 + 0.22 (11-1) = 3.2

Jadi posisi untuk mata kuliah A dosen 1 adalah x(1.1 , 5.4 , 3.2). *Velocity* pada iterasi pertama diinisialisasi sama dengan posisi v(1.1 , 5.4 , 3.2). Perhitungan di atas diulangi sampai sejumlah partikel (semua *records* pada tabel pengampu dan pesanan), kemudian dihitung nilai *fitness* masing-masing partikel. Berikut adalah hasil pembangkitan posisi partikel pada iterasi pertama:

Tabel 2 Pembangkitan Posisi Partikel pada Iterasi Pertama

Pengampu				Hari	Jam	Ruangan
Id	Makul	Dosen	Kelas			
1	1	1	A	1.1	5.4	3.2
2	1	2	B	2.6	4.6	4.0
3	1	3	C	2.1	4.6	4.5

4	1	4	D	2.8	6.4	4.2

Berikut adalah ilustrasi pembangkitan partikel dalam bentuk jadwal:

Tabel 3. Ilustrasi Pembangkitan Partikel dalam Bentuk Jadwal

Hari	Jam	Ruangan										
		1	2	3	4	5	6	7	8	9	10	11
1	1											
	2											
	3											
	4											
	5			MK A Dsn 1								
	6											
2	1											
	2											
	3											
	4				MK B Dsn 2 MK C Dsn 3							
	5											
	6				MK D Dsn 4							

*MK = Mata Kuliah & Dsn = Dosen

2.2. Evaluasi *Fitness*

Dalam penjadwalan ini nilai *fitness* ini menentukan banyaknya pelanggaran *constraint* yang harus dioptimasi. *Constraint-constraint* yang digunakan untuk pengoptimalan aplikasi penjadwalan ini antara lain:

Tidak boleh ada bentrok dengan jadwal lain.
Mata kuliah kelas "reguler" hanya dijadwalkan mulai hari senin sampai jum'at dan mulai jam 08:00 hingga 17:00 WITA.
Mata kuliah kelas "ekstensi" dijadwalkan pada hari sabtu dan minggu, mulai jam 08:00 hingga 21:00 WITA dan tidak boleh ditempatkan di ruangan FHIS.
Mata kuliah kelas "malam" dijadwalkan mulai hari senin sampai jum'at dan mulai jam 17:00 hingga 21:00 WITA.
Mata kuliah kelas fakultas "hukum" dijadwalkan pada hari sabtu dan minggu, mulai jam 08:00 hingga 19:00 WITA dan diprioritaskan ditempatkan di ruangan FHIS.
Mata kuliah "praktikum" harus ditempatkan di ruangan Lab.
Jadwal yang dipesan, harus ditempatkan pada blok/slot jadwal sesuai hari, waktu/jam, dan ruangan yang dipilih.
Jadwal mengajar dosen tidak boleh dijadwalkan pada jadwal tidak bersedia dosen.

Jika pada masing-masing partikel terjadi pelanggaran terhadap *constraint-constraint* diatas, maka nilai *fitness*-nya adalah dijumlahkan setiap pelanggaran yang terjadi dalam satu iterasi kemudian dihitung dengan fungsi fitness minimasi menggunakan persamaan (3).

$$f = h \tag{3}$$

Semakin kecil jumlah pelanggaran yang terjadi, maka nilai fitness yang dihasilkan juga semakin kecil. Sebaliknya semakin besar jumlah pelanggaran yang terjadi, maka nilai fitness yang dihasilkan semakin besar. Sehingga partikel terbaik adalah partikel dengan nilai fitness terkecil atau dengan kata lain partikel terbaik adalah partikel dengan jumlah pelanggaran paling sedikit. Misalkan, jumlah pelanggaran yang terjadi pada partikel setelah pembangkitan posisi dan kecepatan partikel adalah 10. Maka nilai fitness-nya dapat dihitung dengan persamaan (3):

$$f = 10$$

2.3. Menentukan *Local/Personal Best* dan *Global Best*

Local/Personal Best (p_i^t)
Local Best (p_i^t) pada iterasi pertama diset sama dengan posisi awal setelah pembangkitan. Misalnya pada mata kuliah A, dosen 1, dengan posisi harinya adalah 1.1, maka nilai *local best* (p_i^t) diset 1.1. Demikian seterusnya hingga setiap partikel diset sama dengan posisi awal partikel. Sedangkan untuk iterasi selanjutnya, maka akan dibandingkan antara posisi sekarang (x_i^t) dengan *local best* (p_i^t). Jika posisi sekarang (x_i^t) lebih kecil *fitness*-nya atau sama dengan *local best* (p_i^t), maka *local best* (p_i^t) akan di-*update* dengan nilai posisi sekarang (x_i^t).

Global Best (g^t)
Setelah didapat nilai *fitness* terbaik pada iterasi pertama, maka langkah selanjutnya adalah menentukan partikel terbaik dalam satu iterasi, yaitu partikel dengan nilai *fitness* paling kecil dari partikel-partikel lain dalam satu iterasi. Setelah didapat nilai *fitness* pada iterasi pertama adalah 10, maka nilai *global best* (g^t) diset 10 untuk setiap partikel. Partikel terbaik tersebut kemudian disimpan sebagai *global best particle*. Sedangkan untuk iterasi selanjutnya, maka akan dibandingkan antara posisi sekarang (x_i^t) dengan *global best* (g^t). Jika posisi sekarang (x_i^t) *fitness*-nya

lebih kecil atau sama dengan *global best* (g^t), maka *global best* (g^t) akan di-*update* dengan nilai posisi sekarang (x_i^t).

Proses *Update Velocity* dan Posisi

Untuk mencari update velocity maka kita menggunakan persamaan (4):

$$v_i^{t+1} = \omega \cdot v_i^t + rand \cdot c1(p_i^t - x_i^t) + rand \cdot c2(g^t - x_i^t) \quad (4)$$

Keterangan:

v_i^{t+1} = *velocity*/kecepatan baru

v_i^t = *velocity*/kecepatan lama

ω = *inertia factor*, Nilai memiliki rentang 0.4 – 0.9

$rand$ = nilai acak antara 0 dan 1

c1, c2 = *self confidence* dan *swarm confidence*, Nilai dari c1 dan c2 biasanya adalah 2, 1.5 atau 1.

p_i^t = *personal best*

g^t = *global best*

x_i^t = posisi lama

Langkah terakhir dari setiap iterasi adalah update posisi setiap partikel dengan vektor velocity, seperti yang ditunjukkan pada persamaan (5):

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (5)$$

Keterangan:

x_i^{t+1} = posisi baru

x_i^t = posisi lama

v_i^{t+1} = *velocity*/kecepatan baru

Dengan parameter ujicoba sebagai berikut:

c1, c2 = 1.5 & ω = 0.5

Parameter uji coba ini diambil dari hasil penelitian yang dilakukan oleh Dian (2011) yang mana dengan parameter ini dapat menghasilkan rata-rata jadwal yang lebih optimal.

Menentukan posisi hari

Misal hasil rand 1 = 0.3 dan rand 2 = 0.2

$v_1^2 = 0.5 \cdot 1.1 + 0.3 \cdot 1.5(1.1 - 1.1) + 0.2 \cdot 1.5(1.1 - 1.1) = 0.55$

$x_1^2 = 1.1 + 0.55 = 1.65$

Menentukan posisi jam

Misal hasil rand 1 = 0.3 dan rand 2 = 0.2

$v_1^2 = 0.5 \cdot 5.4 + 0.3 \cdot 1.5(5.4 - 5.4) + 0.2 \cdot 1.5(5.4 - 5.4) = 2.7$

$x_1^2 = 5.4 + 2.7 = 8.1$

Menentukan posisi ruangan

Misal hasil rand 1 = 0.3 dan rand 2 = 0.2

$v_1^2 = 0.5 \cdot 3.2 + 0.3 \cdot 1.5(3.2 - 3.2) + 0.2 \cdot 1.5(3.2 - 3.2) = 1.6$

$x_1^2 = 3.2 + 1.6 = 4.8$

Berikut ini ditampilkan hasil perhitungan manual *update velocity* dan posisi pada iterasi pertama:

Tabel 4 Hasil Perhitungan *Update Velocity* dan Posisi pada Iterasi Pertama

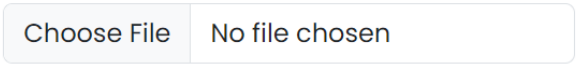
Pengampu			1	2	Hari	Jam	Ruangan
Makul	Dosen	Kelas					
1	1	A	0.3	0.2	1.6	8.1	4.8
1	2	B	0.3	0.4	3.9	6.9	6.0
1	3	C	0.1	0.2	3.2	4.9	7.8
1	4	D	0.1	0.3	4.2	9.6	7.8

3. Hasil dan Pembahasan

Hasil dari penelitian ini adalah aplikasi yang berfungsi untuk menyusun jadwal perkuliahan yang optimal sesuai dengan kondisi perkuliahan pada Universitas Handayani Makassar. Aplikasi ini dapat membantu proses pembuatan jadwal kuliah menjadi lebih efektif dan efisien.

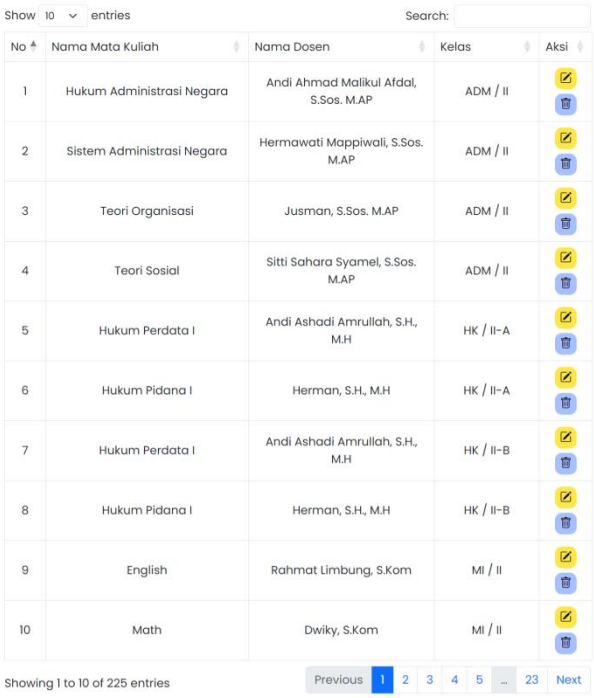
Gambar 1 Tampilan Sistem

Saat pertama kali menjalankan sistem penjadwalan ini, maka kita akan melihat tampilan sistem seperti pada gambar 1 diatas. Komponen pertama berupa *banner* besar yang memuat logo UHM dan *Header* sistem serta memuat semua komponen fungsional pada sistem ini.



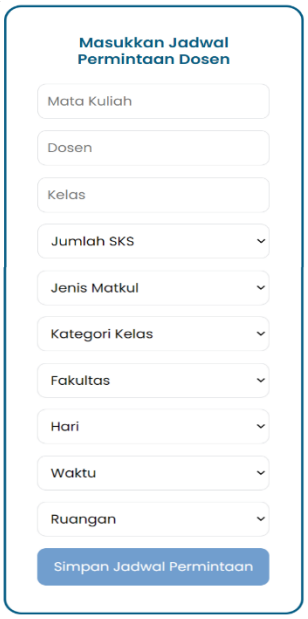
Gambar 2 Form *Input* Data Perkuliahan

Tugas pertama saat menjalankan sistem ini adalah menginput daftar perkuliahan. Proses penginputan bisa dilakukan dengan mengimpor file data perkuliahan dalam bentuk excel seperti pada gambar 2. File tersebut memuat informasi terkait data perkuliahan, meliputi nama mata kuliah, nama dosen, nama kelas, jumlah SKS mata kuliah, semester, jenis mata kuliah, kategori kelas dan fakultas.



Gambar 3 Tabel Data Perkuliahan

Pada bagian 3, sistem akan menampilkan data yang berhasil diinput pada form data perkuliahan dalam bentuk data table. Kita juga bisa menyesuaikan jumlah daftar yang ditampilkan pada data table ini dan kita juga bisa melakukan pencarian data perkuliahan tertentu melalui kotak pencarian. Pada tabel ini memungkinkan juga kita untuk melakukan fungsi *delete* untuk menghapus data tertentu pada tabel ini dan juga fungsi *edit* untuk memodifikasi atau mengubah detail informasi data tertentu pada tabel ini.



Gambar 4 Form *Input* Data Jadwal Permintaan Dosen

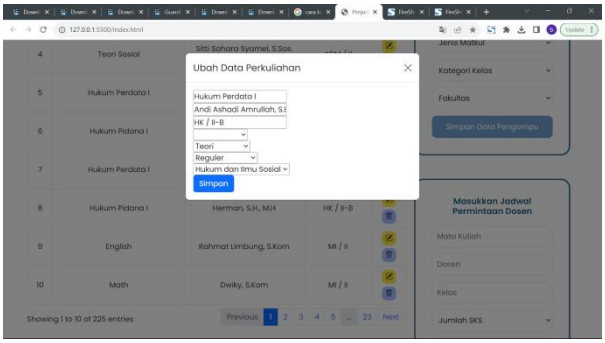
Gambar 4, menampilkan tampilan form jadwal permintaan dosen. Form ini berfungsi untuk mengakomodasi preferensi jadwal mengajar dosen.

Gambar 5 Tabel Data Jadwal Permintaan Dosen

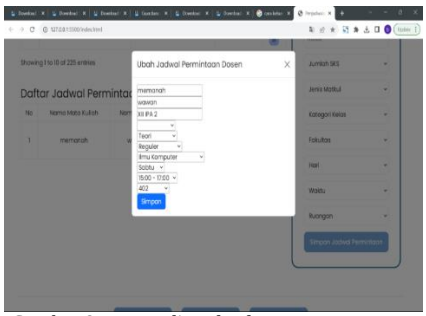
Gambar 5, menampilkan seluruh data jadwal permintaan dosen yang berhasil diinput dalam bentuk tabel. Pada tabel ini juga memuat informasi detail jadwal yang dipilih dosen yaitu terdiri dari hari, waktu/jam dan ruangan. Pada tabel ini memungkinkan juga kita untuk melakukan fungsi *delete* untuk menghapus data tertentu pada tabel ini dan juga fungsi *edit* untuk memodifikasi atau mengubah detail informasi data tertentu pada tabel ini.

Gambar 6 Form *Input* Data Jadwal Tidak Bersedia Dosen

Gambar 6 di atas menampilkan tampilan form jadwal tidak bersedia dosen beserta tampilan seluruh data jadwal tidak bersedia dosen yang berhasil diinput dalam bentuk tabel. Form ini berfungsi untuk mengakomodasi preferensi jadwal tidak bersedia mengajar dosen. Pada tabel ini juga memuat informasi detail jadwal yang dipilih dosen yaitu terdiri dari hari, dan nama dosen yang bersangkutan. Pada tabel ini memungkinkan juga kita untuk melakukan fungsi *delete* untuk menghapus data tertentu pada tabel ini dan juga fungsi *edit* untuk memodifikasi atau mengubah detail informasi data tertentu pada tabel ini.



Gambar 7 Form *Edit* Data Perkuliahan



Gambar 8 Form *Edit* Jadwal Permintaan Dosen

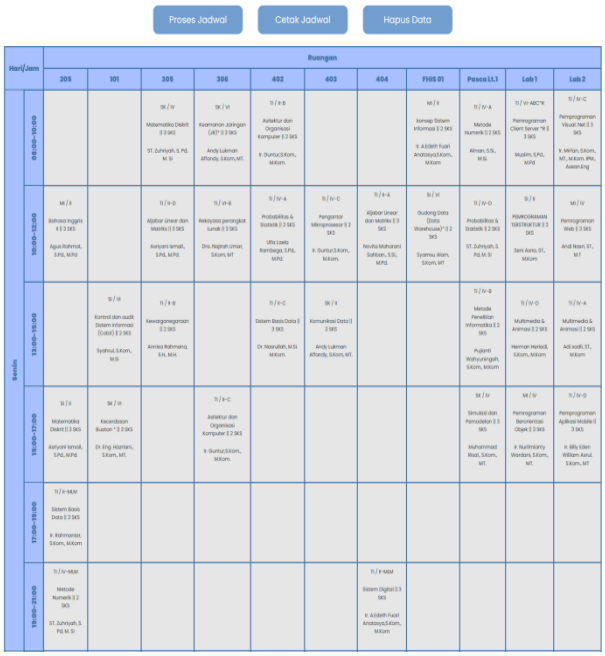
Gambar 9 Form *Edit* Jadwal Tidak Bersedia Dosen

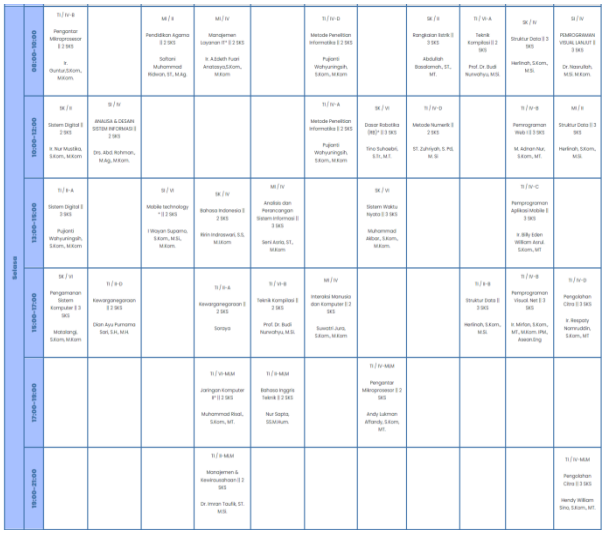
Gambar 7, 8, 9 diatas menampilkan tampilan form edit data perkuliahan, data jadwal permintaan dosen dan data jadwal tidak bersedia dosen. Pada form tersebut kita bisa memodifikasi atau mengubah detail informasi terkait data perkuliahan, data jadwal permintaan dosen, ataupun data tidak bersedia dosen yang kita pilih.

Gambar 10 Tampilan Sebelum Proses Jadwal

Gambar 10 menunjukkan tampilan tabel jadwal perkuliahan sebelum fungsi proses jadwal dijalankan. Sehingga terlihat tampilan masih kosong.

Sementara untuk tampilan tabel jadwal perkuliahan sesudah fungsi proses jadwal dijalankan adalah sebagai berikut:





Gambar 11 Hasil Proses Penjadwalan

Tabel hasil penjadwalan akan menampilkan hasil penjadwalan berdasarkan kriteria/batasan yang telah ditentukan sebelumnya dan dengan implementasi algoritma *PSO*. Tabel ini dilengkapi tombol proses jadwal, untuk mulai membuat jadwal perkuliahan menggunakan PSO dan juga terdapat tombol download jadwal yang berfungsi untuk mengunduh jadwal dalam bentuk pdf. Jika kebetulan perangkat kita terhubung dengan printer maka tombol download jadwal ini bisa langsung mencetak hasil penjadwalan yang siap dipasang/ditempel di papan informasi UHM. Pada tabel tersebut memungkinkan user untuk mengedit jadwal yang sudah dibuat dengan cara mengklik salah satu cell tabel jadwal yang ingin diedit. User perlu mengklik ruang kosong di luar tabel bila sudah melakukan pengeditan.

Gambar 12 Edit Tabel Jadwal

Daftar Mata Kuliah yang Belum Memiliki Jadwal

Nama Mata Kuliah	Nama Dosen	Kelas	Jumlah SKS	Jenis Matkul	Kategori Kelas	Fakultas
Struktur Data	Herlinah, S.Kom., M.Si.	TI / II-A	3	Praktikum	Reguler	Ilmu Komputer

Gambar 13 Edit Tabel Jadwal

PSO memiliki beberapa keunggulan seperti mudah diimplementasikan dan memiliki performa yang baik dalam menemukan solusi optimal. Meskipun merupakan metode yang kuat untuk menyelesaikan berbagai masalah optimisasi, PSO memiliki beberapa keterbatasan dan kelemahan seperti konvergensi yang tidak terjamin. PSO adalah algoritma heuristik yang mungkin atau mungkin tidak mencapai solusi terbaik. Keberhasilan PSO juga bergantung pada file excel (data perkuliahan) yang diimpor user. Jika terdapat kesalahan pada data di dalam file excel, maka kemungkinan

beberapa data tidak berhasil diolah ke dalam bentuk jadwal. Karena itu, pada sistem penjadwalan ini, terdapat tabel seperti gambar 13 yang menampung seluruh data perkuliahan yang tidak berhasil diolah ke dalam bentuk jadwal.

Sebagai bahan kelayakan atas aplikasi yang dihasilkan pada penelitian ini, maka dilakukan pengujian sistem dan validasi oleh staf BAAK UHM.

Tabel 5 Pengujian *Black Box Beta*

Indikator	Responden				
	SS	S	CS	TS	STS
Desain dan tampilan sistem ini mudah dipahami	0	2	0	0	0
Sistem ini mudah dioperasikan	0	2	0	0	0
Semua tombol pada sistem ini berfungsi dengan baik	0	2	0	0	0
Semua fitur pada sistem ini berjalan dengan baik	0	2	0	0	0
Sistem ini dapat membuat jadwal yang sesuai ketentuan/kebijakan UHM	2	0	0	0	0
Sistem dapat menghasilkan jadwal yang tidak bentrok	0	2	0	0	0
Sistem sangat membantu proses pembuatan jadwal perkuliahan di UHM	2	0	0	0	0
Sistem dapat menghemat waktu pembuatan jadwal perkuliahan	2	0	0	0	0

Pengujian dilakukan secara langsung oleh Kepala BAAK dan Staff BAAK UHM. Setelah melalui proses perhitungan untuk mendapatkan jumlah persentase jawaban dari 2 responden maka didapatkan 37,5% responden yang sangat setuju, 62,5% responden setuju, 0% responden cukup setuju, 0% responden tidak setuju, dan 0% responden sangat tidak setuju. Seperti data pada tabrl 5

Pengujian performansi dilakukan untuk mengetahui performa sistem dalam menghasilkan jadwal kuliah yang optimals. Data uji coba yang digunakan adalah data mata kuliah semester genap TA. 2022/2023. Total data sebanyak 226 data perkuliahan. Sebanyak 225 data perkuliahan dan 1 data jadwal permintaan dosen dapat dilihat pada tabel 6.

Tabel 6 Pengujian Performansi

Pengujian ke-	Jumlah Iterasi	Jumlah Pengujian
1	27	Berhasil
2	34	Berhasil
3	38	Berhasil
4	14	Berhasil
5	15	Berhasil
6	24	Berhasil
7	62	Berhasil
8	51	Berhasil
9	17	Berhasil
10	21	Berhasil

Pengujian performansi dilakukan untuk mengetahui performa sistem dalam menghasilkan jadwal kuliah yang optimal. Data uji coba yang digunakan adalah data mata kuliah semester genap TA. 2022/2023. Total data sebanyak 226 data perkuliahan. Sebanyak 225 data perkuliahan dan 1 data jadwal permintaan dosen.

4. Kesimpulan

Sistem optimasi penjadwalan perkuliahan di Universitas Handayani Makassar menggunakan *Particle Swarm Optimization* (PSO) berhasil dirancang untuk menghasilkan jadwal perkuliahan yang optimal.

Hasil penjadwalan berhasil memenuhi semua kriteria atau batasan (*constraint*) yang ditentukan, sehingga jadwal yang dihasilkan sesuai kondisi/kebijakan perkuliahan yang ada di UHM. Serta berhasil mengakomodasi preferensi jadwal mengajar dosen dan jadwal tidak bersedia mengajar dosen.

Pada penelitian ini berhasil mengurangi jumlah iterasi untuk memproses data yang lebih banyak yaitu 226, sehingga waktu komputasi yang dibutuhkan berkurang secara signifikan yaitu dibutuhkan waktu komputasi selama kurang dari 5 detik.

Aplikasi penjadwalan mata kuliah ini mampu menangani proses input data, melakukan proses pembuatan jadwal perkuliahan secara otomatis dan menghasilkan jadwal yang dapat ditampilkan dan dapat dicetak.

Walaupun berhasil menghasilkan jadwal yang optimal, sistem ini masih kurang fleksibel untuk menyesuaikan perubahan *constraint* dan jumlah ruangan kedepannya.

Daftar Rujukan

[1] S. and B. Prasetyo, D. Mustari, Putu Linda primandari, W. H. N. Putra, and A. R. Perdanakusuma, "Kepemerintahan Berskala Small-Medium Dengan Metode Use Case Point (Ucp) Cost Estimate in Small To Medium Government Software Development Projects With Use Case Point (Ucp)," *J. Tecnoscienza*, vol. 4, no. 2, 2020.

[2] A. K. I. Triana, "Komparasi Metode Ant Colony dengan Tabu Search untuk Penjadwalan Perkuliahan," *Matrix J. Manaj. Teknol. dan Inform.*, vol. 6, no. 3, 2016.

[3] H. W. Pangestu, K. Yulianti, and R. Marwati, "Implementasi Algoritma Fuzzy Evolusi Pada Penjadwalan Perkuliahan Herny," *Eureka Matika*, vol. 4, no. 1, 2016.

[4] N. M. W. Suryati, N. K. D. A. Jayanti, and I. K. D. Suryawan,

"Implementasi Algoritma Genetika pada Perancangan Sistem Penjadwalan Perkuliahan di STIKOM Bali," *Eksplora Inform.*, vol. 3, no. 2, 2014.

[5] A. B. Saputra, "Implementation Of Genetic Algorithm In College Scheduling System Ftti Unjani Yogyakarta," *Compiler*, vol. 9, no. 1, 2020, doi: 10.28989/compiler.v9i1.656.

[6] D. Wungguli, "Jurnal Ilmu Matematika dan Terapan Penerapan Model Integer Linear Programming Dalam Optimasi Penjadwalan Perkuliahan Secara Otomatis Application of Integer Linear Programming Model in Automatic Lectures Scheduling Optimization," *Barekeng J. Mat.*, vol. 14, 2020.

[7] Ayu Ratna Juwita, "Penjadwalan Perkuliahan menggunakan Algoritma Particle Swarm Optimization (PSO)," *J. SISFOTEK Glob.*, vol. 10, 2020.

[8] A. R. Juwita, A. R. Pratama, and T. Triono, "Implemntasi Algoritma Particle Swarm Optimization untuk Penjadwalan Perkuliahan di Fakultas Teknik Dan Ilmu Komputer Universitas Buana Perjuangan Karawang," *J. SISFOTEK Glob.*, vol. 10, no. 1, 2020, doi: 10.38101/sisfotek.v10i1.270.

[9] E. Sugiarto, S. Winarno, and A. Fahmi, "Penjadwalan Perkuliahan Otomatis Berbasis Fuzzy Logic Dan Genetic Algorithm," *Techno.COM*, vol. 14, no. 4, 2015.

[10] N. R. Radliya, "Pemodelan Sistem Penjadwalan Perkuliahan Menggunakan Algoritma Genetika," *Jamika*, vol. 1, no. 12, 2014.

[11] V. Witary, N. Rachmat, and Inayatullah, "Optimasi Penjadwalan Perkuliahan dengan Menggunakan Algoritma Genetika (Studi Kasus : AMIK MDP , STMIK GI MDP dan STIE MDP)," *J. STMIK GI MDP*, 2013.

[12] P. D. Aida, S. Aries, "Penjadwalan Perkuliahan Menggunakan Algoritma Hill Climbing," *Pros. Semin. Nas. Inform. dan Sist. Inf.*, vol. 1, no. 3, 2017.

[13] A. Rahman, E. M. Yuniarno, and I. K. E. Purnama, "Optimasi Penjadwalan Perkuliahan Menggunakan Metode Harmony Search," *Al-Khwarizmi J. Pendidik. Mat. dan Ilmu Pengetah. Alam*, vol. 2, no. 2, 2018, doi: 10.24256/jpmipa.v2i2.111.

[14] A. Josi, "Implementasi Algoritma Genetika pada Aplikasi Penjadwalan Perkuliahan Berbasis Web dengan mengadopsi model Waterfall (Studi Kasus: STMIK Prabumulih)," *J. Inform. J. Pengemb. IT*, vol. 2, no. 2, 2017, doi: 10.30591/jpit.v2i2.517.