

**SKRIPSI**

**PENYELESAIAN CAR SEQUENCING PROBLEM DENGAN  
GENETIC ALGORITHM DAN SIMULATED ANNEALING  
ALGORITHM**



**Andreas Ario Christian**

**NPM: 2015730055**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS  
UNIVERSITAS KATOLIK PARAHYANGAN  
2022**



**UNDERGRADUATE THESIS**

**SOLVING CAR SEQUENCING PROBLEM WITH GENETIC  
ALGORITHM AND SIMULATED ANNEALING ALGORITHM**



**Andreas Ario Christian**

**NPM: 2015730055**

**DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES  
PARAHYANGAN CATHOLIC UNIVERSITY  
2022**



LEMBAR PENGESAHAN

PENYELESAIAN CAR SEQUENCING PROBLEM DENGAN  
GENETIC ALGORITHM DAN SIMULATED ANNEALING  
ALGORITHM

Andreas Ario Christian

NPM: 2015730055

Bandung, 1 Februari 2022

Menyetujui,

Pembimbing

Digitally signed  
by Cecilia Esti  
Nugraheni

Dr.rer.nat. Cecilia Esti Nugraheni

Ketua Tim Penguji

Digitally signed  
by Lionov

Lionov, Ph.D.

Anggota Tim Penguji

Digitally  
signed by  
Husnul  
Hakim

Husnul Hakim, M.T.

Mengetahui,

Ketua Program Studi

Digitally signed  
by Mariskha Tri  
Adithia

Mariskha Tri Adithia, P.D.Eng

## PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

### **PENYELESAIAN CAR SEQUENCING PROBLEM DENGAN GENETIC ALGORITHM DAN SIMULATED ANNEALING ALGORITHM**

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,  
Tanggal 1 Februari 2022



Andreas Ario Christian  
NPM: 2015730055

## ABSTRAK

Lini produksi berjalan yang diperkenalkan ke dalam industri otomotif oleh Henry Ford untuk pembuatan Ford Model T merevolusi manufaktur kendaraan bermotor, khususnya mobil. Pada awal perkembangannya, kecepatan produksi dan kuantitas mobil yang diproduksi perhari menjadi hal utama. Produksi yang lebih cepat dan lebih banyak mobil yang dihasilkan membuat harga Ford Model T lebih murah daripada mobil lain di pasaran. Bahkan harga mobil tersebut dapat diturunkan secara bertahap. Seiring berjalannya waktu, manufaktur mobil terbagi menjadi dua, yaitu manufaktur kendaraan mewah dan massal. Kendaraan yang tergolong mewah pada umumnya produksinya dibatasi, waktu produksinya cukup lama, dan banyak pilihan fitur yang disediakan bagi pembeli. Sedangkan, kendaraan massal memiliki produksi yang dibatasi perkembangan model, waktu produksinya cepat, dan fitur pada mobil tersebut ditentukan oleh produsen.

Mengikuti perkembangan zaman, para produsen mobil mulai menyediakan fitur-fitur yang dapat dipilih oleh konsumen, seperti motif jok, lampu kabut, dan *sunroof*. Tentunya mobil dengan fitur yang dipilih oleh konsumen dapat menyebabkan masalah baru, seperti waktu produksi yang lebih lama. Hal ini disebabkan oleh penyesuaian urutan mobil pada lini produksi.

Permasalahan itu disebut dengan *Car Sequencing Problem*, yaitu pencarian urutan paling optimal untuk lini produksi berdasarkan tipe mobil yang ada. Tipe mobil yang dimaksud adalah sebuah mobil yang dibeli konsumen dengan pilihan fitur tertentu, seperti contoh sebuah sedan dengan lampu kabut. *Car Sequencing Problem* tentunya sudah lama dipecahkan oleh para produsen mobil, salah satunya adalah Renault. Perusahaan tersebut menggunakan algoritma *Simulated Annealing* untuk memecahkan *Car Sequencing Problem*. Pada penelitian-penelitian beberapa tahun setelahnya, ternyata *Car Sequencing Problem* dapat dipecahkan juga oleh *Genetic Algorithm*.

Terdapat sebuah penelitian lain yang membuktikan bahwa gabungan dari *Genetic Algorithm* dan *Simulated Annealing* dapat menghasilkan solusi yang lebih baik dari kedua algoritma jika dijalankan masing-masing. Oleh karena itu, skripsi ini akan mencoba menggunakan gabungan tersebut untuk menyelesaikan *Car Sequencing Problem* dan membuktikan bahwa gabungan algoritma tersebut dapat bekerja dengan baik juga pada permasalahan ini.

Setelah melakukan penjabaran dasar teori, pemodelan dan analisis algoritma, perancangan, implementasi perangkat lunak, dan pengujian, dapat ditarik kesimpulan dari skripsi ini. *Car Sequencing Problem* dapat diselesaikan dengan beberapa metode penggabungan *Genetic Algorithm* dan *Simulated Annealing*, yaitu *Hybrid GA-SA*, *Hybrid SA-GA*, dan *Cyclic GA-SA*. Selain itu, perangkat lunak yang dibuat berhasil mengimplementasikan *Car Sequencing Problem* pada masing-masing algoritma. Algoritma terbaik berdasarkan hasil pengujian pada skripsi ini adalah *Genetic Algorithm*, dengan *Simulated Annealing*, *Hybrid GA-SA*, dan *Cyclic GA-SA* menjadi urutan selanjutnya sebagai algoritma terbaik. *Hybrid SA-GA* memiliki kinerja terburuk untuk menyelesaikan *Car Sequencing Problem* dibandingkan algoritma-algoritma lainnya.

**Kata-kata kunci:** mobil, lini produksi, *Car Sequencing Problem*, *Genetic Algorithm*, *Simulated Annealing*, kombinasi, *hybrid*





## ABSTRACT

The moving production line introduced into the automotive industry by Henry Ford for the manufacture of the Ford Model T revolutionized motor vehicle manufacturing, particularly automobiles. At the beginning of its development, the speed of production and the quantity of cars produced per day became the main thing. Faster production and more cars were produced making the Ford Model T cheaper than any other car on the market. Even the price of the car can be lowered gradually. Over time, car manufacturing is divided into two, namely luxury and mass vehicle manufacturing. Vehicles that are classified as luxury are generally limited in production, the production time is quite long, and there are many choices of features provided to buyers. Meanwhile, mass vehicles have production that is limited by model development, production time is fast, and the features of the car are determined by the manufacturer.

As the time goes by, car manufacturers began to provide features that consumers could choose from, such as upholstery motifs, fog lamps, and sunroof. Of course, cars with features chosen by consumers can cause new problems, such as longer production times. This is due to the adjustment of the car sequence on the production line.

This problem is called the Car Sequencing Problem, which is the search for the most optimal sequence for the production line based on the type of car available. The type of car in question is a car purchased by consumers with a choice of certain features, such as a sedan with fog lights. The Car Sequencing Problem has certainly been solved by car manufacturers, one of which is Renault. The company used the Simulated Annealing algorithm to solve the Car Sequencing Problem. In researches several years later, it turns out that Car Sequencing Problem can also be solved by Genetic Algorithm.

There is another study which proves that the combination of Genetic Algorithm and Simulated Annealing can produce a better solution than the two algorithms if run separately. Therefore, this thesis will try to use the combination to solve the Car Sequencing Problem and prove that the combined algorithm can work well on this problem as well.

After describing the basic theory, modeling and analyzing algorithms, designing, implementing software, and testing, conclusions can be drawn from this thesis. Car Sequencing Problem can be solved by combining Genetic Algorithm and Simulated Annealing, namely Hybrid GA-SA, Hybrid SA-GA, and Cyclic GA-SA. In addition, the software made successfully implements Car Sequencing Problem on each algorithm. The best algorithm based on the test results in this thesis is Genetic Algorithm, with Simulated Annealing, Hybrid GA-SA, and Cyclic GA-SA being the next best algorithm. Hybrid SA-GA has the worst performance for solving Car Sequencing Problem compared to other algorithms.

**Keywords:** car, production line, Car Sequencing Problem, Genetic Algorithm, Simulated Annealing, combination, hybrid



## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan karunia-Nya, penulis dapat menuntaskan skripsi yang berjudul "*Penyelesaian Car Sequencing Problem dengan Genetic Algorithm dan Simulated Annealing*". Penulis berharap setiap bagian dari skripsi ini dapat membantu mereka yang tertarik untuk belajar serta membantu penelitian lebih lanjut tentang topik-topik yang bersinggungan dengan skripsi ini. Penulis juga mengucapkan terima kasih sebesar-besarnya kepada Ibu Dr.rer.nat. Cecilia Esti Nugraheni yang telah membimbing dengan sepenuh hati dalam proses pembuatan skripsi dan pembuatan perangkat lunak, keluarga di rumah yang senantiasa memberi bantuan dan semangat pada pembuatan skripsi, Keluarga P369 yang turut membimbing dan menyemangati proses pembuatan skripsi ini, Edo dan teman-teman SMA lainnya yang turut mendoakan, serta dan pihak-pihak lain yang tidak dapat penulis sebutkan satu-satu.

Bandung, Februari 2022

Penulis



# DAFTAR ISI

<b>KATA PENGANTAR</b>	<b>xv</b>
<b>DAFTAR ISI</b>	<b>xvii</b>
<b>DAFTAR GAMBAR</b>	<b>xix</b>
<b>DAFTAR TABEL</b>	<b>xxi</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	4
1.3 Tujuan . . . . .	4
1.4 Batasan Masalah . . . . .	4
1.5 Metodologi . . . . .	5
1.6 Sistematika Pembahasan . . . . .	5
<b>2 LANDASAN TEORI</b>	<b>7</b>
2.1 <i>Car Sequencing Problem</i> . . . . .	7
2.2 <i>Genetic Algorithm</i> . . . . .	11
2.3 <i>Simulated Annealing</i> . . . . .	13
2.4 Gabungan <i>Genetic Algorithm</i> dan <i>Simulated Annealing</i> . . . . .	14
2.4.1 <i>Hybrid GA-SA</i> . . . . .	14
2.4.2 <i>Hybrid SA-GA</i> . . . . .	15
2.4.3 <i>Cyclic GA-SA</i> . . . . .	15
<b>3 PEMODELAN DAN ANALISIS</b>	<b>17</b>
3.1 Pemodelan dan Analisis Algoritma . . . . .	17
3.1.1 <i>Genetic Algorithm</i> . . . . .	17
3.1.2 <i>Simulated Annealing</i> . . . . .	19
3.1.3 Gabungan <i>Genetic Algorithm</i> dan <i>Simulated Annealing</i> . . . . .	19
3.2 Pemodelan dan Analisis <i>Input</i> (Masukan) dan <i>Output</i> (Keluaran) . . . . .	20
3.2.1 <i>Input</i> (Masukan) . . . . .	20
3.2.2 <i>Output</i> (Keluaran) . . . . .	20
3.3 <i>Use Case Diagram</i> dan Skenario . . . . .	20
3.4 Diagram Kelas . . . . .	22
<b>4 PERANCANGAN</b>	<b>25</b>
4.1 Perancangan <i>Input</i> (Masukan) dan <i>Output</i> (Keluaran) . . . . .	25
4.1.1 <i>Input</i> (Masukan) . . . . .	25
4.1.2 <i>Output</i> (Keluaran) . . . . .	26
4.2 Diagram Kelas Rinci . . . . .	26
4.2.1 <i>Package</i> Kalkulasi <i>Average Utility</i> dari <i>Car Sequencing Problem</i> . . . . .	26
4.2.2 <i>Package</i> Komponen <i>Genetic Algorithm</i> dan <i>Simulated Annealing</i> . . . . .	27

4.2.3	<i>Package</i> Algoritma Dasar . . . . .	29
4.2.4	<i>Package</i> Kombinasi Algoritma . . . . .	32
4.3	<i>Pseudocode</i> . . . . .	35
4.3.1	<i>Pseudocode</i> untuk <i>Genetic Algorithm</i> . . . . .	35
4.3.2	<i>Pseudocode</i> untuk <i>Simulated Annealing</i> . . . . .	36
4.3.3	<i>Pseudocode</i> untuk <i>Hybrid GA-SA</i> . . . . .	37
4.3.4	<i>Pseudocode</i> untuk <i>Hybrid SA-GA</i> . . . . .	38
4.3.5	<i>Pseudocode</i> untuk <i>Cyclic GA-SA</i> . . . . .	39
<b>5</b>	<b>IMPLEMENTASI DAN PENGUJIAN</b> . . . . .	<b>41</b>
5.1	Implementasi . . . . .	41
5.1.1	Lingkungan Implementasi . . . . .	41
5.1.2	Implementasi Perangkat Lunak . . . . .	41
5.2	Pengujian . . . . .	56
5.2.1	Pengujian Fungsional . . . . .	56
5.2.2	Pengujian Eksperimental . . . . .	63
5.2.3	Kesimpulan Pengujian Eksperimental . . . . .	67
<b>6</b>	<b>KESIMPULAN DAN SARAN</b> . . . . .	<b>69</b>
6.1	Kesimpulan . . . . .	69
6.2	Saran . . . . .	69
	<b>DAFTAR REFERENSI</b> . . . . .	<b>71</b>
	<b>A KODE PROGRAM</b> . . . . .	<b>73</b>
	<b>B HASIL EKSPERIMEN</b> . . . . .	<b>89</b>

## DAFTAR GAMBAR

2.1	Ilustrasi lini produksi. . . . .	9
3.1	<i>Use Case Diagram</i> . . . . .	20
3.2	Diagram Kelas Sederhana. . . . .	22
4.1	Kelas Perhitungan <i>Average Utility</i> dari <i>Car Sequencing Problem</i> . . . . .	27
4.2	Kelas Kromosom. . . . .	28
4.3	Kelas Populasi. . . . .	29
4.4	Kelas <i>Genetic Algorithm</i> . . . . .	30
4.5	Kelas <i>Simulated Annealing</i> . . . . .	31
4.6	Kelas <i>Hybrid GA-SA</i> . . . . .	32
4.7	Kelas <i>Hybrid SA-GA</i> . . . . .	33
4.8	Kelas <i>Cyclic GA-SA</i> . . . . .	34
5.1	Mengisi nilai variabel utama saat menjalankan kelas <i>QueueFinder</i> . . . . .	48
5.2	Memilih format <i>input</i> dan <i>output</i> . . . . .	48
5.3	Memilih <i>file input</i> yang tersedia. . . . .	49
5.4	Tampilan <i>file datainput.txt</i> . . . . .	49
5.5	Tampilan <i>file datainputcsplib.txt</i> . . . . .	50
5.6	Petunjuk pengisian <i>input</i> secara manual. . . . .	50
5.7	Isi variabel yang ditunjukkan oleh perangkat lunak. . . . .	51
5.8	Tampilan hasil perhitungan pertama dari kelas <i>QueueFinder</i> . . . . .	52
5.9	Tampilan hasil perhitungan kedua dari kelas <i>QueueFinder</i> . . . . .	52
5.10	Tampilan hasil perhitungan ketiga dari kelas <i>QueueFinder</i> . . . . .	53
5.11	Tampilan hasil perhitungan keempat dari kelas <i>QueueFinder</i> . . . . .	53
5.12	Tampilan hasil perhitungan terakhir dari kelas <i>QueueFinder</i> . . . . .	54
5.13	Tampilan konsol pada format <i>output</i> otomatis (dengan file). . . . .	55
5.14	Tampilan <i>file hasilprogram.txt</i> . . . . .	55
5.15	Pengujian fungsional pengisian variabel nomor 1. . . . .	56
5.16	Pengujian fungsional pengisian variabel nomor 2. . . . .	57
5.17	Pengujian fungsional pengisian variabel nomor 3. . . . .	57
5.18	Pengujian fungsional pemilihan <i>input</i> dan <i>output</i> nomor 1. . . . .	57
5.19	Pengujian fungsional pemilihan <i>input</i> dan <i>output</i> nomor 2. . . . .	58
5.20	Pengujian fungsional format <i>input</i> manual nomor 1. . . . .	59
5.21	Pengujian fungsional format <i>input</i> manual nomor 2. . . . .	59
5.22	Pengujian fungsional format <i>input</i> manual nomor 3. . . . .	59
5.23	Pengujian fungsional format <i>input</i> manual nomor 4. . . . .	60
5.24	Pengujian fungsional format <i>input</i> manual nomor 5. . . . .	60
5.25	Pengujian fungsional format <i>input</i> manual nomor 6. . . . .	60
5.26	Pengujian fungsional format <i>input</i> manual nomor 7. . . . .	61
5.27	Pengujian fungsional format <i>input</i> manual nomor 8. . . . .	61
5.28	Pengujian fungsional format <i>input</i> otomatis nomor 1. . . . .	62
5.29	Pengujian fungsional format <i>input</i> otomatis nomor 2. . . . .	62

5.30	Pengujian fungsional format <i>input</i> otomatis nomor 3. . . . .	63
5.31	Pengujian fungsional format <i>input</i> otomatis nomor 4. . . . .	63
5.32	Grafik nilai rata-rata dari semua tahap pengujian eksperimental. . . . .	66
5.33	Grafik perbandingan rata-rata keseluruhan setiap algoritma. . . . .	66



## DAFTAR TABEL

1.1	Jenis mobil Toyota berdasarkan segmennya. . . . .	1
1.2	Tabel ilustrasi jenis mobil dan tipe-tipenya. . . . .	3
2.1	Contoh perhitungan dari <i>Car Sequencing Problem</i> . . . . .	9
2.2	Contoh pengurutan mobil. . . . .	9
2.3	Hubungan antara <i>workstation</i> dan urutan tipe mobil di dalam tabel. . . . .	10
2.4	Contoh terjadinya <i>work overload</i> . . . . .	10
3.1	Hubungan kromosom dan ID. . . . .	17
3.2	Urutan matriks tipe dan opsi. . . . .	18
4.1	Contoh format <i>input</i> perangkat lunak. . . . .	25
5.1	Tabel pengujian fungsional untuk tampilan awal. . . . .	56
5.2	Tabel pengujian fungsional untuk tampilan awal. . . . .	57
5.3	Tabel pengujian fungsional untuk format <i>input</i> manual. . . . .	58
5.4	Tabel pengujian fungsional untuk format <i>input</i> otomatis. . . . .	62
5.5	Tabel perubahan parameter pengujian. . . . .	64
5.6	Tabel hasil pengujian dengan parameter <i>crossover rate</i> sebesar 75%, <i>mutation rate</i> sebesar 0.5%, dan <i>acceptance probability</i> sebesar 0.8. . . . .	64
B.1	Tabel hasil pengujian dengan parameter <i>crossover rate</i> sebesar 75%, <i>mutation rate</i> sebesar 0.5%, dan <i>acceptance probability</i> sebesar 0.8. . . . .	89
B.2	Tabel hasil pengujian dengan parameter <i>crossover rate</i> sebesar 70%. . . . .	90
B.3	Tabel hasil pengujian dengan parameter <i>crossover rate</i> sebesar 80%. . . . .	91
B.4	Tabel hasil pengujian dengan parameter <i>mutation rate</i> sebesar 0.1%. . . . .	92
B.5	Tabel hasil pengujian dengan parameter <i>mutation rate</i> sebesar 1%. . . . .	92
B.6	Tabel hasil pengujian dengan parameter <i>acceptance probability</i> sebesar 0.5. . . . .	93
B.7	Tabel hasil pengujian dengan parameter <i>acceptance probability</i> sebesar 1. . . . .	94



# BAB 1

## PENDAHULUAN

Bab ini membahas bagian-bagian skripsi yang akan dijabarkan lebih lanjut sekaligus menjadi pedoman dan acuan penulisan pada bab-bab selanjutnya, yaitu Latar Belakang, Rumusan Masalah, Tujuan, Batasan Masalah, Metodologi, dan Sistematika Pembahasan.

### 1.1 Latar Belakang

Pada kurun waktu tertentu, sebuah perusahaan yang memproduksi kendaraan bermotor, khususnya mobil, menjual beberapa segmen mobil berbeda sesuai dengan kebutuhan pasar. Sebagai contoh, sebuah produsen mobil akan menyediakan mobil bersegmen *hatchback* untuk konsumen yang membutuhkan kemampuan membawa penumpang maksimal empat hingga lima orang, sedan untuk konsumen yang membutuhkan kemampuan membawa barang yang lebih banyak dari *hatchback*, MPV (*Multi Purpose Van*) untuk konsumen yang membutuhkan kemampuan membawa penumpang maksimal tujuh hingga delapan orang, dan sebagainya. Setiap segmen mobil yang ditawarkan dapat dibagi ke dalam beberapa jenis mobil tertentu. Pabrik tempat jenis-jenis mobil tersebut diproduksi akan memiliki satu lini produksi untuk setiap jenis mobil.

Sebagai ilustrasi, Tabel 1.1 menunjukkan jenis-jenis mobil yang ditawarkan Toyota untuk pasar Indonesia berdasarkan segmennya per awal Januari 2022<sup>1</sup>:

Tabel 1.1: Jenis mobil Toyota berdasarkan segmennya.

MPV	Sedan	SUV	Hybrid	Hatchback	Komersial	Sport
All New Veloz	New Vios	New Fortuner GR Sport	All New Corolla Cross Hybrid	New Agya GR Sport	New Hilux D Cab	GR Yaris
All New Avanza	All New Corolla Altis	New Rush GR Sport	All New Corolla Altis Hybrid	New Yaris GR Sport	Dyna	Toyota GR Supra
New Alphard	All New Camry	All New Raize GR Sport	All New CHR Hybrid		All New Hiace Premio	New Toyota 86
New Vellfire		All New Corolla Cross	All New Camry Hybrid		New Hilux S Cab	
New Venturer		All New CHR				

---

<sup>1</sup><https://www.toyota.astra.co.id/home>

New Kijang Innova		Land Cruiser				
New Calya						
New Sienta						
All New Voxy						

Pada Tabel 1.1, terdapat beberapa jenis mobil yang cukup dikenal masyarakat Indonesia, seperti New Kijang Innova dan New Calya. Pada jenis New Calya, terdapat dua tipe yang tersedia, yaitu tipe 1.2 G dan tipe 1.2 E<sup>2</sup>. Masing-masing tipe tersebut memiliki perbedaan fitur yang terpasang, yang disebut juga opsi. Khusus untuk New Calya tipe 1.2 G, Toyota memberikan opsi *LED Headlamp* dan *Touchscreen Head Unit* yang tidak ada pada tipe 1.2 E<sup>2</sup>.

Pada masa-masa awal industri otomotif menerapkan lini produksi, produsen mobil mengutamakan kuantitas mobil yang dapat dihasilkan. Hal tersebut bertujuan untuk meningkatkan ketersediaan mobil di *dealer* dan menurunkan biaya produksi. Efek langsung dari turunnya biaya dan waktu produksi mobil adalah harga jual mobil yang menurun, sehingga daya beli masyarakat dapat meningkat. Hal tersebut dibuktikan dengan penurunan harga mobil Ford Model T seiring waktu produksinya, karena diproduksi melalui lini produksi berjalan. Penurunan harga jual ini dapat dibuktikan dengan turunnya harga jual Ford Model T seiring waktu, mulai dari \$850 saat peluncurannya hingga \$260 untuk harga terendahnya<sup>3</sup>. Di balik kelebihan yang didapatkan, jumlah tipe mobil yang diproduksi dan opsi yang akan terpasang pada masing-masing tipe ditentukan produsen. Pembatasan tersebut menyebabkan jumlah tipe mobil tidak berubah dalam beberapa tahun produksi, bahkan hingga jenis mobil tersebut diperbaharui atau berhenti produksi.

Perkembangan selanjutnya dari lini produksi adalah penambahan opsi pada tipe tertentu yang ditentukan oleh produsen seiring perkembangan teknologi. Pemasangan opsi tersebut akan merambah kepada setiap tipe mobil yang diproduksi seiring berjalannya produksi jenis mobil tersebut. Sebagai contoh, opsi radio yang termasuk fitur baru untuk mobil pada tahun 1930-an hanya dipasang pada tipe mobil tertinggi oleh produsen. Seiring produksi jenis mobil tersebut, akhirnya opsi radio terpasang pada setiap mobil yang diproduksi. Opsi radio yang sudah dipasang kepada semua tipe disebut sebagai fitur standar dalam produksi dan penjualan mobil.

Seiring waktu dan perkembangan teknologi, produsen mulai menyediakan beberapa opsi khusus di luar opsi yang ditentukan produsen pada tipe mobil. Opsi khusus ini biasa disebut fitur opsional di dalam brosur atau dokumen penjualan mobil. Konsumen diberikan kebebasan untuk memilih fitur opsional dengan penambahan harga mobil sesuai biaya pemasangan opsi tersebut, sehingga tipe mobil yang masuk ke lini produksi akan bertambah. Dengan adanya fitur opsional, tipe mobil akan terbagi menjadi beberapa tipe mobil dengan opsi yang sudah ditentukan produsen dan tipe mobil yang dipasangkan opsi yang ditentukan oleh konsumen. Penambahan opsi yang menjadi fitur-fitur opsional tersebut yang akan dibahas pada permasalahan ini. Ilustrasi pembagian fitur opsional (opsi pilihan konsumen) pada tipe mobil dari sebuah jenis mobil dijelaskan pada Tabel 1.2 berikut ini:

<sup>2</sup><https://www.toyota.astra.co.id/product/calya>

<sup>3</sup><https://www.nydailynews.com/autos/ford-assembly-line-turns-100-changed-society-article-1.1478331>

Tabel 1.2: Tabel ilustrasi jenis mobil dan tipe-tipenya.

Jenis mobil: Sedan X	Ops			Harga
	<i>Anti-lock Braking System</i>	<i>Sunroof</i>	<i>Airbag</i>	
Sedan X Tipe 1	Tidak	Tidak	Ada	100
Sedan X Tipe 2	Tidak	Ada	Ada	115
Sedan X Tipe 3	Ada	Tidak	Ada	125
Sedan X Tipe 4	Ada	Ada	Ada	135

Dari Tabel 1.2, terdapat tiga opsi untuk setiap tipe Sedan X. Masing-masing tipe memiliki opsi yang berbeda-beda, sehingga harga dari masing-masing tipe akan berbeda juga. Sedan X Tipe 1 adalah tipe sedan dengan harga termurah karena opsi yang terpasang hanya *Sunroof*, sedangkan Sedan X Tipe 4 adalah tipe sedan termahal karena memiliki ketiga opsi yang tersedia.

Dalam lini produksi Sedan X ini, akan terdapat beberapa *workstation* yang memasangkan *option-option* tersebut untuk tipe sedan yang berbeda. Setiap *workstation* akan mengerjakan satu opsi saja. *Workstation* pun akan memiliki batasan pengerjaan yang disebut *capacity constraint*, yaitu batas mobil yang dapat dikerjakan dari semua mobil yang melewati *workstation* tersebut. Misalkan, untuk pemasangan ABS (*Anti-lock Braking System*), *workstation* yang bersangkutan dapat dilewati lima buah mobil tapi dapat memasang ABS pada tiga mobil saja.

Jika hal-hal di atas sudah diketahui, maka *Car Sequencing Problem* dapat diselesaikan. *Car Sequencing Problem* adalah sebuah masalah pencarian urutan pengerjaan sebuah jenis mobil dalam suatu lini produksi. Tujuan utamanya adalah menghindari terjadinya kelebihan beban pekerjaan (*work overload*) dan mengoptimalkan banyaknya *part* yang diperlukan. Jika urutan yang paling sedikit tundaan pekerjaannya (*delay*) ditemukan, maka tujuan *Car Sequencing Problem* telah tercapai.

*Car Sequencing Problem* pertama kali dijelaskan oleh Parello dkk. pada tahun 1986 dan dibahas kembali oleh *Société française de Recherche Opérationnelle et d'Aide à la Décision* (ROADEF), lembaga riset operasional di Perancis, sebagai subjek studi kasus pada tahun 2005. Subjek tersebut diajukan oleh Renault, sebuah produsen mobil yang telah memecahkan masalah ini pada tahun 1993 [1].

Dalam pembahasan *Car Sequencing Problem*, *capacity constraint* adalah suatu variabel tetap. Sebenarnya *capacity constraint* ditentukan berdasarkan variabel-variabel seperti kemampuan kerja setiap pekerja dalam *workstation* tersebut, waktu yang dihabiskan untuk memasang opsi, waktu berjalannya setiap mobil untuk masuk ke dalam *workstation*, dan sebagainya. Tetapi *Car Sequencing Problem* mengesampingkan variabel-variabel tersebut. *Car Sequencing Problem* hanya mencari urutan mobil yang memiliki tundaan kerja (*delay*) yang paling kecil saja.

Oleh karena itu, pada kasus ini urutan mobil yang akan memasuki lini produksi sedan menjadi penentu utama optimal atau tidaknya jalan lini produksi tersebut. Sebagai contoh, jika dalam urutan mobil yang masuk ke dalam *workstation* ABS terdapat dua Sedan Tipe 3 dan satu Sedan Tipe 4, maka *workstation* tersebut memenuhi *capacity constraint* yang telah ditetapkan. Jika Sedan Tipe 4 bertambah satu saja, maka *workstation* tersebut akan memiliki *delay*, karena Sedan Tipe 4 terakhir harus menunggu Sedan Tipe 4 sebelumnya untuk selesai dikerjakan.

Permasalahan *Car Sequencing Problem* ini pernah dipecahkan oleh Renault dengan mengoptimisasi perangkat lunak internal berdasarkan *Simulated Annealing* [1], dan tentu dapat diselesaikan dengan beberapa algoritma lainnya. Sebelum ROADEF membahas kembali permasalahan ini, terdapat beberapa peneliti yang telah memecahkan *Car Sequencing Problem* dengan algoritma-algoritma lainnya. Salah satu contoh penyelesaiannya adalah menggunakan *Genetic Algorithm* yang telah dilakukan oleh Terry Warwick dan Edward Tsang pada tahun 1995 [2].

Pada kasus di luar *Car Sequencing Problem*, *Genetic Algorithm* dan *Simulated Annealing* dapat digabungkan dengan tujuan tertentu, seperti mendapatkan hasil yang lebih baik atau performa yang lebih cepat. Penggabungan kedua algoritma tersebut telah dibuat dan diuji oleh Alfarisy dkk. pada jurnalnya di tahun 2016 [3]. Pada jurnal tersebut, penggabungan dibuat untuk menyelesaikan

kasus *Function Optimization*. Terdapat tiga bentuk penggabungan yang dibuat, yaitu *Hybrid GA-SA*, *Hybrid SA-GA*, dan *Cyclic GA-SA*. Pada skripsi ini, bentuk-bentuk penggabungan tersebut akan diimplementasikan untuk mencari urutan mobil terbaik pada *Car Sequencing Problem* dan dibandingkan performanya dengan *Genetic Algorithm* dan *Simulated Annealing* tanpa penggabungan.

Pada perangkat lunak yang akan dibuat, *Genetic Algorithm*, *Simulated Annealing*, dan kombinasinya akan diimplementasikan dalam bahasa Java. Proses berjalannya perangkat lunak dimulai dari memasukkan variabel-variabel yang akan digunakan dalam perhitungan *Car Sequencing Problem*, *Genetic Algorithm*, *Simulated Annealing*, dan kombinasinya. Cara memasukkan variabel pun akan dibagi dua, yaitu melalui *input* secara manual melalui konsol dan *input* secara otomatis melalui *file input* dengan ekstensi `.txt`. *Input* secara otomatis akan mempermudah pengujian yang memerlukan pengulangan soal-soal yang akan dikerjakan.

Soal-soal tersebut berasal dari situs CSPLib<sup>4</sup>, sebuah situs yang menyimpan soal-soal khusus untuk *Car Sequencing Problem*. Beberapa soal tersebut berasal dari soal pengujian pada artikel A Filtering Algorithm for Global Sequencing Constraints yang ditulis oleh Régis dan Puget [4]. Dari sekian banyak soal akan dipilih sepuluh soal untuk pengujian.

Hasil yang diperoleh dari setiap tahap pengujian adalah pelanggaran *capacity constraint* dari urutan yang dihasilkan masing-masing algoritma. Setelah dilakukan perulangan berjalannya perangkat lunak, nilai pelanggaran masing-masing algoritma akan dihitung rata-ratanya. Rata-rata tersebut yang menentukan performa dari masing-masing algoritma pada setiap tahap pengujian.

## 1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah:

1. Apakah *Car Sequencing Problem* dapat diselesaikan dengan menggunakan kombinasi *Genetic Algorithm* dan *Simulated Annealing*?
2. Bagaimana cara mengimplementasikan *Genetic Algorithm*, *Simulated Annealing*, dan kombinasi keduanya ke dalam perangkat lunak untuk menyelesaikan *Car Sequencing Problem*?
3. Adakah algoritma terbaik di antara *Genetic Algorithm*, *Simulated Annealing*, dan kombinasinya, untuk menyelesaikan *Car Sequencing Problem*?

## 1.3 Tujuan

Penelitian ini dilakukan dengan tujuan sebagai berikut:

1. Mempelajari metode yang tepat untuk menyelesaikan *Car Sequencing Problem* dengan kombinasi *Genetic Algorithm* dan *Simulated Annealing*.
2. Mengimplementasikan *Genetic Algorithm*, *Simulated Annealing*, dan kombinasi keduanya ke dalam perangkat lunak untuk menyelesaikan *Car Sequencing Problem*.
3. Mempelajari perbandingan kinerja *Genetic Algorithm*, *Simulated Annealing*, dan kombinasi kedua algoritma tersebut dalam penyelesaian *Car Sequencing Problem*.

## 1.4 Batasan Masalah

Beberapa batasan masalah untuk skripsi ini adalah:

1. Penentu terjadinya *delay* pada urutan hanya berdasarkan *capacity constraint*, tidak dipengaruhi oleh waktu berjalannya lini produksi, kemampuan kerja pekerja pada *workstation*, dan sebagainya (diperjelas pada Bab 2).

---

<sup>4</sup><https://www.csplib.org/>

## 1.5 Metodologi

Tahapan-tahapan yang akan dilakukan untuk menyelesaikan skripsi ini adalah:

1. Melakukan studi literatur mengenai makalah-makalah dan penelitian-penelitian yang membahas dasar-dasar *Car Sequencing Problem*, *Genetic Algorithm*, *Simulated Annealing*, dan kombinasi antara *Genetic Algorithm* dan *Simulated Annealing*.
2. Melakukan pemodelan implementasi *Genetic Algorithm*, *Simulated Annealing*, dan kombinasinya ke dalam perangkat lunak.
3. Menganalisis rancangan implementasi perangkat lunak yang telah dibuat.
4. Membuat perangkat lunak yang melakukan perhitungan *Car Sequencing Problem*.
5. Mengimplementasikan pencarian urutan terbaik dengan *Genetic Algorithm*, *Simulated Annealing*, dan kombinasinya ke dalam perangkat lunak.
6. Melakukan pengujian dan evaluasi terhadap perangkat lunak yang telah dibuat.
7. Menyusun dokumen skripsi.

## 1.6 Sistematika Pembahasan

- Bab 1: Pendahuluan  
Bab 1 menjelaskan Latar Belakang, Rumusan Masalah, Tujuan, Batasan Masalah, Metodologi, dan Sistematika Pembahasan untuk skripsi ini. Pada bagian Latar Belakang, akan disajikan tabel contoh dan pembahasan singkat mengenai *Car Sequencing Problem* yang akan dilanjutkan di bagian Landasan Teori.
- Bab 2: Landasan Teori  
Bab 2 menjelaskan teori-teori yang mendasari skripsi ini, yaitu teori mengenai *Car Sequencing Problem*, *Genetic Algorithm*, dan *Simulated Annealing*. Dijelaskan juga mengenai cara mengkombinasikan kedua algoritma tersebut.
- Bab 3: Pemodelan dan Analisis  
Bab 3 menguraikan cara-cara mengimplementasikan perhitungan *Car Sequencing Problem* dan penyelesaian *Genetic Algorithm*, *Simulated Annealing*, dan kombinasinya di dalam perangkat lunak. Implementasi ini kemudian akan dianalisis satu per satu.
- Bab 4: Perancangan  
Bab 4 menunjukkan perancangan perangkat lunak yang akan melakukan perhitungan *Car Sequencing Problem* dan menyelesaikan permasalahan tersebut dengan *Genetic Algorithm*, *Simulated Annealing*, dan kombinasinya.
- Bab 5: Implementasi dan Pengujian  
Bab 5 menjelaskan implementasi dari perancangan pada bab sebelumnya ke dalam perangkat lunak. Selanjutnya akan dijelaskan cara-cara dan hasil pengujian perangkat lunak tersebut.
- Bab 6: Kesimpulan dan Saran  
Bab 6 berisi kesimpulan dari skripsi ini dan saran dari penulis untuk perbaikan selanjutnya.