

# Perbaikan Kondisi Keseimbangan Lintasan Produksi Dengan Penerapan Metode Genetic Algorithm Pada Lintasan Produksi Muffler K-59J PT

Febryan Sutomo\*, Trifenaus Prabu Hidayat, Karel Octavianus Bachri

Program Studi Teknik Industri, Fakultas Teknik, Universitas Katolik Indonesia Atma Jaya  
Jalan Raya Cisauk, Sampora, Cisauk, Tangerang, Banten 15345

## Article Info

## Abstract

### Article history:

Received  
02 November 2021

Accepted  
05 Desember 2021

### Keywords:

*bottleneck, Line  
Balancing, Genetic  
Algorithm*

PT. XYZ is an automotive manufacturing company that has been working as a supplier of automotive parts to several company in Indonesia. Most of the production line is already good enough, but there is one of their sub product that has problems in their production process. The sub product is muffler K-59J. There are problems on its production line such as bottleneck, overtime. Another problem can be seen from the data, that its line efficiency is only 58,81 %. Thus, this study aims to improve the performance of the production line. Therefore, a computerized Genetic Algorithm method is proposed to improve its performance. The Genetic Algorithm process is made using matlab software. In the process, several stages are carried out starting from making a genetic representation, selecting parents, making the recombination and mutation process, and regenerating the population. But in this study, the generate population process is filtered into the two process. So the output of the population is already qualified for the line balancing requirements. From the results of the Genetic Algorithm method, it is found that the efficiency of the path is 74.07 % with a total of 18 work stations. thus the value of the efficiency of the path was successfully increased by 15.26%.

## 1. PENDAHULUAN

Saat ini, perkembangan teknologi bergerak dengan cepat. Perkembangan teknologi yang cepat ini, membawa industri-industri khususnya industri manufaktur juga mengalami kemajuan. Hal tersebut dapat dilihat dengan mulainya era industri 4.0 di berbagai belahan dunia. Meskipun industri 4.0 masih terbilang sedikit yang diterapkan dalam perindustrian di Indonesia, tetapi berbagai industri manufaktur di Indonesia sudah mulai banyak yang memanfaatkan teknologi dalam proses produksinya mulai dari penggunaan sistem komputerisasi yang bagus, penggunaan robot-robot produksi, proses produksi yang mulai saling terintegrasi satu sama lain, dan lain sebagainya. Hal tersebut yang juga dipertahankan dan ditingkatkan oleh PT XYZ dalam perusahaannya, dimana PT XYZ ini merupakan salah satu perusahaan *supplier* kebutuhan otomotif yang ada di Indonesia dan sudah melayani berbagai kebutuhan konsumennya dari berbagai perusahaan otomotif terkemuka di Indonesia. PT XYZ ini memiliki berbagai lini produksi dari beragam produk otomotif mulai dari roda 2 hingga roda 4. Dengan beragam produk tersebut, tentu PT XYZ sangat memperhatikan kondisi lintasan produksi

masing-masing bagian agar tidak mengganggu hasil produksi setiap produknya dan tetap memenuhi kebutuhan konsumennya tepat waktu. Tetapi, dalam proses perkembangan industri manufaktur, tentu butuh waktu agar setiap bagian dapat berkembang sesuai harapan. Dengan demikian, memungkinkan untuk terdapat beberapa lintasan produksi yang masih berada di bawah standard perusahaan. Hal tersebut terjadi pada salah satu lintasan produksi *muffler* di PT XYZ. Pada lintasan produksi tersebut, diketahui masih terdapat berbagai kendala mulai dari ditemukannya *bottleneck* pada beberapa bagian stasiun kerja, hasil produksi yang kadang menurun, hingga besar efisiensi lintasan yang lebih rendah dari lintasan produksi lainnya. Data-data kondisi ini didapatkan dari hasil pengamatan langsung dan proses diskusi dengan beberapa bagian yang dilakukan selama proses penelitian.

Proses observasi pada penelitian ini berfokus pada satu produk yaitu *muffler* K-59J pada sektor 2-*Wheel*. Pemilihan produk tersebut didasarkan pada kendala dari perusahaan yang sudah dijelaskan sebelumnya yaitu lintasan produksi yang masih belum bisa bekerja secara optimal karena memiliki efisiensi lintasan yang cukup rendah. Data kinerja pada lintasan produksi *muffler* K-59J dapat dilihat

\*Corresponding author. Febryan Sutomo  
Email address: febr্যানsutomo99@gmail.com

pada tabel 1. Data-data tersebut didapatkan dari hasil pengamatan langsung yang dilakukan selama kerja praktik pada PT. XYZ yang saat itu menghasilkan penelitian kerja praktik menggunakan metode RPW dan J-Wagon. Pada pengamatan tersebut, terdapat 25 elemen kerja yang terbagi menjadi 13 stasiun kerja awal.

**Tabel 1.**  
Waktu Siklus setiap Elemen Kerja

Stasiun Kerja	No.	Waktu Baku (s)	Total Waktu Stasiun Kerja (s)	Efisiensi SK
1	1	64,32	117,53	76,33
	2	53,21		
2	3	60,96	95,08	63,06
	4	34,11		
	5	16,89		
3	6	16,11	92,54	64,20
	7	44,22		
	8	15,33		
4	9	37,99	75,54	51,72
	10	37,55		
5	11	24,93	83,27	56,98
	12	58,34		
6	13	30,66	102,24	65,37
	14	32,31		
	15	39,26		
7	16	66,25	123,14	78,45
	17	56,89		
8	18	31,27	67,09	43,51
	19	35,82		
9	20	67,32	154,01	100,00
	21	86,69		
10	22	36,44	36,44	26,14
11	23	88,22	88,22	56,35
12	24	76,16	76,16	41,40
13	25	64,91	64,91	41,06
Total			1176,17	
<i>Line Efficiency</i>			58,74	
<i>Balance Delay</i>			41,26	
<i>Smoothness Index</i>			251,18	

Berdasarkan tabel di atas, dapat dilihat bahwa nilai efisiensi lintasan pada lintasan produksi *muffler* K-59J masih terbilang sangat rendah yaitu sebesar 58,81%. Selain itu juga waktu antar stasiun kerja terendah dan terbesar masih cukup jauh yaitu sebesar 36,44 detik dan 154, 01 detik. Hal ini menyebabkan kemungkinan terjadinya penumpukan pada stasiun kerja tertentu.

Berdasarkan permasalahan di atas, penelitian ini berusaha untuk memberikan usulan perbaikan pada lintasan produksi tersebut dan dibandingkan dengan kondisi semula. Metode yang akan diusulkan adalah metode *Genetic Algorithm*. Dalam proses pembuatannya, metode *Genetic Algorithm* ini akan dibantu dengan *software Matlab*. Penggunaan metode *Genetic Algorithm* ini

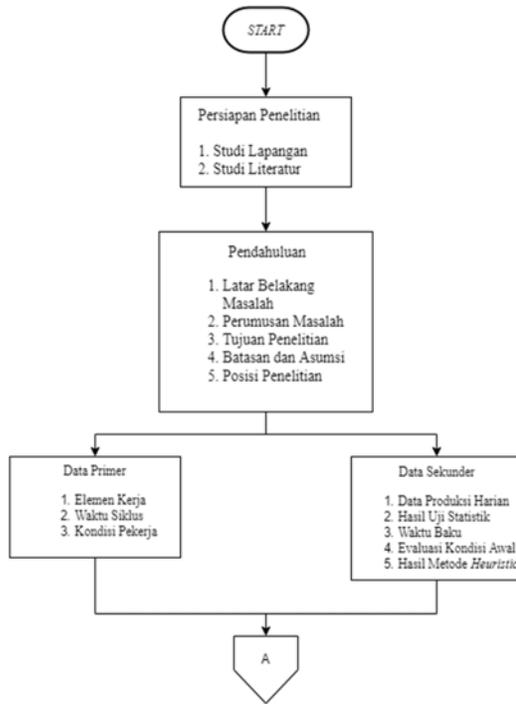
didasarkan karena metode tersebut merupakan salah satu metode berjenis *meta-heuristic* yang berbasis *computer science* (Manuf, 2008). Dengan kata lain, metode ini dapat melakukan proses perhitungannya dengan bantuan komputasi sehingga proses akan dijalankan terus-menerus hingga hasil optimal tercapai. Dengan permasalahan yang sama, metode ini dapat melakukan proses terus-menerus dan menghasilkan hasil secara konstan berdasarkan setiap kemungkinan yang ada. Sehingga akan memungkinkan banyak hasil yang akan di bandingkan satu sama lain (Brahim R. & Alain D., 2005).

## 2. METODOLOGI

Bagian metodologi penelitian akan menjelaskan secara lebih rinci setiap proses penelitian yang dilakukan, mulai dari latar belakang permasalahan sampai penarikan kesimpulan. Dengan membuat metodologi penelitian yang lebih terstruktur, maka penelitian yang dibuat akan lebih terarah dan dapat memecahkan permasalahan yang ada. Gambaran alur penelitian hingga proses pengolahan data menggunakan metode *Genetic Algorithm* akan ditunjukkan pada Gambar 1.

Proses awal penelitian ini dilakukan dengan beberapa tahapan. Tahap awal dalam penelitian ini adalah melakukan studi lapangan. Studi lapangan dilakukan selama masa kerja praktik. Studi lapangan dilakukan dengan kunjungan dan observasi langsung pada area pabrik khususnya pada lintasan produksi *muffler* K-59J. Selama proses studi lapangan, dilakukan observasi langsung dan proses pengambilan data dengan menggunakan bantuan *stopwatch*. Data waktu produksi diambil dengan menghitung proses kerja dari masing-masing elemen kerja dan diulang sesuai replikasi yang dibuat. Studi lapangan ini diperkuat dengan adanya interaksi langsung dengan beberapa bagian pada bagian produksi seperti *department head 2-Wheel*, *Section Head muffler 2-Wheel*, dan *Group Head line muffler K-59J*. Interaksi ini berupa wawancara dan diskusi yang dilakukan saat proses kerja praktik. Selain itu, selama proses observasi juga dilakukan pengamatan terhadap alur proses produksi yang ada sehingga dapat dibuat *precedence diagramnya* sesuai kondisi nyata.

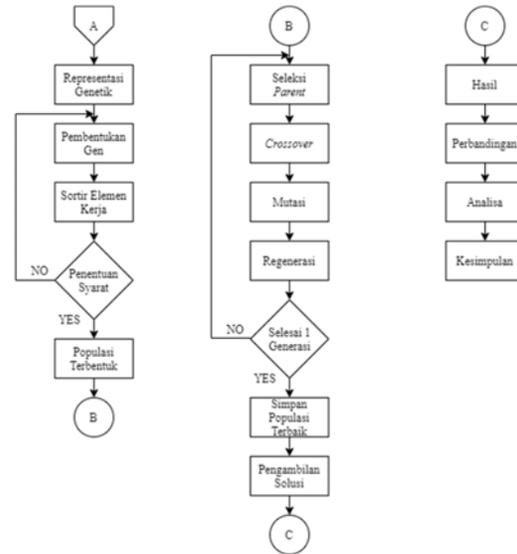
Selain itu, juga dilakukan studi literatur untuk mencari data tambahan dan berbagai pengetahuan tambahan yang berasal dari jurnal-jurnal, buku, dan berbagai sumber lain untuk mendukung proses penyelesaian penelitian. Data-data yang dimaksud adalah data-data atau teori mengenai *line balancing*, metode *heuristic* dan metode *random* keseimbangan lintasan, dan *Genetic Algorithm*.



**Gambar 1.**  
Alur Penelitian

Proses penelitian ini, adalah lanjutan dari penelitian kerja praktik sebelumnya, di mana pada penelitian tersebut memberikan usulan perbaikan dengan metode *heuristic* seperti RPW dan J-Wagon. Oleh karena itu setiap proses pengolahan data waktu sudah dilakukan. Dengan demikian pada penelitian ini, menggunakan data waktu hasil perhitungan statistik yang sudah ada, dan data-data tambahan pada penelitian tersebut akan menjadi data sekunder pada penelitian ini untuk mendukung penelitian.

Penelitian ini akan menekankan pada penggunaan metode *Genetic Algorithm* dalam penyelesaian masalah keseimbangan lintasan yang terjadi pada lintasan produksi *muffler K-59J*. proses pembuatan metode *Genetic Algorithm* ini akan dibantu dengan menggunakan bantuan *software Matlab*. Proses pembuatan metode *Genetic Algorithm* ini sesuai dengan alur pada Gambar 2.



**Gambar 2.**  
Alur *Genetic Algorithm*

Langkah-langkah yang dilakukan antara lain:

1. Representasi Genetik

Representasi genetik merupakan tahapan awal yang perlu dibuat sebagai *input* data program. Representasi genetik merupakan proses penyesuaian setiap data-data awal ke dalam kondisi atau bentuk genetik. Pada tahap ini, perlu dijabarkan setiap data pendukung dan data awal untuk penyelesaian masalah dan dikonversikan ke dalam gen, kromosom, dan populasinya serta menambahkan setiap kebutuhan-kebutuhan data tambahan untuk menjalankan program. Sehingga dengan membuat representasi genetik ini, program yang dijalankan nantinya akan mengenal dan dapat mengidentifikasi data-data yang ada kedalam metode *Genetic Algorithm* (Anderson dan Ferris, 2001).

2. Menentukan *Fitness Value*

*Fitness value* merupakan hasil kombinasi yang tercipta dan melambangkan besar ketepatan dari hasil yang dicapai dengan hasil yang diinginkan. Sehingga dari besar *fitness value* ini dapat diketahui besar ketepatan dari hasil pemecahan metode yang sudah dibuat (Anderson dan Ferris, 2001). Dalam penelitian ini, *fitness value* yang diinginkan adalah besar efisiensi lintasan. Oleh karena itu, semakin besar *fitness value* berarti tingkat kecocokannya juga semakin besar. Sehingga penelitian ini akan mencari secara terus menerus kombinasi yang memungkinkan hingga mendapat nilai efisiensi lintasan terbesar.

3. Populasi

Pada tahap ini akan dibentuk sebuah inisiasi populasi awal berdasarkan hasil terjemahan permasalahan atau keadaan yang ada ke dalam bentuk kromosom yang nantinya menjadi sebuah

populasi. Pembentukan populasi ini dilakukan dengan menentukan terlebih dahulu banyak populasi yang ingin dibentuk lalu setelah itu membangkitkan populasi berdasarkan banyak populasi tersebut.

#### 4. Seleksi

Setelah sudah terbentuk populasi awal, dilanjutkan dengan menyeleksi populasi tersebut untuk mendapatkan 2 kromosom terbaik yang akan dijadikan sebagian kromosom parentalnya. Proses seleksi ini dilakukan dengan menyeleksi *fitness function* dan mencari *fitness function* tertinggi. 2 kromosom terbaik tersebut nantinya akan dikeluarkan dari populasinya dan menjadi *parent chromosome 1* dan *parent chromosome 2* yang kemudian akan digunakan kedalam proses rekombinasi.

#### 5. Crossover

*Crossover* merupakan proses persilangan atau rekombinasi yang akan dilakukan pada populasi yang sudah dibuat sebelumnya. pada *crossover* ini akan digunakan metode *Single Point Crossover*. Menurut Oguzhan & Fuat (2000), Metode *Single Point Crossover* ini akan menentukan sebuah poin pusat untuk dilakukan persilangan seperti yang dijelaskan pada bagian tinjauan pustaka.

#### 6. Mutasi

Mutasi dilakukan untuk meningkatkan kemungkinan dan probabilitas pada proses pencarian solusi dan menghasilkan individu baru hasil proses mutasi. Proses mutasi ini dilakukan dengan metode *Random Resetting*. Metode ini berjalan dengan menggunakan pilihan gen secara acak untuk ditukar. Oleh karena itu, perlu bantuan bilangan *random*.

#### 7. Regenerasi

Regenerasi ini dilakukan untuk menyilangkan atau mengganti populasi dengan nilai *fitness value* terkecil dengan gen anakan yang sudah dibuat sebelumnya. sebelum melakukan tahapan ini, setiap kromosom akan dikumpulkan dan diurutkan. Kemudian proses regenerasi dilakukan dengan memilih kromosom dengan nilai *fitness value* terendah untuk ditukar dengan gen anakan yang sudah dibuat sebelumnya.

#### 8. Finalisasi

Setelah dilakukan regenerasi, maka proses *Genetic Algorithm* ini sudah selesai. Proses akhir yang perlu dilakukan adalah membuat program yang sudah dibuat melakukan tugasnya berulang sampai mendapat hasil terbaiknya. Proses ini dilakukan dengan bantuan *looping*. program yang sudah dibuat akan di *looping* Kembali ke proses seleksi karena pada proses regenerasi, kromosom anakan akan dikembalikan ke populasi awal sehingga setelah itu perlu diseleksi Kembali dan proses akan berlanjut seperti awal hingga mendapatkan hasil yang diinginkan. Untuk

menghentikan proses *looping* ini perlu dibuat terminasi. Terminasi pada program ini adalah ketika proses mendapat hasil *fitness function* tertinggi dan kromosom yang dihasilkan sesuai dengan syarat keseimbangan lintasan.

### 3. HASIL DAN PEMBAHASAN

Proses penelitian dilakukan dengan bantuan *software Matlab*, dan proses pembuatan metode *Genetic Algorithm* hanya akan dijelaskan urutan dan logika prosesnya saja karena untuk proses-proses komputasi sudah dijalankan ke dalam bahasa pemrograman. Proses usulan perbaikan menggunakan metode *Genetic Algorithm* adalah sebagai berikut.

#### 1. Representasi Genetik

Pada tahap ini, setiap *input* data yang ada diubah ke dalam bentuk genetik sehingga terbentuk gen, kromosom, dan selanjutnya membentuk populasi. Proses pembentukan gen elemen kerja dan gen stasiun kerja ini dilakukan dengan cara membuat sebuah matriks yang berisi data elemen kerja dan data waktu yang akan di *input*. Matriks tersebut sebagai data awal yang akan digunakan dalam program. Pada bagian ini juga dimasukan matriks tambahan yang merepresentasikan alur diagram dari proses produksi dengan mengurutkan banyak ketergantungan elemen kerja pada masing-masing elemen kerja. Kemudian setiap populasi yang terbentuk akan mengikuti urutan alur tersebut sehingga populasi yang terbentuk akan sesuai dengan *precedence diagram*.

#### 2. Penentuan *Fitness Function*

Setelah membuat representasi genetiknya, ditentukan *fitness value* yang akan menjadi pembatas dari pencarian ragam solusi tersebut. Dalam kasus penelitian ini, *fitness function* yang digunakan adalah besar efisiensi lintasan sehingga akan dipilih besar *fitness function* tertinggi yang dapat dihasilkan dari ragam solusi yang ada.

#### 3. *Generate* Populasi

Setelah melalui proses representasi genetik dan menentukan *fitness function* sebagai fungsi batasannya, dilanjutkan dengan proses pembangkitan populasi sebanyak jumlah yang sudah ditentukan. Permulaan pembangkitan populasi ini dengan membangkitkan terlebih dahulu gen elemen kerja dan stasiun kerja dengan bantuan pembangkitan bilangan acak. Untuk gen elemen kerja, diharapkan dapat ditampilkan setiap bilangan yang mewakili masing-masing elemen kerja di mana dalam penelitian ini terdapat 25 elemen kerja. sedangkan untuk stasiun kerja, diharapkan akan tertampil setiap bilangan dengan Batasan 14 sebagai stasiun kerja terendah dan 18 sebagai stasiun kerja tertinggi. Dengan demikian digunakan bilangan acak permutasi agar kondisi tersebut dapat terpenuhi. Hasil pembangkitan bilangan acak ini,

akan menghasilkan gen elemen kerja dan stasiun kerja yang kemudian akan digabungkan menjadi 1 populasi beserta *fitness value* nya menggunakan bantuan *structure array*.

4. Seleksi

Proses seleksi dilakukan untuk memilih 2 kromosom terbaik yang akan dijadikan kromosom induk. Penentuan ini didasarkan pada pemilihan *fitness function* terbesar. Dengan dilakukannya proses seleksi ini, maka populasi akan berkurang di mana sebelumnya terdapat 100 populasi, setelah diseleksi akan menjadi 98 populasi karena 2 populasi terbaiknya sudah diambil dan akan di proses ke tahap berikutnya.

5. Crossover

Proses rekombinasi atau *crossover* adalah proses persilangan yang dilakukan antara dua kromosom agar dapat dihasilkan dua kromosom baru dengan hasil yang baru sesuai turunan yang sudah ditukar. Proses rekombinasi ini dilakukan dengan metode *single point crossover* yang merupakan metode *crossover* dengan acuan 1 titik sebagai titik rekombinasinya. *Crossover point* yang dipilih adalah berada di tengah kromosom. Sehingga nantinya kromosom *parent* 1 akan dipotong menjadi 2 dan ditukarkan satu sama lain.

6. Mutasi

Proses mutasi adalah proses lanjutan setelah proses rekombinasi di mana pada proses ini beberapa bagian gen akan bermutasi berubah menjadi gen lain sesuai dengan ketentuan yang sudah dibuat. Proses mutasi ini dilakukan beracuan dengan laju mutasi yang ditentukan terlebih dahulu, yaitu sebesar 0,3. Proses mutasi ini dilakukan dengan membangkitkan bilangan *random* sebanyak panjang gen sebagai representasi dari masing-masing gen. setelah itu akan dibandingkan bilangan *random* yang sudah dibangkitkan dengan laju mutasi yang ada. Bila bilangan *random* lebih kecil sama dengan laju mutasi, maka gen tersebut akan bermutasi dengan cara dibangkitkan gen yang baru

7. Regenerasi dan Finalisasi

Proses regenerasi adalah proses pengembalian hasil persilangan dan mutasi kedalam populasi awal sehingga populasi tersebut kembali menjadi 100 populasi.

Setelah regenerasi, program *Genetic Algorithm* sudah selesai. Tetapi untuk menuntaskan sepenuhnya, perlu dilakukan beberapa kali pengulangan untuk mencari hasil terbaik. Proses ini dilakukan dengan tujuan membuat pengulangan tersebut dan terminasi yang dibutuhkan untuk menyelesaikan pengulangan tersebut. Proses tersebut merupakan banyak generasi yang akan dibentuk, di mana pada penelitian ini banyak generasi yang dibentuk adalah 100 generasi.

Proses pengulangan tersebut dilakukan dengan bantuan *looping*. setelah dilakukan rekombinasi dan mutasi, dilanjutkan dengan proses regenerasi di mana hasil kromosom anakan akan dikembalikan ke populasi awalnya. Oleh karena itu proses *looping* ini akan dilakukan kembali ke proses seleksi. Kemudian, proses *looping* tersebut harus diberhentikan dengan sebuah nilai tujuan. Nilai tujuan tersebut adalah *fitness function* terbaik dan urutan gen sesuai dengan *precedence diagram*.

Hasil terbaik yang dihasilkan pada metode *Genetic Algorithm* ini akan ditampilkan dalam Tabel2.

**Tabel 2.**  
Hasil *Genetic Algorithm*

SK	EK	Waktu (s)	Jumlah (s)	Efisiensi SK
1	9	37,99	54,88	62,21
	5	16,89		
	10	37,55		
2	6	16,11	78,59	89,09
	11	24,93		
3	7	44,22	44,22	50,12
4	12	58,34	58,34	66,14
5	13	30,66	45,99	52,13
	8	15,33		
6	14	32,31	71,58	81,14
	15	39,26		
7	16	66,25	66,25	75,10
8	1	64,32	64,32	72,91
9	17	56,89	56,89	64,49
10	2	53,21	84,48	95,77
	18	31,27		
11	3	60,96	60,96	69,11
	19	35,82		
12	4	34,11	69,93	79,27
	20	67,32		
13	20	67,32	67,32	76,31
14	21	86,69	86,69	98,28
15	22	36,44	36,44	41,31
16	23	88,22	88,22	100,00
17	24	76,16	76,16	86,33
18	25	64,91	64,91	73,59
<b>Total</b>			1176,17	
<b>Line Efficiency</b>			74,07	100,00
<b>Balance Delay</b>			25,93	
<b>Smoothness Index</b>			162,98	

Dari hasil tersebut, terlihat dengan lebih jelas susunan stasiun kerja dan elemen kerja hasil usulan metode *Genetic Algorithm*. Dari Tabel diatas dapat dilihat bahwa hasil solusi terbaik metode *Genetic Algorithm* menghasilkan 18 stasiun kerja dengan rata-rata efisiensi stasiun kerja sebesar 70%. Dari hasil tersebut dapat diketahui besar efisiensi lintasan yang dihasilkan adalah sebesar 74,07 % dengan diikuti besar nilai *balance delay* sebesar 25,93 % dan *smoothness index* sebesar 114,39 %. Hasil tersebut membuktikan bahwa peningkatan efisiensi lintasan produksi *muffler* K-59J adalah sebesar 15,33 % di mana besar efisiensi lintasan awal sebesar 58,74 %.

#### 4. PEMBAHASAN

Proses pengambilan data dilakukan selama masa kerja praktik pada PT XYZ yang merupakan salah satu perusahaan manufaktur yang bergerak dibidang industri otomotif dan sudah menjadi *supplier* bagi banyak *brand* otomotif terkemuka di Indonesia saat ini. proses pengambilan data ini diambil dengan banyak diskusi yang dilakukan bersama beberapa bagian dan pekerja di area produksi tersebut. Sehingga diketahui bahwa terdapat salah satu lintasan produksi yang bekerja di bawah rata-rata atau proses produksi yang dihasilkan di bawah persentase lintasan produksi lainnya. Berdasarkan hasil diskusi yang terjadi, terdapat beberapa penjelasan mengenai perbedaan performa pada lintasan produksi tersebut. Beberapa alasan tersebut adalah proses produksi yang sering berjalan tidak lancar atau terjadi *bottleneck* pada beberapa elemen kerja, sering kali target produksi harian tidak tercapai sehingga lintasan produksi tersebut sering melakukan *overtime*, selain itu juga nilai efisiensi lintasan produksinya yang rendah sebesar 58,74%. Pada penelitian kerja praktik sebelumnya, telah dilakukan usulan perbaikan serupa dengan metode *heuristic* RPW dan J-Wagon. Di mana pada usulan tersebut didapatkan hasil metode *Ranked Position Weight* (RPW) menunjukkan nilai efisiensi lintasan sebesar 74,07% dan metode J-Wagon menunjukkan nilai efisiensi lintasan sebesar 74,07%. Hasil nilai efisiensi tersebut sudah dapat dikatakan baik selain karena sudah terdapat peningkatan yang cukup jauh, juga besar efisiensi lintasan sudah melebihi 70% dan mendekati besar efisiensi lintasan rata-rata pada perusahaan yang memiliki besar efisiensi lintasan di kisaran 80%.

Penelitian ini berusaha meningkatkan efisiensi dari proses produksi *muffler* K-59J semakin baik. Oleh karena itu pada penelitian ini digunakan metode *Genetic Algorithm* untuk memaksimalkan besar efisiensi lintasan pada lintasan produksi *muffler* K-59J. Metode *Genetic Algorithm* ini merupakan salah satu metode yang berjalan secara komputerisasi. Sehingga hasil *output*nya akan terus dihasilkan sampai kondisi idealnya terpenuhi (Goldberg, 1989). Oleh karena itu, metode *Genetic Algorithm* merupakan metode yang sesuai untuk digunakan jika ingin mencari atau memaksimalkan efisiensi lintasan pada sebuah lintasan produksi. Proses pengolahan metode *Genetic Algorithm* ini, dilakukan dengan bantuan software Matlab.

Pada proses *Genetic Algorithm* ini, dibuat terlebih dahulu representasi genetiknya. Di mana pada proses representasi genetik ini akan didefinisikan setiap data yang digunakan ke dalam kondisi genetik yang sesuai. Pada proses ini juga dimasukan setiap *input* data yang diperlukan untuk mendukung program yang dibuat. Setelah proses

representasi genetik dibuat, perlu ditentukan *fitness value* yang diinginkan sebagai fungsi tujuan dari penelitian ini. dengan diketukannya *fitness value*, maka hasil penelitian dapat memiliki parameter tersendiri yang nantinya akan dibandingkan dengan hasil yang diharapkan agar terlihat seberapa besar ketepatan solusi yang dihasilkan. Di mana dalam penelitian ini adalah besar efisiensi lintasan yang dihasilkan.

Kemudian dilanjutkan dengan membuat *generate* populasi menggunakan pembangkitan bilangan *random* yang sudah diatur sesuai kebutuhan. Proses *generate* populasi ini diawali dengan membentuk gen elemen kerja dan stasiun kerja terlebih dahulu sebagai representasi dari data-data yang ingin diolah. Pembentukan gen elemen kerja dan stasiun kerja ini dilakukan dengan membangkitkan bilangan acak dengan bilangan acak permutasi karena menyesuaikan dengan kebutuhan dari kondisi bilangan acak yang ingin dibentuk. Untuk gen elemen kerja, diharapkan dapat ditampilkan setiap bilangan yang mewakili masing-masing elemen kerja di mana dalam penelitian ini terdapat 25 elemen kerja. sedangkan untuk stasiun kerja, diharapkan akan tertampil setiap bilangan dengan batasan 14 sebagai stasiun kerja terendah dan 18 sebagai stasiun kerja tertinggi. Di mana batas minimal 14 stasiun kerja didapatkan dari hasil perhitungan stasiun kerja minimal, dan batas maksimal 18 stasiun kerja didapat dari hasil pengolahan metode *heuristic* pada penelitian sebelumnya sebagai acuan. Setelah dibentuk masing-masing gen, maka akan digabungkan dengan *fitness function*nya menjadi sebuah populasi dengan menggunakan bantuan *structure array*.

Setelah terbentuk 1 populasi ini, populasi ini akan melalui proses penyaringan terlebih dahulu. Proses penyaringan ini bertujuan untuk memastikan populasi yang terbentuk memiliki besar nilai stasiun kerja tidak melebihi waktu elemen kerja tertinggi atau nilai *cycle time* nya. Setelah melalui proses tersebut, dapat dikatakan populasi ini adalah "populasi terpilih" sehingga menghasilkan 1 populasi baru. Proses tersebut akan diulang sampai mendapatkan 100 populasi baru yang berisi "populasi terpilih" tersebut. Setelah didapat 100 populasi, dilanjutkan dengan menyaring 100 populasi tersebut untuk diambil 2 populasi terbaik sebagai induk barunya berdasarkan *fitness value* terbesar.

Setelah itu, dilanjutkan dengan proses *crossover* dan mutasi. Pada proses *crossover*, dilakukan dengan metode *single point crossover*. Pemilihan metode ini adalah agar proses *crossover* tidak terlalu banyak merubah hasil susunan kromosom yang ada karena sebelumnya populasi tersebut sudah melalui penyaringan prasyarat keseimbangan lintasan. Pada *crossover* ini,

ditentukan 1 *crossover point* yang berada di tengah kromosom. Sehingga nanti akan ditukarkan setengah kromosom satu sama lain. Selanjutnya pada proses mutasi, dilakukan dengan bantuan rasio mutasi. Besar rasio mutasi ini ditentukan sebesar 3% dengan alasan yang sama seperti penentuan metode *crossover* sebelumnya. sehingga pada proses mutasi ini akan dibangkitkan Kembali bilangan acak antara 0 sampai 1 yang mewakili setiap elemen kerja. kemudian akan dicocokkan dengan besar rasio yang ada. Jika hasil bilangan acak lebih kecil dari rasio mutasi, maka gen tersebut akan diganti dengan membangkitkan bilangan acak baru. Setelah selesai proses mutasi, maka didapatkan hasil anakan baru yang siap dikembalikan ke populasi awalnya. Dengan demikian proses 1 generasi sudah selesai dan dapat diambil solusi terbaik dari 1 generasi tersebut. Tetapi, pada penelitian ini proses tersebut akan diulang sebanyak 100 kali sehingga didapatkan 100 generasi dengan solusi terbaiknya masing-masing. Setelah didapatkan 100 solusi dari masing-masing generasi, diambil 1 hasil terbaik sebagai solusi terbaiknya.

Solusi terbaik yang dihasilkan oleh metode *Genetic Algorithm* ini diambil dari 100 generasi yang sudah disimpan sebelumnya. Dari hasil tersebut, dapat dilihat solusi terbaik yang dihasilkan memiliki nilai *fitness* sebesar 74,07 yang berarti efisiensi lintasan sebesar 74,07 % dengan banyak stasiun kerja sebesar 18 stasiun kerja. dari hasil ini dapat diketahui kenaikan efisiensi lintasan sebesar 15,33 % dari kondisi awal. Selain itu jika diperhatikan pada setiap metode yang dibuat, menunjukkan besar efisiensi lintasan yang sama, tetapi jika dibandingkan, metode *Genetic Algorithm* memiliki hasil yang lebih baik karena memiliki nilai *smoothness index* yang lebih kecil yaitu sebesar 114,39. Sebagai perbandingan, dapat dilihat perbandingan hasil dari metode awal dan kondisi perbaikan yang ada pada Tabel 3.

**Tabel 3.**  
Perbandingan Hasil Penelitian

Parameter	Kondisi			
	Awal	RPW	J-Wagon	GA
<i>Line Efficiency</i>	58,74	74,07	74,07	74,07
<i>Balance Delay</i>	41,26	25,93	25,93	25,93
<i>Smoothness Index</i>	251,18	117,76	136,92	162,98
Jumlah Stasiun Kerja	13	18	18	18

Jika dilihat dari besar stasiun kerja minimum yaitu sebesar 14 stasiun tetapi yang dihasilkan terbaik sebesar 18 stasiun kerja, hal ini dapat terjadi karena beberapa hal. Hal tersebut antara lain pembagian beban kerja yang kurang merata di mana dapat dilihat dari waktu proses yang cenderung

tidak merata atau berbeda terlalu jauh antar elemen kerja.

Bagi perusahaan, usulan perbaikan ini dapat memberikan manfaat. Diantaranya dengan keseimbangan lintasan yang lebih baik, dapat mengurangi kemungkinan *bottleneck* dan *overtime*. Dengan demikian waktu lembur yang mungkin terjadi akibat target produksi yang tidak tercapai bisa dihindari dan perusahaan tidak perlu mengeluarkan uang lembur dan biaya listrik dan lainnya untuk memenuhi target produksi yang tidak tercapai tersebut. Selain itu, jika *bottleneck* pada beberapa stasiun kerja dapat dikurangi, maka hasil produksi pun dapat meningkat. Dengan kata lain, target produksi bisa tercapai tepat waktu bahkan dapat tercapai sebelum jam kerja selesai sehingga dapat menghemat biaya yang dikeluarkan dari konsumsi listrik, *maintenance* mesin, dll.

**5. KESIMPULAN DAN SARAN**

Kesimpulan yang didapat dari hasil penelitian ini adalah sebagai berikut.

- a. Kondisi awal lintasan produksi banyak ditemukan kendala dan memiliki besar efisiensi lintasan sebesar 58,81 %, selisih waktu yang cukup besar, dan sering terjadi *overtime* atau waktu kerja yang lebih karena target produksi harian yang tidak terpenuhi sehingga dibutuhkan evaluasi pada lintasan produksi tersebut.
- b. Solusi terbaik yang dihasilkan dengan metode *Genetic Algorithm* dibuat dengan bantuan software Matlab dan proses algoritma dapat dilihat pada Lampiran, dimana solusi terbaik nilai efisiensi lintasan sebesar 74,07 %, *ballance delay* sebesar 25,93 %, dan *smoothness index* sebesar 114,39. Dari hasil tersebut, dapat dilihat kenaikan nilai efisiensi lintasan sebesar 15,33 % dengan penambahan stasiun kerja menjadi 18 stasiun kerja.

Saran yang dapat diberikan dari hasil penelitian ini adalah sebagai berikut.

- a. Lintasan produksi *muffler* K-59J dapat memanfaatkan solusi yang dihasilkan dari metode *Genetic Algorithm* dengan menyesuaikan kondisi lingkungan produksi dengan susunan stasiun kerja yang diberikan. Dan menyesuaikan keadaan tata letak mesin agar mengikuti susunan stasiun kerja yang ada.
- b. Menentukan logika program pada pembangkitan bilangan acak, khususnya pada pembangkitan populasi awal dengan tepat dan efisien. Dengan logika yang tepat, dapat dihasilkan pembangkitan gen elemen kerja dan stasiun kerja dengan susunan yang lebih baik, sehingga variasi susunan elemen kerja dan stasiun kerja lebih beragam.

- c. Membuat *filter* program yang lebih sederhana tetapi tetap menyaring kedua fungsi Batasan. Sehingga proses iterasi tidak berjalan terlalu panjang dan program dapat dijalankan dengan waktu lebih singkat.

## 6. DAFTAR PUSTAKA

1. Anderson, E. J., & Ferris, M. C. (1994). Genetic Algorithms for Combinatorial Optimization: The Assembly Line Balancing Problem. *Operational Research Society of America*, 0899-1499.
2. Boysen, N., Fließdner, M., & Scholl, A. (2007). A classification of assembly line balancing problems. *European Journal of Operational Research*, 674-693.
3. Djunaidi, M., & Angga. (2017). Analisis Keseimbangan Lintasan (Line Balancing) pada Proses Perakitan Body Bus pada Karoseri Guna Meningkatkan Efisiensi Lintasan. *Jurnal Ilmiah Teknik Industri*, 77-84.
4. Goldberg, D. E. (1989). *Genetic Algorithms: In Search Optimization & Machine Learning*. Addison-Wesley Publishing Company, Inc.
5. Groover, M. P. (2015). Line Balancing Algorithms. Dalam *Automation, Production Systems, and Computer-Integrated Manufacturing* (hal. 405-410). Pearson.
6. Haryanto, R. M., Suhada, K., & Sapulette, G. L. (2014). Usulan Line Balancing Menggunakan Genetic Algorithm (Studi Kasus di PT Multi Garmenjaya, Bandung). *Jurnal Integra*, 161-174.
7. Hasancebi, O., & Erbatur, F. (2000). Evaluation of crossover techniques in genetic algorithm based optimum structural design. *Computers & Structures*, 435-448.
8. Hutahaean, H. A. (2012). Keseimbangan Lintasan. Dalam *Materi Perkuliahan: Perencanaan Dan Pengendalian Produksi* (hal. 28-51). Jakarta.
9. Lim, T. Y. (2012). Structured population genetic algorithms: a literature survey. *Springer Science*, 385-399.
10. Lisanto, A., Dewi, D. R., & Rahaju, D. E. (2014). Penerapan Model Optimasi Line Balancing dan Genetic Algorithm (Studi Kasus: PT Karya Mekar Dewatamali). *Jurnal Ilmiah Widya Teknik*, 50-63.
11. Mahyudi. (2016). Mengapa Sulit Membedakan Permutasi dan. *AdMathEdu*, 33-44.
12. Mondal, D., Chakrabarti, A., & Sengupta, A. (2020). Optimal and Robust Control. Power System Small Signal Stability Analysis and Control, 243-286.
13. Rikiek, B., & Delchambre, A. (2005). Evolutionary Combinatorial Optimisation. Dalam *Assembly Line Design: The Balancing of Mixed-Model Hybrid* (hal. 31). Brussels: Springer.
14. Rubinovitz, J., & Levitin, G. (1993). Genetic algorithm for assembly line balancing. *International Journal of Production Economics*, 343-354.
15. Santoso, A. (2005). Kajian Beberapa Uji Kenormalan dan Kaitannya dengan Asumsi Kenormalan pada Beberapa Uji Statistika. *Jurnal Matematika, Sains, dan teknologi*, 1 - 13.
16. Sari, T. N., Herlina, L., & Kurniawan, B. (2017). Analisa Line Balancing Dengan Metode Genetika Algoritma Pada Lintasan Produksi Di PT. X. *Jurnal Teknik Industri Untirta*.
17. Scholl, A., & Becker, C. (2006). State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*, 666-693.
18. Stephanus, D. (2019). *Peningkatan Produktivitas dan Penyeimbangan Lintasan Produksi di PT. Morelli Mitra Mandiri*. Jakarta: Universitas Katolik Indonesia Atma Jaya.
19. Tasan, S. O., & Semra, T. (2007). A review of the current applications of genetic algorithms A review of the current applications of genetic algorithms. *Springer Science*, 49-69.
20. Trisnajaya, E. V. (2020). *Penyeimbangan Lintasan dan Peningkatan Kapasitas Produksi pada PT. Fajar Inti Plasindo*. Jakarta: Universitas Katolik Indonesia Atma Jaya.
21. Virgy, R. (2006). *Analisis Keseimbangan Lini Perakitan Refrigerator Twin Door Bagian Pre-Assy Cabinet B Menggunakan Algoritma Genetika (Studi Kasus: "PT. Topjaya Antariksa Electronics")*. Jakarta: Universitas Katolik Indonesia Atma Jaya.
22. Wahyuniardi, R., Zalynda, P. M., & Pamungkas, S. (2012). Perbaikan Keseimbangan Lintasan Perakitan Dengan Algoritma Genetika (Studi Kasus di CV. Jaya Pratama Bandung). *Seminar Nasional Mesin Dan Industri (SNMI7) 2012*, 356 - 365.