

Naskah Publikasi

PROYEK TUGAS AKHIR

**SISTEM PENJADWALAN *SHIFT* KERJA ANGGOTA KEPOLISIAN
MENGUNAKAN ALGORITMA GENETIKA
(Studi Kasus: Kepolisian Sektor Magelang Tengah)**



Disusun oleh:
HARI PRANATA
5150411219

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN ELEKTRO
UNIVERSITAS TEKNOLOGI YOGYAKARTA
2019**

Naskah Publikasi

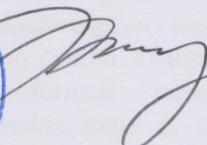
PROYEK TUGAS AKHIR

**SISTEM PENJADWALAN *SHIFT* KERJA ANGGOTA KEPOLISIAN
MENGUNAKAN ALGORITMA GENETIKA
(Studi Kasus: Kepolisian Sektor Magelang Tengah)**



Disusun oleh:
HARI PRANATA
5150411219

Telah disetujui oleh pembimbing

Pembimbing:

Dony Avianto, S.T., M.T.

Tanggal: 7/5/19

SISTEM PENJADWALAN *SHIFT* KERJA ANGGOTA KEPOLISIAN MENGUNAKAN ALGORITMA GENETIKA (Studi Kasus: Kepolisian Sektor Magelang Tengah)

Hari Pranata

Program Studi Teknik Informatika, Fakultas Teknologi Informasi dan Elektro
Universitas Teknologi Yogyakarta
E-mail : hariprant@gmail.com

ABSTRAK

Penjadwalan *shift* kerja anggota kepolisian merupakan hal yang berpengaruh dalam produktifitas proses pelaksanaan pelayanan masyarakat. Permasalahannya adalah bagaimana menyusun jadwal yang sesuai kriteria dan meminimalisir adanya benturan jadwal yang sering terjadi ketika penyusunan dilakukan secara konvensional. Salah satu langkah yang diambil adalah dengan membangun sistem penjadwalan anggota kepolisian menggunakan algoritma genetika. Penggunaan algoritma genetika ditujukan agar hasil penjadwalan tersusun secara alami oleh sistem melalui beberapa iterasi perhitungan sehingga dapat membantu kepolisian dalam menyusun jadwal *shift* kerja anggota polisi secara efektif dan tepat. Dengan menggunakan Algoritma Genetika, didapatkan hasil pada pengujian pertama dengan nilai awal *mutation rate* 0,3 dan batas generasi 1000 menghasilkan nilai *fitness* tertinggi 665 dengan tingkat kesesuaian 89,38% pada generasi 10000 keatas. Sedangkan pengujian dengan nilai *mutation rate* awal 0.5 dan batas generasi 5000 menghasilkan nilai *fitness* tertinggi 660 dengan tingkat kesesuaian 87,37% pada generasi 500000 keatas. Dalam penelitian ini sistem dibangun berbasis web dengan bahasa pemrograman PHP (*Pear Hypertext Preprocessor*) dan MySQL sebagai DBMS.

Kata Kunci : Sistem, Penjadwalan, Algoritma Genetika, *Shift*, Website

I. Pendahuluan

1.1 Latar Belakang

Perkembangan teknologi informasi saat ini cukup pesat dan dapat dikatakan semakin maju. Banyak pekerjaan manusia mulai digantikan oleh teknologi baik sebagian maupun menyeluruh. Adanya teknologi informasi sangat membantu manusia dalam melakukan pekerjaan yang membutuhkan perhitungan dan ketelitian yang cukup tinggi. Teknologi informasi dapat masuk ke segala bidang termasuk Kepolisian. Saat ini kepolisian dituntut agar dapat menggunakan teknologi informasi untuk membantu berbagai tugas polisi. Meski belum secara menyeluruh, penggunaan teknologi informasi di Kepolisian semakin dikembangkan mulai dari tingkat pusat maupun daerah. Dengan memanfaatkan teknologi informasi tersebut, kepolisian menjadi semakin maju di bidang pelayanan masyarakat.

Kepolisian Sektor Magelang Tengah (Polsek Magelang Tengah) merupakan salah satu instansi Kepolisian tingkat sektor Kecamatan Magelang Tengah. Pihak Polsek

Magelang Tengah menggunakan teknologi informasi untuk melakukan berbagai aktivitas. Salah satu pemanfaatan teknologi informasi yang diterapkan di Polsek Magelang Tengah adalah dalam penyusunan jadwal *shift* kerja. Meskipun sudah menggunakan bantuan teknologi informasi, tetap saja masih ada kendala dalam proses penyusunan jadwal *shift* kerja. Polsek Magelang Tengah melakukan penyusunan jadwal menggunakan bantuan aplikasi Microsoft Word, tetapi dalam proses pemilihan anggota jaga masih konvensional. Anggota kepolisian yang akan mendapat giliran *shift* jaga dipilih secara manual dengan melihat apakah anggota polisi tersebut sudah mendapat giliran jadwal jaga atau belum. Jika jumlah anggota kepolisian yang akan disusun jadwal sedikit tentu tidak akan masalah, tetapi jika jumlahnya cukup banyak pasti akan menimbulkan kebingungan dalam pemilihan anggota jaga dan rentan terhadap kesalahan penyusunan oleh petugas (*human error*). Penyusunan jadwal yang seharusnya dapat dilakukan sekali tetapi dalam prakteknya terkadang dapat dilakukan berkali-kali karena terdapat jadwal yang bentrok. Jika tidak segera

dilakukan revisi jadwal maka akan mengakibatkan anggota kepolisian yang seharusnya beristirahat harus kembali terkena shift kerja.

Algoritma Genetika (GA) merupakan algoritma yang menggunakan seleksi alami untuk menemukan solusi optimal dan dimanfaatkan dalam optimasi berbagai permasalahan. Pada aplikasinya, GA biasanya digunakan untuk memperoleh solusi optimal seperti penjadwalan ataupun solusi pendekatan dari persoalan optimasi yang mempunyai banyak sekali solusi yang mungkin. Persoalan seperti ini biasanya tidak dapat atau sulit diselesaikan dengan metode-metode eksak. Setelah mengalami evolusi pada beberapa generasi, GA pada umumnya akan mampu memberikan solusi yang baik, dalam hal ini susunan jadwal yang baik. Salah satu pemanfaatan GA adalah bisa digunakan dalam penjadwalan secara otomatis *shift* anggota kepolisian di Polsek Magelang Tengah. GA digunakan dalam proses generate otomatis jadwal *shift* kerja anggota polisi sesuai dengan *constraint* (batasan) yang ditentukan dalam jangka waktu tertentu. Beberapa penelitian yang memanfaatkan GA dalam penyusunan jadwal antara lain adalah: Penelitian (Ilmi, Mahmudy, & Ratnawati, 2015) dengan judul Optimasi Penjadwalan Perawat Menggunakan Algoritma Genetika dan penelitian (Ginting, 2017) dengan judul Implementasi Algoritma Genetika dalam Penjadwalan Shift Kerja di Call Center Telkomsel.

Berdasarkan beberapa penelitian yang telah dibahas di atas, sudah ada penelitian yang menggunakan GA sebagai metode untuk memecahkan masalah penjadwalan. Akan tetapi dari beberapa penelitian tersebut belum ada yang meneliti untuk penjadwalan *shift* kerja anggota kepolisian. Oleh sebab itu peneliti tertarik untuk melakukan penelitian dengan menggunakan GA untuk mendapatkan susunan jadwal *shift* yang optimal di Polsek Magelang Tengah. Tujuan dari penelitian ini adalah agar kepolisian khususnya di Polsek Magelang Tengah menjadi lebih mudah ketika menyusun jadwal shift kerja dalam jangka waktu tertentu sesuai ketentuan atau batasan. Sistem penjadwalan menggunakan GA ini juga dapat meminimalkan terjadinya kesalahan penyusunan jadwal jika dilakukan penyusunan secara konvensional. Penyusunan jadwal shift kerja yang nantinya juga berdampak pada

pelayanan masyarakat yang lebih cepat dan tepat.

II. Tinjauan Pustaka

2.1 Landasan Teori

a. Pengertian Algoritma Genetika

Menurut Haupt (2004), Algoritma genetika adalah suatu proses optimasi yang dikembangkan berdasarkan prinsip genetika dan proses seleksi alamiah. Algoritma Genetika banyak dipakai pada aplikasi bisnis, teknik maupun pada bidang keilmuan lainnya. Algoritma ini dimulai dengan kumpulan solusi yang disebut dengan populasi. Solusi-solusi dari sebuah populasi diambil dan digunakan untuk membentuk solusi yang baru. Hal ini dimotivasi dengan harapan bahwa populasi yang dibentuk tersebut akan lebih baik daripada yang lama. Solusi-solusi yang dipilih untuk membentuk solusi-solusi yang baru sesuai dengan fitness mereka masing-masing.

b. Operasional Algoritma Genetika

Struktur dasar dari GA untuk penjadwalan ditunjukkan pada pseudocode berikut:

Pseudocode Algoritma Genetika

```
populasi <= Ukuran Populasi
maxGen <= Maksimal Generasi
MR <= Mutation Rate
bangkitkanGenerasiAwal (jmlHari)
evaluasiKromosom(fitness function())
while (g < maxGen)
  seleksiKromosom(fitness)
  random [1,n]
if (new fitness > old fitness) then
  kromosom dipilih
  proses crossover ()
  if (new fitness > old fitness) then
  posisi_x = random [1,
jml_gen]
  crossover[posisi_x]
  proses mutasi (MR)
  while (k < jumlahGen) do
  r = random bilangan [0,1]
  if (r < MR) then
  posisi_a = rand [1, jml_gen]
  posisi_b = rand [1, jml_gen]
  mutasi_gen[posisi_a, posisi_b]
  k = k +1
g = g+1
```

1. Membangkitkan Generasi Awal

Pada umumnya generasi awal pada proses GA berisikan kromosom yang dibangkitkan acak (random). Untuk menghindari diperolehnya solusi yang tidak layak,

kromosom generasi awal harus diuji kelayakannya untuk semua batasan (*constraint*). Kromosom yang tidak memiliki salah satu batasan dikatakan sebagai kromosom layak.

2. Fungsi *Fitness*

Fungsi *fitness* digunakan untuk proses evaluasi kromosom GA untuk memperoleh kromosom yang diinginkan. Fungsi ini menghitung kualitas dari kromosom untuk mengetahui seberapa baik kromosom yang dihasilkan. Fungsi *fitness* pada GA sebagai berikut:

Keterangan:

$$\text{Total Fitness} = \text{Fitness kromosom} - (P1 + P2 + P3)$$

Fitness kromosom = 744 (total *fitness* jika tidak penalti dalam 1 kromosom)

P = pelanggaran / penalti.

3. Seleksi

Salah satu hal penting pada proses GA adalah bagaimana memilih kromosom yang akan diikuti ke generasi berikutnya. Berdasarkan teori evolusi Darwin, hanya kromosom yang terbaik yang dipilih ke generasi berikutnya. Pada proses GA dikenal beberapa metode yang digunakan untuk memilih kromosom.

a. *Roulette Wheel Selection*

Metode ini dimulai dengan memberikan dua nilai peluang pada masing-masing kromosom untuk terpilih ke generasi berikutnya berdasarkan nilai *fitness* kromosom tersebut. Kromosom yang lebih baik tentu saja akan mempunyai peluang yang lebih besar untuk terpilih ke generasi berikutnya.

b. *Tournament Selection*

Metode seleksi ini melakukan pemilihan berdasarkan nilai *fitness*. Pemilihan dimulai dengan memunculkan beberapa nilai random sebagai index untuk memilih beberapa calon induk, kemudian di seleksi dengan nilai *fitness* terbaik.

c. *Rank Selection*

Metode ini dilakukan untuk menghindari kendala pada metode seleksi *roulette-wheel*, seperti yang telah dijelaskan sebelumnya. Metode ini dilakukan dengan memberikan nilai ranking pada masing-masing kromosom. Kromosom dengan nilai *fitness* terburuk diberikan nilai satu, demikian seterusnya hingga kromosom dengan nilai

fitness terbaik diberi nilai N (jumlah kromosom).

4. Crossover

Crossover merupakan suatu proses pembentukan kromosom turunan (*offspring*) dengan menggabungkan elemen dari kromosom induk yang terpilih (*parent*). Proses ini dilakukan dalam upaya mendapatkan kromosom baru dengan solusi yang lebih baik. Berikut disajikan beberapa metode *crossover* yang sering dipergunakan.

a. *Crossover Satu Titik*

Crossover satu titik dimulai dengan menentukan satu titik secara random pada kromosom. Proses pembentukan keturunan (*offspring*) dilakukan dengan mempertukarkan elemen di sebelah kanan dari titik tersebut pada masing-masing induk (*parent*).

b. *Crossover Dua Titik*

Seperti halnya *crossover* satu titik, *crossover* dua titik dilakukan dengan menentukan dua titik secara random pada kromosom. Proses pembentukan keturunan (*offspring*) dilakukan mempertukarkan elemen diantara kedua titik pada masing-masing induk (*parent*).

5. Mutasi

Proses mutasi biasanya dilakukan dengan melakukan perubahan terhadap gen pada suatu kromosom. Proses ini bertujuan meningkatkan keragaman kromosom. Proses mutasi berfungsi untuk menghasilkan individu baru yang memiliki sifat baru dengan cara melakukan perubahan pada satu atau lebih gen.

a. Metode Mutasi Pembalikan

Metode mutasi pembalikan dilakukan dengan mengambil suatu *substring* yang terletak diantara dua titik pada kromosom. Pemilihan dua titik ini dilakukan secara acak. Selanjutnya dilakukan proses pembalikan (*invers*) gen pada *substring* tersebut.

b. Metode Mutasi Penyisipan

Metode mutasi penyisipan dilakukan dengan memilih salah satu gen yang ada pada kromosom. Selanjutnya gen tersebut disisipkan pada posisi yang juga dipilih secara acak.

c. Metode Mutasi Penukaran

Berbeda dengan metode mutasi penyisipan yang salah satu gen dari kromosom. Metode mutasi pemindahan dilakukan dengan

memilih dua titik pada kromosom. Selanjutnya gen yang ada diantara kedua titik tersebut disisipkan pada suatu posisi yang juga dipilih secara acak.

c. Tingkat Kesesuaian

Tingkat kesesuaian jadwal digunakan untuk mengetahui persentase seberapa sesuai jadwal dari hasil generate sistem GA dengan peraturan yang berlaku di Polsek Magelang Tengah. Tingkat kesesuaian dihitung dari nilai *fitness* yang didapat dengan menggunakan rumus sebagai berikut.

$$\text{Tingkat Kesesuaian} = \frac{\text{Jumlah } fitness \text{ yang didapat}}{\text{Total } Fitness} \times 100 \%$$

d. Pengertian Penjadwalan

Menurut Jong Jek (Ginting, 2017), jadwal merupakan sesuatu yang menjelaskan dimana dan kapan orang-orang dan sumber daya berada pada suatu waktu. Sedangkan penjadwalan merupakan proses cara pembuatan menjadwalkan atau memasukkan dalam jadwal. Sedangkan menurut Stevenson & Chuong (2014), yang dialih bahasakan oleh Diana Angelica mengemukakan bahwa “Penjadwalan adalah menetapkan waktu dari penggunaan perlengkapan, fasilitas, dan aktivitas manusia dalam sebuah organisasi”.

III. Pembahasan

3.1 Objek Penelitian

Pada penelitian ini, penulis mengambil objek penelitian di Polsek Magelang Tengah. Data yang digunakan adalah jadwal *shift* kerja anggota kepolisian dalam satu bulan. Data tersebut digunakan sebagai sampel guna membuktikan hasil output yang sesuai dengan harapan. Input yang akan digunakan pada sistem yang akan dibangun adalah input nama anggota kepolisian yang akan mendapat jadwal *shift* kerja.

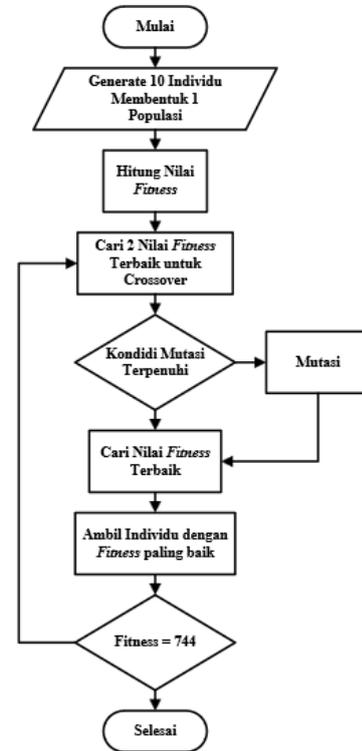
3.2 Metode Pengumpulan Data

Metode penelitian yang dilakukan penulis dalam melakukan penelitian untuk mendapatkan data yang lengkap dan akurat, dilakukan metode pengumpulan data yaitu dengan metode wawancara dengan pihak kepolisian divisi seksi umum (Sium) di Polsek Magelang Tengah.

3.3 Proses Algoritma Genetika

Urutan prosedur di dalam sistem dan apa yang dikerjakan di dalam sistem penjadwalan secara keseluruhan di algoritma genetika ditunjukkan dengan flowchart yang ada pada Gambar 3.1

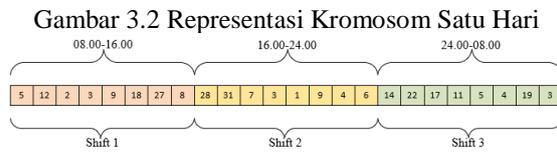
Gambar 3.1 Flowchart Proses GA



1. Pembangkitan Populasi Awal (Representasi Kromosom)

Merupakan proses untuk membentuk satu populasi dengan cara mengambil variabel waktu *shift*, variabel hari, variabel nama anggota Polisi, kemudian direpresentasikan ke dalam bentuk kromosom. Dalam penelitian ini, jumlah waktu yang tersedia sebanyak satu bulan atau setara 31 hari. Anggota Polisi bekerja selama 24 jam dalam 7 hari kerja dari hari senin sampai hari minggu. Dalam satu hari kerja terdapat tiga waktu *shift* dengan durasi 8 jam. *Shift* pagi dimulai pukul 08.00 WIB sampai pukul 16.00 WIB, *shift* sore dimulai pukul 16.00 WIB sampai pukul 24.00 WIB, kemudian *shift* malam dilanjutkan hingga pukul 08.00 WIB keesokan harinya. Setiap waktu *shift* terdapat 8 anggota Polisi yang mendapatkan jadwal *shift* kerja. Banyaknya kromosom yang dibangkitkan sebanyak 10 kromosom, sedangkan dalam satu kromosom terdapat 744 gen. Nilai dari setiap gen didapat secara random dari bilangan integer 0 sampai 43 secara acak, bilangan tersebut mewakili nomor induk pegawai (NIP) anggota Polisi yang mendapat

jadwal *shift*. Contoh representasi gen membentuk kromosom dalam satu hari pada penelitian ini dapat dilihat pada Gambar 3.2



2. Evaluasi Kromosom (Hitung Nilai *Fitness*)

Evaluasi kromosom dilakukan dengan cara menghitung total nilai *fitness* pada setiap kromosom menggunakan fungsi *fitness* dengan menjumlahkan penalti yang dilanggar oleh setiap kromosom. Nilai penalti dihitung berdasarkan kemunculan pelanggaran pada kromosom. Setiap kemunculan dihitung 1 pelanggaran. Jika suatu kromosom memiliki nilai *fitness* yang tinggi (tidak banyak pelanggaran penalti), maka berpeluang besar bertahan ke generasi selanjutnya. Begitupun sebaliknya jika memiliki nilai *fitness* kecil (banyak pelanggaran penalti) maka akan memperkecil peluang untuk lolos ke generasi selanjutnya. Nilai *fitness* dapat dihitung dengan menggunakan rumus *fitness* seperti yang telah dijelaskan pada bab sebelumnya. Jenis pelanggaran beserta nilai pelanggaran dapat dilihat pada Tabel 3.1.

Tabel 3.1 Penalti Pelanggaran

No	Jenis Pelanggaran	Nilai Pelanggaran
1	Anggota Polisi tidak boleh mendapat <i>shift</i> yang sama dalam satu hari.	1
2	Anggota Polisi yang mendapat <i>shift</i> malam tidak boleh mendapat <i>shift</i> keesokan paginya.	1
3	Seorang anggota Polisi tidak boleh muncul dua kali dalam satu <i>shift</i> .	1
4	Anggota Polisi wanita tidak boleh mendapat <i>shift</i> malam.	1

Contoh perhitungan nilai *fitness* kromosom sebagai berikut:

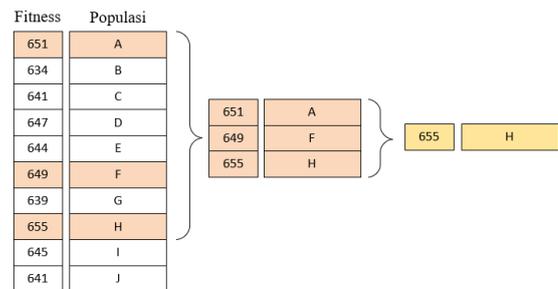
1. Kromosom 1

$$\begin{aligned}
 \text{Penalti satu hari} &= 2 \\
 \text{Penalti malam pagi} &= 1 \\
 \text{Penalti muncul dua kali} &= 3 \\
 \text{Penalti perempuan} &= 0 \\
 \text{Fitness Kromosom 1} &= 744 - 6 = 738
 \end{aligned}$$

3. Proses Seleksi

Setelah mendapatkan nilai *fitness* dilakukan seleksi menggunakan metode *Tournament Selection*. Kromosom dipilih berdasarkan nilai *fitness*. Pemilihan dimulai dengan memunculkan beberapa nilai random sebagai index untuk memilih beberapa calon induk, kemudian di seleksi dengan nilai *fitness* terbaik. Untuk lebih jelasnya mengenai proses seleksi dalam penelitian ini dapat dilihat pada Gambar 3.3

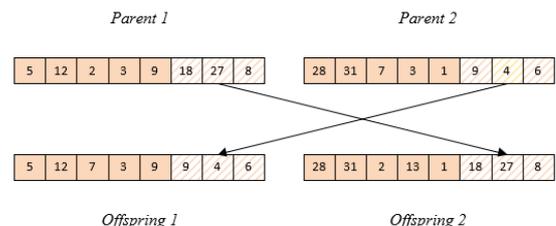
Gambar 3.3 *Tournament Selection*



4. Proses Crossover

Setelah proses seleksi langkah selanjutnya dipilih dua individu (kromosom) yang mempunyai nilai *fitness* terbaik untuk dijadikan induk yang nantinya akan dilakukan *crossover*. Jika terdapat nilai *fitness* yang sama maka diambil salah satu kromosom saja. Turunan (*offspring*) hasil *crossover* akan menggantikan kromosom yang memiliki nilai *fitness* terburuk. Dalam penelitian ini peneliti menggunakan teknik *one-point crossover*. Dimulai dengan menentukan satu titik secara random pada kromosom. Proses pembentukan keturunan (*offspring*) dilakukan dengan mempertukarkan elemen di sebelah kanan dari titik tersebut pada masing-masing induk (*parent*). Untuk lebih jelasnya dapat dilihat pada Gambar 3.4

Gambar 3.4 *Crossover satu titik*

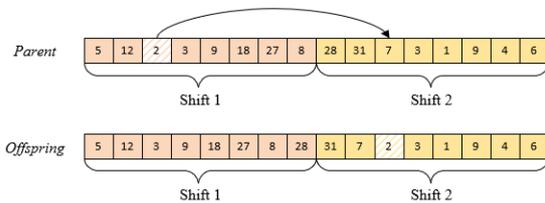


5. Proses Mutasi

Setelah proses *crossover* dilakukan, proses selanjutnya adalah mutasi. Jumlah kromosom yang mengalami mutasi dalam satu populasi ditentukan oleh parameter *mutation rate* (MR). Nilai MR didapat dari random antara bilangan

0-1. Dalam penelitian ini menggunakan teknik mutasi penyisipan. Metode ini dilakukan dengan memilih salah satu gen yang ada pada kromosom. Selanjutnya gen tersebut disisipkan pada posisi yang juga dipilih acak. Untuk lebih jelasnya mengenai proses mutasi penyisipan dalam penelitian ini dapat dilihat pada Gambar 3.5

Gambar 3.5 Mutasi Penyisipan



Proses mutasi dalam penelitian ini dapat dilihat pada pseudocode berikut:

Pseudocode Proses Mutasi

```

jml_gen = 496
k = 0
MR = 0,05
while (k < jumlahGen) do
  r = random bilangan [0,1]
  if (r < MR) then
    posisi_a=rand(1, jml_gen)
    posisi_b=rand(1, jml_gen)
    mutasi_gen[posisi_a, posisi_b]
  k = k +1

```

6. Kondisi Berhenti

Proses penjadwalan yang dilakukan dengan GA akan berhenti setelah suatu syarat berhenti terpenuhi. Syarat yang pertama adalah jika kromosom terbaik telah ditemukan. Proses GA juga akan berhenti jika jumlah generasi (iterasi) telah mencapai maksimal generasi yang telah ditentukan sebelumnya.

3.4 Proses Tahapan Penjadwalan GA

Tahapan penjadwalan GA dengan menggunakan contoh permasalahan sebagai berikut:

1. Pembentukan populasi awal

Misalkan ditentukan jumlah populasi adalah 10 dengan rincian sebagai berikut:

- Kromosom[1] = [05, 08, 23, 06, 24, 16, 13, 04, 11, 27, 17, 29, 02, 13, 26, 19]
- Kromosom[2] = [16, 18, 28, 16, 04, 30, 09, 25, 30, 24, 06, 30, 16, 05, 28, 19]
- Kromosom[3] = [16, 28, 16, 02, 02, 21, 25, 30, 02, 06, 02, 30, 25, 12, 01, 17]
- Kromosom[4] = [25, 17, 03, 24, 09, 23, 19, 28, 17, 27, 11, 02, 28, 20, 26, 31]
- Kromosom[5] = [24, 13, 15, 16, 14, 04, 08, 08, 21, 17, 03, 21, 10, 05, 04, 10]

2. Seleksi Kromosom

Hitung nilai fitness sesuai aturan penalti dari setiap kromosom yang telah dibangkitkan:

- fitness[1] = 744 - 13 = 731
- fitness[2] = 744 - 21 = 723
- fitness[3] = 744 - 7 = 737
- fitness[4] = 744 - 17 = 727
- fitness[5] = 744 - 23 = 721

Pada penelitian ini metode seleksi yang digunakan adalah Tournament Selection, untuk itu akan dibangkitkan nilai random sebanyak jumlah populasi sebagai index untuk memilih beberapa calon orang tua. Misalkan nilai random yang di generate untuk populasi pertama ada 3 (1,3,5), maka kromosom ke (1,3,5) yang dipilih kemudian dibandingkan nilai fitness dan diseleksi satu yang memiliki nilai fitness terbaik.

- kromosom[1] fitness[1] = 731
- kromosom[2] fitness[2] = 723
- kromosom[3] fitness[3] = 737
- kromosom[4] fitness[4] = 727
- kromosom[5] fitness[5] = 721

kromosom yang dipilih:

- kromosom[1] fitness[1] = 731
- kromosom[3] fitness[3] = 737
- kromosom[5] fitness[5] = 721

kromosom dengan nilai fitness terbaik:

- kromosom[3] fitness[3] = 737

langkah seleksi tersebut dilakukan sampai pada pupulasi terakhir. Dari proses seleksi diatas maka populasi kromosom baru yang didapat adalah:

- kromosom[3] fitness[3] = 737
- kromosom[5] fitness[5] = 721
- kromosom[1] fitness[1] = 731
- kromosom[3] fitness[3] = 737
- kromosom[2] fitness[2] = 723

Kromosom baru yang dihasilkan:

- kromosom[1] fitness[1] = 737
- kromosom[2] fitness[2] = 721
- kromosom[3] fitness[3] = 731

kromosom[4] fitness[4] = 737
kromosom[5] fitness[5] = 723

Populasi kromosom baru hasil seleksi:

kromosom[1] = [16, 28, 16, 02, 02, 21, 25, 30, 02, 06, 02, 30, 25, 12, 01, 17]
kromosom[2] = [24, 13, 15, 16, 14, 04, 08, 08, 21, 17, 03, 21, 10, 05, 04, 10]
kromosom[3] = [05, 08, 23, 06, 24, 16, 13, 04, 11, 27, 17, 29, 02, 13, 26, 19]
kromosom[4] = [16, 28, 16, 02, 02, 21, 25, 30, 02, 06, 02, 30, 25, 12, 01, 17]
kromosom[5] = [16, 18, 28, 16, 04, 30, 09, 25, 30, 24, 06, 30, 16, 05, 28, 19]

3. Proses Crossover

Pada proses crossover dalam penelitian ini menggunakan 2 induk terbaik yang dipilih berdasarkan nilai fitnessnya. Jika ada nilai fitness yang sama maka diambil salah satu kromosom. Dari populasi hasil seleksi di atas induk yang dipilih adalah:

kromosom[1] fitness[1] = 737
kromosom[3] fitness[3] = 731

Metode crossover yang digunakan adalah one-point crossover, sehingga posisi cut-point crossover yang dipilih menggunakan bilangan acak sebanyak jumlah gen dalam satu kromosom.

kromosom[1]  kromosom[3]

Misalkan posisi cut-point crossover didapat dari hasil random nilai [1- jumlah_gen] mendapatkan nilai 12, maka crossover akan dilakukan dari gen ke 12 sampai ke gen terakhir dalam kromosom.

kromosom[1] = [16, 28, 16, 02, 02, 21, 25, 30, 02, 06, 02, 30, 25, 12, 01, 17]



kromosom[3] = [05, 08, 23, 06, 24, 16, 13, 04, 11, 27, 17, 29, 02, 13, 26, 19]

Hasil turunan dari crossover yang dilakukan adalah:

kromosom[3] = [16, 28, 16, 02, 02, 21, 25, 30, 02, 06, 02, 29, 02, 13, 26, 19]

kromosom[1] = [05, 08, 23, 06, 24, 16, 13, 04, 11, 27, 17, 30, 25, 12, 01, 17]

Kromosom turunan akan menggantikan kromosom yang memiliki nilai fitness terjelek pada satu populasi. Dengan demikian populasi kromosom setelah mengalami proses crossover menjadi:

kromosom[1] = [16, 28, 16, 02, 02, 21, 25, 30, 02, 06, 02, 30, 25, 12, 01, 17]

kromosom[2] = [16, 28, 16, 02, 02, 21, 25, 30, 02, 06, 02, 29, 02, 13, 26, 19]

kromosom[3] = [05, 08, 23, 06, 24, 16, 13, 04, 11, 27, 17, 29, 02, 13, 26, 19]

kromosom[4] = [16, 28, 16, 02, 02, 21, 25, 30, 02, 06, 02, 30, 25, 12, 01, 17]

kromosom[5] = [05, 08, 23, 06, 24, 16, 13, 04, 11, 27, 17, 30, 25, 12, 01, 17]

4. Proses Mutasi

Misalkan nilai mutation rate (MR) yang ditentukan adalah 0,2, kemungkinan sebanyak 20% dari populasi akan termutasi. Prosesnya adalah dengan membangkitkan nilai acak R dalam range 0-1. Jika $R[1] < MR$ maka pilih kromosom 1 untuk dimutasi. Misal didapat R setiap kromosom:

R[1] = 0,103
R[2] = 0,231
R[3] = 0,642
R[4] = 0,394
R[5] = 0,117

Maka kromosom yang mengalami proses mutasi adalah kromosom[1] dan kromosom[5]. Pada penelitian ini metode mutasi yang digunakan adalah mutasi penyisipan, oleh karena ini pada setiap kromosom yang terpilih untuk termutasi akan dibangkitkan dua nilai random [1-jumlah_gen]. Misal pada kromosom pertama mendapatkan nilai 7 dan 15, maka proses mutasi akan dilakukan dengan memindahkan gen index ke 7 ke gen posisi gen index ke 15.

kromosom[1] = [16, 28, 16, 02, 02, 21, 25, 30, 02, 06, 02, 30, 25, 12, 01, 17]

Pada kromosom ke dua mendapatkan 6 dan 11.

kromosom[5] = [05, 08, 23, 06, 24, 16, 13, 04, 11, 27, 17, 30, 25, 12, 01, 17]

kromosom hasil turunan mutasi:

kromosom[1] = [16, 28, 16, 02, 02, 21, 01, 30, 02, 06, 02, 30, 25, 12, 25, 17]

kromosom[5] = [05, 08, 23, 06, 24, 17, 13, 04, 11, 27, 16, 30, 25, 12, 01, 17]

Kromosom turunan akan menggantikan kromosom induknya. Dengan demikian populasi kromosom setelah mengalami proses mutasi:

kromosom[1] = [16, 28, 16, 02, 02, 21, 01, 30, 02, 06, 02, 30, 25, 12, 25, 17]

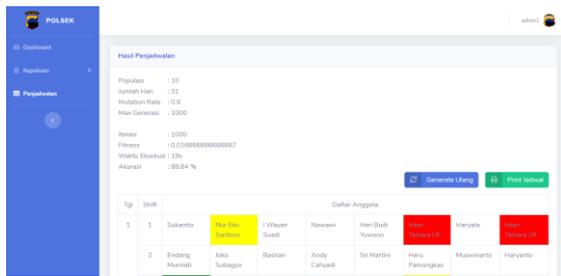
kromosom[2] = [16, 28, 16, 02, 02, 21, 25, 30, 02, 06, 02, 29, 02, 13, 26, 19]

kromosom[3] = [05, 08, 23, 06, 24, 16, 13, 04, 11, 27, 17, 29, 02, 13, 26, 19]
 kromosom[4] = [16, 28, 16, 02, 02, 21, 25, 30, 02, 06, 02, 30, 25, 12, 01, 17]
 kromosom[5] = [05, 08, 23, 06, 24, 17, 13, 04, 11, 27, 16, 30, 25, 12, 01, 17]

Semua proses di atas akan berulang sampai ditemukan nilai fitness paling baik yaitu paling mendekati 0 dan akan berhenti sampai sejumlah generasi yang ditentukan sebelumnya.

IV. Implementasi

Berikut adalah tampilan halaman atau form antar muka dari sistem penjadwalan shift kerja anggota kepolisian yang sudah dirancang dalam bab sebelumnya.



Gambar 4.1 Halaman Hasil Penjadwalan

Gambar 4.1 merupakan halaman untuk menampilkan hasil generate proses penjadwalan menggunakan GA. Ditampilkan hasil nilai fitness, banyaknya iterasi yang diperlukan, lamanya waktu generate, serta hasil jadwal yang optimal.

4.1 Pengujian

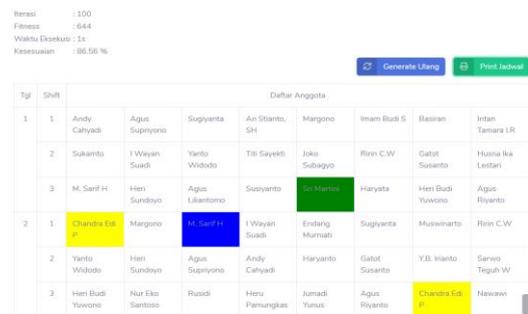
Pengujian pada penelitian ini dilakukan sesuai parameter GA yang digunakan yaitu ukuran populasi, *mutation rate*, dan banyak generasi. Masing-masing parameter diuji dengan beberapa variasi nilai untuk mendapatkan kombinasi jadwal yang paling optimal. Semua pengujian dilakukan mengacu pada hasil nilai *fitness* terbaik yang dihasilkan dan tingkat kesesuaian yang dicapai.

Pada sistem yang dibuat, untuk memudahkan menganalisis hasil dari generate proses GA, peneliti membuat semacam *bot* / sistem pengecekan. *Bot* ini berfungsi mengecek jadwal hasil generate GA sudah sesuai atau masih ada benturan. Kemudian untuk memudahkan mengetahui gen mana yang bertabrakan, peneliti memberi warna gen jadwal yang bertabrakan sesuai dengan *constraint* penalti yang telah ditentukan di awal. Gen jadwal yang tidak terdapat benturan akan

berwarna putih, sedangkan untuk gen jadwal yang masih ada benturan diberi warna sesuai dengan *constraint* penalti yang telah ditentukan.

Penalti	Warna Gen Jadwal
Benturan jadwal <i>shift</i> anggota polisi yang sama dalam satu <i>shift</i> .	Merah
Benturan jadwal <i>shift</i> anggota polisi dalam satu hari.	Kuning
Ada anggota polisi wanita mendapat <i>shift</i> di malam hari.	Hijau
Benturan jadwal <i>shift</i> anggota polisi yang sudah <i>shift</i> malam masih mendapat jadwal kesokan harinya.	Biru

Pada sistem yang dibuat, contoh hasil model jadwal yang masih terdapat benturan setelah dilakukan generate dapat dilihat pada Gambar 4.1.



Gambar 4.1 Halaman Hasil Penjadwalan

Dapat dilihat pada hasil jadwal di atas, pada gen jadwal yang terdapat warna selain warna putih menunjukkan benturan. Ada beberapa gen jadwal yang masih berbenturan dan diberi warna sesuai *constraint* penalti pada tabel sebelumnya. Setelah mengetahui benturan yang masih terjadi, peneliti mencoba memberi alternatif solusi dengan cara mengosongkan gen jadwal yang masih terjadi benturan.

Fitness : 644
 Iterasi : 100
 Waktu Eksekusi : 1s
 Kesesuaian : 86.56 %

Tgl	Shift	Daftar Anggota									
1	1	Andy Cahyadi	Agus Supriyono	Sugiyanta	Ari Siantoro SH	Margono	Imam Budi S	Basiran	Istana Tamara L.R		
	2	Sukanto	I Wayan Suadi	Yanto Widodo	Titi Sayekti	Joko Subagyo	Ririn C.W	Gatot Susanto	Huma Iba Lestari		
	3	M. Sarif H	Heri Sandiyo	Agus Liliantomo	Susiyanto		Haryata	Heri Budi Yuwono	Agus Riyanto		
2	1		Margono		I Wayan Suadi	Endang Murnani	Sugiyanta	Muswinarto	Ririn C.W		
	2	Yanto Widodo	Heri Sandiyo	Agus Supriyono	Andy Cahyadi	Haryanto	Gatot Susanto	Y.B. Irianto	Serwo Teguh W		
	3	Heri Budi Yuwono	Nur Eko Santoso	Rusidi	Heru Pamungkas	Jumadi Yunus	Agus Riyanto		Nawawi		

Gambar 4.1 Halaman Hasil Penjadwalan

ketika jadwal hasil generate GA dicetak maka gen jadwal yang masih terdapat benturan akan kosong. Langkah tersebut dilakukan agar petugas yang membagi jadwal *shift* tetap dapat mengisi nama anggota polisi yang masih berbenturan jadwalnya dengan nama anggota lain.

4.2 Hasil Pengujian Pemilihan Bibit Populasi Terbaik

Pengujian pertama yaitu memilih bibit populasi yang paling baik agar menghasilkan solusi jadwal yang sesuai atau dapat dikatakan paling optimal. Pengujian ini dilakukan dengan memilih bibit populasi yang memiliki nilai *fitness* dan tingkat kesesuaian terbaik untuk dijadikan calon populasi pada proses GA selanjutnya. Peneliti menggunakan ukuran populasi sebanyak 10 populasi, *mutation rate* 0.3 dengan generasi sebanyak 1000 dan *mutation rate* 0.5 generasi sebanyak 5000. Pengujian tersebut dilakukan sebanyak 10 kali untuk mendapatkan satu bibit populasi dengan nilai *fitness* dan tingkat kesesuaian terbaik.

Tabel 4.1 Hasil pengujian bibit dengan *mutation rate* 0.3 dan 1000 generasi

Pengujian	<i>Fitness</i>	Kesesuaian (%)	Waktu Eksekusi
1	647	86,96	15s
2	631	84,81	16s
3	641	86,16	15s
4	644	86,56	15s
5	656	88,17	15s
6	650	87,37	16s
7	660	88,71	15s
8	650	87,37	16s
9	651	87,5	15s
10	652	87,63	17

Berdasarkan hasil pengujian pada Tabel 4.1, pengujian ke 7 menghasilkan nilai paling tinggi dari 10 pengujian yang dilakukan dengan *fitness* sebesar 660 dan tingkat kesesuaian 88.71%. Dari hasil nilai tersebut, populasi pada pengujian ke 7 menjadi bibit yang paling baik untuk calon populasi pada proses pengujian selanjutnya.

Tabel 4.2 Hasil pengujian bibit dengan *mutation rate* 0.5 dan 5000 generasi

Pengujian	<i>Fitness</i>	Kesesuaian (%)	Waktu Eksekusi
1	640	86,02	1m 2s
2	634	85,22	1m 5s
3	648	87,1	1m 8s
4	643	86,42	1m 2s

5	642	86,29	1m 3s
6	646	86,83	1m 3s
7	647	86,96	1m 2s
8	638	85,75	1m 3s
9	640	86,2	1m 6s
10	647	86,96	1m 5s

Berdasarkan hasil pengujian pada Tabel 4.2, pengujian ke 3 menghasilkan nilai paling tinggi dari 10 pengujian yang dilakukan dengan *fitness* sebesar 648 dan tingkat kesesuaian 87,1 %. Dari hasil nilai tersebut, populasi pada pengujian ke 3 menjadi bibit yang paling baik untuk calon populasi pada proses pengujian selanjutnya.

4.3 Hasil Pengujian *Mutation Rate*

Pengujian kedua yaitu menguji beberapa nilai *mutation rate* terhadap nilai *fitness* yang dihasilkan serta tingkat kesesuaian yang dicapai. Pengujian ini dilakukan untuk mengetahui *mutation rate* terbaik yang dapat menghasilkan solusi jadwal yang optimal. Nilai generasi digunakan sebesar 1000 dan 5000 generasi dan jumlah populasi sebesar 10 populasi. Populasi yang dibunakan sesuai dengan hasil pengujian sebelumnya dengan harapan dapat menghasilkan solusi jadwal yang lebih optimal. Nilai *mutation rate* yang diuji mulai dari 0.1, 0.3, 0.5, 0.7 dan 0.9.

Tabel 4.3 Hasil pengujian *mutation rate* dengan 1000 generasi

<i>Mutation Rate</i>	<i>Fitness</i>	Kesesuaian (%)	Waktu Eksekusi
0,1	660	88,71	13s
0,3	660	88,71	15s
0,5	664	89,25	17s
0,7	663	89,11	19s
0,9	662	88,98	19s

Pengujian pada Tabel 4.3, *mutation rate* dengan nilai 0.5 dapat menghasilkan nilai paling tinggi daripada nilai *mutation rate* lainnya yaitu dengan hasil *fitness* 664 dan tingkat kesesuaian 89.25 %. Berdasarkan hasil tersebut, nilai *mutation rate* 0.5 menjadi acuan untuk parameter pengujian berikutnya agar mendapat hasil yang lebih optimal lagi.

Tabel 4.4 Hasil pengujian *mutation rate* dengan 5000 generasi

<i>Mutation Rate</i>	<i>Fitness</i>	Kesesuaian (%)	Waktu Eksekusi
0.1	648	87.1	1m 8s

0.3	648	87.1	1m 15s
0.5	648	87.1	1m 20s
0.7	648	87.1	1m 26s
0.9	648	87.1	1m 33s

Pengujian pada Tabel 4.4, tidak ada peningkatan nilai *fitness* dan tingkat kesesuaian meskipun *mutation rate* telah diubah mulai dari 0,1 hingga 0,9. Meskipun tidak ada peningkatan nilai *fitness*, dari hasil ini peneliti mencoba untuk mengambil nilai *mutation rate* tertinggi yang ada yaitu 0,9 untuk diuji di penelitian selanjutnya.

4.4 Hasil Pengujian Generasi

Pengujian terakhir yaitu menguji beberapa generasi terhadap nilai *fitness* yang dihasilkan dan tingkat kesesuaian yang dicapai. Pengujian ini dilakukan untuk mengetahui seberapa banyak generasi yang diperlukan agar mencapai hasil jadwal yang optimal. Populasi yang digunakan berjumlah 10 populasi hasil dari pengujian pemilihan bibit populasi dan nilai *mutation rate* yang digunakan sebesar 0,5 untuk bibit populasi dengan awal 1000 generasi dan *mutation rate* 0,9 untuk bibit populasi dengan awal 5000 generasi. Jumlah generasi yang diuji mulai dari 100 sampai 1000000 generasi.

Tabel 4.5 Hasil pengujian generasi dengan *mutation rate* 0.5

Generasi	<i>Fitness</i>	Kesesuaian (%)	Waktu Eksekusi
100	660	88,71	1s
500	660	88,71	8s
1000	664	89,25	13s
5000	663	89,11	1m 9s
10000	665	89,38	2m 4s
50000	665	89,38	10m 11s
100000	665	89,38	22m 47s
500000	665	89,38	1h 57m 58s
1000000	665	89,38	4h 11m 29s

Berdasarkan pengujian Tabel 4.5, dari 100 hingga 10000 generasi terjadi peningkatan nilai *fitness* dan tingkat kesesuaian. Tetapi ketika generasi melewati 10000 nilai *fitness* dan tingkat kesesuaian belum menunjukkan peningkatan nilai. Pengujian generasi terus dilakukan sampai 1000000 generasi dengan nilai *fitness* 665 dan tingkat kesesuaian 89,38%.

Tabel 4.6 Hasil pengujian generasi dengan *mutation rate* 0.9

Generasi	<i>Fitness</i>	Kesesuaian (%)	Waktu Eksekusi
100	644	86,56	1s
500	646	86,83	7s
1000	648	87,1	14
5000	648	87,1	1m 15s
10000	648	87,1	2m 50s
50000	648	87,1	11m 50s
100000	648	87,1	23m 36s
500000	650	87,37	2h 9m 24s
1000000	650	87,37	4h 38m 29s

Berdasarkan pengujian Tabel 4.6, dari 100 hingga 500 generasi terjadi peningkatan nilai *fitness* dan tingkat kesesuaian. Tetapi ketika generasi melewati 500 nilai *fitness* dan tingkat kesesuaian belum menunjukkan peningkatan nilai. Pengujian terus dilakukan sampai pada generasi 500000 nilai *fitness* dan tingkat kesesuaian kembali naik. Pengujian dilakukan hingga 1000000 generasi dengan nilai *fitness* 650 dan tingkat kesesuaian 87,37%.

V. Penutup

5.1 Kesimpulan

Kesimpulan dari implementasi algoritma genetika dalam penjadwalan *shift* kerja anggota kepolisian di Polsek Magelang Tengah adalah sebagai berikut:

1. Algoritma genetika terbukti dapat digunakan untuk menyelesaikan permasalahan penjadwalan *shift* kerja di Polsek Magelang Tengah.
2. Hasil jadwal dari generate sistem yang menggunakan algoritma genetika cukup sesuai dengan peraturan yang ada di Polsek Magelang Tengah dan dibuktikan dengan hasil pengujian yang telah peneliti lakukan.
3. Pada penelitian ini algoritma genetika menggunakan representasi kromosom dengan panjang 744 gen yang mewakili jadwal dalam satu bulan atau 31 hari. Populasi yang digunakan berjumlah 10 populasi diambil dari hasil pengujian selama 10 kali. Metode seleksi yang digunakan yaitu *Tournament Selection*, metode *crossover* menggunakan *one-cut-point* dan metode mutasi menggunakan penyisipan. Dari ketiga metode tersebut, peneliti hanya melakukan variasi pada nilai *mutation rate* mulai dari rentan 0,1, 0,3, 0,5, 0,7 dan 0,9 serta batas generasi mulai dari rentan 1000 –

- 1000000 generasi. nilai *crossover rate* tidak divariasikan dalam penelitian ini karena sudah pasti kromosom yang dikawin silangkan merupakan 2 kromosom terbaik dalam setiap generasi.
4. Berdasarkan hasil penelitian, pada pengujian pertama dengan nilai awal *mutation rate* 0,3 dan batas generasi 1000 menghasilkan nilai *fitness* tertinggi 665 dengan tingkat kesesuaian 89,38% pada generasi 10000 keatas. Sedangkan pengujian dengan nilai *mutation rate* awal 0.5 dan batas generasi 5000 menghasilkan nilai *fitness* tertinggi 660 dengan tingkat kesesuaian 87,37% pada generasi 500000 keatas.
 5. Pemilihan bibit populasi awal mempengaruhi kualitas populasi ketika dilakukan pengujian dengan variasi nilai *mutation rate* dan batas generasi. Semakin tinggi *mutation rate* belum tentu hasil yang didapatkan optimal. Sementara semakin tinggi batas generasi, hasil nilai *fitness* dan tingkat kesesuaian bisa semakin baik. Tetapi hal tersebut mengakibatkan proses generate jadwal memakan waktu eksekusi yang lama.

5.2 Saran

Beberapa saran yang dapat untuk pengembangan sistem yang telah dibuat diantaranya adalah:

1. Pada penelitian penjadwalan menggunakan GA selanjutnya, untuk mendapatkan hasil yang semakin optimal perlu dilakukan lebih banyak variasi lagi pada metode seleksi, metode *crossover* dan metode mutasi yang digunakan. Ketiga metode tersebut juga dikombinasikan dengan banyaknya populasi serta batas maksimal generasi dapat divariasikan lagi untuk mencapai hasil optimal yang diinginkan.
2. Untuk mendapatkan hasil jadwal yang optimal, waktu eksekusi program cukup memakan waktu yang lama. Maka dari itu perlu dipertimbangkan hardware / komputer yang digunakan untuk melakukan generate jadwal.
3. Peneliti memberi alternatif solusi dengan cara mengosongkan gen jadwal yang masih terjadi benturan agar petugas yang membagi jadwal *shift* tetap dapat mengisi nama anggota polisi yang masih berbenturan jadwalnya dengan nama anggota lain

DAFTAR PUSTAKA

- [1] Astiti, W. F., Ratnawati, D. E., & Fauzi, M. A. (2018). Penjadwalan Dinas Pegawai Menggunakan Algoritma Evolution Strategies pada PT . Kereta Api Indonesia (KAI) DAOP 7 Stasiun Besar Kediri. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2473–2479.
- [2] Effriandi, A. R., Zenda, A., & Rachmansyah. (2010). Rancang Bangun Aplikasi Penjadwalan Shift Kerja Lembur Menggunakan Algoritma Ant Colony Optimization (Studi Kasus : PUSRI 1B). *Jurnal Teknik Informatika*, 1–12.
- [3] Ginting, W. C. (2017). *Implementasi Algoritma Genetika dalam Penjadwalan Shift Kerja di Call Center Telkomsel*. Universitas Sumatera Utara.
- [4] Haupt, R. L., & Haupt, S. E. (2004). *Practical Genetic Algorithms* (2 ed.). John Wiley & Sons.
- [5] Ilmi, R. R., Mahmudy, W. F., & Ratnawati, D. E. (2015). Optimasi Penjadwalan Perawat Menggunakan Algoritma Genetika. *Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya*, 5(13), 1–8.
- [6] Jogiyanto, H. M. (2006). *Analisis dan Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: ANDI Publisher.
- [7] Stevenson, W. J., & Chuong, S. C. (2014). *Manajemen Operasi Perspektif Asia* (9 ed.). Jakarta Salemba Empat.
- [8] Susandi, D., & Milana, L. (2015). Perancangan dan Pembuatan Aplikasi Penyusunan Jadwal Kerja Dinas Jaga Perawat IGD Menggunakan Algoritma TPB. *Metris*, 16, 29–34. <https://doi.org/1411-3287>
- [9] Sutanta, E. (2011). *Basis Data Dalam Tinjauan Konseptual*. Yogyakarta: ANDI Publisher.
- [10] Turban, E. (1995). *Decision support and expert systems: management support systems* (4 ed.). Prentice Hall.
- [11] Yanto, R. (2016). *Manajemen Basis Data Menggunakan MySQL*. Yogyakarta: Deepublish.