

# Optimalisasi dan Penyelesaian *Job Shop Scheduling Problem* dengan Metode Algoritma Genetik (Studi Kasus *Injection Mould Base Tumbler Upper Cap* di PT. Berlina Tbk Pandaan, Pasuruan)

Niko Aditya<sup>1\*</sup>, Rina Sandora<sup>2</sup>, Endang Pudji Purwanti<sup>3</sup>

Program studi Teknik Desain dan Manufaktur, Jurusan Teknik Permesinan Kapal, Politeknik Perkapalan Negeri Surabaya, Indonesia<sup>1\*</sup>

Program studi Teknik Perpipaan, Jurusan Teknik Permesinan Kapal, Politeknik Perkapalan Negeri Surabaya, Indonesia<sup>2</sup>

Program studi Teknik Perancangan dan Konstruksi Kapal, Jurusan Teknik Bangunan Kapal, Politeknik Perkapalan Negeri Surabaya, Indonesia<sup>3</sup>

Email: [nikoaditya@student.ppns.ac.id](mailto:nikoaditya@student.ppns.ac.id)<sup>1\*</sup>; [rinasandora@ppns.ac.id](mailto:rinasandora@ppns.ac.id)<sup>2\*</sup>; [endangpudjip@ppns.ac.id](mailto:endangpudjip@ppns.ac.id)<sup>3\*</sup>;

---

**Abstract** – PT. Berlina Tbk Pandaan is a plastic packaging and mold maker company with a make to order production system especially at the Mold Shop Department with job shop production flow. From the scheduling system, production delays often occur. Injection Mold Base for Tumbler Upper Cap products was chosen as the object of research with a delay of 7 weeks. The purpose of this study is to determine the sequence of job operations flow by minimizing tardiness and makespan using the Genetic Algorithm method so as to produce optimum scheduling. The analysis of this study involved 37 jobs with 24 different machines. Calculations on the Genetic Algorithm method using Microsoft Excel and MATLAB R2012b media. From the calculation of the Genetic Algorithm method produces tardiness value of 0 hours, makespan 134.125 hours, and the highest fitness value of 2.3126. While scheduling the company produces a value of tardiness of 589.8750 hours and makespan of 195.125 hours. Then this method is able to minimize 100% tardiness and 31% makespan of company scheduling. While the operating flow of the resulting job is: 25-14-11-26-6-18-23-36-31-5-4-1-13-33-8-35-32-12-34-7-3-28-17-15-29-2-22-27-9-30-19-21-37-16-24-20-10.

**Keywords:** Genetic Algorithm, job shop, machine scheduling, makespan, tardiness.

---

## Nomenclature

<b>P</b>	= Populasi
<b>f</b>	= nilai kesesuaian
<b>q</b>	= nilai probabilitas kumulatif
<b>P</b>	= nilai probabilitas bertahan hidup
<b>F</b>	= nilai total keseluruhan dari kesesuaian
<b>P<sub>crossover</sub></b>	= persentase crossover
<b>P<sub>mutasi</sub></b>	= persentase mutasi

## 1. PENDAHULUAN

PT. Berlina Tbk Pandaan merupakan perusahaan yang bergerak di bidang industri manufaktur *moulding* baik untuk *injection moulding* maupun *blow moulding* yang diproduksi oleh *Mould Shop Departement* berdasarkan *job* yang datang (*make to order*) dengan jenis aliran produksi *job shop*. Namun dari sistem tersebut seringkali terjadi keterlambatan. Faktor keterlambatan tersebut bisa disebabkan sistem penjadwalan yang tidak sesuai dengan rencana maupun sumber daya lainnya. Sedangkan untuk waktu penyelesaian *mould*, setiap komponen memiliki waktu

penyelesaian yang berbeda. Perusahaan harus memiliki perencanaan dan pengendalian yang baik mengenai penjadwalan. Dari latar belakang tersebut diperoleh rumusan masalah yaitu bagaimana optimalisasi dari sistem penjadwalan perusahaan serta urutan penjadwalan permesinan pada *Injection Mould Base* produk *Tumbler Upper Cap* dengan menggunakan metode Algoritma Genetik. Hal tersebut diupayakan untuk mengoptimalkan proses produksi dan mengetahui urutan penjadwalan *job shop* secara optimal pada proses manufaktur *Injection Mould Base* produk *Tumbler Upper Cap* menggunakan metode Algoritma Genetik.

Namun ada beberapa batasan masalah pada penelitian ini antara lain nilai *tardiness* dan *makespan* dihitung hanya pada *injection moulding*, analisa yang dilakukan hanya pada pembuatan *mould* yang sudah siap *assembly* dan *trial and error* pada proses *manufacturing mould shop*, data yang digunakan periode tahun 2019, tidak ada perubahan jumlah variabel proses serta hanya mengacu berdasarkan *Time Ticket*.

## 2. METODOLOGI

### 2.1 Job Shop Scheduling Problem

*Job shop scheduling problem* bisa dibilang salah satu masalah penjadwalan yang memiliki kendala urutan pemrosesan tugas. Di dalam *Job shop scheduling problem*, diberikan suatu masalah dengan  $n$  pekerjaan dan  $m$  mesin. Setiap pekerjaan harus diproses tepat satu kali pada setiap mesin sesuai dengan urutannya masing-masing. Karena tugas adalah kombinasi dari pekerjaan dan mesin, maka dua tugas dari suatu pekerjaan tidak boleh dikerjakan pada waktu yang bersamaan, dan setiap mesin hanya dapat memproses paling banyak satu pekerjaan pada satu waktu. Tujuan dari *job shop scheduling problem* adalah mencari suatu jadwal yang meminimumkan waktu yang dibutuhkan untuk menyelesaikan semua pekerjaan (*makespan*) [1].

### 2.2 Metode First Come First Serve (FCFS)

Metode *First Come First Serve* (FCFS) merupakan aturan *sequencing* yang umum digunakan dengan memproses *job* sesuai dengan *job* yang datang terlebih dahulu [2]. Secara umum FCFS merupakan metode penyelesaian waktu produksi dengan memprioritaskan urutan kedatangan *job*, karena pada metode ini dijelaskan bahwa *job* yang datang akan segera diproses dahulu dengan memperhatikan tanggal waktu proses awal. Metode FCFS juga termasuk dalam golongan algoritma. Algoritma ini merupakan algoritma dalam penjadwalan yang paling sederhana dan mudah dipahami, maka setiap *job* yang berada pada status *ready* atau siap dikerjakan akan dimasukkan ke dalam antrian dengan prinsip *first in first out*, sesuai dengan waktu kedatangan *job* tersebut. Kekurangan dari penerapan prioritas FCFS sebagai algoritma dalam penjadwalan adalah *waiting time* rata-rata bagi *job* cukup lama serta terjadinya *convoy effect*, yaitu proses-proses menunggu yang lama untuk 1 proses besar yang sedang dieksekusi [5].

### 2.3 Algoritma Genetik

Algoritma sendiri dapat melakukan proses peniruan terhadap mekanisme evolusi alam. Pencarian populasi dilakukan dengan memunculkan kemungkinan-kemungkinan solusi untuk mendapatkan solusi terbaik atas masalah yang terjadi [3]. Penggunaan algoritma ini adalah untuk mencari ukuran performansi yang lebih baik dengan melakukan pencarian titik - titik baru, seperti halnya pencarian generasi baru yang lebih baik sesuai dengan teori genetik. Dari suatu populasi awal, individu - individu melakukan beberapa operasi genetik tertentu. Kemudian daya tahan setiap individu tersebut dinilai untuk menentukan apakah individu - individu tersebut

dapat dipilih pada generasi berikutnya atau tidak [2].

### 2.4 Langkah - Langkah Optimasi dengan Metode Algoritma Genetik

#### 1. Tahap Kodifikasi

Tahap kodifikasi dilakukan dengan menggunakan permutasi *encoding* tahap ini dilakukan untuk menentukan permasalahan pengurutan data. Kemudian dilakukan tahap inialisasi populasi awal dengan membentuk kromosom, pada kromosom tersebut dibentuk gen-gen secara acak. Kromosom merupakan kumpulan dari beberapa gen yang merepresentasikan operasi pada setiap mesin dari keseluruhan *job*. Setelah dilakukan perhitungan dengan inialisasi awal populasi maka tahap selanjutnya dilakukan perhitungan nilai *fitness* pada tiap kromosom sesuai dengan fungsi tujuan awal.

#### 2. Tahap Dekodifikasi

Proses dekodifikasi merupakan proses mengubah kromosom menjadi solusi, sehingga kromosom dapat dievaluasi nilai kesesuaiannya, pada tahap penelitian ini dilakukan untuk mentransformasikan kromosom ke dalam bentuk *gant chart* [4].

#### 3. Tahap Fungsi Kesesuaian (*fitness function*)

Pada tahap fungsi kesesuaian atau *function fitness* merupakan fungsi tujuan dari penelitian Algoritma Genetika yang mengindikasikan seberapa baik suatu solusi dan seberapa dekat individu dengan solusi optimal. Sesuai dengan permasalahan yang diteliti pada tugas akhir ini yaitu meminimasi *tardiness* dan meminimasi *makespan*. Setelah dilakukan tahap proses dekodifikasi kemudian dapat dilakukan evaluasi menggunakan fungsi kesesuaian seperti berikut :

$$f_p = \left( \frac{\text{total tardiness awal} - \text{total tardiness kromosom ke - n}}{\text{total tardiness awal}} \right) + \left( \frac{\text{total makespan awal} - \text{total makespan kromosom ke - n}}{\text{total makespan awal}} \right) \quad (3.1)$$

Catatan : Formula fungsi kesesuaian dapat diubah sesuai dengan tujuan penelitian yang dicari.

#### 4. Tahap Proses Seleksi

Pada tahap seleksi digunakan untuk memilih calon induk baru dari proses *crossover* dan mutasi. Jika diketahui ukuran populasi adalah  $P$ , maka jumlah kromosom yang akan diseleksi adalah ukuran populasi dikurangi dengan satu kromosom (*elitisme*) atau  $(P-1)$ . Proses seleksi berisi Daftar Calon Induk (DCI) yang berisi  $(P - 1)$  kromosom *crossover* dan mutasi [4]. Tahap seleksi ini menggunakan metode *roulette wheel*, metode *roulette wheel* merupakan metode yang digunakan untuk menghitung nilai probabilitas bertahan hidup [4]. Jika  $P_p$  merupakan nilai probabilitas bertahan hidup individu  $p$ ,  $f_p$  merupakan nilai kesesuaian individu  $p$ , dan  $F$  merupakan nilai total keseluruhan dari kesesuaian semua individu dalam satu

populasi.  $P_p = \frac{f_p}{F}$  kemudian untuk nilai probabilitas kumulatif ( $q_p$ ) dapat dihitung berdasarkan  $P_p$  setiap individu. Individu  $p$  terseleksi jika nilai variabel random berdistribusi uniform (0,1) berada diantara  $q_{p-1}$  dan  $q_p$ . Pada kasus minimasi kemungkinan besar individu dengan nilai kesesuaian kecil memiliki peluang lebih besar untuk dipilih. Hal ini berarti individu dengan nilai kesesuaian kecil memiliki peluang untuk diganti dengan menggunakan *crossover* dan mutasi.

#### 5. Tahap Elitis

Proses elitis atau elitisme merupakan proses pemilihan nilai kesesuaian terbaik disuatu populasi. Individu yang memiliki nilai kesesuaian terbaik akan masuk dan dikumpulkan pada Daftar Individu Elit (DIE). Kemudian satu individu dalam DIE akan diacak secara random sehingga dapat dipilih dan dimasukkan ke dalam Daftar Induk Baru (DIB). Tujuan dari dilakukan pemilihan individu elit secara acak untuk DIB adalah mempertahankan individu terbaik terhadap populasi / generasi berikutnya dan memiliki setidaknya sama dengan populasi / generasi sebelumnya.

#### 6. Tahap Proses Crossover

Proses *crossover* merupakan proses pindah silang antara dua induk untuk menghasilkan dua individu anakan yang saling membagi karakteristik induk mereka [5]. Hal kritis pada tahap *crossover* adalah penentuan operator *crossover*. Pemilihan *crossover* yang salah akan berakibat pada turunnya kualitas individu anakan. Selain itu operator *crossover* harus menghasilkan operator yang layak. Individu yang mengalami *crossover* diambil dari Daftar Calon Induk (DCI). Kemudian dua induk diambil secara acak dengan syarat induk 1 dan induk kedua tidak sama. Jika  $P_c$  adalah persentase *crossover* dalam satu populasi,  $P_{crossover}$  adalah jumlah kromosom yang mengalami *crossover* dan  $P$  adalah ukuran populasi, sehingga dapat diketahui sebagai berikut :

$$P_{crossover} = \text{Integer} (P - 1) * P_c \quad (3.2)$$

Apabila  $P_{crossover} \text{ Mod } 2 = 0$ , maka  $P_{crossover}$  dapat langsung digunakan. Sedangkan, apabila  $P_{crossover} \text{ Mod } = 1$ , maka  $P_{crossover}$  harus dikurangi satu sehingga menghasilkan bilangan genap.

Operator yang digunakan pada penelitian ini adalah *Partially Matched Crossover* (PMX). Operator ini memiliki prinsip yaitu menukarkan nilai gen – gen yang berada diantara dua titik *crossover* yang dipilih secara acak. Operator (PMX) merupakan modifikasi dari operator dua titik yang dilanjutkan dengan prosedur tambahan. Karakteristik representasi kromosom yang dapat menggunakan *crossover* PMX adalah tidak

dibolehkan dalam satu *string* atau satu kromosom memiliki nilai gen yang sama.

#### 7. Tahap Proses Mutasi

Proses mutasi memiliki peran untuk membuat perubahan kecil secara acak terhadap solusi. Perubahan ini akan menambahkan beberapa karakteristik baru pada populasi secara bertahap. Selain itu proses mutasi dilakukan guna suatu solusi yang dihasilkan tidak terjebak dalam lokal optimal. Individu yang akan mengalami mutasi diambil dari Daftar Calon Induk (DIC). Kemudian individu yang digunakan merupakan individu yang tidak terpilih untuk proses *crossover*. Jika  $P_{crossover}$  adalah jumlah kromosom yang mengalami *crossover*  $P$  adalah ukuran populasi,  $P_{mutasi}$  adalah jumlah individu yang mengalami mutasi. Sehingga rumus yang digunakan adalah sebagai berikut :

$$P_{mutasi} = P - 1 - P_{crossover} \quad (3.3)$$

#### 8. Tahap Kondisi Terminasi

Proses terminasi merupakan populasi dilakukan hingga mencapai kondisi terminasi [5]. Kondisi penelitian tercapai apabila mengalami *Steady State* telah terpenuhi. *Steady State* berarti memiliki pengulangan nilai *fitness* terbaik sebanyak beberapa kali. Pada penelitian ini dilakukan penetapan pengulangan untuk mencapai nilai *steady* sebanyak tiga puluh lima kali.

#### 9. Tahap Penyelesaian

Penyelesaian dilakukan dengan menggunakan *software Microsoft Excel* dan *MATLAB R2012b*. Tahap ini dilakukan dengan cara membuat *coding* berdasarkan tahapan yang telah dijelaskan di atas. Tahap berikutnya adalah dilakukan verifikasi kesesuaian data dengan model yang diharapkan, berdasarkan fungsi yang ditetapkan.

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Pengumpulan Data

Peneliti telah menyediakan data – data yang telah dirangkum dalam sebuah tabel yang menjelaskan informasi mengenai jenis – jenis *part* yang dikerjakan, jumlah *job* yang dikerjakan, inisialisasi tiap – tiap *job*, tanggal pemesanan, tanggal pengiriman, tanggal mulai hingga akhir pengerjaan tiap – tiap *job*, dan *due date*, sehingga dapat mengetahui urutan *job* yang dikerjakan di awal hingga akhir.

Tabel 3.1 *Job* dan *due date Tumbler Upper Cap*

No	Part	Job ke-	Tanggal Pesa-an	Start Proses-sing	Finish Proses-sing	Tanggal Kirim	Due Date (Hours)
1	KK-BAUT PENAHAN RUNNER PIN	J1	5/16/2019	6/20/2019	7/11/2019	7/13/2019	148
		J2	5/16/2019	6/20/2019	7/11/2019	7/13/2019	148
2	KKP-FEMALE BLOCK	J3	5/16/2019	6/17/2019	8/28/2019	7/18/2019	164
		J4	5/16/2019	6/17/2019	8/28/2019	7/18/2019	164

Tabel 3.1 Job dan due date Tumbler Upper Cap (Lanjutan)

No	Part	Job ke-	Tanggal Pesan	Start Proses	Finish Proses	Tanggal Kirim	Due Date (Hours)
3	KKP-INNER MALE	J5	5/16/2019	6/25/2019	8/26/2019	7/25/2019	184
		J6	5/16/2019	6/25/2019	8/26/2019	7/25/2019	184
4	KKP-INSERT FEMALE BLOCK	J7	5/16/2019	5/21/2019	8/29/2019	7/25/2019	184
		J8	5/16/2019	5/21/2019	8/29/2019	7/25/2019	184
5	KK-PIPA PENGIN	J9	5/16/2019	5/16/2019	7/15/2019	7/13/2019	74
		J10	5/16/2019	5/16/2019	7/15/2019	7/13/2019	74
		J11	5/16/2019	5/16/2019	7/15/2019	7/13/2019	74
		J12	5/16/2019	5/16/2019	7/15/2019	7/13/2019	74
6	KKP-MALE BLOCK	J13	5/16/2019	6/20/2019	8/28/2019	7/25/2019	184
		J14	5/16/2019	6/20/2019	8/28/2019	7/25/2019	184
7	KK-RUNNER BUSHING	J15	5/16/2019	5/16/2019	7/23/2019	6/28/2019	108
		J16	5/16/2019	5/16/2019	7/23/2019	6/28/2019	108
8	KP-EJECTOR PIN	J17	5/16/2019	5/16/2019	8/29/2019	7/25/2019	184
		J18	5/16/2019	5/16/2019	8/29/2019	7/25/2019	184
9	KP-KLEMATROL	J19	5/16/2019	6/18/2019	6/25/2019	6/24/2019	184
10	KP-PULLER BOLT	J20	5/16/2019	7/9/2019	7/15/2019	7/13/2019	74
		J21	5/16/2019	7/9/2019	7/15/2019	7/13/2019	74
		J22	5/16/2019	7/9/2019	7/15/2019	7/13/2019	74
		J23	5/16/2019	7/9/2019	7/15/2019	7/13/2019	74
11	KP-SPRUE BUSHING	J24	5/16/2019	5/16/2019	7/18/2019	7/18/2019	328
12	KP-SPRUE INSERT	J25	5/16/2019	5/16/2019	7/18/2019	7/18/2019	328
13	KP-SUPPORT PILLAR	J26	5/16/2019	5/16/2019	6/21/2019	6/21/2019	88
		J27	5/16/2019	5/16/2019	6/21/2019	6/21/2019	88
14	KP-TOP LOCATING RING	J28	5/16/2019	7/5/2019	7/12/2019	7/13/2019	296
15	MB-BOTTOM PLATE	J29	5/16/2019	6/13/2019	6/17/2019	6/21/2019	176
16	MB-EJECTOR PLATE	J30	5/16/2019	6/14/2019	7/10/2019	6/28/2019	216
17	MB-EJECTOR RETAINER PLATE	J31	5/16/2019	6/14/2019	7/12/2019	7/18/2019	328
18	MB-FEMALE PLATE	J32	5/16/2019	6/13/2019	7/15/2019	7/13/2019	296
19	MB-MALE PLATE	J33	5/16/2019	6/14/2019	7/4/2019	7/13/2019	296
20	MB-SPACE BLOCK	J34	5/16/2019	7/9/2019	7/10/2019	7/13/2019	148
		J35	5/16/2019	7/9/2019	7/10/2019	7/13/2019	148
21	MB-STRIP PER RUNNER PLATE	J36	5/16/2019	6/19/2019	7/22/2019	7/13/2019	296
22	MB-TOP PLATE	J37	5/16/2019	6/19/2019	7/22/2019	7/18/2019	328

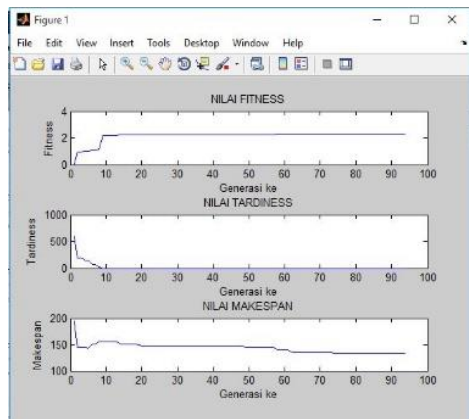
### 3.2 Analisa Penjadwalan dengan Urutan Job oleh Algoritma Genetik

Pada tahap analisa ini menggunakan beberapa parameter pengujian untuk mengetahui hasil optimum dari penjadwalan mesin. Seperti yang digunakan pada pengujian validasi program, parameter yang digunakan antara lain ukuran populasi, persentase *crossover*, dan *steady state*. Ukuran populasi yang digunakan yaitu 35 dan 70. Sedangkan persentase *crossover* menggunakan 0,5 dan 1. Sementara untuk *steady state* menggunakan parameter yang sama sebesar 20 perulangan. Pada pengujian ini menggunakan 37 job yang berjalan di 24 operasi mesin yang berbeda sehingga didapat pengurutan job yang menghasilkan nilai *tardiness* dan *makespan* minimum dan nilai *fitness* yang optimum. Variasi dan kombinasi dari parameter – parameter tersebut dimaksudkan untuk menguji ketepatan dari eksperimen agar memperoleh ketelitian yang lebih akurat. Hasil dari kombinasi parameter – parameter tersebut ditampilkan pada Tabel 3.2.

Tabel 3.2 Hasil pengujian dari setiap parameter

No	Parameter			Hasil Optimum			
	nPop	Pc	SS	Jumlah Generasi	Tardiness	Makespan	Fitness
1	35	0.2	20	83	0	137.5	2.295
2	35	0.4	20	60	0	137.25	2.297
3	35	0.6	20	75	0	139.75	2.284
4	35	0.8	20	72	0	141	2.277
5	35	1	20	65	0	138.5	2.290
6	70	0.2	20	46	0	139.75	2.284
7	70	0.4	20	93	0	134.125	2.313
8	70	0.6	20	97	0	142	2.272
9	70	0.8	20	67	0	140.25	2.281
10	70	1	20	85	0	134.5	2.311

Berdasarkan pada Tabel 3.2 didapatkan hasil yang paling optimum dari serangkaian pengujian kombinasi yaitu pada pengujian nomor 7. Dengan ukuran populasi 70, persentase *crossover* 0,4, jumlah *steady state* 20 menghasilkan jumlah generasi sebesar 93, nilai *tardiness* 0 jam, nilai *makespan* 134,125 jam, dan nilai *fitness* tertinggi sebesar 2,3126. Sehingga menghasilkan aliran grafik nilai *fitness*, *makespan*, dan *tardiness* terbaik yang disajikan pada Gambar 3.1.



Gambar 3.1 Grafik nilai fitness, makespan, dan tardiness terbaik

#### 4. KESIMPULAN

Penjadwalan Algoritma Genetik yang diterapkan untuk mengerjakan *Injection Mould Base Tumbler Upper Cap* menghasilkan jumlah generasi sebesar 93, nilai *tardiness* 0 jam, nilai *makespan* 134,125 jam, dan nilai *fitness* tertinggi sebesar 2,3126 dengan aliran urutan operasi *job* beserta detail *job* sebagai berikut :

*Job* 25 (KP *Sprue Insert Q1*) - *Job* 14 (KKP *Male Block Q1*) - *Job* 11 (KK *Pipa Pendingin Q1*) - *Job* 26 (KP *Support Pillar Q1*) - *Job* 6 (KKP *Inner Male Q1*) - *Job* 18 (KP *Ejector Pin Q1*) - *Job* 23 (KP *Puller Bolt Q1*) - *Job* 36 (MB *Stripper Runner Plate Q1*) - *Job* 31 (MB *Ejector Retainer Plate Q1*) - *Job* 5 (KKP *Inner Male Q2*) - *Job* 4 (KKP *Female Block Q1*) - *Job* 1 (KK *Baut Penahan Runner Pin Q1*) - *Job* 13 (KKP *Male Block Q2*) - *Job* 33 (MB *Male Plate Q1*) - *Job* 8 (KKP *Insert Female Block Q1*) - *Job* 35 (MB *Spacer Block Q1*) - *Job* 32 (MB *Female Plate Q1*) - *Job* 12 (KK *Pipa Pendingin Q2*) - *Job* 34 (MB *Spacer Block Q2*) - *Job* 7 (KKP *Insert Female Block Q2*) - *Job* 3 (KKP *Female Block Q2*) - *Job* 28 (KP *Top Locating Ring Q1*) - *Job* 17 (KP *Ejector Pin Q2*) - *Job* 15 (KK *Runner Bushing Q1*) - *Job* 29 (MB *Bottom Plate Q1*) - *Job* 2 (KK *Baut Penahan Runner Pin Q2*) - *Job* 22 (KP *Puller Bolt Q2*) - *Job* 27 (KP *Support Pillar Q2*) - *Job* 9 (KK *Pipa Pendingin Q3*) - *Job* 30 (MB *Ejector Plate Q1*) - *Job* 19 (KP *Klem Katrol Q1*) - *Job* 21 (KP *Puller Bolt Q3*) - *Job* 37 (MB *Top Plate Q1*) - *Job* 16 (KK *Runner Bushing Q2*) - *Job* 24 (KP *Sprue Bushing Q1*) - *Job* 20 (KP *Puller Bolt Q4*) - *Job* 10 (KK *Pipa Pendingin Q4*).

Sedangkan penjadwalan perusahaan dengan menerapkan metode FCFS (*First Come First Serve*) menghasilkan nilai *tardiness* sebesar 589,8750 jam dan nilai *makespan* sebesar 195,125 jam. Metode Algoritma Genetik mampu meminimasi 100% keterlambatan positif (*tardiness*) dan meminimasi *makespan* sekitar 31% dari hasil metode FCFS perusahaan

#### 5. DAFTAR PUSTAKA

- [1] Baker, K. R. (2009). Principles of Sequencing and Scheduling. *Jhon Wiley & Sons, Inc.*
- [2] Ginting, R. (2009). Penjadwalan Mesin. *Graha Ilmu*, 1-271.
- [3] Kushartantya, S. A. (2010). Penyelesaian Masalah Job Shop Menggunakan Algoritma Genetika. *Jurnal Masyarakat Informatika, Volume 1, Nomor 1, ISSN 2086 – 4930*, 31-42.
- [4] Paksi, A. B. (2014). Penjadwalan Fleksibel Job-Shop dengan Dual-Resource Constraints untuk Meminimasi Tardiness Menggunakan Algoritma Genetika. *Institut Teknologi Bandung*.
- [5] Yulianto, A. (2018). Penjadwalan Pembuatan Inboard Outer Fixed Leading Edge di PT. Dirgantara Indonesia (Persero) Menggunakan Algoritma Genetika. *Tugas*