

APLIKASI PENJADWALAN MATA PELAJARAN MENGGUNAKAN ALGORITMA ARTIFICIAL BEE COLONY BERBASIS WEB

Aris Fajrianto^[1]; Ilhamsyah^[2]; Rahmi Hidayati^[3]

^{1,3}Jurusan Rekayasa Sistem Komputer, FMIPA Universitas Tanjungpura Pontianak^{[1][3]}

²Jurusan Sistem Informasi, FMIPA Universitas Tanjungpura, Pontianak^[2]

*Corresponding author: ilhamsyah@sisfo.untan.ac.id

INFO ARTIKEL	INTISARI
Diajukan : 5 Maret 2022	<i>Penjadwalan merupakan masalah umum yang terjadi pada institusi pendidikan. Dalam pembuatan jadwal mata pelajaran diperlukan ketelitian dengan mempertimbangkan berbagai faktor seperti pembagian tugas mengajar guru, jumlah ruangan yang tersedia, dan slot waktu agar tidak terjadi bentrok antar jadwal. Waktu yang dibutuhkan juga cukup lama jika pembuatan jadwal dilakukan secara manual. Berdasarkan permasalahan tersebut diperlukan sebuah sistem yang dapat mempermudah pembuatan jadwal untuk meminimalisir kejadian bentrok antar jadwal. Aplikasi yang dibangun berbasis web dengan mengimplementasikan algoritma Artificial Bee Colony yang mampu menyelesaikan permasalahan optimasi seperti penjadwalan. Pengujian dilakukan dengan menggunakan data sebanyak 1195 jadwal. Hasil pengujian menggunakan parameter colony size sebesar 6, limit sebesar 5, dan max iterations sebanyak 5 iterasi menghasilkan solusi optimal dengan nilai rata-rata fitness sebesar 0.95. Dari hasil pengujian menunjukkan jadwal paling optimal yang dihasilkan memiliki jumlah bentrok sebanyak 0 jadwal dan tidak ada jadwal yang melanggar hard constraints.</i>
Diterima : 19 Juni 2022	
Diterbitkan: 25 Juni 2022	
Kata Kunci : Penjadwalan, Mata Pelajaran, Artificial Bee Colony, Web	
Keywords : Scheduling, Subjects, Artificial Bee Colony, Web	Abstract- <i>Scheduling is a common problem that occurs in educational institutions. In making subject schedules, accuracy is needed by considering various factors such as the division of teacher teaching tasks, the number of rooms available, and time slots so that there are no clashes between schedules. The time required is also quite long if the schedule is made manually. Based on these problems, a system is needed that can facilitate the making of schedules to minimize the occurrence of clashes between schedules. The application is built on a web-based basis by implementing the Artificial Bee Colony algorithm which is able to solve optimization problems such as scheduling. The test was carried out using 1195 data schedules. The test results using the colony size parameter of 6, limit of 5, and max iterations of 5 iterations produce an optimal solution with an average fitness value of 0.95. From the test results, it is shown that the optimal schedule produced has a total of 0 clashes and no schedule violates the hard constraints.</i>

I. PENDAHULUAN

Penjadwalan merupakan masalah yang umum terjadi pada institusi pendidikan seperti di perguruan tinggi maupun di sekolah. Kegiatan proses belajar mengajar antara guru dan siswa di sekolah mengacu kepada jadwal yang telah ditetapkan. Seringkali dalam kegiatan belajar mengajar tersebut terdapat permasalahan seperti terjadinya bentrok antara jadwal yang satu dan jadwal lainnya. Dalam pembuatan jadwal diperlukan ketelitian dengan mempertimbangkan berbagai faktor seperti pembagian tugas mengajar guru, jumlah ruangan yang tersedia, dan slot waktu agar tidak terjadi masalah jadwal yang saling

bentrok.

Madrasah Ibtidaiyah Negeri (MIN) 2 Pontianak beralamat di Jalan Husein Hamzah Kelurahan Pal Lima, Kecamatan Pontianak Barat, Kota Pontianak. Saat ini MIN 2 Pontianak memiliki 1267 siswa, 48 guru, 30 kelas, 25 ruang belajar dan 13 mata pelajaran. Kegiatan belajar mengajar di MIN 2 Pontianak dimulai pukul 06.45 WIB hingga pukul 15.30 WIB pada hari Senin sampai hari Jumat.

Untuk menunjang kelancaran kegiatan belajar mengajar diperlukan suatu jadwal. Saat ini penyusunan jadwal di Madrasah Ibtidaiyah Negeri (MIN) 2 Pontianak masih dilakukan secara konvensional atau manual sehingga membutuhkan waktu yang cukup lama karena memperhatikan

constraint yaitu aturan atau batasan yang harus dipenuhi dalam pembuatan jadwal. Setelah jadwal berhasil dibuat dan dijalankan selama beberapa hari sering kali masih terdapat masalah yang terjadi di lapangan seperti seorang guru yang memiliki jadwal mengajar di dua atau lebih ruangan kelas yang berbeda dalam satu waktu. Hal ini berdampak pada kegiatan belajar mengajar menjadi tidak tertib dan tidak efektif pada awal tahun ajaran baru. Jadwal yang telah dibuat tersebut harus direvisi dengan menukar masing-masing jadwal yang saling bentrok agar dihasilkan jadwal yang optimal. Berdasarkan permasalahan tersebut, *Swarm Intelligence* atau Kecerdasan Kelompok dapat menjadi solusi atas permasalahan optimasi seperti penjadwalan. Pada penelitian ini akan digunakan algoritma *Artificial Bee Colony* (ABC) yang merupakan salah satu bagian dari *Swarm Intelligence*.

Penelitian menggunakan algoritma ABC pernah dilakukan oleh Luthfina (2018) dengan judul "Implementasi Algoritma *Artificial Bee Colony* (ABC) Untuk Penjadwalan Mata Pelajaran Sekolah". Penelitian ini menggunakan parameter jumlah iterasi, jumlah populasi, dan jumlah onlooker. Uji coba pada penelitian ini dilakukan sebanyak 7 kali dengan kombinasi nilai masukan parameter yang berbeda-beda pada setiap pengujian. Hasil penelitian ini dapat mengurangi bentrok antar jadwal dengan rata-rata akurasi sebesar 97,189%.

Penelitian terkait juga pernah dilakukan oleh Fitriawati(2018) dengan judul "Analisis Dan Pemodelan Prototipe Sistem Informasi Jadwal Kuliah Dengan Algoritma ABC (*Artificial Bee Colony*)". Pada proses penjadwalan penelitian ini terdapat dua jenis pelanggaran atau bentrok yaitu bentrok DHJ (Dosen, Hari, Jam) dan bentrok RHJ (Ruang, Hari, Jam) sehingga perhitungan nilai *fitness*-nya adalah hasil penjumlahan dari banyaknya bentrok DHJ dan bentrok RHJ. Penelitian ini berhasil membuat model sistem informasi jadwal kuliah dengan menggunakan algoritma ABC dengan harapan dapat menjadi dasar untuk membangun sistem informasi jadwal kuliah.

Selanjutnya penelitian terkait dilakukan oleh Nugroho(2013) tentang Algoritma *Artificial Bee Colony* pada aplikasi penjadwalan pelajaran. Penelitian ini menghasilkan sebuah aplikasi yang dapat membuat jadwal pelajaran menggunakan metode Algoritma ABC untuk menghindari jumlah bentrokan antar kelas, waktu, dan guru. Pada penelitian ini terdapat dua macam metode pada tahap *Onlooker Bee* yaitu eksploitasi *random* dan *sequential*. Berdasarkan perbandingan hasil pengujian yang dilakukan menunjukkan bahwa pengujian eksploitasi dengan metode *sequential yang* mampu menghasilkan jadwal pelajaran

dengan *fitness* yang terbaik dari hasil pengujian eksploitasi secara *random* dengan parameter jumlah generasi atau perulangan yang lebih sedikit.

Penelitian mengenai penjadwalan juga dilakukan oleh Fadhli(2016) yang membahas pengoptimasian penjadwalan pekerjaan *software* pada *Software House* dengan *Flow-Shop Problem* menggunakan metode *Artificial Bee Colony*. Penelitian ini melakukan tiga kali percobaan terhadap parameter iterasi dan *limit*. Berdasarkan hasil pengujian terdapat penggunaan algoritma pada penelitian yang dapat mengurangi waktu pengerjaan dengan jumlah iterasi dan jumlah *colony yang* diperbesar.

Berdasarkan penelitian terkait, pada penelitian ini dibuat sebuah aplikasi penjadwalan menggunakan algoritma ABC berbasis web sebagai solusi untuk mempermudah dalam pembuatan jadwal mata pelajaran di MIN 2 Pontianak. Algoritma ABC digunakan karena mampu menyelesaikan permasalahan optimasi seperti penjadwalan.

II. BAHAN DAN METODE

Analisis Kebutuhan

Adapun data yang digunakan pada penelitian ini menggunakan data yang diperoleh dari Madrasah Ibtidaiyah Negeri 2 Pontianak. Data tersebut di antaranya adalah data guru, data mata pelajaran, data kelas, data jam pelajaran, data pembagian tugas mengajar, dan data pembagian slot yang digunakan di Madrasah Ibtidaiyah Negeri 2 Pontianak tahun pelajaran 2019/2020.

Setelah data tersebut diperoleh selanjutnya dilakukan analisis terhadap data tersebut mengenai batasan-batasan (*constraints*) atau aturan dalam pembuatan jadwal mata pelajaran di Madrasah Ibtidaiyah Negeri 2 Pontianak. *Constraint* pada penelitian ini dibagi menjadi dua macam yaitu *hard constraint* dan *soft constraint*. *Hard constraint* ialah batasan yang wajib dipenuhi, sedangkan *soft constraint* ialah batasan yang apabila dipenuhi akan lebih baik, namun jika tidak dapat dipenuhi tidak menjadi masalah.

Hard constraint dalam pembuatan jadwal pada penelitian ini dapat dijabarkan sebagai berikut.

1. Seorang guru tidak boleh dijadwalkan di dua atau lebih kelas yang berbeda pada hari dan jam yang sama.
2. Dua atau lebih dari satu guru tidak boleh dijadwalkan pada kelas, hari, dan jam yang sama.
3. Masing-masing mata pelajaran tidak boleh dijadwalkan melebihi dari kuota slot per minggu untuk satu kelas.

4. Kelas 2 menggunakan ruangan yang sama dengan kelas 1 sehingga jadwal kelas 2 harus dijadwalkan pada jam setelah kelas 1 pulang.

Soft constraint dalam pembuatan jadwal pada penelitian ini yaitu mata pelajaran yang hanya memiliki kuota slot 2 jam per minggu dijadwalkan secara berurutan dalam satu hari.

1.1. Perancangan Sistem

Perancangan yang dilakukan pada penelitian ini yaitu membuat rancangan sistem atau aplikasi yang dibangun untuk pembuatan jadwal mata pelajaran yang kemudian rancangan tersebut dijadikan sebagai dasar dalam implementasi sistem. Adapun Rancangan Sistem menggunakan algoritma ABC yang menjadi kecerdasan buatan dengan menirukan koloni lebah untuk mencari sari bunga (sumber nektar). Koloni lebah mampu untuk menentukan sumber makanan dengan masing-masing kemampuannya dibagi menjadi tiga kelompok yaitu lebah penjelajah, lebah pekerja, dan lebah pengintai. Lebah-lebah ini dapat membuat keputusan tentang posisi dan besar suatu sumber nektar, serta dapat mengingat dan dapat membandingkan dengan sumber nektar lainnya. Pada akhir fungsi dipilih suatu lokasi dengan sumber nektar yang paling optimal (Fadhila, 2016).

Berikut adalah langkah-langkah pada algoritma ABC (Cholissodin dan, Riyandani, 2016).

1. Inisialisasi parameter yang digunakan di antaranya:
 - a. *Colony Size*, yaitu jumlah dari keseluruhan lebah yang akan digunakan dalam sistem.
 - b. *Limit*, yaitu batasan untuk setiap solusi yang tidak ada peningkatan kualitas pada sejumlah iterasi.
 - c. *Max Iterations*, yaitu kriteria berhenti berdasarkan banyaknya iterasi yang ditetapkan.
2. Fase *Initial*
 - a. Pada tahap ini akan dilakukan proses inisialisasi solusi awal secara *random* untuk tiap *Employed Bee* kemudian menghitung nilai *fitness* untuk setiap solusi tersebut. Untuk menghitung nilai *fitness* digunakan Persamaan 1.

$$fitness_i = \frac{1}{1+f(x_i)} \quad (1)$$

Keterangan:

x_i = index solusi ke-*i*

- b. Semakin tinggi nilai *fitness* maka akan semakin baik solusi yang diperoleh. Nilai *fitness* untuk solusi terbaik adalah 1, artinya tidak ada slot jadwal yang bentrok.

3. Fase *Employed Bee*

- a. Melakukan *improvement solution* yaitu pembaruan terhadap solusi dari tiap *Employed Bee*. Pembaruan solusi tersebut dilakukan dengan menggunakan fungsi *neighborhood operator* yaitu *random swap operator*. *Random swap operator* dilakukan dengan cara menukar slot jadwal yang saling bentrok.
- b. Menghitung nilai *fitness* setiap solusi menggunakan Persamaan 1. Jika terdapat solusi baru lebih baik dari solusi sebelumnya maka solusi yang lama akan digantikan dengan solusi baru, jika tidak maka *trial* akan ditambahkan dengan 1.

4. Menghitung Probabilitas

Pada langkah ini masing-masing solusi akan dihitung probabilitasnya menggunakan Persamaan 2.

$$P_i = \frac{fitness(x_i)}{\sum_{k=1}^S fitness(x_k)} \quad (2)$$

Keterangan:

P_i = Probabilitas *Employed Bee* ke-*i*

S = Jumlah *Employed Bee*

5. Fase *Onlooker Bee*

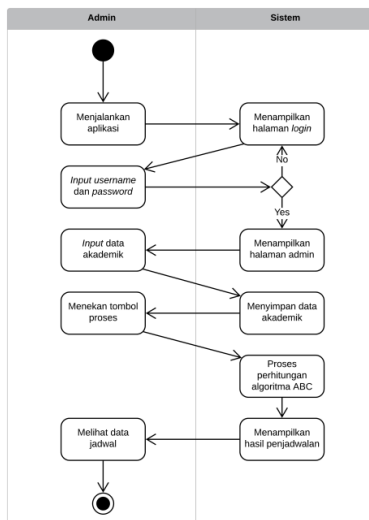
- a. Tiap *Onlooker Bee* akan memilih solusi berdasarkan nilai probabilitasnya dengan menggunakan teknik *roulette wheel selection*.
- b. Solusi yang telah terpilih tersebut akan dilakukan pembaruan kembali menggunakan *random swap operator* dengan cara menukar slot jadwal yang saling bentrok.
- c. Menghitung kembali nilai *fitness* setiap solusi yang telah diperbarui menggunakan Persamaan 1. Jika terdapat solusi baru lebih baik dari solusi sebelumnya maka solusi yang lama akan digantikan dengan solusi baru, jika tidak maka *trial* akan ditambahkan dengan 1.

6. Fase *Scout Bee*

- a. Menghitung jumlah *trial* untuk tiap *bee* yang tidak ada peningkatan solusi.

7. Membandingkan maksimum *trial* tersebut dengan parameter *limit* yang telah ditetapkan. Jika maksimum *trial* melebihi *limit* maka solusi yang tidak mengalami perbaikan akan diganti dengan solusi baru secara *random* seperti pada fase *initial* dan *trial* di-*reset* kembali menjadi 0. Simpan solusi terbaik yang didapatkan sementara dan mengulangi langkah 3 sampai kriteria berhenti. Apabila sampai batas *max iterations* tidak mendapatkan solusi terbaik atau nilai *fitness* tidak sama dengan 1 maka solusi pada iterasi tersebut akan disimpan sebagai solusi terbaik.

Perancangan sistem pada penelitian ini menggunakan pemodelan *Unified Modeling Language* (UML). Diagram UML yang digunakan dalam perancangan ini yaitu *activity diagram*. Rancangan *activity diagram* penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Rancangan Activity Diagram

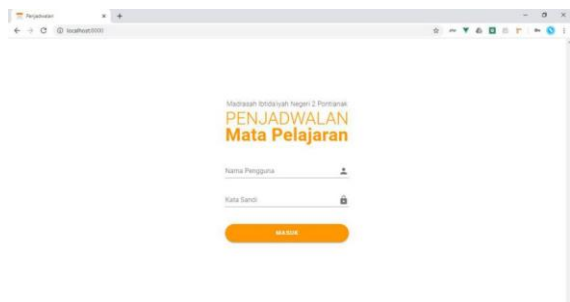
III. HASIL DAN PEMBAHASAN

Implementasi Aplikasi

Implementasi aplikasi pada penelitian ini merupakan tahap berikutnya setelah perancangan. Pada bagian ini akan dibahas mengenai implementasi dari aplikasi Penjadwalan Mata Pelajaran Pada Sekolah Dasar Menggunakan Algoritma *Artificial Bee Colony* Berbasis Web. Dalam aplikasi ini terdapat beberapa halaman menu dan submenu. Berikut adalah implementasi setiap halaman yang terdapat di dalam aplikasi.

1. Halaman Login

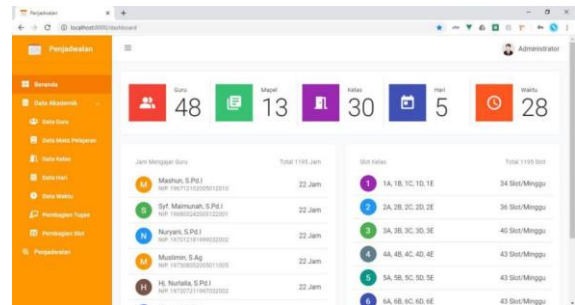
Halaman *login* adalah halaman awal yang ditampilkan pada saat pertama kali menjalankan aplikasi. Implementasi halaman *login* dapat dilihat pada Gambar 2.



Gambar 2. Halaman Login

2. Halaman Beranda

Halaman beranda adalah halaman yang pertama kali ditampilkan setelah admin berhasil *login* ke dalam aplikasi. Implementasi halaman beranda dapat dilihat pada Gambar 3.

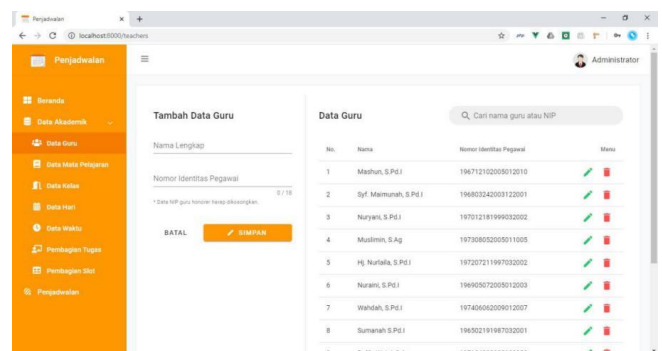


Gambar 3. Halaman Beranda

3. Halaman Data Akademik

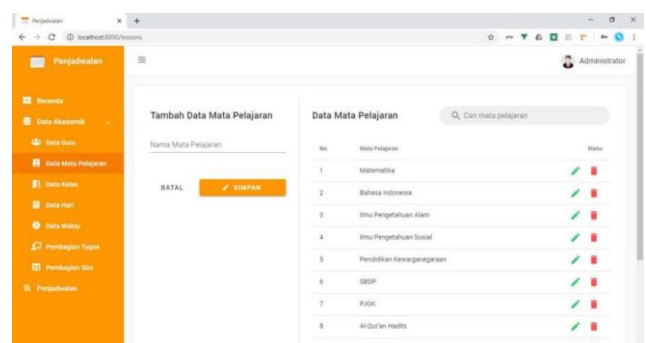
Data akademik terdiri dari data mata pelajaran, kelas, guru, hari, dan waktu, pembagian tugas mengajar, dan pembagian slot. Berikut implementasi setiap halaman pada data akademik.

a. Data Guru



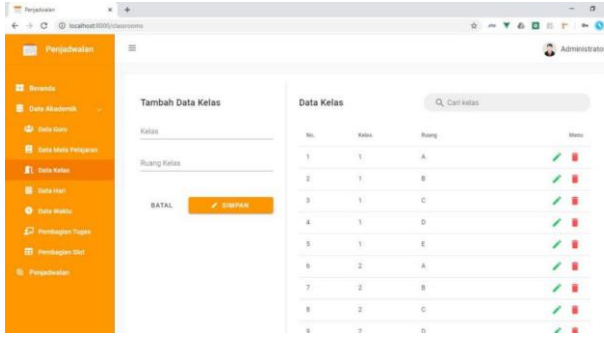
Gambar 4. Halaman Data Guru

b. Data Mata Pelajaran



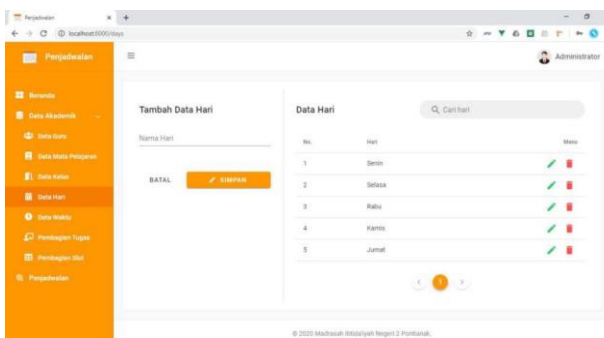
Gambar 5. Halaman Data Mata Pelajaran

c. Data Kelas



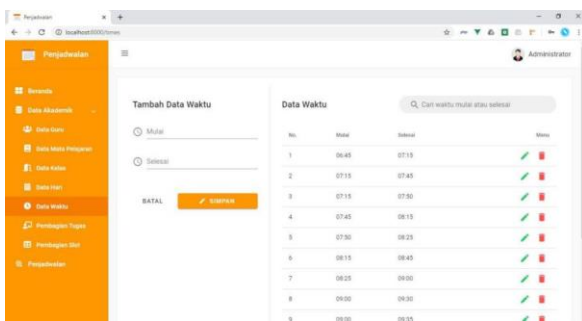
Gambar 6. Halaman Data Kelas

d. Data Hari



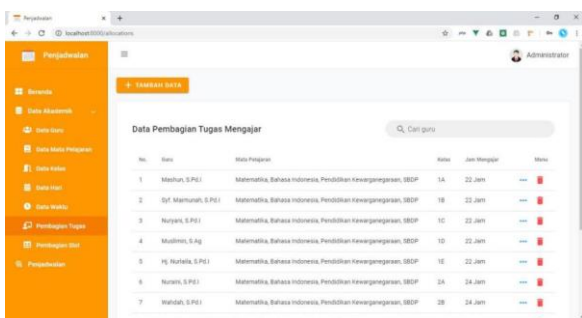
Gambar 7. Halaman Data Hari

e. Data Waktu



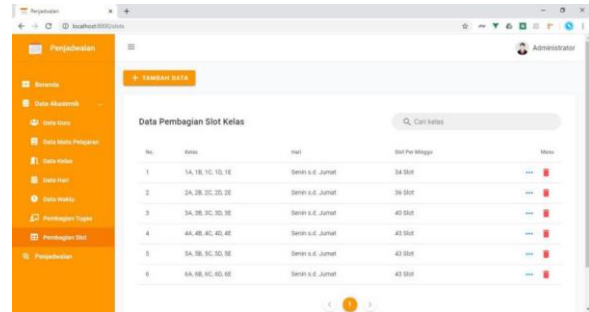
Gambar 8. Halaman Data Waktu

f. Data Pembagian Tugas Mengajar



Gambar 9. Halaman Data Pembagian Tugas Mengajar

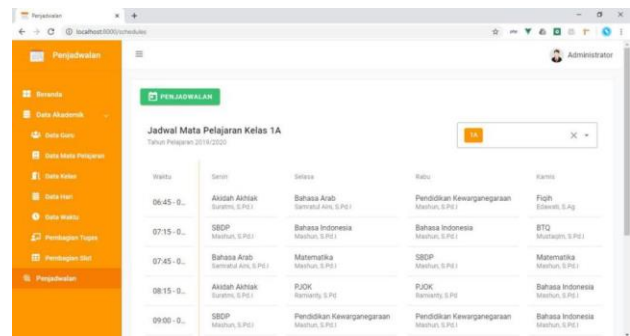
g. Data Pembagian Slot



Gambar 10. Halaman Data Pembagian Slot

4. Halaman Penjadwalan

Halaman penjadwalan merupakan halaman untuk memproses penjadwalan yang dilakukan oleh sistem berdasarkan parameter algoritma *Artificial Bee Colony* yang telah ditetapkan. Parameter tersebut di antaranya *colony size*, *limit*, dan *max iterations*. Nilai dari parameter yang akan diproses ditetapkan berdasarkan hasil pengujian yang memiliki *output* optimal. Pada halaman ini admin dapat melihat data jadwal yang telah diproses. Implementasi halaman penjadwalan dapat dilihat pada Gambar 11.



Gambar 11. Halaman Penjadwalan

1.2. Pengujian

Pada bagian ini dilakukan pengujian terhadap algoritma *Artificial Bee Colony*. Pengujian ini bertujuan untuk mengetahui apakah algoritma *Artificial Bee Colony* yang diimplementasikan pada sistem penjadwalan mata pelajaran dapat berjalan sesuai dengan tujuan dan menghasilkan keluaran yang diinginkan yaitu jadwal mata pelajaran yang optimal dengan memenuhi batasan atau *constraint* yang ditetapkan.

Data masukan yang digunakan pada pengujian ini adalah data akademik yang diperoleh dari MIN 2 Pontianak. Data akademik tersebut berupa data guru yang berjumlah 48 orang, 13 data

mata pelajaran, 30 data kelas, 5 data hari, 28 data waktu, 1195 data pembagian tugas mengajar dan data pembagian slot yang berjumlah sama dengan data pembagian tugas mengajar yaitu sebanyak 1195 slot.

Pengujian yang dilakukan yaitu dengan menggunakan nilai parameter *colony size* yang berbeda-beda yaitu 2, 4, dan 6. Masing-masing nilai *colony size* tersebut akan diuji sebanyak 10 kali kemudian akan dihitung rata-rata nilai *fitness* yang dihasilkan. Untuk nilai parameter *limit* yang digunakan yaitu sebesar 5, sedangkan jumlah *max iterations* yang digunakan yaitu sebanyak 5 iterasi berdasarkan analisis terhadap uji coba jumlah iterasi dari 2 sampai dengan 6 iterasi mendapatkan hasil *fitness* yang cukup baik saat menggunakan *max iterations* sebanyak 5 iterasi.

Pada pengujian yang pertama yaitu dengan menggunakan nilai *colony size* sebesar 2 yang diuji sebanyak 10 kali kemudian menghitung rata-rata nilai *fitness* yang dihasilkan. Hasil pengujian nilai *colony size* sebesar 2 dapat dilihat pada Tabel 1.

Tabel 1. Hasil Pengujian *Colony Size* Sebesar 2

No.	Solusi Terbaik	Jumlah Bentrok	Fitness
1	Bee ke 2	1	0.5
2	Bee ke 1	0	1
3	Bee ke 2	1	0.5
4	Bee ke 1	0	1
5	Bee ke 1	0	1
6	Bee ke 1	2	0.33
7	Bee ke 2	2	0.33
8	Bee ke 1	2	0.33
9	Bee ke 1	4	0.2
10	Bee ke 1	2	0.3
Rata-rata			0.553

Sumber : Fajrianto (2021)

Pada Tabel 1 dapat dilihat setelah nilai *colony size* sebesar 2 diuji sebanyak 10 kali maka didapatkan rata-rata nilai *fitness* sebesar 0.553333333. Selanjutnya dilakukan pengujian sebanyak 10 kali menggunakan nilai *colony size* sebesar 4 kemudian menghitung rata-rata *fitness* yang dihasilkan. Hasil pengujian nilai *colony size* sebesar 4 dapat dilihat pada Tabel 2.

Tabel 2. Hasil Pengujian *Colony Size* Sebesar 4

No.	Solusi Terbaik	Jumlah Bentrok	Fitness
1	Bee ke 1	1	0.5

2	Bee ke 4	1	0.5
3	Bee ke 1	2	0.33
4	Bee ke 1	1	0.5
5	Bee ke 2	1	0.5
6	Bee ke 1	1	0.5
7	Bee ke 4	1	0.5
8	Bee ke 3	0	1
9	Bee ke 1	1	0.5
10	Bee ke 4	0	1
Rata-rata			0.583

Sumber : Fajrianto (2021)

Pada Tabel 2 dapat dilihat setelah nilai *colony size* sebesar 4 diuji sebanyak 10 kali maka didapatkan rata-rata nilai *fitness* sebesar 0.583333333. Selanjutnya dilakukan pengujian sebanyak 10 kali menggunakan nilai *colony size* sebesar 6 kemudian menghitung rata-rata *fitness* yang dihasilkan. Hasil pengujian nilai *colony size* sebesar 6 dapat dilihat pada Tabel 3.

Tabel 3. Hasil Pengujian *Colony Size* Sebesar 6

No.	Solusi Terbaik	Jumlah Bentrok	Fitness
1	Bee ke 5	0	1
2	Bee ke 1	0	1
3	Bee ke 2	0	1
4	Bee ke 1	1	0.5
5	Bee ke 2	0	1
6	Bee ke 1	0	1
7	Bee ke 3	0	1
8	Bee ke 1	0	1
9	Bee ke 1	0	1
10	Bee ke 2	0	1
Rata-rata			0.95

Sumber : Fajrianto (2021)

Pada Tabel 3 dapat dilihat setelah nilai *colony size* sebesar 6 diuji sebanyak 10 kali maka didapatkan rata-rata nilai *fitness* sebesar 0.95.

Berdasarkan pengujian yang telah dilakukan dengan menggunakan nilai parameter *colony size* yang berbeda-beda yaitu sebesar 2, 4, dan 6 yang diuji masing-masing sebanyak 10 kali menghasilkan rata-rata nilai *fitness* untuk *colony size* 2 sebesar 0.553, untuk *colony size* 4 menghasilkan rata-rata nilai *fitness* sebesar 0.583, dan untuk *colony size* 6 menghasilkan rata-rata nilai *fitness* sebesar 0.95. Dari ketiga rata-rata *fitness* tersebut dapat dilihat bahwa nilai rata-rata *fitness* tertinggi adalah 0.95 yang dihasilkan

menggunakan parameter *colony size* bernilai 6. Berdasarkan hasil tersebut, maka semakin banyak jumlah *colony size* maka akan semakin optimal jadwal yang dihasilkan.

IV. KESIMPULAN

Berdasarkan hasil pengujian aplikasi penjadwalan mata pelajaran pada sekolah dasar dengan menggunakan algoritma Artificial Bee Colony diambil kesimpulan bahwa Algoritma Artificial Bee Colony dapat diimplementasikan pada aplikasi penjadwalan mata pelajaran sekolah dasar berbasis web yang diuji menggunakan data sebanyak 1195 slot jadwal dengan menggunakan parameter colony size sebesar 6, limit sebesar 5, dan max iterations sebanyak 5 iterasi yang menghasilkan solusi paling optimal dengan rata-rata nilai fitness sebesar 0,95. Sedangkan berdasarkan pengujian yang dilakukan, hasil jadwal yang paling optimal memiliki jumlah bentrok sebanyak 0 jadwal dan tidak ada jadwal yang melanggar hard constraints.

V. REFERENSI

Cholissodin, I., Riyandani, E. (2016). *Swarm Intelligence*, Malang : Penerbit : Fakultas Ilmu Komputer. Universitas Brawijaya.

Fadila, J. N. (2016). Persebaran Multi NPC Menggunakan Algoritma Bee Colony Untuk Game 3D Space Shooter. *Tesis*. FAKULTAS TEKNOLOGI INDUSTRI. INSTITUT TEKNOLOGI SEPULUH SEPTEMBER, SURABAYA.

Fhadli, M. (2016). Optimasi Penjadwalan Pengerjaan Software pada Software House dengan Flow-shop Problem Menggunakan Artificial Bee Colony. *Skripsi*. FAKULTAS ILMU KOMPUTER, UNIVERSITAS BRAWIJAYA. MALANG.

Fitriawati, M. (2018). Analisis Dan Pemodelan Prototipe Sistem Informasi Jadwal Kuliah Dengan Algoritma ABC (Artificial Bee Colony). *Skripsi*. PROGRAM STUDI SISTEM INFORMASI. UNIVERSITAS KOMPUTER INDONESIA BANDUNG.

Luthfina, L. (2018). Implementasi Algoritma Artificial Bee Colony (ABC) Untuk Penjadwalan Mata Pelajaran Sekolah. *Skripsi*. FAKULTAS SAINS DAN TEKNOLOGI. UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG.

Nugroho, R. F. (2013). Penerapan Algoritma Artificial Bee Colony dalam Aplikasi Penjadwalan Pelajaran untuk Sekolah Menengah Pertama.

Skripsi. FAKULTAS TEKNOLOGI INFORMASI, UNIVERSITAS KRISTEN MARANATHA BANDUNG.