

# WIKARA

WIKARA NATIONAL ECONOMIC AND SOCIAL RESEARCH DEVELOPMENT

Laman Jurnal: [ejournal.stie-wikara.ac.id](http://ejournal.stie-wikara.ac.id)

ISSN : 2721-8120 (p) ,2797-3247 (e)

## MANAJEMEN PENJADWALAN DIGITAL UNTUK OPTIMASI ANTRIAN SERVICE MOBIL TOYOTA DI AUTO2000 PASTEUR

Soca Ramdhani<sup>1\*</sup>, Phitsa Mauliana<sup>2</sup>, Wildan Wiguna<sup>3</sup>

Universitas Sangga Buana YPKP<sup>1,2,3</sup>, Indonesia, Jawa Barat

Email: [ramdhanisocal15@gmail.com](mailto:ramdhanisocal15@gmail.com)

ARTICLE INFO	ABSTRACT
<p><b>Artikel History:</b></p> <p>Received:</p> <p>Revised:</p> <p>Published:</p> <hr/> <p><b>Keywords:</b>                      Digital Scheduling, Genetic Algorithm, Service Queue, Optimization, Web System</p>	<p><i>This study aims to develop and implement a digital scheduling management system to optimize the Toyota car service queue at Auto2000 Pasteur using a Genetic Algorithm approach. The workshop frequently experiences morning rush conditions and bottlenecks due to uneven technician workload distribution and overlapping service schedules involving mechanics, stalls, vehicles, and service time. Although appointment services are available, scheduling is still manually estimated, leading to inefficiencies and uncertainty in service lead time. This research employs a system development methodology using the waterfall model, including requirement analysis, system design, implementation, testing, and support. Data were collected through observation, interviews, and literature study. The optimization technique applies a Genetic Algorithm by representing service schedules as chromosomes, evaluating fitness values based on clashes between mechanics, stalls, vehicles, and time, and performing selection using the roulette wheel method, crossover, mutation, and replacement processes. The objective function maximizes scheduling feasibility by minimizing conflicts. The system is implemented as a web-based application using PHP, HTML, CSS, JavaScript, and MySQL database. Testing through black-box methods confirms that all system functions operate as expected. The results indicate that the proposed system successfully generates optimized service schedules without conflicts and distributes workload more evenly among technicians. Therefore, the digital scheduling system enhances operational efficiency, reduces service queue bottlenecks, and improves customer service performance at Auto2000 Pasteur.</i></p>
INFO ARTIKEL	ABSTRAK
<p><b>Riwayat Artikel:</b></p> <p>Diterima :</p>	<p>Penelitian ini bertujuan untuk mengembangkan dan mengimplementasikan sistem manajemen penjadwalan digital guna mengoptimasi antrian service mobil Toyota di Auto2000</p>

# WIKARA

WIKARA NATIONAL ECONOMIC AND SOCIAL RESEARCH DEVELOPMENT

Laman Jurnal: [ejournal.stie-wikara.ac.id](http://ejournal.stie-wikara.ac.id)

ISSN : 2721-8120 (p) ,2797-3247 (e)

Direvisi:

Dipublikasikan :

**Kata kunci:**

Penjadwalan Digital, Algoritma Genetika, Antrian Service, Optimasi, Sistem Web

Pasteur menggunakan pendekatan Algoritma Genetika. Bengkel sering mengalami kondisi morning rush dan bottleneck akibat pembagian pekerjaan teknisi yang tidak merata serta bentrok jadwal antara mekanik, stall, kendaraan, dan waktu pelayanan. Meskipun telah tersedia layanan appointment, proses estimasi penjadwalan masih dilakukan secara manual sehingga menimbulkan ketidakefisienan dan ketidakpastian lead time pelayanan. Metode penelitian menggunakan model pengembangan sistem waterfall yang meliputi analisis kebutuhan, perancangan, implementasi, pengujian, dan dukungan sistem. Teknik pengumpulan data dilakukan melalui observasi, wawancara, dan studi pustaka. Proses optimasi menggunakan Algoritma Genetika dengan merepresentasikan jadwal service sebagai kromosom, menghitung nilai fitness berdasarkan jumlah bentrok mekanik, stall, kendaraan, dan waktu, serta melakukan seleksi menggunakan metode roulette wheel, crossover, mutasi, dan replace. Fungsi objektif digunakan untuk memaksimalkan kelayakan jadwal dengan meminimalkan konflik. Sistem diimplementasikan berbasis web menggunakan PHP, HTML, CSS, JavaScript, dan basis data MySQL. Pengujian dengan metode black box menunjukkan seluruh fungsi sistem berjalan sesuai harapan. Hasil penelitian menunjukkan bahwa sistem mampu menghasilkan jadwal service yang optimal tanpa bentrok serta mendistribusikan pekerjaan teknisi secara lebih merata. Dengan demikian, sistem penjadwalan digital ini meningkatkan efisiensi operasional, mengurangi bottleneck antrian, dan meningkatkan kualitas pelayanan di Auto2000 Pasteur.

## PENDAHULUAN

Servis mobil merupakan jenis jasa yang dikategorikan sebagai *low contact services* yang membutuhkan kontak pelanggan dengan intensitas yang rendah (Fatihudin & Firmansyah, 2019). Salah satu jenis servis mobil yaitu perawatan berkala dengan jadwal yang ditetapkan oleh beberapa faktor seperti model, usia, atau status kendaraannya. Sedangkan interval servis ditentukan sesuai dengan jarak tempuh dan periode yang telah dilalui sejak servis sebelumnya (Tampubolon, Fahmi & Batu, 2020). Bengkel dalam industri otomotif merupakan tempat layanan jasa perbaikan dan perawatan kendaraan. Sebuah antrian tidak bisa dihindari dikarenakan layanannya memerlukan waktu yang relatif lama dan tahapan pekerjaannya yang cukup banyak, sehingga diperlukan *appointment* sebelum hari pengerjaan (Zulfa, Mujibah & Rajaguguk, 2020). Pada antrian terdapat suatu kondisi dimana penerima layanan yang jumlahnya lebih besar dibandingkan dengan pemberi layanan. Kondisi ini dapat mengakibatkan penumpukan penerima layanan dan ujungnya akan terjadi *bottleneck* (Setyawati & Maulachela, 2020).

Auto2000 Pasteur merupakan salah satu dealer resmi mobil Toyota di Bandung yang menyediakan layanan perbaikan kendaraan yang berlokasi di Jl. Dr. Djunjunan No.192, Sukagalih, Kec. Sukajadi, Kota Bandung, Jawa Barat 40163. Pada *dealer* ini terdapat bengkel dengan fasilitas *General Repair* (GR) yaitu layanan purna jual yang menawarkan jasa perbaikan berupa servis perawatan berkala maupun perbaikan umum berkaitan dengan mesin, sasis atau kerangka kendaraan, serta kelistrikan untuk seluruh tipe mobil Toyota. Layanan GR mengedepankan kepuasan pelanggan dengan terus menjaga kualitas, biaya, serta *lead time* pengerjaan. Pada kegiatan observasi didapatkan beberapa kendala dari pengamatan secara langsung di Auto 2000 Pasteur terkait dengan penjadwalan *service*. Seorang *Foreman* terkadang memperhatikan sebagian mekanik atau teknisi yang terlalu sibuk melakukan perawatan kendaraan, namun di sisi lain ada juga teknisi yang tidak kebagian pekerjaan. Kondisi ini biasa disebut dengan *morning rush* atau kesibukan di waktu tertentu saja, kemudian diperparah dengan kondisi *bottleneck* yang menyebabkan terjadinya antrian kendaraan beserta pelanggannya di ruang tunggu dengan ketidakpastian waktu pelayanan. Pada bengkel Auto2000 Pasteur sebenarnya sudah disediakan layanan *appointment* via telepon yang bisa dilakukan beberapa sehari sebelum kunjungan, namun perlu dilakukan estimasi lagi mengenai *lead time* pengerjaan layanannya oleh *Service Advisor*. Berdasarkan hal tersebut, diperlukan sebuah solusi untuk mengatasi permasalahan penjadwalan antrian servis kendaraan.

Penjadwalan layanan *service* kendaraan dapat diestimasi menggunakan metode komputasi Algoritma Genetika yang ditanamkan dalam sistem terkomputerisasi. Terdapat beberapa penelitian mengenai penjadwalan yang berkaitan dengan layanan bagi pelanggan seperti implementasi

Algoritma Genetika untuk penjadwalan *customer service*. Algoritma Genetika menghabiskan waktu yang lebih sedikit yaitu hanya membutuhkan 30% dalam pembuatan jadwal *customer service* (Damayanti, et al, 2017). Kemudian terdapat penerapan Algoritma Genetika dalam optimasi penjadwalan proyek. Algoritma Genetika terbukti dapat memberikan hasil yang cepat dan mendekati optimal dalam membentuk jadwal proyek (Ferdyawan & Hajjah, 2020). Dari beberapa permasalahan yang telah dijelaskan sebelumnya, maka dapat dirangkum bahwa dibutuhkan suatu sistem penjadwalan *service* kendaraan berbasis digital di Auto2000 Pasteur. Kemudian solusi untuk optimasi yang diajukan pada kasus penjadwalan yang diteliti akan diterapkan Algoritma Genetika pada system penjadwalan tersebut. Suatu sistem penjadwalan dapat diimplementasikan menggunakan pemrograman *website* (Kurnia & Ardiansyah, 2020). Suatu sistem *website* biasanya disimpan di dalam komputer *web server* (Wiguna & Mauliana, 2021). Sehingga pada penelitian ini bertujuan untuk menerapkan Algoritma Genetika pada sistem penjadwalan digital untuk *service* kendaraan berbasis *web* di *dealer* resmi mobil Toyota Auto2000 Pasteur.

## METODE PENELITIAN

Terdapat beberapa jenis metode penelitian yang digunakan pada penelitian ini yaitu teknik pengumpulan data, metode pengembangan sistem penjadwalan, teknik optimasi, Algoritma Genetika, hingga contoh kasus perhitungannya.

### Teknik Pengumpulan Data

Pengambilan data *service* kendaraan dari bengkel Auto2000 Pasteur menggunakan beberapa teknik pengumpulan data (Hidayat & Wiguna, 2021), yaitu:

#### 1. Observasi

Pengamatan langsung dilakukan pada bengkel resmi Auto2000 Pasteur dan mengamati sistem yang sedang berjalan untuk mendapatkan data penelitian.

#### 2. Wawancara

Proses wawancara dilakukan dengan kepala bengkel, *service advisor*, teknisi, serta beberapa pelanggan *service* Auto2000 Pasteur.

#### 3. Studi Pustaka

Mempelajari beberapa jurnal dan buku yang terkait dengan sistem penjadwalan, Algoritma Genetika, perawatan berkala, *service* kendaraan, serta pemrograman *website* dengan basis data MySQL.

## Metode Pengembangan Sistem

Model pengembangan sistem penjadwalan *service* kendaraan pada penelitian ini menggunakan model *waterfall* yang terbagi menjadi beberapa tahapan (Wiguna & Alawiyah, 2019), yang dapat diuraikan sebagai berikut ini:

### 1. Analisis

Pada tahap ini dianalisa kebutuhan-kebutuhan yang akan digunakan yaitu pendataan perawatan berkala kendaraan yang dilakukan *service* beserta pemilikinya oleh *Service Advisor*, data mekanik dan *stall* bengkel yang dapat diketahui dari *Foreman*, maupun SOP yang sedang berjalan guna menentukan solusi pengembangan sistem penjadwalan.

### 2. Perancangan

Dijelaskan kebutuhan-kebutuhan sistem yang berhubungan dengan desain sistem penjadwalan *service* pada perawatan berkala mobil Toyota. *Tools* yang digunakan yaitu rancangan basis data menggunakan *Entity Relationship Diagram* (ERD), serta arsitektur perangkat lunak atau *software architecture* pada penelitian ini menggunakan *Unified Modeling Language* (UML).

### 3. Implementasi

Pada tahap ini ditentukan *platform* pemrograman *website* yang akan digunakan yaitu menggunakan bahasa pemrograman PHP, *Java Script*, HTML, dan CSS. Kemudian penerapan basis data menggunakan MySQL.

### 4. Pengujian

Proses pengujian yang akan dilakukan dengan *black box testing* yaitu masukan dari program, apakah menghasilkan *output* sesuai dengan yang diharapkan oleh pengguna sistem.

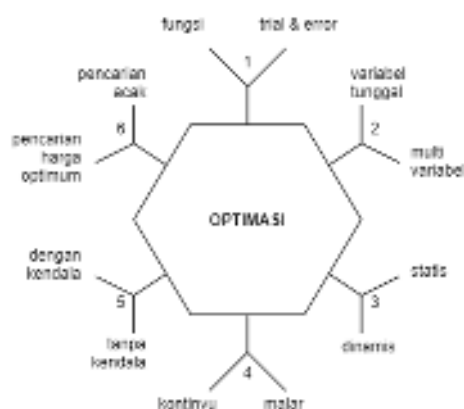
### 5. Dukungan

Tahap akhir pengembangan sistem yang sudah dibuat diperlukan dalam mengantisipasi perubahan sistem yang bersangkutan dengan *hardware* dan *software* yang akan digunakan bagi perusahaan sebagai tempat riset.

## Teknik Optimasi

Optimasi adalah proses menyelesaikan suatu permasalahan tertentu supaya berada pada kondisi yang paling menguntungkan dari suatu sudut pandang. Masalah yang harus diselesaikan berkaitan erat dengan data-data yang dapat dinyatakan dalam satu atau beberapa variabel. Biasanya yang disebut keuntungan berhubungan dengan pencarian minimum dan maksimum, bergantung pada sudut pandang yang digunakan. Proses optimasi dapat diklasifikasikan menjadi enam kategori.

Pembagian tersebut tidak dapat dianggap sepenuhnya membagi-bagi proses oprimasi persis menjadi enam kategori, karena setiap cabang saling dapat terkait (Zukhri, 2014).



Sumber: (Zukhri, 2014)  
Gambar 1. Klasifikasi Optimasi

Pada Gambar 1 di atas menunjukkan enam kategori pada klasifikasi optimasi (Zukhri, 2014), dapat dijelaskan sebagai berikut:

1. Optimasi dengan cara *trial and error* berhubungan dengan proses untuk menyesuaikan nilai variabel input yang membuat perubahan output tanpa pengetahuan yang banyak mengenai prosesnya.
2. Optimasi satu dimensi melibatkan sebuah variabel, sedangkan optimasi yang melibatkan banyak variabel disebut dengan multidimensi. Proses optimasi menjadi bertambah sulit dengan bertambahnya dimensi. Optimasi multidimensi dapat dianggap sebagai rangkaian optimasi dari satu dimensi.
3. Optimasi dinamis mempunyai output yang merupakan fungsi waktu, sedangkan optimasi statis mempunyai output yang independent terhadap waktu.
4. Optimasi diskret adalah optimasi yang melibatkan variabel yang mempunyai sejumlah variasi nilai yang terbatas. Sedangkan optimasi malar adalah optimasi yang banyaknya variasi nilai variabelnya tidak berhingga.
5. Optimasi dengan kendala menggabungkan kesamaan variabel dan ketidaksamaan variabel ke dalam fungsi objektif. Sedangkan optimasi tanpa kendala mengizinkan variabel mempunyai nilai berapapun juga.
6. Optimasi yang penyelesaiannya dicari secara acak disebut dengan optimasi acak. Optimasi yang penyelesaiannya dicari dari sebuah nilai variabel awal tertentu disebut dengan pencarian nilai optimum.

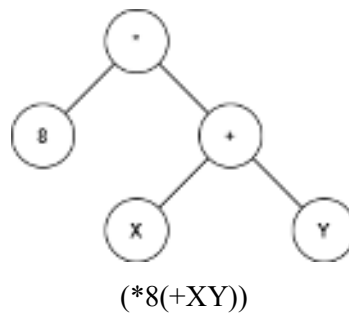
## Algoritma Genetika

Algoritma genetika meniru kejadian alam untuk menyelesaikan suatu masalah yaitu dengan menggabungkan teori reproduksi, seleksi alam, dan teori evolusi Darwin. Algoritma genetika menyimbolkan sebuah kromosom yang menentukan karakter sebagai sebuah individu dan kualitas dari setiap kromosom yang disebut dengan *fitness*. Setiap individu mewakili sebuah solusi dan kumpulan dari sejumlah individu yang disebut dengan populasi (Krisnanda, 2020).

Terdapat langkah-langkah dari desain algoritma genetika yang digunakan untuk menyelesaikan masalah optimasi penjadwalan (Krisnanda, 2020), sebagai berikut:

1. Pengkodean, mengkodekan solusi menjadi individu. Pengkodean dilakukan dengan tujuan untuk membentuk kromosom dari individu, kromosom inilah yang akan dievaluasi terhadap *objective function* untuk menentukan *fitness* suatu individu. Proses pengkodean suatu individu dapat dilakukan dengan beberapa metode, pemilihan metode yang tepat dapat mempermudah proses selanjutnya sesuai dengan karakteristik permasalahan. Macam-macam pengkodean pada algoritma genetika antara lain:

- a. *Binary encoding*, metode ini merepresentasikan kromosom suatu individu dalam bentuk string biner, dimana setiap bit terdiri dari bilangan 0 atau 1. Pada masalah linier, kromosom yang dihasilkan biasanya adalah nilai biner dari sebuah nilai desimal.
- b. *Permutation encoding*, metode pengkodean merepresentasikan kromosom suatu individu dalam bentuk urutan dari angka-angka atau huruf. Inti dari metode pengkodean ini tidak boleh terjadi penghilangan atau pengulangan dari urutan angka atau huruf yang dibentuk.

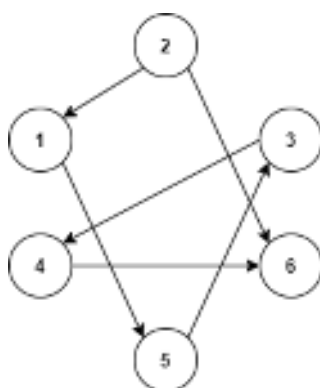


Sumber: (Krisnanda, 2020)

Gambar 2. *Permutation Encoding*

Pada Gambar 2. merupakan contoh kromosom yang dihasilkan oleh *permutation encoding* dari rute kunjungan dengan node 1 sebagai node asal.

- c. *Tree encoding*, bentuk pengkodean untuk kromosom dengan setiap kromosom akan mewakili sebuah *tree* dari sebuah objek.
- d.



1 - 5 - 3 - 4 - 6 - 2

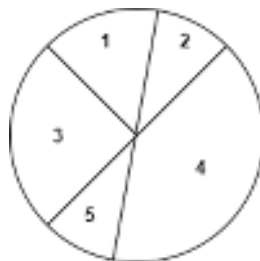
Sumber: (Krisnanda, 2020)

Gambar 3. *Tree Encoding*

Pada Gambar 3 menunjukkan contoh kromosom yang dihasilkan oleh *route tree encoding* dari sebuah objek *tree*.

- e. *Value encoding*, bentuk pengkodean untuk kromosom dengan setiap kromosom yang diwakili oleh suatu nilai. Nilai dapat berupa apa saja yang berhubungan dengan masalah yang ada misalnya bilangan bulat, *real*, karakter, atau obyek lainnya.
2. Populasi awal, untuk populasi awal sebanyak N individu. Setiap individu dalam populasi awal harus memiliki *feasible solution* dari *search space* permasalahan yang akan diselesaikan. Setiap individu yang dihasilkan harus sesuai dengan karakteristik dari metode pengkodean.
3. *Fitness*, menghitung nilai *fitness* dari setiap individu dalam populasi. Setiap individu memiliki *fitness* yang menunjukkan kualitas dari individu tersebut. Nilai *fitness* suatu individu diperoleh dengan mengevaluasi kromosom suatu individu terhadap *objective function*. Suatu individu tersebut memiliki kesempatan yang lebih besar untuk berpartisipasi pada proses reproduksi dan menghasilkan individu baru untuk generasi berikutnya.
4. Seleksi, pilih individu yang berpartisipasi dalam proses seleksi berdasarkan nilai *fitness* masing-masing. Seleksi bertujuan untuk memilih individu yang akan menjadi parent dari populasi yang ada sesuai dengan *fitness* individu tersebut. Proses seleksi dilakukan berdasarkan *fitness* dari setiap individu di dalam populasi. Individu yang mempunyai *fitness* baik mempunyai kemungkinan untuk terpilih lebih besar. Terdapat beberapa metode seleksi yang sering digunakan pada algoritma genetika diantaranya:
  - a. *Elistism Selection*, ketika membentuk populasi dengan *crossover* dan mutasi, ada kemungkinan kromosom yang paling baik hilang. Untuk mengatasi masalah ini, metode elistism memasukkan kromosom dengan nilai *fitness* paling baik atau beberapa kromosom dengan nilai *fitness* yang tinggi dari generasi lama ke generasi baru. Kemudian sisa kromosom dalam generasi baru diperoleh dengan cara reproduksi biasa.

- b. *Roulette wheel selection*, kromosom dipilih berdasarkan nilai *fitness*, semakin besar nilai *fitness* maka kromosom tersebut mempunyai peluang untuk dipilih lebih besar, serta mungkin dapat terpilih beberapa kali.



Sumber: (Krisnanda, 2020)  
Gambar 4. *Roulette Wheel*

Pada Gambar 4 kemungkinan suatu individu untuk terpilih digambarkan dengan sebuah juring pada lingkaran. Lebarnya juring tersebut tergantung dari nilai *fitness* individu yang bersangkutan. Semakin baik *fitness*, maka juringnya semakin besar.

- c. *Rank selection*, semua individu dalam populasi diurutkan berdasarkan *fitness* mulai dari yang besar ke yang kecil. Setiap individu diberikan probabilitas untuk terpilih berdasarkan distribusi probabilitas yang dipakai, nilainya dapat dicari dengan memasukkan nilai *fitness* ke fungsi distribusi yang telah ditentukan.
- d. *Tournament selection*, dapat dilakukan dalam dua tahap yaitu memilih grup yang terdiri N individu dan memilih individu yang mempunyai *fitness* dari grup dengan membuang individu yang lain. Individu-individu yang terpilih akan menjadi populasi baru.
5. *Crossover*, menyilangkan dua individu terpilih. Proses *crossover* melibatkan dua individu, dihasilkan oleh proses seleksi yang paling menyumbangkan sebagian kromosomnya untuk mendapatkan individu yang baru. Individu baru yang dihasilkan diharapkan memiliki *fitness* yang lebih baik.
6. Mutasi, berdasarkan parameter probabilitas mutasi, ubah kromosom dari individu yang dipilih. Untuk mencegah agar algoritma genetika tidak menghasilkan lokal optimal, maka diperlukan proses mutasi. Tidak seperti *crossover* yang melibatkan dua individu, proses mutasi hanya melibatkan satu individu. Proses mutasi akan mengubah setiap *gen* pada kromosom suatu individu, sehingga *fitness* yang dimiliki juga berubah. Jumlah individu yang terkena mutasi dalam suatu populasi tergantung dari nilai probabilitas mutasi yang ditetapkan, sedangkan individu yang terkena mutasi dipilih secara acak.
7. *Replace*, ganti populasi lama dengan populasi yang baru terbentuk.

8. *Test*, periksa apakah jumlah populasi sudah sama dengan parameter jumlah populasi yang sudah ditentukan. Jika kondisi tersebut sudah terpenuhi, maka proses berhenti dan solusi dari populasi tersebut ditampilkan. Jika belum terpenuhi, kembali ke langkah ke-3 dan jumlah populasi ditambah satu.

### Perhitungan Algoritma Genetika

Pada penelitian ini diberikan contoh kasus antrian *service* menggunakan Algoritma Genetika pada suatu antrian *service*.

#### Langkah Ke-1: Inisialisasi Populasi

Inisialisasi populasi adalah suatu proses menentukan jumlah individu atau kromosom didalam suatu populasi. Data diambil dari data antrian *service* kendaraan sebagai data *sample* penjadwalan pada Tabel 1.

Tabel 1. Antrian *Service* Kendaraan

No.	Layanan	Waktu	Nopol	Teknisi
1	<i>Service</i> 1.000 km	1 Jam	D8841HG	Mulyadi
2	<i>Service</i> 5.000 km	1 Jam	D4742BV	R Septiawan
3	<i>Service</i> 8.000 km	1 Jam	D888BN	Dede
4	<i>Service</i> 7.000 km	1 Jam	D9711FF	Agus P
5	<i>Service</i> 10.000 km	1 Jam	D1188CM	Iyus
6	<i>Service</i> 20.000 km	1 Jam	D1728LA	Ribdan

Tabel 2. Data *Stall* Bengkel

No	Kode <i>Stall</i>	Nama <i>Stall</i>
1	R01	<i>Stall</i> 1
2	R02	<i>Stall</i> 2
3	R03	<i>Stall</i> 3
4	R04	<i>Stall</i> 4

Tabel 3. Data Waktu Pelayanan

No	Hari	Jam
1	Mingg u	08:00:0 0
2	Mingg u	09:00:0 0
3	Mingg u	10:00:0 0
4	Mingg u	11:00:0 0

#### Langkah Ke-2: *Generate* Kromosom

*Generate* Kromosom adalah suatu proses pembangkitan nilai acak kromosom awal yang didapat dari data inialisasi populasi. Pada langkah kedua ini dilakukan pembangkitan 4 individu dan 4 kromosom yang di *generate* secara acak yang merupakan kombinasi dari inialisasi populasi [tabel antrian *service*, tabel *stall*, dan tabel waktu].

Tabel 4. Random Generate Populasi ke-1

Individu Ke	Total Gen
Individu 1	{[1,4,4],[2,4,4],[3,2,2],[4,1,1]}
Individu 2	{[1,2,2],[3,4,4],[2,1,2],[4,4,4]}
Individu 3	{[1,1,1],[1,3,3],[1,1,1],[4,3,3]}
Individu 4	{[1,1,2],[2,4,1],[4,2,4],[3,4,1]}

Tabel 5. Individu ke-1

Kromosom Ke:	Gen1 [Nopol, Teknisi]	Gen 2 [Nama Stall]	Gen 3 [Hari, Jam]
1	D8841HG Mulyadi	Stall 4	Minggu 11:00:00
2	D4742BV R Septiawan	Stall 4	Minggu 11:00:00
3	D8881BN Dede	Stall 2	Minggu 09:00:00
4	D9711FF Agus P	Stall 1	Minggu 08:00:00

Tabel 6. Individu ke-2

Kromosom Ke:	Gen1 [Nopol, Teknisi]	Gen 2 [Nama Stall]	Gen 3 [Hari, Jam]
1	D8841HG Mulyadi	Stall 2	Minggu 09:00:00
2	D8888BN Dede	Stall 4	Minggu 11:00:00
3	D4742BV R Septiawan	Stall 1	Minggu 10:00:00
4	D9711FF Agus P	Stall 4	Minggu 11:00:00

Tabel 7. Individu ke-3

Kromosom Ke:	Gen1 [Nopol, Teknisi]	Gen 2 [Nama Stall]	Gen 3 [Hari, Jam]
1	D4742BV R Septiawan	Stall 1	Minggu 08:00:00
2	D8841HG Mulyadi	Stall 3	Minggu 10:00:00
3	D8881N Dede	Stall 2	Minggu 11:00:00
4	D9711FF Agus P	Stall 3	Minggu 10:00:00

Tabel 8. Individu ke-4

Kromosom Ke:	Gen1 [Nopol, Teknisi]	Gen 2 [Nama Stall]	Gen 3 [Hari, Jam]
1	D8841HG Mulyadi	Stall 1	Minggu 09:00:00
2	D4742BV R Septiawan	Stall 4	Minggu 08:00:00
3	D9711FF Agus P	Stall 2	Minggu 11:00:00
4	D8881N Dede	Stall 4	Minggu 08:00:00

### Langkah Ke-3: *Fitness Function*

Nilai *fitness function* ditentukan sebagai parameter penentuan baik atau tidaknya suatu individu dalam algoritma genetika. Nilai fitness ditentukan sebagai parameter yang menentukan nilai optimum dalam masalah optimasi.

Pada langkah ketiga ini dilakukan perhitungan *fitness function* dengan persamaan fungsi objektif untuk proses maksimasi sebagai berikut:

$$Fitness = 1/(1+CM+CS+CK)$$

Keterangan:

Clash Mekanik = Jadwal Bentrok Mekanik dengan waktu yang sama

Clash Stall = Jadwal bentrok Stall dengan waktu yang sama

Clash Waktu = Jadwal bentrok kendaraan dengan waktu yang sama

Parameter *Fitness* terbaik = 1

Menghitung *fitness* berdasarkan Individu dari generasi awal yang dibangkitkan:

- Fitness* Individu 1 =  $1/(1+1+1+1) = 0.33333$
- Fitness* individu 2 =  $1/1+(1+0+0+0) = 0.5$
- Fitness* Individu 3 =  $1/1 = (1+0+0+0) = 0.5$
- Fitness* Individu 4 =  $1/1 = (1+1+0+0) = 0.33333$

Total Nilai *Fitness* = 1,666

### Langkah Ke-4: Seleksi Generasi 1

Seleksi berfungsi untuk memilih individu-individu mana saja yang akan dipilih untuk proses *crossover*, teknik yang dilakukan adalah dengan menggunakan teknik *roulette wheel*. teknik ini berprinsip semakin besar nilai *fitness* setiap individu semakin besar individu tersebut akan di pilih dalam setiap kemungkinan yang ada.

Tahapan- tahapan pada seleksi dengan metode *roulette wheel* akan di hitung pada tabel berikut 3.8 berikut ini.

1.  $Probability = Fitness[i] / Total\_Fitness$

Tabel 9. Menghitung Probabilitas

Jumlah Area	Range Area
Probabilitas 1	$0.333/1.666 = 0.200$
Probabilitas 2	$0.5/1.666 = 0.300$
Probabilitas 3	$0.5/1.666 = 0.300$
Probabilitas 4	$0.333/1.666 = 0.200$

2. Langkah selanjutnya yaitu hitung nilai kumulatif yang bisa dilihat pada Tabel 10.

Tabel 10. Menghitung Kumulatif

Kumulatif Ke	Perhitungan Kumulatif
Kumulatif 1	$0.200 + 0 = 0.200$
Kumulatif 2	$0.200 + 0.300 = 0.500$
Kumulatif 3	$0.500 + 0.300 = 0.800$
Kumulatif 4	$0.800 + 0.200 = 1$

Setelah dilakukan proses menghitung nilai kumulatif maka didapat nilai area *roulette wheel* yang dimana area *roulette wheel* didapat diantara angka 0 sampai dengan 1 yang bisa dilihat pada Tabel 11.

Tabel 11. Menghitung Area Roulette Wheel

Jumlah Area	Range Area
Area 1	(0.2-0.5)
Area 2	(0.5-0.8)
Area 3	(0.8-1)
Area 4	(1-0.2)

3. Langkah selanjutnya yaitu membangkitkan bilangan acak dalam *range* (0-1) dimana nomor acak ini dibangkitkan sesuai jumlah individu yang dimiliki nilai *random* dapat dilihat pada Tabel 12.

Tabel 12. Pembangkitan Bilangan Acak

Angka Ke	Range Area
Random 1	0.396
Random 2	0.822
Random 3	0.211

Random 4	0.720
----------	-------

Selanjutnya melakukan pengelompokan dari nilai acak yang di bangkitkan pada tabel dengan ketentuan jika nilai *Random* < 1 maka pilih area *roulette wheel* pada tabel hasil yang didapat pada Tabel 13.

Tabel 13. Memasukkan Individu ke Area *Roulette Wheel*

Individu Sebelumnya	Individu Setelah Seleksi ke area <i>Roulette Wheel</i>
Individu 1	Individu 2
Individu 2	Individu 4
Individu 3	Individu 2
Individu 4	Individu 3

Tabel 14. Populasi Setelah Masuk ke Area *Roulette Wheel*

Individu Ke	Total Gen
Individu 1	{[1,4,4],[2,4,4],[3,2,2],[4,1,1]}
Individu 2	{[1,1,2],[2,4,1],[4,2,4],[3,4,1]}
Individu 3	{[1,4,4],[2,4,4],[3,2,2],[4,1,1]}
Individu 4	{[1,2,2],[3,4,4],[2,1,2],[4,4,4]}

#### Langkah Ke-5: *Crossover*

*Crossover* atau kawin silang adalah proses dalam algoritma genetika yang mengawinkan dua buah individu yang bertujuan untuk menghasilkan keturunan yang baru dari individu yang terseleksi. Pada proses ini melakukan pertukaran secara random dua buah *parent*.

1. Penentuan laju *crossover* atau *crossover rate* yang di tentukan yaitu sebesar 0.75 atau 75%. Selama kromosom kurang kurang dari populasi lalu lakukan *random* bilangan acak dari range (0-1) serta jika nilai *random* dan kromosom kurang dari *crossover rate* yang ditentukan maka selanjutnya dipilih sebagai *parent*. Pembangkitan bilangan acak (0-1) sebanyak jumlah individu dapat dilihat pada Tabel 15.

Tabel 15. Bilangan random *Crossover Rate*

Nomor	Nilai <i>Random</i>
Random 1	0.259
Random	0.673

2	
Random	0.870
3	
Random	0.483
4	

2. Lalu memilih individu sesuai laju *crossover* 75% yang dapat dilihat pada Tabel 16.

Tabel 16. Individu yang terpilih sesuai laju *Crossover*

<b>Individu Terpilih</b>
Individu 1
Individu 2
Individu 4

3. Setelah itu individu dikawinkan dengan individu yang lain yang dapat dilihat pada Tabel 17.

Tabel 17. Perkawinan Individu

<b>Individu yang dikawinkan</b>
Individu 1 >< Individu 2
Individu 2 >< Individu 4
Individu 4 >< Individu 1

4. Menentukan Posisi *crossover* dengan metode *one cut point crossover*

Bangkitkan bilangan acak antara 1 s.d (Panjang Kromosom -1) sebanyak jumlah *parent* dengan ketentuan sebagai berikut 1 s/d (4-1) = 1 sampai dengan 3. Lalu didapat hasil bilangan *random* untuk penentuan proses bagian *one cut point crossover* yang dapat dilihat pada Tabel 18.

Tabel 18. Hasil Posisi *One Cut Point Crossover*

<b>Nilai Random Penentuan One Cut Point Crossover</b>
<i>Crossover</i> 1 = 3
<i>Crossover</i> 2 = 2
<i>Crossover</i> 3 = 1

5. Pada tahap ini akan dilakukan proses pengkawinan *crossover* dengan metode *one cut crossover* dengan mengawinkan individu terpilih.

Tabel 19. Perkawinan Individu Terpilih ke-1

<b>Parent</b>	<b>Kromosom</b>
Parent 1	{[1,4,4],[2,4,4],[3,2,2],[4,1,1]}
Parent 2	{[1,1,2],[2,4,1],[4,2,4],[3,4,1]}
<i>Offspring</i> 1	{[1,4,4],[2,4,4],[3,2,2],[3,4,1]}

Tabel 20. Perkawinan Individu Terpilih ke -2

Parent	Kromosom
Parent 1	{[1,1,2],[2,4,1], [4,2,4],[3,4,1]}
Parent 2	{[1,2,2],[3,4,4],[2,1,2],[4,4,4]}
Offspring 2	{[1,2,2],[3,4,4],[3,2,2],[3,4,1]}

Tabel 21. Perkawinan Individu Terpilih ke-3

Parent	Kromosom
Parent 1	{[1,2,2],[3,4,4],[2,1,2],[4,4,4]}
Parent 2	{[1,4,4],[2,4,4],[3,2,2],[4,1,1]}
Offspring 3	{[1,2,2],[2,4,4],[3,2,2],[4,1,1]}

6. Setelah dilakukan proses *crossover* dengan metode *one cut point crossover* maka didapat populasi baru hasil *crossover* yang dapat dilihat pada Tabel 22.

Tabel 22. Populasi Hasil *Crossover*

Individu Ke	Total Kromosom
Individu 1	{[1,4,4], [2,4,4], [3,2,2], [3,4,1]}
Individu 2	{[1,2,2], [3,4,4], [3,2,2], [3,4,1]}
Individu 3	{[1,4,4], [2,4,4], [3,2,2], [4,1,1]}
Individu 4	{[1,2,2], [2,4,4], [3,2,2], [4,1,1]}

7. Lalu langkah selanjutnya menghitung kembali *fitness function* dengan hasil metode *crossover* *Fitness* individu yang dapat dilihat pada Tabel 23.

Tabel 23. *Fitness Function* Hasil Proses *Crossover*

Individu Ke	Total Gen
<i>Fitness</i> individu 1	$Fitness = 1/(1+1+1) = 0.333$
<i>Fitness</i> individu 2	$Fitness = 1/(1+1+1) = 0.333$
<i>Fitness</i> individu 3	$Fitness = 1/(1+1+1) = 0.333$
<i>Fitness</i> individu 4	$Fitness = 1/(1+1+1) = 0.333$
Total Nilai <i>Fitness</i>	0.9

Langkah Ke-6: *Mutation*

Mutasi adalah proses penggantian nilai *gen* yang ada dalam individu. Mutasi dilakukan dengan menggunakan laju mutasi sebagai parameter mutasi. lalu penentuan nilai *random* individu yang akan dimutasi. Langkah awal dalam melakukan proses mutasi yaitu dengan menentukan total kromosom = jumlah kromosom \* jumlah individu. Maka didapat  $4 \times 4 = 16$  adalah jumlah kromosom yang bisa mutasi dalam populasi sample ini.

1. Menentukan Nilai *Random*

Penentuan nilai *random* ditentukan dengan laju mutasi sebesar  $1/100$ . Laju Mutasi( $0.05 \times 16 = 0.8$ ) dengan 1 kromosom yang akan dimutasi

2. Lalu langkah selanjutnya menentukan kromosom yang akan di mutasi dengan menentukan 1 nilai random dengan range antara (1-16). nilai yang telah didapat dengan angka acak yang didapat adalah 9.
3. Langkah selanjutnya melakukan melakukan pembangkitan nilai acak dari *range* (1-4) sebanyak jumlah index yang ada didalam kromosom yang dengan nilai acak yang didapat yaitu [1,3,3] yang selanjutnya akan di mutasi kedalam kromosom ke 9 yang telah dilakukan *generate*.

Tabel 24. Proses Mutasi Dari Generasi Hasil *Crossover*

Nomor	Total Kromosom
Random 1	{[1,4,4],[2,4,4],[3,2,2],[3,4,1]}
Individu 2	{[1,2,2],[3,4,4],[3,2,2],[3,4,1]}
Individu 3	{[1,3,3],[2,4,4],[3,2,2],[4,1,1]}
Individu 4	{[1,2,2],[2,4,4],[3,2,2],[4,1,1]}

4.Lalu langkah selanjutnya menghitung kembali *fitness function* dengan hasil mutasi individu yang dapat dilihat pada Tabel 25.

Tabel 25. Hasil Mutasi Individu

Individu Ke	Total Gen
<i>Fitness</i> individu 1	$Fitness = 1/(1+1+1) = 0.333$
<i>Fitness</i> individu 2	$Fitness = 1/(1+1+1) = 0.333$
<i>Fitness</i> individu 3	$Fitness = 1/(1+0+0) = 1$
<i>Fitness</i> individu 4	$Fitness = 1/(1+1+1) = 0.333$
Total Nilai <i>Fitness</i>	1.9

Langkah Ke-7: *Replace*

Setelah dilakukan proses mutasi maka didapatkan hasil *fitness* terbaik dengan telah mencapai nilai angka tertinggi yaitu nilai 1 yang menandakan tidak ada lagi clash atau bentrok maka telah didapat generasi baru lalu setelah itu proses *replace* selesai.

Langkah Ke-8: *Test*

Lalu dilakukan proses *testing* dimana parameter inisialisasi awal tetap sama sampai proses testing harus sama lalu ditunjukkan juga hasil individu yang sudah tidak mengalami bentrok data populasi bisa ditampilkan dapat dilihat pada Tabel 26.

Tabel 26. Hasil Akhir Perhitungan Algoritma Genetika

Nomor	Total Kromosom
Random 1	{[1,4,4],[2,4,4],[3,2,2],[3,4,1]}
Individu 2	{[1,2,2],[3,4,4],[3,2,2],[3,4,1]}
Individu 3	{[1,3,3],[2,4,4],[3,2,2],[4,1,1]}
Individu 4	{[1,2,2],[2,4,4],[3,2,2],[4,1,1]}

## HASIL DAN PEMBAHASAN

### Analisis Kebutuhan Sistem

Pada bagian ini menjelaskan kebutuhan fungsional sistem penjadwalan antrian *service* yang akan dibangun.

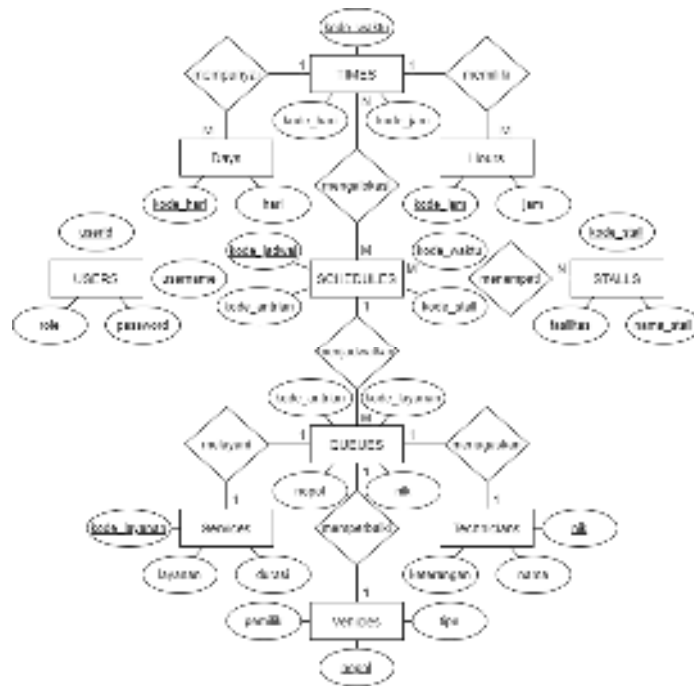
### Tahapan Analisis

Spesifikasi kebutuhan dari sistem penjadwalan antrian *service* yang akan dibangun sebagai berikut:

Halaman *Service Advisor* (SA):

- A1. SA dapat melakukan *login*.
- A2. SA dapat mengelola data *timing* atau waktu (jam, dan hari).
- A3. SA dapat mengelola data antrian *service*.
- A4. SA dapat mengelola data *stall*.
- A5. SA dapat melakukan optimasi jadwal pada menu *optimization*.
- A6. SA dapat memeriksa hasil perhitungan penjadwalan pada menu *schedule*.
- A7. SA dapat memperbaharui *profile*.

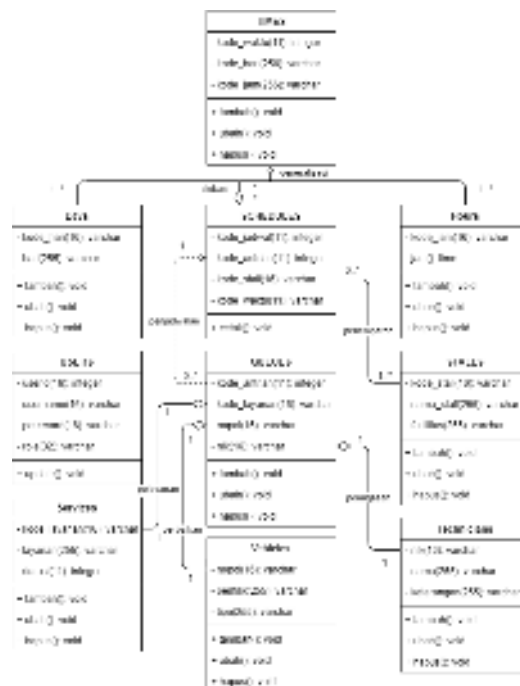




Gambar 6. ERD Sistem Penjadwalan Antrian *Service*

**B. Class Diagram**

Desain *class diagram* pada penelitian ini menggambarkan struktur pemrograman yang menunjukkan struktur statis pengklasifikasian dari sistem antrian *service* yang dirancang.



Gambar 7. *Class Diagram* Sistem Antrian *Service*

Pada Gambar 7 adalah *class diagram* sistem antrian *service* yang menggambarkan struktur sistem antrian *service* dengan menunjukkan *class*, atribut, metode atau operasi, beserta hubungan antar objeknya.

## Penerapan

Pada tahap penerapan ditampilkan *user interface* yang merupakan gambar antarmuka pengguna dari perancangan sistem penjadwalan antrian *service* yang dibuat. Perancangan antarmuka berfungsi untuk memudahkan interaksi antara pengguna dengan sistem yang telah dibuat.

### 1. Halaman *Login* Pengguna

Tampilan halaman *login* merupakan halaman untuk akses ke halaman utama dari sistem penjadwalan antrian *service* dapat dilihat pada Gambar 8.



Gambar 8 Tampilan Halaman *Login* Pengguna

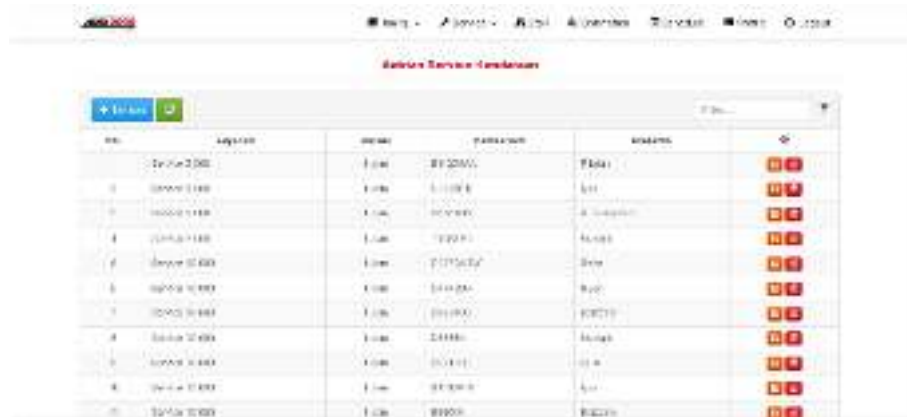
### 2. Halaman Data Waktu

Halaman yang digunakan dalam mengelola data waktu dengan tampilannya yang dapat dilihat pada Gambar 9.

Gambar 9. Tampilan Halaman Data Waktu

### 3. Halaman Data Antrian

Halaman yang digunakan dalam mengelola data antrian *service* yang ditampilkan pada Gambar 10.



no.	adanya	waktu	pelayanan	antrian
1	1000000000	1000	1000000	1000
2	1000000000	1000	1000000	1000
3	1000000000	1000	1000000	1000
4	1000000000	1000	1000000	1000
5	1000000000	1000	1000000	1000
6	1000000000	1000	1000000	1000
7	1000000000	1000	1000000	1000
8	1000000000	1000	1000000	1000
9	1000000000	1000	1000000	1000
10	1000000000	1000	1000000	1000

Gambar 10. Tampilan Data Antrian

### 4. Halaman Optimasi Jadwal


Halaman yang digunakan dalam menghitung optimasi penjadwalan dengan tampilan antarmuka pengguna yang dapat dilihat pada Gambar 11.



Gambar 11. Tampilan Optimasi Jadwal

### 5. Halaman Jadwal Antrian *Service*

Halaman yang digunakan dalam memeriksa jadwal antrian *service* dengan tampilan dari halaman akun pengguna dapat dilihat pada Gambar 12.



ID	No	Jam	Lokasi	Status	Barang	Waktu	Waktu
1	01001	08:00 - 09:00	08:00 - 09:00	1	100	1000	1000
2	01002	09:00 - 10:00	09:00 - 10:00	1	200	2000	2000
3	01003	10:00 - 11:00	10:00 - 11:00	1	300	3000	3000
4	01004	11:00 - 12:00	11:00 - 12:00	1	400	4000	4000
5	01005	12:00 - 13:00	12:00 - 13:00	1	500	5000	5000
6	01006	13:00 - 14:00	13:00 - 14:00	1	600	6000	6000
7	01007	14:00 - 15:00	14:00 - 15:00	1	700	7000	7000
8	01008	15:00 - 16:00	15:00 - 16:00	1	800	8000	8000
9	01009	16:00 - 17:00	16:00 - 17:00	1	900	9000	9000
10	01010	17:00 - 18:00	17:00 - 18:00	1	1000	10000	10000

Gambar 12. Tampilan Jadwal Antrian Service

### 6. Halaman Update Profile

Halaman yang digunakan dalam memperbaharui *profile* pengguna. Tampilan dari halaman *profile* pengguna dapat dilihat pada Gambar 13.



Update Profile Pengguna

Username Baru:

Password Baru:

Password Lama:

[Update] [Cancel]

Gambar 13. Tampilan Profile Pengguna

### Testing

Proses evaluasi komponen sistem untuk memverifikasi perbedaan-perbedaan antara hasil yang diharapkan dengan hasil yang terjadi. Pengujian pada sistem penjadwalan antrian *service* menghasilkan validasi sebagai berikut:

1. Pengujian *black box testing login* pengguna pada kolom input *username* dan *password* memberikan hasil pengujian yang sesuai harapan dan kesimpulan yang *valid*.
2. Pengujian *black box testing* pada pengelolaan data waktu yaitu pengguna dapat menginput data, mengubah data dan menghapus data waktu hari dan jam memberikan hasil pengujian yang sesuai harapan dan kesimpulan yang *valid*.

3. Pengujian *black box testing* pada pengelolaan *form* data antrian yaitu pengguna dapat memasukkan data antrian, menghapus data antrian dan mengubah data antrian yang memberikan hasil pengujian yang sesuai harapan dan kesimpulan yang *valid*.
4. Pengujian *black box testing* mengelola *stall* yaitu pengguna dapat menambah data *stall*, menghapus data *stall* dan mengubah data *stall* yang memberikan hasil yang sesuai dan kesimpulan yang *valid*.
5. Pengujian *black box testing form* optimasi yaitu pengguna dapat mengatur optimasi sesuai kromosom, generasi, *crossover rate* dan laju mutasi yang diperlukan yang memberikan hasil sesuai dan kesimpulan yang *valid*.
6. Pengujian *black box testing form* memperbaharui *profile* yaitu pengguna dapat memperbaharui *profile* mengganti *password* dengan mendapatkan hasil yang sesuai dan kesimpulan yang *valid*.

### **Support**

Menjelaskan tentang spesifikasi *hardware* dan *software* yang akan digunakan untuk menjalankan sistem penjadwalan antrian *service* yang telah dibuat beserta dokumentasi formulir *input* dan *output* sistem. Dijelaskan tentang kebutuhan standar *hardware* dan *software* yang akan digunakan untuk dapat menjalankan sistem antrian *service*. Kebutuhan *hardware* dan *software* tersebut meliputi sebagai berikut:

1. *Hardware*
  - a. *Processor* minimal Core 2 Duo
  - b. RAM 2 GB
  - c. VGA Card 1 GB
  - d. *Harddisk* 160 GB
  - e. *Monitor* Samsung LED 14 inch
  - f. *Keyboard* Logitech K120
  - g. *Mouse* Standard
  - h. *Printer* Laser Jet
2. *Software*
  - a. Sistem operasi Microsoft Windows 7
  - b. *Browser* Google Chrome atau Mozilla Firefox.
  - c. *Software* basis data XAMPP v7.0.33

## **KESIMPULAN**

### **Kesimpulan**

Sistem penjadwalan antrian *service* mobil Toyota di Auto2000 Pasteur yang telah dilakukan dalam menjawab permasalahan penelitian dengan diambil beberapa kesimpulan sebagai berikut:

1. Sistem penjadwalan yang telah dibangun mampu meratakan pembagian pekerjaan bagi teknisi oleh *Foreman*. Pada sistem ini *output* jadwal yang dihasilkan sesuai dan konsisten dengan inputan inialisasi data awal yang terdiri dari data pelanggan, layanan, stall, teknisi dan waktu.
2. Pada penelitian ini telah berhasil mengimplementasikan algoritma genetika pada sistem penjadwalan antrian *service* di Auto2000 Pasteur. Algoritma genetika mampu melakukan optimasi jadwal *service* dengan meniru proses seleksi alam berdasarkan aturan *fitness* terkuat yang akan bertahan yaitu penentuan waktu, teknisi dan *stall* diantara jadwal *service* yang tidak berbenturan dengan jadwal lainnya. Metode ini melakukan proses penyeleksian berulang-ulang sampai didapatkan jadwal *service* baru dengan *fitness* terbaik.
3. Sistem penjadwalan antrian *service* mobil Toyota Auto2000 Pasteur berhasil dibangun dengan pemrograman *website* dan *database* MySQL. Sistem tersebut memberikan hak akses bagi *Service Advisor* dalam mengelola data waktu, layanan perbaikan dan pelanggan, stall, hingga optimasi jadwal. Kemudian dihasilkan *output* penjadwalan yang digunakan *Foreman*.

### Saran

Pada bagian ini dipaparkan mengenai kekurangan pada sistem penjadwalan yang telah dibahas pada penelitian ini. Terdapat beberapa saran bagi penelitian selanjutnya yang dapat dijadikan sebagai acuan dalam mengembangkan sistem agar lebih baik lagi, diantaranya:

1. Sistem penjadwalan yang telah dikembangkan diharapkan dapat digunakan juga oleh teknisi, sehingga teknisi dapat melihat langsung jadwal serta jenis pekerjaan *service* yang harus dilakukannya.
2. Algoritma Genetika pada penelitian ini dapat juga diimplementasikan pada sistem antrian *service* yang melibatkan pelanggan dalam mengakses sistem pada saat melakukan *appointment*.
3. Sistem penjadwalan antrian *service* kendaraan yang telah dibuat bagi bengkel Auto2000 Paster ini dapat juga diintegrasikan dengan penggunaan *Application Programming Interface* (API) yang terhubung dengan basis data *spare part*. Hal tersebut dapat memungkinkan ketersediaan *spare part* yang berpengaruh terhadap jadwal maupun durasi layanan.

### DAFTAR PUSTAKA

Fatihudin, D., & Firmansyah, A. (2019). Pemasaran jasa: Strategi, mengukur kepuasan dan loyalitas pelanggan. Deepublish.

- Ferdyawan, & Hajjah, A. (2020). Penerapan algoritma genetika dalam optimasi penjadwalan proyek. *Jurnal Mahasiswa Aplikasi Teknologi Komputer dan Informasi*, 2(1), 50–55.
- Hidayat, H. R., & Wiguna, W. (2021). Aplikasi diagnosa penyakit tuberculosis menggunakan metode certainty factor berbasis Android. *Jurnal Responsif: Riset Sains Informatika*, 3(1), 20–29.
- Krisnanda, M. (2020). Penjadwalan pilot maskapai penerbangan menggunakan algoritma genetika. Lakeisha.
- Kurnia, D., & Ardiansyah, H. (2020). Sistem informasi reservasi penjadwalan service pada PT Mentari Alam Semesta berbasis web. In *Prosiding Seminar Informatika dan Sistem Informasi* (Vol. 5, No. 1, pp. 101–108).
- Setyawati, R., & Maulachela, A. B. (2020). Penerapan algoritma dynamic priority scheduling pada antrian pencucian mobil. *JTIM: Jurnal Teknologi Informasi dan Multimedia*, 2(1), 29–35. <https://doi.org/10.35746/jtim.v2i1.85>
- Tampubolon, K., Fahmi, A., & Batu, F. L. (2020). Elemen-elemen mesin bensin pada mobil dan perawatannya. *Inteligensia Media*.
- Wiguna, W., & Alawiyah, T. (2019). Sistem reservasi paket wisata pelayaran menggunakan mobile commerce di Kota Bandung. *Jurnal VOI (Voice of Informatics)*, 8(2), 49–62.
- Wiguna, W., Mauliana, P., & Permana, A. Y. (2020). Pengembangan e-helpdesk support system berbasis web di PT Akur Pratama. *Jurnal Responsif: Riset Sains Informatika*, 2(1), 19–29.
- Zukhri, Z. (2020). Algoritma genetika: Metode komputasi evolusioner untuk menyelesaikan masalah optimasi.
- Zulfa, L. L., Mujibah, E. M., & Rajaguguk, Z. F. (2020). Pelatihan penggunaan perangkat berbasis internet dalam pengumpulan data penelitian masa pandemi COVID-19. *Educivilia: Jurnal Pengabdian pada Masyarakat*, 1(2), 143–150. <https://doi.org/10.30997/ejpm.v1i2.2835>