

Penjadwalan Mata Pelajaran Menggunakan Algoritma Particle Swarm Optimization (PSO) Pada SMPIT Mufidatul Ilmi

Muhammad Muhardeny^{1*}, Muhammad Haviz Irfani¹, Juhaini Alie²

^{1,2,3}Program Teknik Informatika
^{1,2,3}Universitas Indo Global Mandiri
^{1,2,3}Palembang

* muhardeny12@gmail.com, ¹m.haviz@uigm.ac.id, ²juhaini@uigm.ac.id,

Abstrak

Penjadwalan memiliki pembagian waktu berdasarkan rencana pengaturan urutan kerja berupa daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci sangat perlu dalam menjalankan proses bisnis institusi/perusahaan. Bagaimana kompleksitas proses dalam menjadwalkan mata pelajaran yang sesuai, penting diperhatikan dari berbagai sudut pandang baik guru, siswa dan ruang kelas. Pemberian jadwal mengajar guru berdasarkan kemampuan di bidang mata pelajaran, waktu yang cocok setiap semesternya sangat penting dipertimbangkan untuk pengaturan jadwal yang sangat kompleks, jumlah ruang kelas yang bisa digunakan dalam kegiatan mengajar terbilang sedikit, dan mencegah terjadi konflik mengajar guru sehingga perlunya pengoptimalan pada penjadwalan mata pelajaran yang ingin dibuat. Selanjutnya pada tahapan pengembangan aplikasi menggunakan metode *Waterfall*. Tujuan dari penelitian ini adalah membangun aplikasi penjadwalan pelajaran di SMPIT Mufidatul Ilmi dengan menerapkan algoritma *particle swarm optimization (PSO)* untuk menyusun jadwal pelajaran. Particle Swarm Optimization merupakan algoritma berbasis populasi yang mengeksplorasi individu dalam pencarian. Dalam PSO populasi disebut *swarm* dan individu disebut *particle*, Tiap partikel berpindah dengan kecepatan yang diadaptasi dari daerah pencarian dan menyimpannya sebagai posisi terbaik yang pernah dicapai. Analisis perancangan meliputi *Use Case Diagram, Activity Diagram, Class Diagram, Sequence Diagram, Entity Relationship Diagram(ERD)*. Hasil penelitian ini memberikan beberapa fitur (layanan) data primer, terutama fitur untuk memberikan hasil penjadwalan dari pemrosesan dengan algoritma PSO.

Kata Kunci: *penjadwalan, particle swarm optimization, waterfall*

Abstract

Scheduling has a division of time based on a work sequence arrangement plan in the form of a list or table of activities or an activity plan with a detailed division of implementation time which is very necessary in carrying out institutional/company business processes. It is important to note the complexity of the process in scheduling appropriate subjects from various perspectives, both teachers, students and classrooms. Provision of teacher teaching schedules based on abilities in the field of subjects, suitable time each semester is very important to consider for very complex schedule arrangements, the number of classrooms that can be used in teaching activities is relatively small, and preventing teacher teaching conflicts so that the need for optimization of

eye scheduling lesson to be made. Furthermore, at the stage of application development using the Waterfall method. The purpose of this research is to build a lesson scheduling application at SMPIT Mufidatul Ilmi by applying the particle swarm optimization (PSO) algorithm to compile lesson schedules. Particle Swarm Optimization is a population-based algorithm that exploits individuals in search. In PSO the population is called a swarm and individuals are called particles. Each particle moves at a speed adapted from the search area and stores it as the best position ever achieved. Design analysis includes Use Case Diagrams, Activity Diagrams, Class Diagrams, Sequence Diagrams, Entity Relationship Diagrams (ERD). The results of this study provide several primary data (service) features, especially features to provide scheduling results from processing with the PSO algorithm.

Keywords: scheduling, particle swarm optimization, waterfall

1. Pendahuluan

Berdasarkan kamus besar bahasa Indonesia, jadwal merupakan pembagian waktu berdasarkan rencana pengaturan urutan kerja, daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci. Sedangkan pengertian penjadwalan adalah proses, cara, perbuatan menjadwalkan atau memasukkan kedalam jadwal (Depdikbud,1995).

Untuk membuat jadwal mata pelajaran yang layak, penting melihat berbagai sudut pandang, yaitu dari sudut pandang guru, siswa dan ruang kelas. Dalam sebuah penelitian yang akan terfokus kepada penjadwalan dari sudut pandang guru dan waktu mengajar, dalam pemberian jadwal guru berdasarkan kemampuan di bidang mata pelajaran dan waktu yang cocok, sangat memerlukan pengaturan jadwal yang sangat kompleks dan juga banyaknya ruang kelas yang bisa digunakan dalam kegiatan mengajar terbilang sedikit dan mencegah terjadi konflik atau tidaknya jadwal mengajar guru sehingga perlunya pengoptimalan pada penjadwalan mata pelajaran yang ingin dibuat.

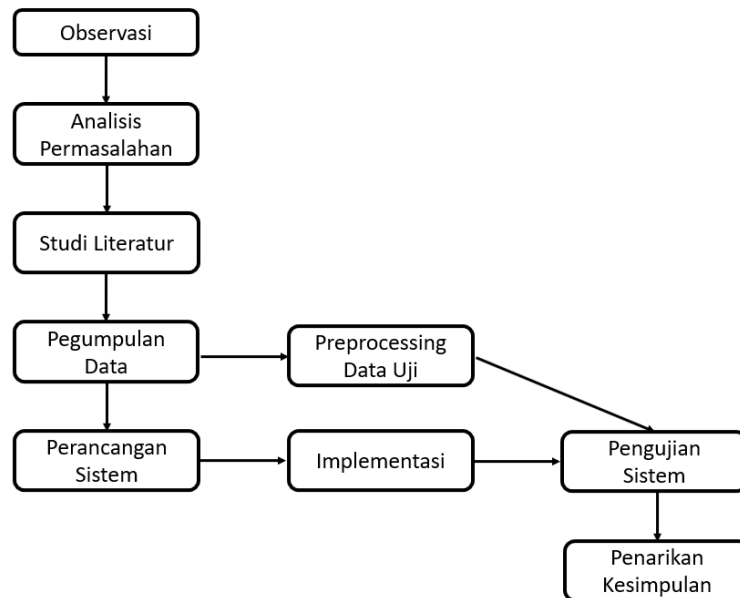
Permasalahan sekolah tersebut penulis mengambil studi kasus di SMPIT Mufidatul Ilmi Palembang yang merupakan jenjang di Sekolah Sekolah Islam Terpadu Mufidatul ilmi sekolah tersebut berlokasi di JL.Dipenogoro Gang Swakarya Rt.027 Rw.06, Sukamoro Kec.Talang Kelapa Kab.Banyuasin Prov.Sumatera selatan yang merupakan lembaga sekolah islam pertama di kabupaten banyuasin yang berbasis islam terpadu. Sit Mufidatul Ilmi telah memiliki tiga jenjang pendidikan yaitu, TKIT (Taman Kanak-Kanak Islam Terpadu), SDIT (Sekolah Dasar Islam Terpadu), SMPIT (Sekolah Menengah Pertama Islam Terpadu).

Penting untuk membuat penjadwalan mata pelajaran dengan menerapkan algoritma *Particle Swarm Optimization* (PSO) di SMPIT Mufidatul Ilmi yang menggunakan pendekatan yang diinspirasi oleh perilaku sosial kawanan burung atau sekelompok ikan yang merupakan bagian dari pencarian *heuristic* (berpola) dan suatu metode optimasi yang menggabungkan *local search* dengan metode *global search* untuk menyeimbangkan antara eksplorasi dan eksploitasi.

2. Metode

1. Tahapan Penelitian

Adapun tahapan penelitian yang digunakan sebagai berikut (Gambar 1).



Gambar 1. Tahapan Penelitian[1]

Observasi – Melakukan pengamatan secara langsung pada tempat objek penelitian dan *output* yang diperoleh berupa diagram *use case* .

Analisis Permasalah - Mengidentifikasi masalah yang terjadi pada objek penelitian.

Studi Literatur – Menggunakan dan mempelajari literatur terpublikasi (artikel) yang terkait dengan penelitian.

Pengumpulan Data - Melakukan pengumpulan data yang diperlukan pada saat tahapan rancangan, implementasi dan testing.

Perancangan Sistem- Melakukan desain kebutuhan, *interface system*, struktur aktivitas.

Implementasi - Melakukan pembuatan kode program sesuai dengan kebutuhan.

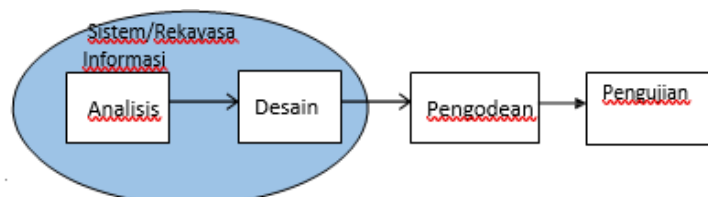
Processing Data Uji- Melakukan pembersihan data (*missing value*) dan kesesuaian tipe data.

Pengujian Sistem- Menentukan fitur/layanan aplikasi yang sesuai dengan kebutuhan dan Data, serta hasil yang diharapkan.

Penarikan Kesimpulan – Tahapan akhir untuk memberikan kesimpulan atas pemecahan masalah yang sudah dilakukan.

2. Metode Waterfall

Model *Waterfall* (air terjun) (Gambar 2) adalah salah satu model dalam *Software Development Life Cycle* (SDLC) yang sering disebut sebagai model sekuensial linier atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, implementasi / pengodean, pengujian dan tahap pendukung[2].

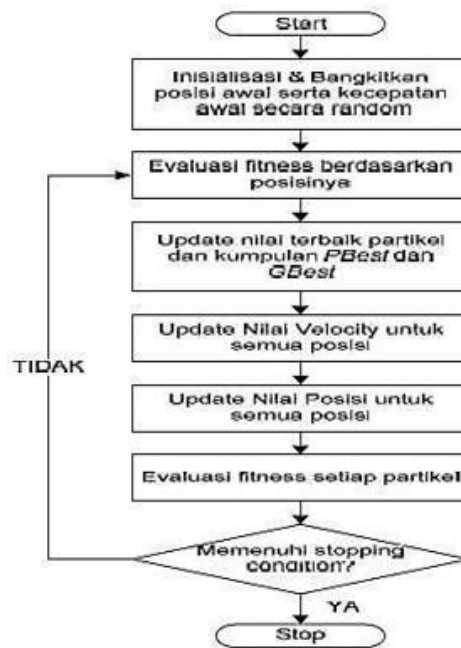


Gambar 2. Tahapan Model Waterfall

3. Penjadwalan

Berdasarkan kamus besar bahasa Indonesia, jadwal merupakan proses, cara, pembagian waktu berdasarkan rencana pengaturan urutan kerja, daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci[3].

4. Particle Swarm Optimization (PSO)



Gambar 3. Tahapan Particle Swarm Optimization[1][3]

Dalam penjadwalan mata pelajaran akan dibahas tentang pembagian jadwal untuk tiap guru dan siswa pada sekolah. Dalam penjadwalan ujian akan dibahas pengaturan guru yang menjaga ujian dan siswa yang menempati ruang ujian yang ada. Dalam penjadwalan karyawan, dilakukan pengaturan karyawan yang akan bekerja pada waktu tertentu dibagian tertentu. Sedangkan penjadwalan job shop dilakukan penjadwalan sejumlah mesin dan sejumlah pekerjaan terkait rute yang telah ditentukan.

Algoritma yang mengeksploitasi individu dalam pencarian, dengan populasi disebut *swarm* dan individu disebut *particle*. Setiap partikel akan berpindah dengan kecepatan yang diadaptasi dari daerah pencarian dan menyimpannya sebagai posisi terbaik yang pernah dicapai[4]. Algoritma PSO ini awalnya diusulkan oleh R. Kennedy J. & Eberhart.(1995). Selain itu, algoritma optimasi yang meniru proses yang terjadi dalam kehidupan populasi burung dan ikan dalam bertahan hidup (Gambar 3). Beberapa istilah umum yang biasa digunakan dalam *Optimisasi Particle Swarm* dapat didefinisikan sebagai berikut [1][3][5]:

1. Swarm : populasi dari suatu algoritma.
2. Particle: anggota (individu) pada suatu swarm. Setiap particle merepresentasikan suatu solusi yang potensial pada permasalahan yang diselesaikan. Posisi dari suatu particle adalah ditentukan oleh representasi solusi saat itu.
3. Pbest (Personal best): posisi Pbest suatu particle yang menunjukkan posisi particle yang dipersiapkan untuk mendapatkan suatu solusi yang terbaik.
4. Gbest (Global best) : posisi terbaik particle pada swarm.
5. Velocity (vektor): vektor yang menggerakkan proses optimisasi yang menentukan arah di mana suatu particle diperlukan untuk berpindah (move) untuk memperbaiki posisinya semula.

Tahapan Algoritma PSO sebagai berikut[5][6][1]:

1. Inisialisasi:

Tentukan jumlah partikel dalam koloni (population size). Inisialisasi posisi dan kecepatan setiap partikel secara acak di dalam ruang pencarian. Tetapkan posisi terbaik (pbest) awal setiap partikel dengan posisi saat ini. Tetapkan posisi terbaik koloni (gbest) awal dengan salah satu posisi terbaik partikel.

2. Evaluasi:

Hitung nilai fitness untuk setiap partikel berdasarkan fungsi objektif yang ingin dioptimalkan.

3. Pembaruan Kecepatan dan Posisi:

Perbarui kecepatan setiap partikel dengan mempertimbangkan faktor kognitif, sosial, dan inersia. Kecepatan partikel dapat diubah menggunakan persamaan[6]:

$$V_i(t+1) = w * V_i(t) + c1 * rand() * (pbest_i(t) - X_i(t)) + c2 * rand() * (gbest(t) - X_i(t)) \dots \dots \dots (1)$$

Di mana $V_i(t)$ adalah kecepatan partikel i pada iterasi t , w adalah faktor inersia, $c1$ dan $c2$ adalah faktor akselerasi kognitif dan sosial, $rand()$ adalah bilangan acak antara 0 dan 1, $pbest_i(t)$ adalah posisi terbaik partikel i pada iterasi t , $X_i(t)$ adalah posisi partikel i pada iterasi t , dan $gbest(t)$ adalah posisi terbaik koloni pada iterasi t . Perbarui posisi setiap partikel berdasarkan kecepatan yang telah diperbarui.

4. Pembaruan pbest dan gbest: Periksa setiap partikel apakah posisi saat ini memberikan solusi yang lebih baik dari pbest sebelumnya. Jika iya, perbarui pbest partikel dengan posisi saat ini. Periksa setiap partikel apakah terdapat partikel dengan pbest yang lebih baik daripada gbest saat ini. Jika iya, perbarui gbest dengan pbest partikel tersebut.

5. Kriteria Berhenti: Algoritma PSO dapat berhenti jika telah mencapai jumlah iterasi maksimum yang ditentukan atau jika solusi yang diinginkan telah ditemukan.

6. Ulangi Langkah 3-5 sampai kriteria berhenti terpenuhi.

Dalam setiap iterasi, partikel-partikel di koloni saling berinteraksi dan berbagi informasi melalui kecepatan dan posisi mereka. Konsep utama dalam PSO adalah partikel bergerak ke arah pbest (posisi terbaik pribadi) mereka sendiri dan juga gbest (posisi terbaik koloni).

Dalam proses random diperlukan slot untuk menampung dari random posisi dan random kecepatan. Melalui proses pembangkitan partikel ini dapat maka kumpulan partikel dapat terdistribusi secara acak. Posisi dan kecepatan dari partikel direpresentasikan melalui vektor.

3. Hasil

Posisi awal setiap partikel dengan kecepatan bernilai nol untuk partikel 1 sampai dengan 10 (Tabel 1).

Tabel 1. Tahap Awal Inisialisasi

Partikel	v	x(t)
1	0	57
2	0	72
3	0	61
4	0	127
5	0	41
6	0	1
7	0	147

8	0	14
9	0	164
10	0	97

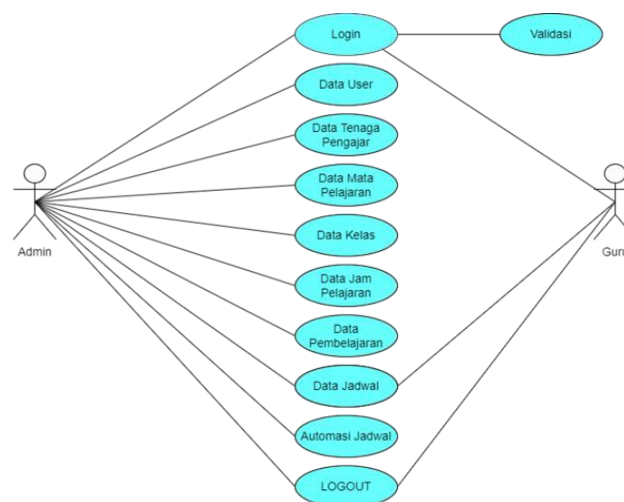
Setiap partikel akan diperiksa satu persatu untuk dibandingkan dengan partikel lainnya sesuai *constraint* yang harus diotimasi. Ketika terjadinya pelanggaran terhadap *constraint* maka nilai pelanggaran akan ditotalkan untuk satu iterasi tersebut disini penulis bertujuan untuk meminimalisasi nilai fitness[7]. Semakin kecil jumlah pelanggaran yang terjadi, maka nilai fitness yang dihasilkan semakin besar, sebaliknya semakin besar jumlah pelanggaran yang terjadi, maka nilai fitness yang dihasilkan semakin kecil (Tabel 2).

Tabel 2. Nilai Fitness Per iterasi

Partikel	v	x(t)	f(x)
1	0	104	0,5
2	0	168	0,5
3	0	64	0,7
4	0	78	0,4
5	0	167	0,4
6	0	65	0,3
7	0	4	0,5
8	0	70	0,7
9	0	62	0,4
10	0	58	0,4

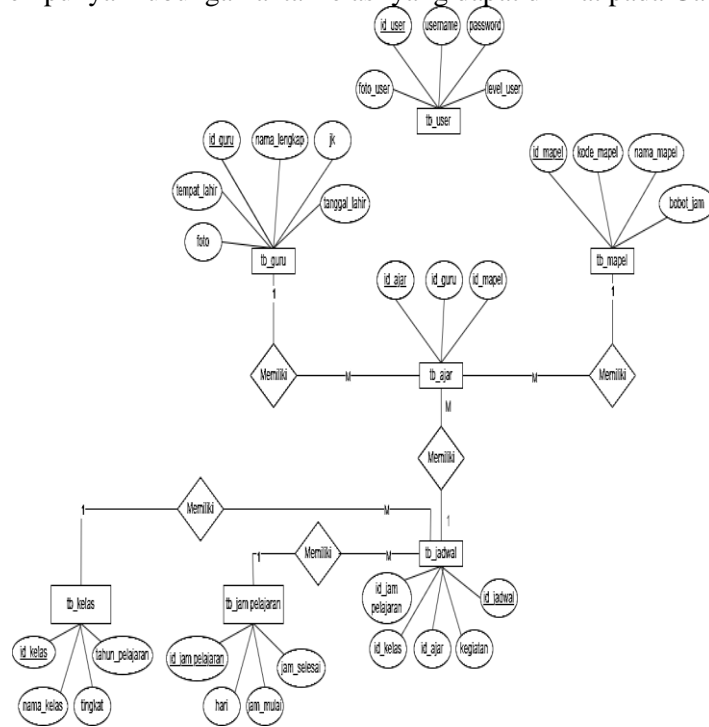
1. Use Case Diagram Penjadwalan

Diagram kebutuhan untuk mendapatkan hasil penjadwalan dengan proses relasi otomatisasi beberapa data seperti data guru, kelas, mata pelajaran, jam pelajaran (Gambar 4).



Gambar 4. Use Case Diagram Penjadwalan

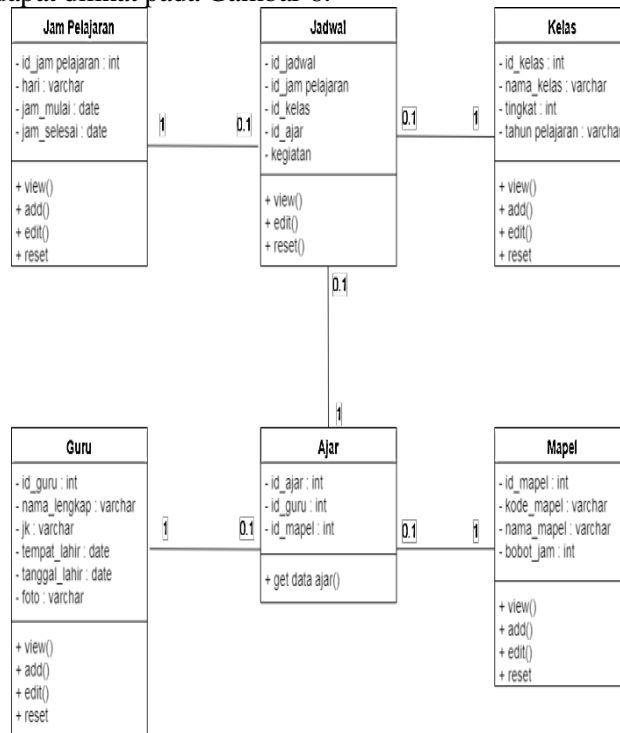
Entity Relationship Diagram (ERD) dengan aktor admin dan guru dan relasi antar objek-objek data yang mempunyai hubungan antar relasi yang dapat dilihat pada Gambar 5 [2][8].



Gambar 5. ERD Penjadwalan[2]

2. Class Diagram Penjadwalan

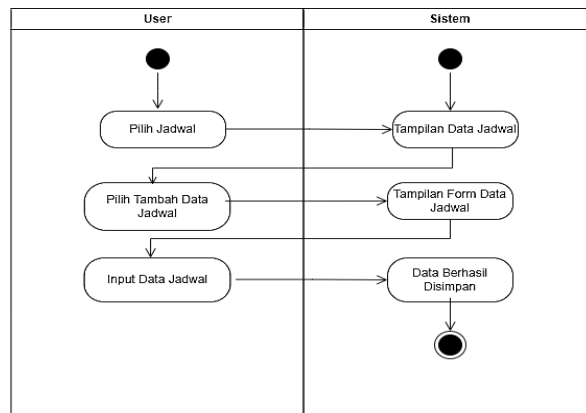
Class jadwal mempunyai relasi terhadap jam, kelas, dan pengajaran. Sementara kelas pengajaran berelasi dengan kelas guru dan mata pelajaran. Diagram yang menggambarkan struktur kelas tersebut dapat dilihat pada Gambar 6.



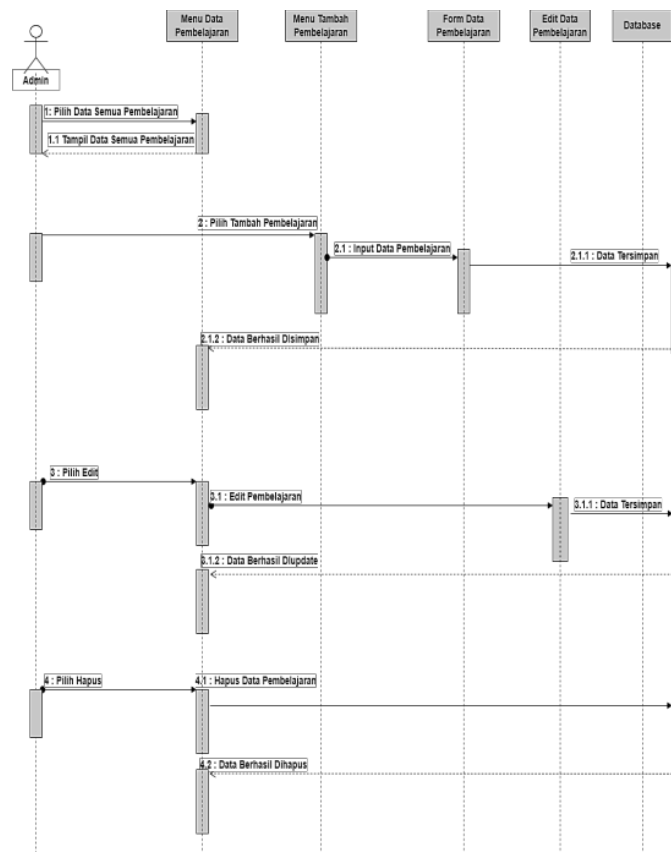
Gambar 6. Class Diagram Penjadwalan[2]

3. Diagram Activity dan Squence Pembelajaran

Diagram *activity* dan *squence* pada menu pembelajaran yang menampilkan pemrosesan pembelajaran yang dilakukan oleh admin dengan *class*, objek, *timeline* dan database dapat dilihat pada Gambar 7 dan Gambar 8.



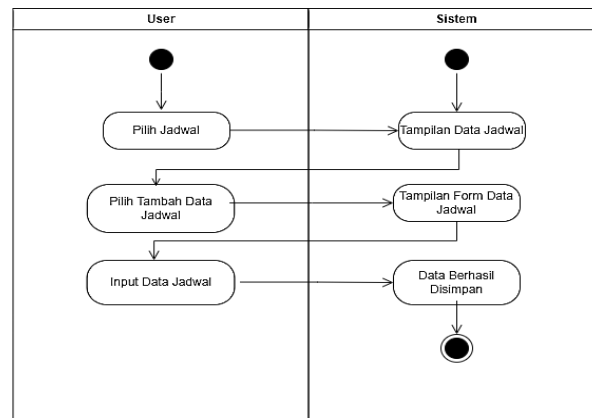
Gambar 7. Activity Diagram Jadwal



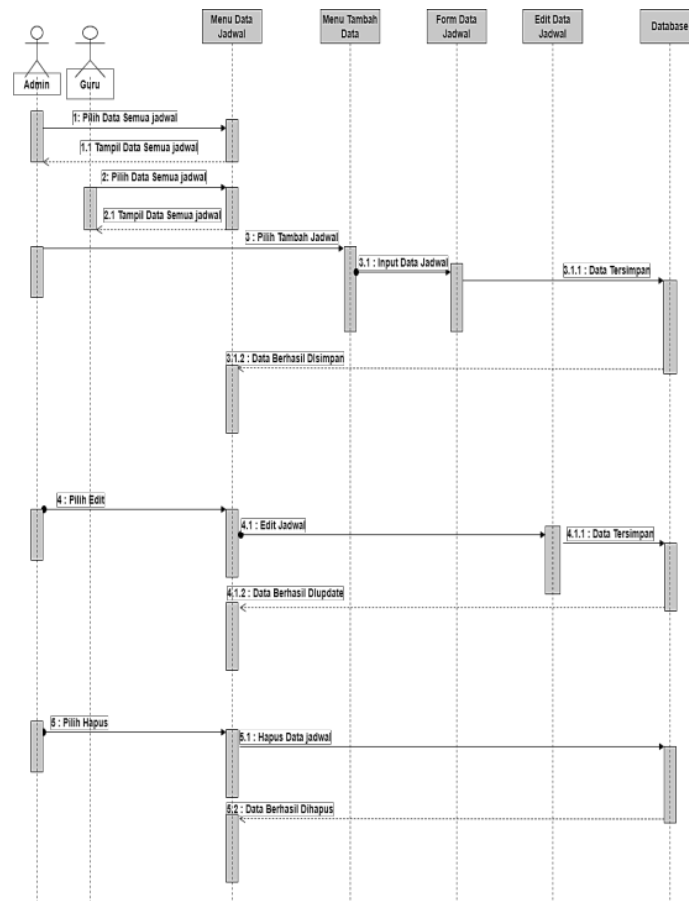
Gambar 8. Sequence Diagram Pembelajaran[2]

4. Diagram Activity dan Squence Jadwal

Terlihat diagram *activity* dan *sequence* untuk menu Jadwal yang menampilkan pemrosesan data jadwal yang dilakukan oleh admin dan guru dapat dilihat pada Gambar 9 dan Gambar 10.



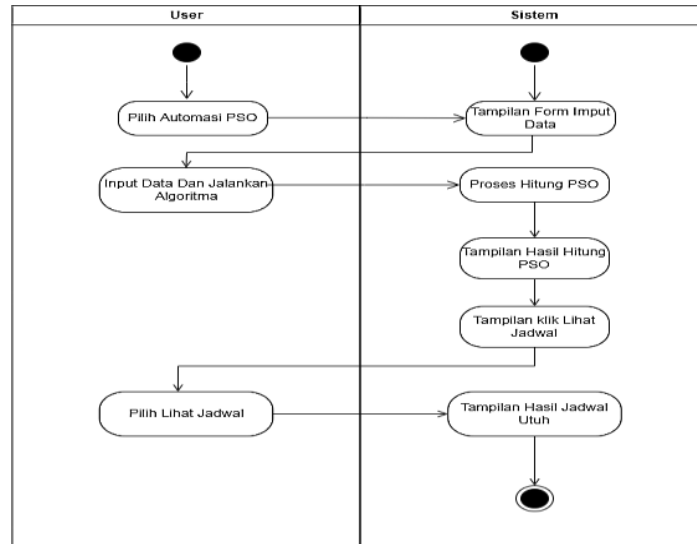
Gambar 9. Activity Diagram Jadwal



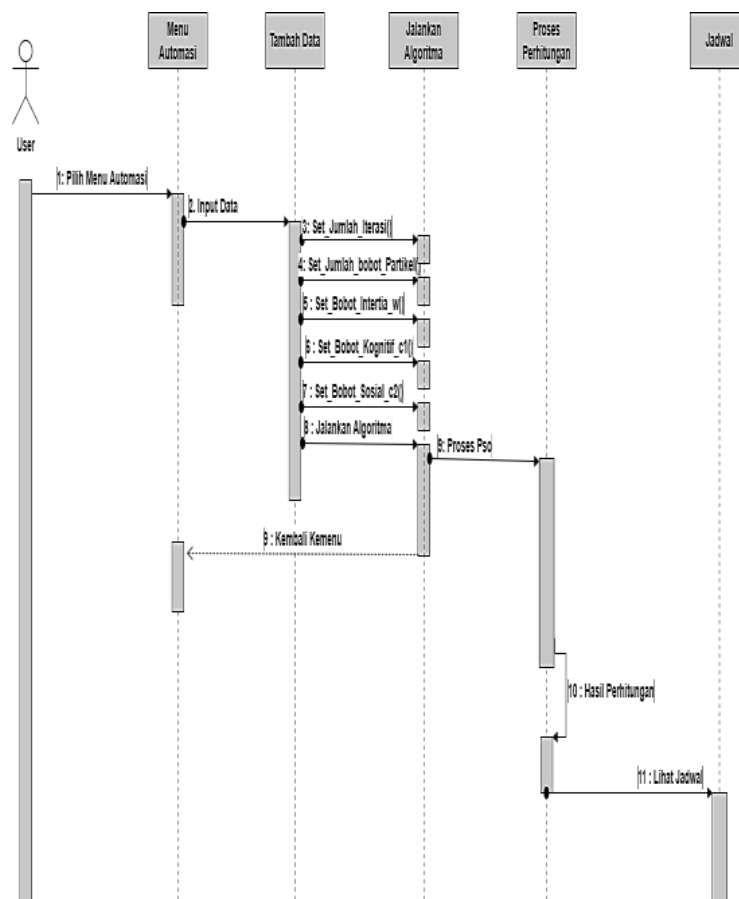
Gambar 10. Sequence Diagram Jadwal

5. Activity dan Sequence Diagram Automasi Jadwal

Pada *activity* dan *Sequence Diagram* Automasi Jadwal yang menampilkan pemrosesan data perhitungan menggunakan algoritma PSO yang dilakukan oleh admin dapat dilihat pada Gambar 11 dan Gambar 12.



Gambar 11. Activity Diagram Automasi Jadwal

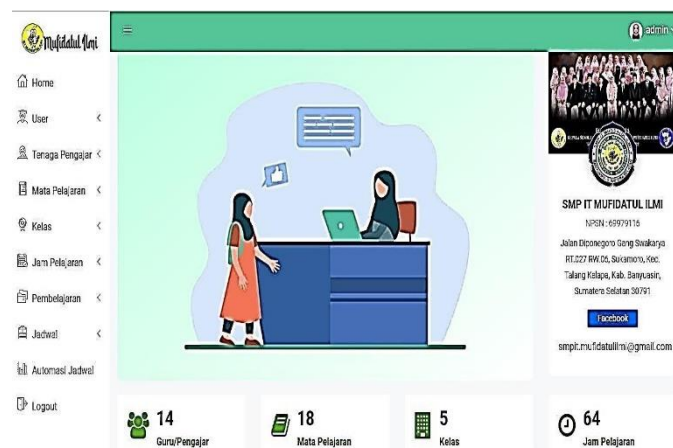


Gambar 12. Sequence Diagram Automasi Jadwal

Tabel 3. Iterasi Fitness Function Pbest dan Gbest [7]

Partikel	v	x	f(x)	Pbest	Gbest	Bestf(x)
1	40.097206686669	140	0.6	140		
2	70.35924946906	130	0.4	130		
3	0	153	0.3	153		
4	50.688921660506	137	0.3	137		
5	80.194413373337	127	0.2	127	127	0,2
6	102.13439439057	120	0.4	120		
7	14.374470321636	158	0.6	158		
8	80.950964442897	127	0.2	127		
9	91.542679416734	124	0.3	124		
10	83.977168721136	126	0.5	126		

Setelah dihitung sebanyak 5 iterasi didapatkan posisi baru dengan nilai Gbest 127 dan nilai constraint minimum yang didapat pada iterasi 1 dari 5 iterasi adalah 0,2 , ada 2 partikel yang mendapati nilai constraint 0,2 maka nilai yang diambil biasanya adalah nilai awal. Jika posisi baru yang telah diperoleh ditampilkan dalam representasi pada posisi jadwal (Tabel 3).



Gambar 13. Halaman Antarmuka Home

Tampilan awal atau Home untuk aplikasi penjadwalan berupa layanan pada bagian sisi kiri. Data primer seperti data guru, mata pelajar, dan kelas diinput langsung oleh admin. Sementara tampilan jadwal yang sudah dilakukan proses automasi jadwal terlebih dahulu, dapat dilihat pada halaman jadwal (Gambar 13 dan Gambar 14).

Gambar 14. Sequence Diagram Automasi Jadwal

No.	Foto	Nama Guru	Mata Pelajaran	Kelas	Hari Mengajar	Aksi
1		Lidia Juli Andriani	Bahasa Arab	VII.1 IX.1 IX.2	Selasa	
2		Lidia Juli Andriani	Mengaji	IX.1 IX.2	Selasa	
3		Siti Zuhriyah, S.Pd.	Mengaji	VII.1 VII.2 VII.3	Selasa Rabu Kamis Jumat	
4		Kartika, S.Pd.	Pend. Agama Islam	VII.1 VII.2 VII.3 IX.1 IX.2	Selasa Selasa Rabu Kamis	
5		Siti Zuhriyah, S.Pd.	Fiqh	VII.1 VII.2 VII.3 IX.1 IX.2	Selasa Rabu Kamis Jumat	
6		Kartika, S.Pd.	Hadis	VII.1 VII.2 VII.3 IX.1 IX.2	Selasa Selasa Rabu Kamis	
7		Siti Zuhriyah, S.Pd.	Ekabah. Kehidupan Islam	VII.1 VII.2 VII.3 IX.1 IX.2	Selasa Rabu Kamis Jumat	

Gambar 15. Interface Halaman Pembelajaran

HARI	JAM	Kelas				
		VII.1	VII.2	VIII	IX.1	IX.2
	07:15 - 07:30	DHURJA				
	07:30 - 08:00	ASMAUL HUSNA				
	08:00 - 08:25	Siti Zuhriyah, S.Pd. Mengaji	Asraruddin, M.H. TIK	Sri Nopriandi, S.Pd. ENG	Lidia Juli Andriani Mengaji	Nadya Putri Setiawati, S.Pd. MTK
	08:25 - 08:50	Fanny Harpani, S.Pd. IPS	Asraruddin, M.H. TIK	Sri Nopriandi, S.Pd. ENG	Siti Zuhriyah, S.Pd. FIQH	Nadya Putri Setiawati, S.Pd. MTK
	08:50 - 09:15	Fanny Harpani, S.Pd. IPS	Juwita Intan Sari, S.Pd. TIQ	Sri Nopriandi, S.Pd. ENG	Siti Zuhriyah, S.Pd. FIQH	Dikky Ardian INA
	09:15 - 09:40	Fanny Harpani, S.Pd. IPS	Juwita Intan Sari, S.Pd. TIQ	Ayudesa Intan Pradana, S.Pd. IPA	Lidia Juli Andriani Arab	Dikky Ardian INA
	09:40 - 09:50	ISTIRAHAT				

Gambar 16. Interface Halaman Penjadwalan

Admin dan pengguna dapat masuk ke halaman jadwal belajar untuk melihat jadwal utuh yang telah diproses oleh perhitungan PSO (Gambar 15 dan Gambar 16).

4. Kesimpulan

1. Aplikasi penjadwalan mata pelajaran menggunakan Particle Swarm Optimization (PSO) menghasilkan jadwal mata pelajaran di SMPIT Mufidatul Ilmi walaupun masih terdapat *soft constraint* yang dilanggar namun untuk *hard constraint* berhasil terpenuhi.
2. Hasil akhir dari penjadwalan mata pelajaran dengan PSO ini menggunakan parameter $C1=1.5$, $C2=1.5$, $W=0.5$ dengan jumlah partikel 10 dan 5 iterasi.
3. Beberapa percobaan yang dilakukan dapat menghasilkan nilai *fitness* minimum pada partikel cukup besar.
4. Aplikasi penjadwalan mata pelajaran menggunakan Particle Swarm Optimization (PSO) memiliki kekurangan yaitu *constraint* yang tidak bisa berubah sesuai kondisi.

Daftar Pustaka

- [1] M. Mansur, T. Prahasto, and F. Farikhin, "Particle Swarm Optimization Untuk Sistem Informasi Penjadwalan Resource Di Perguruan Tinggi," *J. Sist. Inf. Bisnis*, vol. 4, no. 1, pp. 11–19, 2014, doi: 10.21456/vol4iss1pp11-19.
- [2] Sukamto dan and Shalahuddin, *Shalahuddin, M. Rosa A.S 2015. Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Edisi REVISI*, Revisi. Bandung: Informatika Bandung, 2018.
- [3] D. A. R. Wati and Y. A. Rochman, "Model Penjadwalan Matakuliah Secara Otomatis Berbasis Algoritma Particle Swarm Optimization (PSO)," *J. Rekayasa Sist. Ind.*, vol. 2, no. 1, pp. 22–31, 2013.
- [4] L. Istikomah and I. Cholissodin, "Implementasi Algoritma Particle Swarm Optimization (PSO) untuk Optimasi Pemenuhan Kebutuhan Gizi Balita," vol. 1, no. 11, 2017.
- [5] A. Fariza, "Optimasi Penjadwalan Mata Kuliah Di Jurusan Teknik Informatika Pens Dengan Menggunakan Algoritma Particle Swarm Optimization Algoritma Particle Swarm Optimization (Pso)," no. January 2011, 2016.
- [6] I. Kusmarna, L. K. Wardhani, and M. Safrizal, "Aplikasi Penjadwalan Mata Kuliah Menggunakan Algoritma Particle Swarm Optimization (Pso)," *J. Tek. Inform.*, vol. 8, no. 2, pp. 1–8, 2015, doi: 10.15408/jti.v8i2.2441.
- [7] K. R. Surya, "OPTIMASI PENJADWALAN JAM KERJA DIMASA PANDEMI COVID-19 MENGGUNAKAN METODE PARTICLE," vol. 01, pp. 44–50, 2021.
- [8] I. Print, I. Online, and S. Artikel, "Jurnal Teknologi dan Sistem Informasi Perancangan Sistem Informasi Penggajian Terintegrasi Berbasis Web (Studi Kasus di Rumah Sakit St. Elisabeth)," *Peranc. Sist. Inf. Penggajian Terintegrasi Berbas. Web (Studi Kasus di Rumah Sakit St. Elisabeth)*, vol. 3 NO. 2, pp. 225–232, 2017.