

## Sistem Penjadwalan Karyawan dengan Algoritma Genetika

Maria Karmelia Fajarlestari<sup>1</sup>, Mawar Hardiyanti<sup>2</sup>

Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Pignatelli Triputra  
Jl. Duwet Raya No. 1 Karangasem, Laweyan, Kota Surakarta 57145, Jawa Tengah, Indonesia  
Email: <sup>1</sup>mariakarmelia@upitra.ac.id, <sup>2</sup>mawar@upitra.ac.id

**Abstract. Employee Scheduling System Using Genetic Algorithm.** Employee scheduling is a complex problem in Human Resource Management (HRM) that significantly impacts operational efficiency. This study develops an employee scheduling system using a genetic algorithm. The employee schedules are constructed by considering scheduling rules and various components such as the number of days, shifts, employee quality, and scheduling requests. The genetic algorithm, proven effective in solving various optimization problems, is employed to generate optimal schedules through the processes of selection, crossover, and mutation. The results indicate that the genetic algorithm can effectively produce employee schedules, with fitness values indicating improved schedule quality as iterations increase. The findings of this study are anticipated to be useful in HRM, aiming to improve both employee efficiency and satisfaction.

**Keywords:** employee scheduling, genetic algorithm, human resource management

**Abstrak.** Penjadwalan karyawan merupakan masalah kompleks dalam manajemen Sumber Daya Manusia (SDM) yang sangat berpengaruh pada efisiensi operasional. Penelitian ini mengembangkan sistem penjadwalan karyawan menggunakan algoritma genetika. Jadwal karyawan dibangun dengan mempertimbangkan aturan-aturan penjadwalan karyawan dan berbagai komponen penjadwalan seperti jumlah hari, sif, kualitas karyawan, dan permintaan jadwal karyawan. Algoritma genetika yang telah terbukti efektif dalam memecahkan berbagai masalah optimasi digunakan untuk menghasilkan jadwal optimal melalui proses seleksi, crossover, dan mutasi. Penelitian ini menunjukkan bahwa algoritma genetika berhasil menciptakan jadwal karyawan, dengan nilai fitness yang menunjukkan peningkatan kualitas jadwal seiring bertambahnya iterasi. Hasil penelitian ini diharapkan dapat diaplikasikan dalam manajemen SDM untuk meningkatkan efisiensi dan kepuasan karyawan.

**Kata Kunci:** algoritma genetika, manajemen SDM, penjadwalan karyawan

### 1. Pendahuluan

Penjadwalan karyawan merupakan salah satu aspek penting dalam manajemen sumber daya manusia yang sangat berdampak pada operasional [1]. Dalam lingkungan bisnis terutama yang mempekerjakan karyawan paruh waktu, proses penjadwalan menjadi lebih kompleks karena adanya variabel penjadwalan terkait ketersediaan waktu kerja, kebutuhan operasional, dan kebutuhan individu karyawan [2].

Dalam praktiknya, metode penjadwalan karyawan yang umum digunakan meliputi penjadwalan manual, heuristik, dan algoritma deterministik. Metode manual sering kali dianggap tidak efisien karena memakan banyak waktu dan rentan terhadap kesalahan manusia [3]. Selain itu, metode heuristik, seperti metode *First Come First Served* (FCFS) atau aturan prioritas lain, meskipun lebih cepat dibandingkan metode manual, memiliki keterbatasan dalam menangani masalah skala besar dengan kompleksitas tinggi karena tidak dirancang untuk mengoptimalkan semua variabel dalam penjadwalan [4]. Algoritma deterministik, seperti algoritma pemrograman linier, meski dapat menghasilkan solusi optimal pada masalah yang lebih sederhana, mengalami kesulitan saat menghadapi masalah penjadwalan dengan kendala yang dinamis dan variabel yang kompleks, seperti preferensi karyawan [5].

Oleh karena itu, pendekatan algoritma optimasi seperti algoritma genetika semakin banyak dipertimbangkan dalam proses penjadwalan modern. Algoritma genetika, sebuah metode optimasi yang meniru proses seleksi alam, menunjukkan keunggulan dalam menangani masalah penjadwalan yang kompleks dan multi-dimensional [6]. Algoritma ini bekerja dengan

menghasilkan solusi baru melalui proses seleksi, *crossover*, dan mutasi, sehingga memungkinkan eksplorasi yang lebih luas dalam ruang solusi dan meningkatkan peluang menemukan solusi optimal dalam waktu yang relatif singkat dibandingkan metode lainnya [7]. Dalam konteks penjadwalan karyawan, fleksibilitas algoritma genetika dalam menangani berbagai kendala dan tujuan menjadikannya sebagai pendekatan yang potensial untuk menghasilkan solusi penjadwalan yang lebih efisien dan memenuhi preferensi karyawan.

Beberapa penelitian terdahulu juga menunjukkan bahwa algoritma genetika efektif dalam menghasilkan solusi optimal untuk masalah penjadwalan yang kompleks, terutama dalam konteks yang membutuhkan penyesuaian dengan kebutuhan karyawan dan organisasi secara bersamaan [8]. Penelitian ini tidak hanya bertujuan untuk mengembangkan sistem penjadwalan karyawan berbasis algoritma genetika, masih terdapat beberapa pertanyaan penting yang perlu dijawab untuk memastikan efektivitasnya dalam konteks penjadwalan karyawan. Penelitian ini juga akan menganalisis sifat-sifat operator dalam algoritma dalam menghasilkan solusi yang optimal.

## 2. Tinjauan Pustaka

### 2.1. Algoritma Genetika

Algoritma genetika adalah metode optimasi yang didasarkan pada proses seleksi alam dan genetika, menggunakan konsep individu, populasi, seleksi, *crossover*, dan mutasi untuk mencari solusi paling baik dan optimal dalam ruang solusi yang luas [9]. Prinsip dasar teori evolusi, seperti seleksi alam dan adaptasi, diterapkan dalam algoritma genetika untuk mengembangkan solusi yang lebih baik seiring waktu [10]. Operator seleksi, *crossover*, dan mutasi adalah komponen kunci yang mempengaruhi kinerja algoritma genetika [11]. Penelitian terbaru mengeksplorasi berbagai konfigurasi operator ini untuk meningkatkan kinerja algoritma genetika, menemukan bahwa kombinasi tertentu lebih konsisten dalam menghasilkan solusi optimal [12]. Algoritma genetika telah terbukti sebagai metode yang sesuai untuk menangani masalah optimasi yang kompleks [13], termasuk kemampuan mengelola masalah penjadwalan karyawan [14]. Evaluasi kinerja algoritma genetika menunjukkan bahwa algoritma genetika dapat bersaing bahkan melampaui metode optimasi lainnya dalam konteks penjadwalan yang kompleks [15].

### 2.2. Penjadwalan Karyawan

Penelitian sebelumnya telah menunjukkan bahwa penjadwalan karyawan yang efisien merupakan kunci untuk meningkatkan produktivitas dan mengurangi biaya operasional [16]. Hal ini menghadirkan tantangan tersendiri karena banyaknya variabel penjadwalan termasuk kepentingan individu karyawan. Pendekatan yang lebih fleksibel dan adaptif diperlukan untuk mengatasi tantangan ini [17]. Selain itu, teori kepuasan karyawan menunjukkan pentingnya faktor-faktor seperti kesesuaian jumlah jam dalam distribusi jam kerja dan kesesuaian terhadap *request* karyawan, yang berkontribusi terhadap kepuasan kerja [18].

### 2.3. Algoritma Alternatif

Beberapa algoritma optimasi lain juga memiliki potensi dalam penjadwalan karyawan, meskipun setiap metode memiliki pendekatan yang berbeda dalam mencapai solusi optimal. *Simulated Annealing* (SA), algoritma ini bekerja dengan prinsip penurunan suhu secara bertahap, menyerupai proses pemadatan logam. SA sering digunakan dalam penjadwalan karena kemampuannya untuk keluar dari solusi lokal menuju solusi yang lebih baik melalui pencarian acak yang terkontrol [19]. Namun, dibandingkan AG, SA lebih lambat dalam konvergensi dan sering kali tidak efektif untuk masalah yang melibatkan kendala multi-dimensi, seperti penjadwalan karyawan dengan banyak variabel preferensi [20].

*Particle Swarm Optimization* (PSO) meniru perilaku sosial burung dalam menemukan solusi optimal melalui partikel yang saling berbagi informasi [21]. Algoritma ini cepat dalam konvergensi pada masalah optimasi yang lebih sederhana, namun kadang-kadang terjebak dalam solusi lokal ketika diterapkan pada permasalahan penjadwalan yang kompleks. PSO umumnya kurang fleksibel dalam menangani kendala multi-dimensi yang dinamis dan sering kali membutuhkan modifikasi tambahan agar sesuai untuk masalah penjadwalan karyawan [22].

*Ant Colony Optimization* (ACO) meniru perilaku koloni semut dalam mencari jalur terpendek, yang cocok untuk masalah penjadwalan rute dan penugasan [22]. ACO bekerja dengan menciptakan dan memperkuat jalur optimal melalui umpan balik positif, namun pada kasus penjadwalan karyawan yang melibatkan variabel waktu kerja dan preferensi individu, ACO seringkali kurang efisien dan cenderung membutuhkan waktu lebih lama untuk mencapai solusi optimal [23].

Dalam konteks perbandingan tersebut, algoritma genetika menonjol karena fleksibilitasnya dalam menangani kendala dan variabel yang kompleks pada penjadwalan karyawan. Studi-studi terdahulu yang mengkaji efektivitas AG dalam konteks penjadwalan, seperti dalam manufaktur dan layanan, menunjukkan bahwa AG mampu menghasilkan solusi yang lebih adaptif dan sesuai dengan preferensi karyawan serta kebutuhan operasional dibandingkan metode lain.

### 3. Metodologi Penelitian

#### 3.1. Pengumpulan Data

Dalam penelitian ini, kami akan menganalisis data dari sebuah kedai kopi yang beroperasi 24 jam di Yogyakarta. Fokus utama adalah merancang sistem penjadwalan yang efisien untuk karyawan paruh waktu. Beberapa variabel yang akan dipertimbangkan meliputi durasi periode penjadwalan, jumlah sif per hari di setiap lokasi, jumlah total karyawan, dan kualitas kinerja masing-masing karyawan. Selain itu, kami juga akan memperhitungkan jadwal kerja sebelumnya sebagai salah satu input dalam algoritma penjadwalan. Untuk mengakomodasi preferensi karyawan, sistem ini juga akan mempertimbangkan permintaan jadwal yang diajukan, meskipun tidak semua permintaan dapat dipenuhi sepenuhnya. Tujuan akhir dari penelitian ini adalah menyusun jadwal kerja yang efektif, dengan meminimalkan konflik jadwal.

#### 3.2. Perancangan Algoritma Genetika

Pada tahap ini, algoritma genetika akan dirancang khusus untuk masalah penjadwalan karyawan. Tahapan ini meliputi: (1) Representasi masalah: menentukan representasi kromosom yang sesuai untuk jadwal karyawan. (2) Inisialisasi populasi: membuat populasi awal dari solusi, yaitu jadwal karyawan. (3) *Fitness function*: mengembangkan *fitness function* yang mengukur kualitas setiap solusi berdasarkan kriteria seperti kebutuhan operasional dan aturan-aturan penjadwalan. (4) Operator seleksi: memilih individu untuk reproduksi berdasarkan nilai *fitness*. (5) Operator *crossover*: mengombinasikan dua individu untuk menghasilkan keturunan baru. (6) Operator mutasi: memodifikasi individu untuk menjadi individu baru dalam populasi.

#### 3.3. Representasi Masalah

Setiap kemungkinan jadwal kerja digambarkan sebagai sebuah individu atau kromosom. Panjang kromosom ini ditentukan oleh perkalian antara jumlah sif yang tersedia dalam sehari dengan jumlah hari yang akan dijadwalkan. Dengan kata lain, setiap gen dalam kromosom mewakili satu keputusan penjadwalan. Misalnya, dalam kasus penjadwalan empat shift selama 31 hari, kromosom akan memiliki panjang 124 gen, yang dapat dihitung menggunakan Persamaan 1.

$$\text{jumlah gen} = \text{jumlah sif dalam satu hari} \times \text{jumlah hari} \quad (1)$$

#### 3.4. Pembentukan Kromosom dan Pemodelan Kromosom

| HARI 1 |     |     |     | HARI 2 |     |     |     | • • • | HARI 31 |     |     |     |
|--------|-----|-----|-----|--------|-----|-----|-----|-------|---------|-----|-----|-----|
| KRM    | JAP | PAE | STY | NLC    | JAP | STY | HNK |       | STY     | JAP | PAE | AFR |
| S1     | S2  | S3  | S4  | S1     | S2  | S3  | S4  |       | S1      | S2  | S3  | S4  |

Gambar 1. Model Kromosom

Kromosom awal dihasilkan secara acak untuk memetakan setiap karyawan ke suatu sif dan hari kerja tertentu. Representasi kromosom ini divisualisasikan pada Gambar 1. Pada kotak ke- $n$  mempunyai variabel hari, sif dan karyawan. Gen pertama menjelaskan bahwa pada hari pertama kode karyawan KRM berada pada sif pertama. Begitu seterusnya sampai dengan gen terakhir. Gen terakhir menjelaskan bahwa karyawan dengan kode AFR berada di sif empat pada hari ke-31.

### 3.5. Pembangkitan Kromosom

Populasi awal algoritma genetika terdiri dari 10 individu, di mana setiap individu direpresentasikan sebagai sebuah *string* biner sepanjang 124 bit. Setiap bit dalam *string* ini merepresentasikan satu gen yang mengodekan informasi tentang penugasan karyawan pada sif tertentu di hari tertentu.

### 3.6. Pencarian Nilai *Fitness*

Kualitas suatu jadwal kerja (atau kromosom) ditentukan oleh seberapa baik jadwal tersebut memenuhi semua aturan yang telah ditetapkan pada Tabel 1. Semakin sedikit pelanggaran terhadap aturan, maka nilai kualitas (*fitness*) jadwal tersebut semakin tinggi. Dalam penelitian ini, dilakukan modifikasi terhadap nilai *fitness* sehingga nilai yang lebih tinggi menunjukkan jadwal yang lebih baik, yang dirumuskan pada Persamaan 2. Nilai *fitness'* atau nilai *fitness* akhir suatu jadwal dihitung berdasarkan perbandingan antara total pelanggaran jadwal tersebut dengan total pelanggaran seluruh jadwal dalam populasi. Nilai *fitness* adalah nilai setiap pelanggaran yang terjadi pada proses pembuatan jadwal, setiap terjadi suatu pelanggaran maka nilai *fitness* akan bertambah satu. Total nilai *fitness* adalah akumulasi nilai *fitness* pada suatu jadwal karyawan.

$$\text{nilai fitness}' = \frac{\text{total nilai fitness}}{\text{nilai fitness}} \quad (2)$$

**Tabel 1. Aturan-aturan dalam Pembuatan Jadwal Karyawan**

| No | Aturan                                                                                                                                     | Pelanggaran                                                                                                                                                      |
|----|--------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | Seorang pegawai tidak dapat ditugaskan pada lebih dari satu jadwal kerja dalam sehari.                                                     | Ada pegawai yang ditugaskan lebih dari satu jadwal kerja dalam sehari, maka nilai <i>fitness</i> bertambah 1.                                                    |
| 2  | Pegawai dengan peringkat kualitas tertinggi (sangat baik) tidak dapat bertugas secara bersamaan dalam suatu sif.                           | Ada pegawai dengan peringkat kualitas tertinggi (sangat baik) bertugas secara bersamaan dalam suatu sif, maka nilai <i>fitness</i> bertambah 1.                  |
| 3  | Pegawai yang menyelesaikan tugas pada akhir hari (sif terakhir) tidak dapat memulai tugas pada awal hari (sif pertama) di hari berikutnya. | Ada pegawai menyelesaikan tugas pada sif terakhir kemudian memulai tugas pada awal hari (sif pertama) di hari berikutnya, maka nilai <i>fitness</i> bertambah 1. |
| 4  | Jumlah sif yang dimiliki setiap pegawai dalam sebulan berkisar antara 14 hingga 16 kali.                                                   | Ada pegawai yang memiliki sif tidak berkisar antara 14 sampai 16 kali dalam sebulan, maka nilai <i>fitness</i> bertambah 1.                                      |
| 5  | Jadwal kerja yang diberikan kepada pegawai sesuai dengan preferensi yang telah diajukan sebelumnya (opsional).                             | Jadwal kerja yang diberikan kepada pegawai tidak sesuai dengan preferensi yang telah diajukan sebelumnya (opsional), maka nilai <i>fitness</i> bertambah 1.      |

Setiap jadwal kerja (kromosom) dievaluasi berdasarkan seberapa baik memenuhi semua aturan yang telah ditentukan. Awalnya, setiap jadwal diberikan skor nol. Kemudian, skor ini disesuaikan sehingga jadwal yang paling sesuai dengan aturan mendapatkan bobot yang lebih besar dalam proses seleksi. Jadwal dengan skor tertinggi akan memiliki peluang lebih kecil untuk dipilih, sedangkan jadwal dengan skor terendah memiliki peluang lebih besar. Hasil evaluasi untuk 10 jadwal awal dapat dilihat pada Tabel 2.

**Tabel 2. Hasil Nilai *Fitness* dan Hasil Konversi Nilai *Fitness* Masing-masing Kromosom**

| Kromosom              | 1    | 2    | 3    | 4     | 5     | 6     | 7   | 8    | 9     | 10   |        |
|-----------------------|------|------|------|-------|-------|-------|-----|------|-------|------|--------|
| Nilai <i>Fitness</i>  | 120  | 124  | 110  | 69    | 98    | 75    | 100 | 104  | 85    | 115  |        |
| Nilai <i>Fitness'</i> | 8.33 | 8.06 | 9.09 | 14.49 | 10.20 | 13.33 | 10  | 9.62 | 11.76 | 8.70 | 103.59 |

### 3.7. Seleksi Kromosom

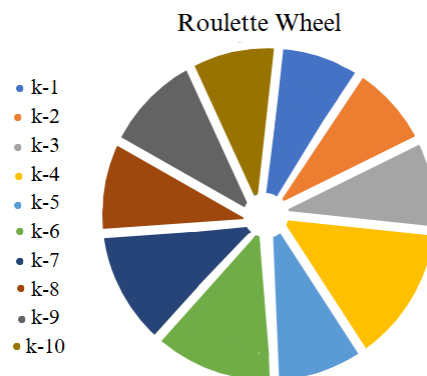
Proses seleksi bertujuan memilih jadwal kerja terbaik untuk menjadi dasar generasi berikutnya. Seleksi individu dalam penelitian ini dilakukan menggunakan metode roda *roulette*. Semakin baik kualitas sebuah jadwal, semakin besar bagiannya pada roda. Jadwal dengan bagian yang lebih besar akan lebih sering terpilih untuk menjadi induk dan menghasilkan jadwal-jadwal baru melalui proses persilangan dan mutasi. Probabilitas seleksi suatu individu berbanding lurus dengan nilai *fitness*. Individu dengan nilai *fitness* lebih tinggi memiliki kemungkinan lebih besar untuk dipilih sebagai induk dalam proses reproduksi, yang mencakup tahapan persilangan dan mutasi. Persamaan untuk menghitung probabilitas seleksi ( $P$ ) dapat dihitung dengan Persamaan 3.

$$P = \frac{t}{s} \times 100 \quad (3)$$

Probabilitas yang telah dihitung untuk setiap individu ditampilkan pada Tabel 3 akan digunakan untuk menentukan peluang mereka terpilih dalam proses seleksi. Semakin tinggi probabilitas suatu individu, semakin besar kemungkinan akan dipilih. Besar probabilitas ini akan divisualisasikan dalam bentuk diagram lingkaran (roda *roulette*) pada Gambar 2. Ukuran setiap sektor pada diagram lingkaran sebanding dengan probabilitas seleksi individu tersebut. Proses seleksi dilakukan dengan memilih secara acak satu titik pada lingkaran. Individu yang berada pada sektor yang terpilih akan dipilih untuk menjadi induk. Untuk memilih jadwal yang akan menjadi induk, proses seleksi dilakukan dengan memutar roda dan melihat jadwal mana yang terpilih. Sebagai contoh roda berhenti pada k-7 maka kromosom 7 yang terpilih untuk digunakan dalam proses mutasi atau *crossover*.

**Tabel 3. Probabilitas Total Nilai *Fitness* Masing-masing Kromosom**

| Kromosom         | 1    | 2    | 3    | 4     | 5    | 6     | 7    | 8    | 9     | 10   |
|------------------|------|------|------|-------|------|-------|------|------|-------|------|
| Probabilitas (%) | 8.04 | 7.79 | 8.78 | 13.99 | 9.85 | 12.87 | 9.65 | 9.28 | 11.36 | 8.39 |



**Gambar 2. Roda Roulette Probabilitas Total Nilai *Fitness* Tiap Kromosom**

### 3.8. Proses Reproduksi

Untuk menghasilkan jadwal kerja baru yang lebih baik, dilakukan proses reproduksi, yaitu persilangan dan mutasi. Persilangan adalah menggabungkan dua jadwal yang sudah ada menjadi jadwal baru. Mutasi adalah mengubah sedikit bagian dari sebuah jadwal. Kita akan melakukan persilangan pada setiap tahap, kecuali jika ada jadwal yang perlu diubah secara acak (mutasi). Jumlah dan waktu terjadinya mutasi ditentukan berdasarkan peluang yang telah ditentukan sebelumnya dengan Persamaan 4 dan Persamaan 5. Setelah proses persilangan dan mutasi, akan diperoleh jadwal-jadwal baru yang diharapkan lebih baik dari jadwal sebelumnya.

$$\text{jumlah mutasi} = \frac{\text{probabilitas mutasi}}{100} \times \text{jumlah iterasi} \quad (4)$$

$$\text{iterasi mutasi} = \frac{\text{jumlah iterasi}}{\text{jumlah mutasi}} \quad (5)$$

### 3.9. Pembaruan Populasi

Proses pembaruan populasi bertujuan untuk menjaga individu-individu berkualitas tinggi dan menyingkirkan yang berkualitas rendah. Setelah reproduksi, individu baru dibandingkan dengan individu berkualitas terendah dari populasi sebelumnya. Apabila individu baru memiliki nilai *fitness* lebih tinggi, individu yang mempunyai kualitas terendah akan digantikan. Proses ini terus berulang secara iteratif hingga kriteria penghentian tercapai.

### 3.10. Kondisi Berhenti

Proses algoritma genetika akan berhenti ketika sudah melakukan iterasi sebanyak inputan awal iterasi yang sudah ditentukan. Jumlah iterasi bisa ditentukan sebelumnya sebelum proses penjadwalan dilakukan. Selain itu proses algoritma genetika akan berhenti ketika menghasilkan hasil optimal yaitu nilai *fitness* = 0 atau tidak ada pelanggaran yang terjadi pada jadwal karyawan yang sudah terbentuk.

## 4. Hasil dan Diskusi

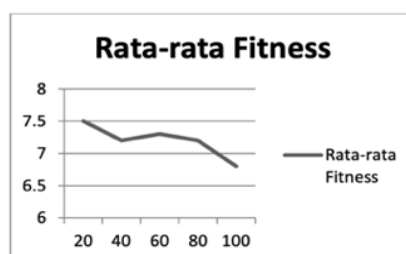
Pengujian dilakukan untuk mengevaluasi kinerja algoritma genetika dalam menyelesaikan masalah penjadwalan karyawan. Dalam proses penjadwalan, digunakan 10 individu dalam satu populasi. Parameter-parameter lain yang digunakan dalam proses penjadwalan ini dibuat fleksibel, sistem yang dikembangkan dapat menentukan jumlah iterasi, tipe *crossover*, serta tingkat probabilitas mutasi yang akan diterapkan sebelum memulai proses penjadwalan.

### 4.1. Pengaruh Jumlah Iterasi terhadap Hasil Penjadwalan

Program diuji sepuluh kali, *crossover* yang digunakan dalam pengujian ini adalah *crossover* empat titik. Besar probabilitas mutasi yang digunakan dalam pengujian adalah 5% dan aturan permintaan jadwal karyawan tidak digunakan dalam pengujian ini. Tabel 4 menunjukkan hubungan antara jumlah iterasi dan nilai kekuatan yang dihasilkan. Fokus utama adalah mengevaluasi pengaruh jumlah iterasi terhadap nilai kekuatan solusi yang dihasilkan. Gambar 3 menunjukkan hasil dari sepuluh percobaan dengan parameter *crossover* empat titik dan probabilitas mutasi 5%.

**Tabel 4. Hasil Nilai *Fitness* Berdasarkan Jumlah Iterasi**

| Jumlah Iterasi | Nilai <i>Fitness</i> Terbaik (Percobaan ke-n) |   |   |    |   |   |   |   |   |    | Rerata <i>Fitness</i> |
|----------------|-----------------------------------------------|---|---|----|---|---|---|---|---|----|-----------------------|
|                | 1                                             | 2 | 3 | 4  | 5 | 6 | 7 | 8 | 9 | 10 |                       |
| 20             | 8                                             | 8 | 7 | 6  | 5 | 9 | 9 | 8 | 7 | 8  | 7.5                   |
| 40             | 7                                             | 7 | 8 | 6  | 7 | 8 | 7 | 7 | 6 | 9  | 7.2                   |
| 60             | 8                                             | 7 | 6 | 10 | 8 | 6 | 5 | 6 | 9 | 8  | 7.3                   |
| 80             | 7                                             | 5 | 5 | 8  | 7 | 8 | 9 | 8 | 9 | 6  | 7.2                   |
| 100            | 5                                             | 6 | 7 | 6  | 9 | 5 | 8 | 8 | 5 | 9  | 6.8                   |



**Gambar 3. Grafik Hasil Nilai *Fitness* Berdasarkan Jumlah Iterasi**

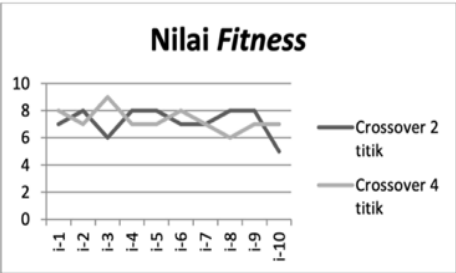
Analisis data pada Tabel 4 dan Gambar 4 menunjukkan bahwa nilai *fitness* yang diperoleh pada setiap percobaan bersifat acak. Meskipun tidak ada hubungan pasti antara banyaknya perhitungan dengan kualitas solusi terbaik, tren umum menunjukkan bahwa semakin banyak perhitungan, kemungkinan mendapatkan solusi yang lebih baik cenderung meningkat.

**4.2. Pengaruh Jenis Crossover terhadap Hasil Penjadwalan**

Program diuji sepuluh kali, jumlah iterasi dalam pengujian ini sebanyak 100 kali. Besar probabilitas mutasi yang digunakan dalam pengujian adalah 1% dan aturan permintaan jadwal karyawan tidak digunakan dalam pengujian ini. Fokus utama adalah mengevaluasi pengaruh jenis *crossover* terhadap nilai kekuatan solusi yang dihasilkan. Tabel 5 menunjukkan hubungan antara jenis *crossover* dan nilai *fitness* yang dihasilkan.

**Tabel 5. Hasil Nilai *Fitness* Berdasarkan Jenis Crossover**

| Jenis Crossover   | Nilai <i>Fitness</i> Terbaik (Percobaan ke-n) |   |   |   |   |   |   |   |   |    | Rerata <i>Fitness</i> |
|-------------------|-----------------------------------------------|---|---|---|---|---|---|---|---|----|-----------------------|
|                   | 1                                             | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |                       |
| Crossover 2 titik | 7                                             | 8 | 6 | 8 | 8 | 7 | 7 | 8 | 8 | 5  | 7.2                   |
| Crossover 4 titik | 8                                             | 7 | 9 | 7 | 7 | 8 | 7 | 6 | 7 | 7  | 7.3                   |



**Gambar 4. Grafik Hasil Nilai *Fitness* Berdasarkan Jenis Crossover**

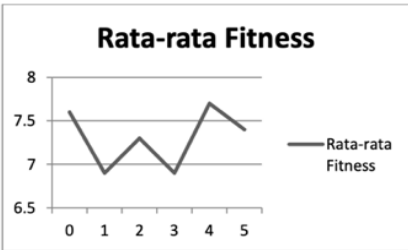
Analisis data pada Tabel 5 dan Gambar 4 menunjukkan, nilai *fitness* yang diperoleh untuk masing-masing jenis *crossover* bersifat acak. Meskipun tidak ada hubungan pasti antara jenis *crossover* dengan kualitas solusi terbaik, hasil secara keseluruhan menunjukkan bahwa *crossover* dua titik cenderung menghasilkan solusi yang lebih baik dibandingkan *crossover* empat titik.

**4.3. Pengaruh Probabilitas Mutasi terhadap Hasil Penjadwalan**

Program diuji sepuluh kali, jumlah iterasi dalam pengujian ini sebanyak 100 kali. *Crossover* yang digunakan dalam pengujian ini adalah *crossover* empat titik dan aturan permintaan jadwal karyawan tidak digunakan dalam pengujian ini. Fokus utama adalah mengevaluasi pengaruh probabilitas mutasi terhadap nilai kekuatan solusi yang dihasilkan. Tabel 6 menunjukkan hubungan antara probabilitas mutasi dan nilai *fitness* yang dihasilkan.

**Tabel 6. Hasil Nilai *Fitness* Berdasarkan Probabilitas Mutasi**

| Probabilitas Mutasi (%) | Nilai <i>Fitness</i> Terbaik (Percobaan ke-n) |    |   |   |   |   |   |   |   |    | Rerata <i>Fitness</i> |
|-------------------------|-----------------------------------------------|----|---|---|---|---|---|---|---|----|-----------------------|
|                         | 1                                             | 2  | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |                       |
| 0                       | 9                                             | 7  | 9 | 9 | 7 | 9 | 8 | 7 | 6 | 5  | 7.6                   |
| 1                       | 8                                             | 8  | 7 | 5 | 9 | 6 | 8 | 8 | 4 | 6  | 6.9                   |
| 2                       | 7                                             | 10 | 6 | 8 | 6 | 9 | 4 | 8 | 7 | 8  | 7.3                   |
| 3                       | 8                                             | 7  | 8 | 7 | 6 | 5 | 8 | 4 | 9 | 7  | 6.9                   |
| 4                       | 8                                             | 8  | 9 | 9 | 9 | 6 | 8 | 8 | 6 | 6  | 7.7                   |
| 5                       | 8                                             | 9  | 8 | 7 | 6 | 8 | 9 | 6 | 6 | 7  | 7.4                   |



**Gambar 5. Grafik Hasil Nilai *Fitness* Berdasarkan Probabilitas Mutasi**

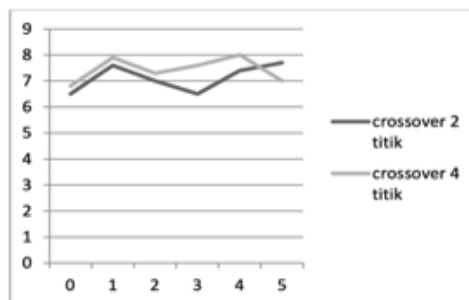
Analisis data pada Tabel 6 dan Gambar 5 menunjukkan, nilai *fitness* yang diperoleh untuk setiap nilai probabilitas mutasi bersifat acak. Meskipun tidak ada hubungan pasti antara besarnya probabilitas mutasi dengan kualitas solusi terbaik, hasil secara keseluruhan percobaan ini menunjukkan bahwa tidak ada nilai probabilitas mutasi yang secara konsisten menghasilkan solusi yang lebih baik.

#### 4.4. Pengaruh Jenis *Crossover* dan Probabilitas Mutasi terhadap Hasil Penjadwalan

Program diuji sepuluh kali, jumlah iterasi dalam pengujian ini sebanyak 100 kali. Aturan permintaan jadwal karyawan tidak digunakan dalam pengujian ini. Fokus utama adalah mengevaluasi pengaruh jenis *crossover* dan besaran probabilitas mutasi terhadap nilai kekuatan solusi yang dihasilkan. Tabel 7 menunjukkan hubungan antara jenis *crossover* dan probabilitas mutasi dengan nilai *fitness* yang dihasilkan.

**Tabel 7. Hasil Nilai *Fitness* Berdasarkan Jenis *Crossover* dan Probabilitas Mutasi**

| Jenis <i>Crossover</i> | Probabilitas Mutasi (%) | Nilai <i>Fitness</i> Terbaik (Percobaan ke-n) |   |    |    |    |    |   |    |   |    | Rerata <i>Fitness</i> |
|------------------------|-------------------------|-----------------------------------------------|---|----|----|----|----|---|----|---|----|-----------------------|
|                        |                         | 1                                             | 2 | 3  | 4  | 5  | 6  | 7 | 8  | 9 | 10 |                       |
| Crossover 2 titik      | 0                       | 4                                             | 7 | 6  | 8  | 6  | 6  | 6 | 9  | 7 | 6  | 6.5                   |
|                        | 1                       | 6                                             | 8 | 7  | 6  | 8  | 9  | 9 | 6  | 8 | 9  | 7.6                   |
|                        | 2                       | 7                                             | 9 | 6  | 8  | 5  | 6  | 6 | 7  | 6 | 10 | 7                     |
|                        | 3                       | 8                                             | 6 | 8  | 7  | 7  | 6  | 6 | 4  | 4 | 9  | 6.5                   |
|                        | 4                       | 6                                             | 5 | 8  | 8  | 8  | 7  | 9 | 7  | 8 | 8  | 7.4                   |
|                        | 5                       | 8                                             | 7 | 5  | 7  | 10 | 8  | 8 | 8  | 9 | 7  | 7.7                   |
| Crossover 4 titik      | 0                       | 8                                             | 9 | 6  | 7  | 8  | 8  | 3 | 7  | 5 | 7  | 6.8                   |
|                        | 1                       | 9                                             | 8 | 3  | 10 | 6  | 11 | 8 | 7  | 9 | 8  | 7.9                   |
|                        | 2                       | 9                                             | 8 | 10 | 5  | 8  | 8  | 6 | 6  | 5 | 8  | 7.3                   |
|                        | 3                       | 6                                             | 9 | 6  | 10 | 6  | 7  | 6 | 10 | 9 | 7  | 7.6                   |
|                        | 4                       | 6                                             | 9 | 10 | 7  | 8  | 7  | 8 | 5  | 8 | 12 | 8                     |
|                        | 5                       | 6                                             | 8 | 5  | 9  | 7  | 5  | 6 | 9  | 8 | 7  | 7                     |



**Gambar 6. Grafik Hasil Nilai *Fitness* Berdasarkan Jenis *Crossover* dan Probabilitas Mutasi**

Analisis data pada Tabel 7 dan Gambar 6 menunjukkan bahwa nilai *fitness* yang diperoleh untuk setiap kombinasi jenis *crossover* dan probabilitas mutasi bersifat acak. Tidak ada korelasi yang signifikan antara jenis *crossover* dan probabilitas mutasi dengan kualitas solusi terbaik.

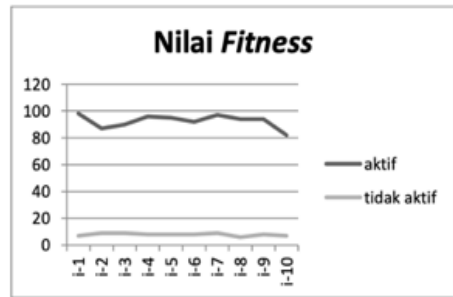
#### 4.5. Pengaruh Aturan Permintaan Jadwal terhadap Hasil Penjadwalan

Program diuji sepuluh kali, jumlah iterasi dalam pengujian ini sebanyak 100 kali. *Crossover* yang digunakan dalam pengujian ini adalah *crossover* empat titik. Besar probabilitas mutasi yang digunakan dalam pengujian adalah 2%. Fokus utama adalah mengevaluasi pengaruh aktif dan tidaknya aturan permintaan jadwal karyawan terhadap nilai kekuatan solusi yang dihasilkan. Tabel 8 menunjukkan hubungan antara aktif dan tidaknya aturan permintaan jadwal dan nilai *fitness* yang dihasilkan.

**Tabel 8. Hasil Nilai *Fitness* Berdasarkan Aturan Permintaan Jadwal**

| Aturan Permintaan Jadwal | Nilai <i>Fitness</i> Terbaik (Percobaan ke-n) |    |    |    |    |    |    |    |    |    | Rerata <i>Fitness</i> |
|--------------------------|-----------------------------------------------|----|----|----|----|----|----|----|----|----|-----------------------|
|                          | 1                                             | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 |                       |
| aktif                    | 98                                            | 87 | 90 | 96 | 95 | 92 | 97 | 94 | 94 | 82 | 92.5                  |
| tidak aktif              | 7                                             | 9  | 9  | 8  | 8  | 8  | 9  | 6  | 8  | 7  | 7.9                   |





Gambar 7. Grafik Hasil Nilai *Fitness* Berdasarkan Aturan Permintaan Jadwal

Berdasarkan analisis data pada Tabel 8 dan Gambar 7, dapat disimpulkan bahwa percobaan tanpa aturan permintaan jadwal menghasilkan solusi yang lebih baik dibandingkan percobaan dengan aturan tersebut diaktifkan.

4.6. Perbandingan Algoritma Genetika dengan Metode Manual

Untuk menguji keefektifan algoritma genetika dibandingkan metode manual, proses penjadwalan akan dibandingkan antara menggunakan algoritma genetika dan metode manual. Kemudian hasil penjadwalan diterapkan dan dilakukan survei terhadap semua karyawan yang bekerja di salah satu kedai 24 jam di Yogyakarta.

Tabel 9. Hasil Evaluasi Waktu Komputasi (*Computational Time*)

| Tujuan                                                                                                                  | Indikator                                                           | Hasil Evaluasi                           |                                    |
|-------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|------------------------------------------|------------------------------------|
|                                                                                                                         |                                                                     | Algoritma Genetika                       | Metode Manual                      |
| Mengukur berapa lama waktu yang dibutuhkan oleh AG dan metode manual untuk menyelesaikan penjadwalan satu minggu penuh. | Semakin pendek waktu komputasi, semakin efektif algoritma tersebut. | Membutuhkan waktu pemrosesan 10-20 menit | Membutuhkan tiga jam kerja manajer |

Tabel 10. Hasil Evaluasi Kualitas Solusi (*Solution Quality*)

| Tujuan                                                                              | Indikator                                                                                                                                    | Hasil Evaluasi                                                   |                                                                  |
|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|------------------------------------------------------------------|
|                                                                                     |                                                                                                                                              | Algoritma Genetika                                               | Metode Manual                                                    |
| Mengukur sejauh mana jadwal memenuhi kebutuhan operasional dan preferensi karyawan. | Persentase kesesuaian jadwal karyawan yang dihasilkan dengan aturan-aturan penjadwalan dan preferensi karyawan dengan kebutuhan operasional. | Memenuhi 95% preferensi karyawan dan semua kebutuhan operasional | Memenuhi 80% preferensi karyawan dan semua kebutuhan operasional |

Tabel 11. Hasil Evaluasi Jumlah Kesalahan pada Penjadwalan (*Scheduling Conflicts*)

| Tujuan                                                                                                                 | Indikator                                                                                                   | Hasil Evaluasi                                     |                                                     |
|------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|----------------------------------------------------|-----------------------------------------------------|
|                                                                                                                        |                                                                                                             | Algoritma Genetika                                 | Metode Manual                                       |
| Menghitung kesalahan terhadap jadwal yang dihasilkan, seperti sif ganda atau tumpang tindih shift untuk satu karyawan. | Jumlah kesalahan dalam satu periode penjadwalan, nilai yang lebih rendah menunjukkan hasil yang lebih baik. | Menghasilkan rata-rata satu konflik dalam seminggu | Menghasilkan rata-rata empat konflik dalam seminggu |

Tabel 12. Hasil Evaluasi Kepuasan Karyawan (*Employee Satisfaction*)

| Tujuan                                                                            | Indikator                                                                                                                                                  | Hasil Evaluasi                    |                                        |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|----------------------------------------|
|                                                                                   |                                                                                                                                                            | Algoritma Genetika                | Metode Manual                          |
| Mengukur kepuasan karyawan terhadap jadwal yang dibuat oleh masing-masing metode. | Survei kepuasan atau indeks kepuasan berdasarkan jadwal yang dihasilkan, skor yang lebih tinggi artinya jadwal yang lebih sesuai dengan kepuasan karyawan. | a-rata skor kepuasan karyawan 4,5 | Rata-rata skor kepuasan karyawan 3,8/5 |

Berdasarkan Tabel 9, Tabel 10, Tabel 11, dan Tabel 12, algoritma genetika dapat dinilai lebih efektif dalam menghasilkan jadwal yang memenuhi kebutuhan operasional, meningkatkan

kepuasan karyawan, dan meminimalkan konflik, dengan waktu komputasi yang jauh lebih cepat dibandingkan metode manual.

## 5. Kesimpulan dan Saran

Hasil penelitian menunjukkan bahwa algoritma genetika adalah metode yang efektif untuk mengatasi masalah penjadwalan karyawan. Namun, analisis terhadap berbagai parameter algoritma menghasilkan beberapa temuan penting. Peningkatan kompleksitas fungsi *fitness* cenderung menurunkan peluang memperoleh solusi optimal. Selain itu, semakin banyak iterasi yang digunakan semakin besar pula kemungkinan kualitas solusi yang dihasilkan lebih optimal. Hasil penelitian juga menunjukkan bahwa pemilihan operator algoritma genetika tidak secara langsung menentukan kualitas solusi yang dihasilkan, karena proses pencarian solusi bersifat acak.

## Referensi

- [1] J. Soriano, E. R. Jalao, and I. A. Martinez, "Integrated employee scheduling with known employee demand, including breaks, overtime, and employee preferences," *Journal of Industrial Engineering and Management*, vol. 13, no. 3, p. 451, Sep. 2020, doi: 10.3926/jiem.3126.
- [2] Q. Zhang, "Research on the Construction Schedule and Cost Optimization of Grid Structure based on BIM and Genetic Algorithm," *Journal of Physics: Conference Series*, vol. 1744, no. 2, p. 022065, Feb. 2021, doi: 10.1088/1742-6596/1744/2/022065.
- [3] H. R. O. Rocha, I. H. Honorato, R. Fiorotti, W. C. Celeste, L. J. Silvestre, and J. A. L. Silva, "An Artificial Intelligence based scheduling algorithm for demand-side energy management in Smart Homes," *Applied Energy*, vol. 282, p. 116145, Jan. 2021, doi: 10.1016/j.apenergy.2020.116145.
- [4] H. J. Kaleybar, M. Davoodi, M. Brenna, and D. Zaninelli, "Applications of genetic algorithm and its variants in rail vehicle systems: A bibliometric analysis and comprehensive review," *IEEE Access*, vol. 11, pp. 68972–68993, 2023, doi: 10.1109/ACCESS.2023.3292790.
- [5] R. Chen, B. Yang, S. Li, and S. Wang, "A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem," *Computers & Industrial Engineering*, vol. 149, p. 106778, Nov. 2020, doi: 10.1016/j.cie.2020.106778.
- [6] Z. Wang and A. Sobey, "A comparative review between genetic algorithm use in composite optimisation and the state-of-the-art in evolutionary computation," *Composite Structures*, vol. 233, p. 111739, Feb. 2020, doi: 10.1016/j.compstruct.2019.111739.
- [7] L. Manzoni, L. Mariot, and E. Tuba, "Balanced crossover operators in Genetic Algorithms," *Swarm Evol Comput*, vol. 54, p. 100646, May 2020, doi: 10.1016/j.swevo.2020.100646.
- [8] B. Koohestani, "A crossover operator for improving the efficiency of permutation-based genetic algorithms," *Expert Systems with Applications*, vol. 151, p. 113381, Aug. 2020, doi: 10.1016/j.eswa.2020.113381.
- [9] M. A. Albadr, S. Tiun, M. Ayob, and F. AL-Dhief, "Genetic Algorithm Based on Natural Selection Theory for Optimization Problems," *Symmetry*, vol. 12, no. 11, p. 1758, Oct. 2020, doi: 10.3390/sym12111758.
- [10] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 8091–8126, Feb. 2021, doi: 10.1007/s11042-020-10139-6.
- [11] B. Alhijawi, A. Awajan, "Genetic algorithms: theory, genetic operators, solutions, and applications," *Evolutionary Intelligence*, vol. 17, no. 3, pp. 1245–1256, Jun. 2024, doi: 10.1007/s12065-023-00822-6.
- [12] R. Salman, Suprpto, and Irfandi, "Analisis Pengaruh Probabilitas Crossover Terhadap Kinerja Algoritma Genetika Dalam Optimasi Penjadwalan Matakuliah," *Jurnal Teknoif Teknik Informatika Institut Teknologi Padang*, vol. 11, no. 2, pp. 69–74, Oct. 2023, doi: 10.21063/jtif.2023.V11.2.69-74.

- [13] L. P. S. Ardiyani, "Perbandingan Algoritma Genetika dengan Algoritma Steepest Ascent Hill Climbing untuk Optimasi Penjadwalan Kuliah," *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 11, no. 1, p. 63, Apr. 2022, doi: 10.23887/janapati.v11i1.43172.
- [14] M. H. P. Swari, C. A. Putra, and I. P. S. Handika, "Analisis Perbandingan Algoritma Genetika dan Modified Improved Particle Swarm Optimization dalam Penjadwalan Mata Kuliah," *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 11, no. 2, pp. 92–101, Aug. 2022, doi: 10.23887/janapati.v11i2.49061.
- [15] T. G. Bhagya, "Algoritma Genetik Pada Penjadwalan Transportasi Kapal Laut (Studi Kasus PT. Pelni)," *Journal of Industrial and Manufacture Engineering*, vol. 5, no. 2, Nov. 2021, doi: 10.31289/jime.v5i2.5952.
- [16] L. Kletzander and N. Musliu, "Solving the general employee scheduling problem," *Computers & Operations Research*, vol. 113, p. 104794, Jan. 2020, doi: 10.1016/j.cor.2019.104794.
- [17] F. N. Kiwanuka, L. Karadsheh, J. alqatawna, and A. H. Muhamad Amin, "Modeling Employee Flexible Work Scheduling As A Classification Problem," *Procedia Computer Science*, vol. 192, pp. 3281–3290, 2021, doi: 10.1016/j.procs.2021.09.101.
- [18] S. H. Moi, P. Y. Yong, and F. C. Weng, "Genetic Algorithm Based Heuristic for Constrained Industry Factory Workforce Scheduling," in *2021 7th International Conference on Control, Automation and Robotics (ICCAR)*, IEEE, Apr. 2021, pp. 345–348. doi: 10.1109/ICCAR52225.2021.9463323.
- [19] S. Nayeri, R. Tavakkoli-Moghaddam, Z. Sazvar, and J. Heydari, "A heuristic-based simulated annealing algorithm for the scheduling of relief teams in natural disasters," *Soft Computing*, vol. 26, no. 4, pp. 1825–1843, Feb. 2022, doi: 10.1007/s00500-021-06425-6.
- [20] X. Liang and Z. Du, "Genetic Algorithm with Simulated Annealing for Resolving Job Shop Scheduling Problem," in *2020 IEEE 8th International Conference on Computer Science and Network Technology (ICCSNT)*, IEEE, Nov. 2020, pp. 64–68. doi: 10.1109/ICCSNT50940.2020.9305010.
- [21] N. Potu, C. Jatoth, and P. Parvataneni, "Optimizing resource scheduling based on extended particle swarm optimization in fog computing environments," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 23, Dec. 2021, doi: 10.1002/cpe.6163.
- [22] S. Saeedi, R. Khorsand, S. Ghandi Bidgoli, and M. Ramezanpour, "Improved many-objective particle swarm optimization algorithm for scientific workflow scheduling in cloud computing," *Computers & Industrial Engineering*, vol. 147, p. 106649, Sep. 2020, doi: 10.1016/j.cie.2020.106649.
- [23] X. Wei, "Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing," *Journal of Ambient Intelligence and Humanized Computing*, Oct. 2020, doi: 10.1007/s12652-020-02614-7.