

**PENERAPAN MODEL OPTIMASI *LINE BALANCING* DAN *GENETIC ALGORITHM*
(STUDI KASUS: PT. KARYA MEKAR DEWATAMALI)**

Andy Lisanto, Dian Retno Sari Dewi*, Dini Endah Setyo Rahaju
Jurusan Teknik Industri, Fakultas Teknik, Universitas Katolik Widya Mandala, Surabaya, Indonesia

ABSTRAK

Keseimbangan dalam suatu lintasan perakitan (Line Balancing) sangatlah penting bagi industri perakitan. Sebuah produk yang memiliki N elemen kerja memerlukan suatu keseimbangan lintasan karena jika keseimbangan tidak tercapai, maka keterlambatan akan terjadi akibat dari kemacetan. Dua permasalahan utama dalam lintasan perakitan adalah menyeimbangkan stasiun kerja sekaligus mempertahankan lintasan perakitan agar terus menerus berproduksi.

PT. Karya Mekar Dewatamali (PT. KMD) adalah perusahaan yang bergerak di bidang industri perakitan sepatu. Pada perusahaan ini akan diterapkan model Algoritma Genetik yang kemudian akan divalidasi oleh model Subiantoro. Penelitian ini bertujuan untuk meminimasi jumlah stasiun kerja dan mengalokasikan elemen kerja yang ada agar menjadi optimal sehingga biaya pengadaan stasiun kerja, biaya waktu siklus, biaya operator, dan biaya idle operator dapat diminimalkan. Adapun penelitian dimulai dari tahap pengamatan awal, studi literatur, pengumpulan data, input data (model optimasi Subiantoro dan Algoritma Genetik), verifikasi dan validasi model, analisa dan pembahasan, serta kesimpulan. Hasil dari penerapan model ini adalah Algoritma genetik hanya menyimpang 0,0001% dari hasil model optimasi dan terbentuk 5 stasiun kerja dengan total biaya sebesar Rp 75.017.690,98.

Kata Kunci: *Line Balancing*, Subiantoro, Algoritma Genetik.

I. Pendahuluan

Keseimbangan dalam suatu lintasan perakitan sangatlah penting bagi industri perakitan. Sebuah produk yang memiliki N elemen kerja memerlukan suatu keseimbangan lintasan karena jika keseimbangan tidak tercapai, maka keterlambatan akan terjadi akibat dari kemacetan. Dua permasalahan utama dalam lintasan perakitan adalah menyeimbangkan stasiun kerja sekaligus mempertahankan lintasan perakitan agar terus menerus berproduksi^[1].

Dalam meneliti keseimbangan lintasan, terdapat beberapa faktor yang mempengaruhinya, diantaranya adalah volume produksi, jenis urutan operasi pada lintasan dalam proses produksi, dan waktu penyelesaian masing-masing elemen kerja pada masing-masing produksi. Manfaat dari menyeimbangkan lintasan produksi adalah menyeimbangkan beban kerja pada tiap stasiun kerja, mengoptimalkan jumlah pekerja (operator), meminimasi jumlah stasiun kerja, meminimasi waktu penyelesaian produk, dan meminimasi biaya pengadaan stasiun kerja maupun operator.

Algoritma genetik adalah sebuah metode yang didasarkan pada prinsip-prinsip genetika dan seleksi alam^[2]. Untuk mengatasi kelemahan waktu komputasi model optimasi *line balancing* yang sangat lama, maka digunakan model Algoritma genetik yang waktu komputasinya lebih cepat.

PT. Karya Mekar Dewatamali (PT. KMD) adalah perusahaan yang bergerak di bidang

industri perakitan sepatu. Merek sepatu yang di produksi adalah Speed. Daerah pemasaran industri sepatu ini hanya melingkupi dalam negeri (lokal). Pada perusahaan ini diterapkan model Algoritma Genetik yang bertujuan untuk meminimasi jumlah stasiun kerja dan mengoptimalkan pengalokasian elemen kerja pada stasiun kerja sehingga biaya pengadaan stasiun kerja, biaya waktu siklus, biaya operator, dan biaya *idle* operator dapat diminimalisasi.

Model optimasi *line balancing* adalah NP *hard problem* sehingga waktu komputasi akan naik secara eksponensial seiring dengan penambahan input data. Untuk menanggulangi permasalahan kecepatan komputasi tersebut dikembangkan model pencarian *heuristic*. Metode yang dipilih adalah *Genetic Algorithm*, dimana pada penelitian sebelumnya pernah digunakan untuk menyelesaikan permasalahan optimasi *line balancing*^[3]. Penelitian sebelumnya bertujuan untuk membandingkan performansi skala dengan berbagai parameter algoritma genetika dan dihasilkan bahwa parameter yang baik untuk *crossover* adalah 0,7 dan mutasi adalah 0,035. Pada penelitian yang lain, algoritma genetik juga digunakan pada kasus lintasan perakitan dengan tujuan untuk membandingkan populasi awal yang dipilih secara acak dengan populasi awal yang dihasilkan dari individu *heuristic*^[4]. Hasilnya adalah populasi awal individu *heuristic* lebih baik digunakan untuk masalah yang sangat rumit

*corresponding author

E-mail : dianretnosd@yahoo.com (Dian Retno Sari Dewi)

dibandingkan dengan secara acak. Metode algoritma genetik dipilih karena kemudahan implementasi dan kemampuannya untuk menemukan solusi yang bagus (bisa diterima) secara cepat.

Penelitian dilakukan dengan menerapkan model Algoritma Genetik untuk mengatasi permasalahan waktu komputasi model Subiantoro sehingga waktu komputasi menjadi lebih cepat.

II. Tinjauan Pustaka

II.1. Keseimbangan Lintasan Perakitan

Keseimbangan lintasan perakitan adalah lintasan produksi dimana material berpindah secara kontinu dengan laju rata-rata yang sama melalui sejumlah stasiun kerja, tempat dilakukannya pekerjaan perakitan^[1].

Adapun beberapa tujuan dari keseimbangan lintasan adalah sebagai berikut:

1. Penyetaraan beban kerja antara orang (pekerja), sel, dan departemen.
2. Mengidentifikasi operasi yang *bottleneck*.
3. Menetapkan kecepatan dari jalur perakitan.
4. Menentukan jumlah dari stasiun kerja.
5. Menentukan biaya tenaga kerja perakitan.
6. Menetapkan persentase beban kerja dari masing-masing operator.
7. Mengurangi biaya produksi.

Adapun beberapa definisi yang dapat diterapkan dalam perakitan:

1. *Part* yang dirakit

Part yang melalui suatu rangkaian stasiun kerja dimana pekerjaan dilakukan sampai *part* produksi selesai pada stasiun kerja terakhir.

2. Elemen kerja (i)

Sebagian dari total pekerjaan dalam pada proses perakitan. N adalah jumlah keseluruhan elemen kerja yang harus diselesaikan dalam perakitan, i adalah elemen kerja dalam proses. $1 \leq i \leq N$.

3. Stasiun kerja (S)

Suatu lokasi pada lintas perakitan dimana elemen kerja dilakukan. Nilai minimum jumlah stasiun kerja, s, lebih besar atau sama dengan 1.

4. Waktu siklus (C)

Waktu maksimum yang diperbolehkan untuk menyelesaikan pekerjaan pada setiap stasiun kerja. Nilai minimal dari waktu siklus harus lebih besar atau sama dengan waktu terlama stasiun kerja. Waktu siklus biasanya dicari dengan efisiensi lintasan dibagi dengan *production rate*.

5. Waktu operasi aktual elemen kerja

Waktu dimana *performance rating* operator ikut diukur dalam setiap pekerjaan yang dilakukan. Waktu operasi aktual ini didapatkan dari waktu *standard* dibagi dengan *performance rating* operator.

6. Waktu operasi aktual stasiun kerja (T_s)

Gabungan beberapa waktu aktual elemen kerja yang dialokasikan pada stasiun kerja.

7. Waktu menganggur dari stasiun

Pengurangan antara waktu siklus (C) dan waktu aktual stasiun (T_s).

8. *Precedence Diagram*

Suatu diagram yang menjelaskan urutan dari elemen kerja harus dilakukan. Beberapa pekerjaan tidak bisa dilakukan kecuali pekerjaan sebelumnya telah diselesaikan.

9. Waktu proses / *standard* (W_i)

Waktu proses perakitan elemen kerja.

II.2. Model Optimasi Subiantoro

Bertujuan untuk menggabungkan model optimasi Elsayed Boucher dan Lesmana Hartono dengan fungsi tujuan meminimasi jumlah stasiun kerja dan waktu siklus dengan mempertimbangkan performansi operator pada tiap elemen kerja sehingga dapat meminimasi biaya pengadaan stasiun kerja, biaya waktu siklus, biaya operator dan biaya *idle* operator [5].

Fungsi tujuan :

$$\text{Minimasi } Z = \sum_{s=1}^S f_s X_s + c \cdot (\max T_s) + F_r \cdot$$

$$\sum_{s=1}^S X_s \cdot \max T_s + F_r \cdot \left(\left(\sum_{s=1}^S X_s \max T_s - \sum_{s=1}^S T_s \right) / \max T_s \right) \quad (1)$$

dengan pembatas :

$$\sum_{s=1}^S X_{is} = 1 \quad \forall i = 1 \dots N \quad X_{is} \text{ berdasarkan } E_i \text{ dan } L_i \quad (2)$$

$$T_s = \sum_{i=1}^N \sum_{k=1}^K X_{is} Y_{ks} \left(\frac{W_i}{C_{ki}} \right) \quad \forall s = 1 \dots S \quad (3)$$

$$T_s \leq C \quad \forall s = 1 \dots S \quad (4)$$

$$X_{is} \leq \sum_{r=1}^s X_{jr} \quad \forall i = S_i, \forall j = P_i, \forall s = 1 \dots S \quad (5)$$

$$\sum_{i=1}^N X_{is} - \|W_s\| X_s \leq 0, \forall s = 1, 2, \dots, S \quad (6)$$

$$\sum_{k=1}^K Y_{ks} = 1 \quad \forall s = 1 \dots S \quad (7)$$

$$\sum_{s=1}^S Y_{ks} \leq 1 \quad \forall k = 1 \dots K \quad (8)$$

$$X_{is}, Y_{ks}, X_s \in \{0,1\} \quad (9)$$

Dimana

i, j : Indeks elemen kerja.

s : Indeks stasiun kerja.

K : Indeks operator.

f_s : Biaya tetap pada stasiun kerja s (Rp / Siklus).

X_s : Stasiun kerja s , dimana $s = 1 \dots S$. X_s merupakan jumlah stasiun kerja yang memungkinkan bernilai 1 jika stasiun dipergunakan dan 0 jika tidak.

c : Biaya waktu siklus (Rp / menit).

C : Waktu siklus.

F_r : Gaji operator reguler (Rp / menit).

N : Jumlah elemen kerja.

W_i : Waktu standar elemen kerja i .

P_i : Merupakan elemen kerja *predecessor*.

S_i : Merupakan elemen kerja *successor*.

C_{ki} : *Performance rating* operator k dalam menangani elemen kerja i .

X_{is} : Variabel alokasi elemen kerja i ke stasiun kerja s . 1 jika elemen kerja i dialokasikan ke stasiun kerja s , 0 jika tidak.

W_s : Beberapa elemen kerja dari total elemen kerja yang dapat ditempatkan pada stasiun kerja dengan memperhatikan pembatas pembatas elemen kerja yang fungsinya untuk meminimasi pengadaan stasiun kerja.

E_i : Stasiun kerja awal dimana elemen kerja i akan dialokasikan (indeks).

L_i : Stasiun kerja akhir dimana elemen kerja i akan dialokasikan (indeks).

Y_{ks} : Variabel alokasi operator k ke stasiun kerja s . 1 jika operator k dialokasikan ke stasiun kerja s , 0 jika tidak.

T_s : Waktu operasi aktual stasiun kerja s .

II.3. Algoritma Genetik

Algoritma genetik adalah sebuah metode yang didasarkan pada prinsip-prinsip genetika dan seleksi alam. Algoritma genetik adalah algoritma *iterative* yang menjaga sekumpulan solusi pada tiap iterasi. Pada awalnya, kumpulan solusi dihasilkan secara acak, dan pada tiap iterasi sebuah kumpulan solusi yang baru

dibentuk oleh operator genetik yang meniru prinsip evolusi dan hereditas. Tiap solusi dievaluasi dengan sebuah fungsi obyektif, dan proses ini diulang hingga beberapa bentuk konvergensi dicapai. Kumpulan solusi yang baru dapat terdiri dari solusi-solusi lama yang terpilih karena nilai fungsi obyektifnya yang tinggi, dan juga solusi-solusi baru yang terbentuk dengan mengkombinasikan solusi-solusi lain^[2].

II.4. Crossover atau Persilangan

Persilangan adalah operator genetik yang utama. Operator ini mengoperasikan dua kromosom dalam suatu waktu dan menghasilkan *offspring* dengan mengkombinasikan fitur-fitur kedua kromosom tersebut. Sebuah cara yang sederhana untuk melakukan persilangan adalah dengan memilih titik potong secara acak dan menghasilkan *offspring* dengan mengkombinasikan segmen dari sebuah induk di sisi kiri titik potong dengan segmen induk lain yang berada di sebelah kanan titik potong^[2].

II.5. Mutasi

Mutasi adalah operator *background* yang menghasilkan perubahan acak secara spontan pada beberapa kromosom. Sebuah cara yang sederhana untuk melakukan mutasi adalah dengan cara mengubah satu atau lebih gen^[2].

Sebuah algoritma genetik sederhana dapat dirangkum sebagai berikut:

1. Inisialisasi. Menghasilkan sebuah populasi *string* secara acak.
2. Mengevaluasi *string* berdasarkan fungsi *fitness*.
3. Menggunakan sekumpulan operator genetik untuk menghasilkan populasi *string* yang baru. Hasilkan sebuah populasi baru dari m *string* dengan:
 - (a) Jika $p_c > \text{rand}[0, 1]$, gunakan operator persilangan (*Crossover*) pada sepasang *string*.
 - (b) Jika $p_m > \text{rand}[0, 1]$, gunakan operator mutasi pada sepasang *string*.
4. Mengulangi langkah ke-2 dan ke-3 hingga sebuah solusi yang diinginkan.

III. Metode Penelitian

Adapun urutan rancangan atau prosedur penelitian dapat dilihat pada *flowchart* seperti pada gambar 1.

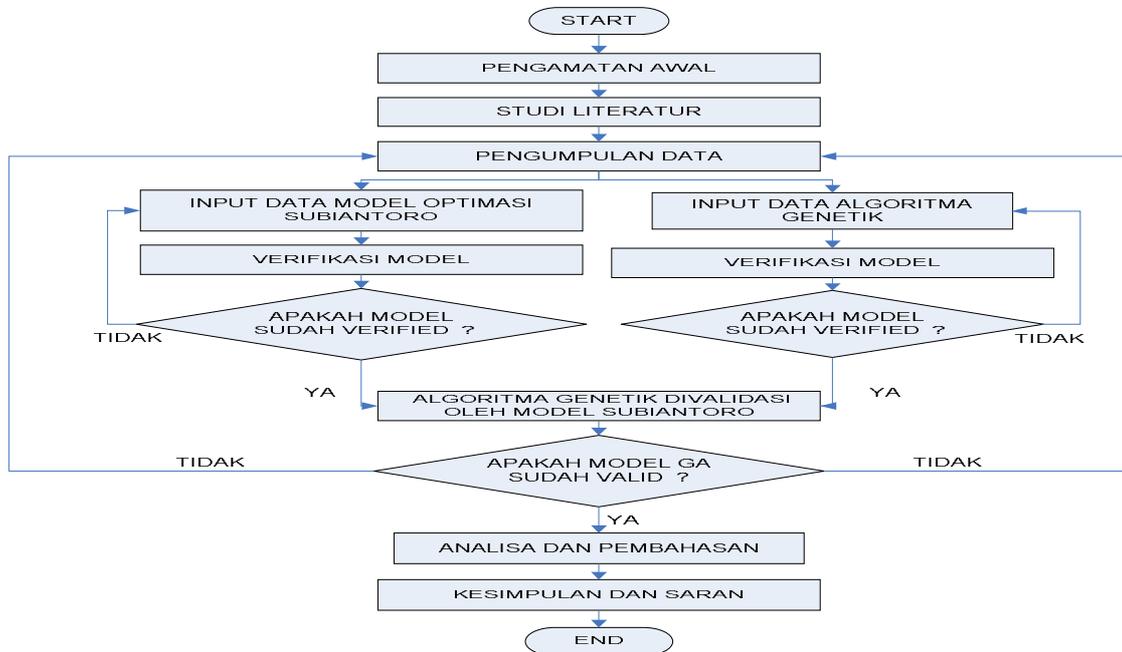
III.1. Pengamatan Awal

Pada tahap pengamatan awal dilakukan pencarian informasi mengenai bisnis proses dari perusahaan. Hal ini dilakukan dengan tujuan untuk mengetahui kondisi perusahaan. Pada tahap ini juga dilakukan pengamatan pada

jalannya aliran produksi dari bahan baku hingga menjadi produk jadi dan memastikan bahwa model yang akan digunakan dapat diterapkan pada sistem.

III.2. Studi Literatur

Pada tahap ini dipelajari beberapa literatur yang sesuai dimana bertujuan untuk mempelajari dan memahami teori-teori yang sesuai dengan penerapan model yang akan dibahas.



Gambar 1. Flowchart prosedur penelitian

III.3. Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data yang diperlukan untuk kemudian dimasukkan dalam model yang sudah terpilih. Data yang dibutuhkan adalah data jumlah elemen kerja, waktu standar elemen kerja, *performance rating* operator, waktu siklus, biaya waktu siklus, gaji operator reguler, jumlah stasiun kerja, biaya tetap pengadaan stasiun kerja, dan waktu operasi aktual stasiun kerja.

III.4. Input Data Model Optimasi Subiantoro

Pada tahap ini, semua data yang diperoleh dari pengumpulan data di masukkan ke dalam model Subiantoro.

III.5. Verifikasi Model Subiantoro

Setelah diperoleh hasil *output* model Subiantoro, maka dilakukan verifikasi model yaitu dengan cara memeriksa hasil tersebut dengan syarat – syarat model, antara lain:

1. Tidak melanggar pembatas-pembatas yang terdapat dalam model.
2. Variabel keputusan bersifat bilangan biner (0 atau 1).

Apabila syarat diatas sudah terpenuhi, maka model dapat dikatakan *verified*. Namun apabila sebaliknya, maka input data modeling harus di cek ulang.

III.6. Input Data Model Algoritma Genetik

Pada tahap ini, semua data yang diperoleh dari pengumpulan data di masukkan ke dalam model Algoritma Genetik dengan prosedur sebagai berikut:

a) Tahap I: Inisialisasi

Pada tahap ini, *Parents* awal yang digunakan digenerate dengan metode Helgeson and Birnie dan secara random.

b). Tahap II: Fitness

Tahap ini merupakan tahap untuk mengevaluasi *fitness* $f(x)$ dari tiap kromosom x dalam populasi.

c). Tahap III: New Population

Tahap ini merupakan tahap untuk menciptakan populasi baru dengan menggunakan metode *Crossover* dan Mutasi sampai terbentuk populasi baru.

d). Loop

Pada tahap ini, akan dilakukan pengulangan ke langkah *fitness*.

III.7. Verifikasi Model Algoritma Genetik

Setelah diperoleh hasil *output* model Algoritma Genetik, maka dilakukan verifikasi model yaitu dengan cara memeriksa hasil tersebut apakah sudah tidak melanggar batasan-batasan yang ada atau tidak.

Apabila tidak melanggar batasan yang ada, maka model dapat dikatakan *verified*. Namun apabila sebaliknya, maka input data modeling harus di cek ulang.

III.8. Validasi Model Algoritma Genetik

Pada tahap ini, model Algoritma Genetik yang sudah di verifikasi akan di validasi dengan model optimasi Subiantoro. Apabila model sudah valid, maka dapat dilanjutkan ke tahap **Analisis** dan pembahasan, namun bila sebaliknya, maka harus kembali ke tahap pengumpulan data.

III.9. Analisis dan Pembahasan

Pada tahap ini, *output software* LINGO dan hasil dari model Algoritma Genetik dianalisis untuk melihat apakah model dapat membantu pihak perusahaan dalam meminimasi biaya pengadaan stasiun kerja dan biaya pengadaan operator.

III.10. Kesimpulan dan Saran

Menarik kesimpulan dari keseluruhan hasil **Analisis** dan pembahasan dan memberi saran kepada perusahaan dari hasil yang ada.

IV. Hasil Penelitian Dan Pembahasan

IV. 1. Pengumpulan Data

Pengumpulan data dilakukan dengan wawancara dan observasi langsung terhadap pihak perusahaan. Data-data yang dibutuhkan antara lain:

- a) Data urutan proses perakitan produk sepatu story kokko warna hijau ukuran 42.
- b) Data Pengamatan Waktu Elemen Kerja.

Elemen-elemen kerja pada proses perakitan story kokko terdiri dari 42 elemen kerja dimana setiap waktu standar tiap elemen dapat dilihat pada tabel 1.

IV.2. Pengolahan Data

1. Uji Keceragaman Data

Dari hasil uji yang dilakukan, dapat dilihat bahwa tidak terdapat data yang diluar batas kontrol 3σ sehingga semua data elemen kerja dapat disimpulkan seragam.

Contoh perhitungan :

Mean elemen 1 = $(0,46 + 0,53 + 0,46 + 0,53 + 0,47 + 0,50 + 0,48 + 0,48 + 0,43 + 0,52)/10 = 0,486$

$$\text{Standar deviasi elemen 1} = \sqrt{\frac{\sum_{i=1}^n (Xi - \text{mean})^2}{n}}$$

$$= \sqrt{\frac{0,009762}{10}} = \sqrt{0,000976} = 0,31245 \text{ detik}$$

BKA (*Upper*) elemen 1 = $x + 3 \text{ SD} = 0,486 - 3 * 0,31245 = 0,578135$

BKB (*Lower*) elemen 1 = $x - 3 \text{ SD} = 0,486 - 3 * 0,31245 = 0,390665$

Keterangan :

$x = \text{Mean}$

BKA: Batas kontrol atas

BKB: Batas kontrol bawah

SD: Standar deviasi

Dari kedua batas kontrol diatas, dapat dilihat bahwa setiap sampel yang diambil tidak ada yang melebihi batas kontrol atas dan batas kontrol bawah sehingga data waktu untuk elemen 1 sudah seragam.

2. Uji Kecukupan Data

Dari hasil uji yang dilakukan, dapat dilihat bahwa tidak terdapat N' data yang melebihi N sehingga dapat disimpulkan bahwa pengambilan data pengamatan sudah cukup.

Contoh perhitungan:

$$N' \text{ sampel ke-1} = \left[\frac{\left(k / s \cdot \sqrt{N \sum_{i=1}^N Xi^2 - \left(\sum_{i=1}^N Xi \right)^2} \right)^2}{\sum_{i=1}^N Xi} \right]^2$$

$$= \left[\frac{\left(2 / 0,05 \cdot \sqrt{10 * 2,35620 - 23,46434} \right)^2}{4,844} \right]^2 = 6,66$$

($N' < N$. sehingga data untuk sampel ke-1 sudah cukup)

Keterangan:

$k =$ Tingkat kepercayaan (95% >> tabel normal

$z = 1,96$ >> dibulatkan 2)

$s =$ Tingkat ketelitian (5% atau 0,05)

$N =$ Jumlah sampel (10)

$Xi =$ Sampel ke-i

3. Waktu Normal dan Waktu Standar

Hasil perhitungan waktu normal tiap elemen kerja dapat dilihat pada Suplemen 2.

Contoh perhitungan:

Waktu Normal elemen ke-1 = Waktu

Observasi elemen ke-1 x *Performance Rating* operator ke-1

Waktu Normal elemen ke-1 = $0,486 \times 1,12 = 0,54$ detik

$$Ws = WnX \frac{100\%}{(100\% - \%allowance)}$$

$$Ws \text{ elemen ke-1} = 0,54 * \frac{100\%}{100\% - 20\%} = 0,68 \text{ detik}$$

IV.3. Input Model Subiantoro

Dalam model ini, akan dimasukkan beberapa *input* selain waktu standar tiap elemen kerja (asumsi 1 bulan = 25 hari, 1 hari = 8 jam =

28800 detik), tabel 2 merupakan data-data yang di inputkan ke dalam model.

Keterangan tabel 2:

E_i : Stasiun kerja awal dimana elemen kerja i akan dialokasikan (indeks).

L_i : Stasiun kerja akhir dimana elemen kerja i akan dialokasikan (indeks).

Jadi untuk elemen ke-1 ditentukan $E_i = 1$ dan $L_i = 2$ (penentuan ditetapkan sendiri oleh penulis berdasarkan perkiraan) karena elemen ke-1 ini dapat dialokasikan pertama kali di stasiun kerja 1 dan diperkirakan dapat dialokasikan terakhir kali di stasiun kerja 2. Elemen ke-36 ditentukan $E_i = 4$ dan $L_i = 5$ karena elemen ke-36 ini dapat dialokasikan pertama kali di stasiun kerja 4 dan diperkirakan dapat dialokasikan terakhir kali di stasiun kerja 5, demikian juga dengan elemen kerja yang lain.

$Production Rate = 1010 \text{ unit/hari} = 0,035069444 \text{ unit/detik}$

$Waktu siklus = (3600 \text{ detik} \times 8 \text{ jam}) / 1010 \text{ unit} = 28,51 \text{ detik}$

$Gaji operator reguler (F_r) = Rp 866.500/bulan = Rp 34.660/hari = Rp 4.332,5/jam = Rp 1,2035/detik$

$Biaya tetap pengadaan stasiun kerja (f_s) = Rp 15.000.000/stasiun$, dimana rinciannya adalah sebagai berikut:

$Biaya listrik = Rp 3.500.000,00$

$Biaya pengadaan alat keselamatan (Masker, sarung tangan) = Rp 1.500.000,00$

$Biaya perawatan mesin-mesin = Rp 8.000.000,00$

$Biaya pengadaan alat-alat bantu (Gunting, silet, paku, benang) = Rp 2.000.000,00$

$Harga Produk = Rp 90.000/unit$

$Laba = 20\% = Rp 18.000/unit$

$Biaya waktu siklus (c) = Production rate \times laba = 0,035069444 \text{ unit/detik} \times Rp 18.000 /unit = Rp 631/detik$

Tabel 3. Ringkasan Hasil Pengujian Model Subiantoro

Biaya Stasiun kerja (Rp/stasiun)	Biaya siklus (Rp / detik)	Gaji operator reguler (Rp /detik)	Jumlah Stasiun Kerja	Waktu Siklus (menit)	Total Biaya (Rp)
15.000.000	631	1,2035	5	28,51	75.017.600,00

IV.4. Input Algoritma Genetik

Dalam membuat algoritma genetik, diperlukan beberapa tahap untuk menghasilkan solusi yang optimal, yaitu:

- 1) *Initialitation*
- 2) *Fitness*
- 3) *New Population*

IV.4.1. Tahap I: Inisialisasi

Dalam model ini, akan digunakan metode Helgeson and Birnie untuk membentuk suatu *initial solution* awal. Tahap pertama untuk

menggunakan metode ini adalah dengan membuat *precedence diagram*, mengurutkan *ranking* berdasarkan *Positional Weight (PW)*, dan membagi elemen-elemen kerja berdasarkan *PW*. Untuk menentukan *parent* yang kedua, digunakan sistem pemilihan secara acak.

IV.4.2. Tahap II: Fitness

Pada tahap ini, *Objective Function* dari *parents* awal dihitung berdasarkan *parents* yang sudah terbentuk. Perhitungan total biaya dari *Initial solution* pertama adalah sebagai berikut:

$Objective Function = (15.000.000 \times 5) + (631 \times 28,50) + (1,2035 \times 5 \times 28,50) + [1,2035 \times ((5 \times 28,50) - (28,45 + 28,50 + 27,66 + 26,72 + 18,92) / 28,50)] = Rp. 75.018.321,00$

Perhitungan total biaya dari *Initial solution* kedua adalah sebagai berikut:

$Objective Function = (15.000.000 \times 5) + (631 \times 28,45) + (1,2035 \times 5 \times 28,45) + [1,2035 \times ((5 \times 28,45) - (21,56 + 25,05 + 26,22 + 28,45 + 20,98) / 28,45)] = Rp. 75.018.289,17$

IV.4.3. Tahap III: New Population

Dalam tahap ini, digunakan 2 metode untuk menghasilkan populasi baru yakni *Crossover* dan kemudian dilanjutkan dengan *Mutasi*. Berikut merupakan perhitungan manual dalam penggunaan *crossover* dan *mutasi* pada *initial solution* yang ada :

Iterasi 1/Generasi I :

Crossover:

Parent I:

1 2 3 10 11 12 13 14 15 16 17 18 4 5 6 19 20 21
22 23 24 25 26 7 8 27 28 29 30 31 9 32 33 34 35
36 37 38 39 40 41 42

Parent II:

1 2 10 11 12 13 14 15 16 3 4 5 6 17 18 19 20 21
22 23 7 8 24 25 26 27 28 29 30 9 31 32 33 34 35
36 37 38 39 40 41 42

Ditentukan probabilitas terjadinya *crossover* sebesar 0,75 ($P_c = 0,75$) dan dengan bantuan *software* Minitab didapat probabilitas r sebesar 0,43. Karena $r < P_c$, maka perlu dilakukan *crossover*.

Langkah selanjutnya adalah merandom 2 angka dari 1-42 (sesuai dengan jumlah elemen) dengan bantuan *software* Minitab dan didapat hasil angka 3 dan 8 yang artinya bahwa Parent I dengan elemen dari urutan ke-3 sampai dengan ke-8 harus di *crossover* dengan Parent II.

Parent I:

1 2 3 10 11 12 13 14 15 16 17 18 4 5 6 19 20 21 22 23 24 25 26 7 8 27 28
29 30 31 9 32 33 34 35 36 37 38 39 40 41 42

Parent II:

1 2 10 11 12 13 14 15 16 3 4 5 6 17 18 19 20 21 22 23 7 8 24 25 26 27 28
29 30 9 31 32 33 34 35 36 37 38 39 40 41 42

Hasil Crossover (Child I):

1 2 3 10 11 12 13 14 15 16 4 5 6 17 18 19 20 21
22 23 7 8 24 25 26 27 28 29 30 9 31 32 33 34 35
36 37 38 39 40 41 42

Setelah dievaluasi, ternyata kondisi *child I* sudah memenuhi pembatas-pembatas yang ada, yakni *precedence diagram* dan waktu siklus, sehingga proses *crossover* berhasil dan dapat dilanjutkan ke proses Mutasi.

Objective Function Child I = $(15.000.000 \times 5) + (631 \times 27,52) + (1,2035 \times 5 \times 27,52) + [1,2035 \times ((5 \times 27,52) - (27,52 + 26,8 + 20,3 + 26,67 + 20,98) / 27,52)] = \text{Rp. } 75.017.690,98$

Mutasi :

Ditentukan terlebih dahulu probabilitas mutasi sebesar 0,25 ($P_m = 0,25$) dan didapat probabilitas r sebesar 0,17. Karena $r < P_m$, maka perlu dilakukan Mutasi.

Dari hasil *software* Minitab, didapat pula 2 angka random antara 1-42 yakni 3 dan 11, sehingga kedua elemen pada *Child I* harus ditukar.

Hasil Crossover (Child I):

1 2 3 10 11 12 13 14 15 16 4 5 6 17 18 19 20 21 22 23 7 8 24 25 26 27 28
29 30 9 31 32 33 34 35 36 37 38 39 40 41 42

Hasil Mutasi (Child II) :

1 2 4 10 11 12 13 14 15 16 3 5 6 17 18 19 20 21
22 23 7 8 24 25 26 27 28 29 30 9 31 32 33 34 35
36 37 38 39 40 41 42

Setelah dievaluasi, ternyata kondisi *child II* sudah memenuhi pembatas-pembatas yang ada, yakni *precedence diagram* dan waktu siklus, sehingga proses Mutasi berhasil.

Objective Function Child II = $(15.000.000 \times 5) + (631 \times 27,52) + (1,2035 \times 5 \times 27,52) + [1,2035 \times ((5 \times 27,52) - (27,52 + 26,8 + 20,3 + 26,67 + 20,98) / 27,52)] = \text{Rp. } 75.017.690,98$

Selanjutnya, akan diulang kembali tahapan-tahapan yang ada mulai dari Tahap II: *Fitness*. Hal ini dilakukan sampai 100 iterasi. Oleh karena hal tersebut, maka untuk iterasi-iterasi berikutnya digunakan program yang dirancang khusus untuk menghitung Algoritma Genetik ini. Berikut merupakan *flowchart* program Algoritma Genetik (Gambar 2).

IV.5. Analisis Data

IV.5.1. Verifikasi Model Subiantoro

1. Tidak terdapat pelanggaran terhadap fungsi-fungsi pembatas yang terdapat dalam model.
2. Semua variabel keputusan sudah bersifat bilangan biner (0 atau 1).

Dengan ini model dapat disimpulkan *Verified*.

IV.5.2. Verifikasi Model Algoritma Genetik

Tidak terdapat pelanggaran terhadap fungsi-fungsi pembatas yang terdapat dalam model (waktu siklus dan *precedence diagram*).

IV.5.3. Validasi Model Algoritma Genetik

Hasil model optimasi Subiantoro menghasilkan 5 stasiun kerja dengan nilai *fitness* 17.017.600,00. Sedangkan model Algoritma Genetik menghasilkan 5 stasiun kerja dengan nilai *fitness* terkecil 75.017.690,98. Dari hasil ini, model algoritma genetik hanya menyimpang 0,00001% dari model optimasi Subiantoro sehingga model algoritma genetik dapat dinyatakan *valid*.

IV.6. Analisis Sensitivitas Model Algoritma Genetik

IV.6.1. Analisis Pengaruh Perubahan Biaya Pengadaan Stasiun Kerja Terhadap Fungsi Objective

Pengaruh perubahan biaya pengadaan stasiun kerja terhadap fungsi *objective* dapat dilihat pada gambar 5.

Dari gambar tersebut, dapat disimpulkan bahwa semakin besar biaya pengadaan stasiun kerja, maka total biayanya juga akan semakin besar.

IV.6.2. Analisis Pengaruh Perubahan Biaya Waktu Siklus Terhadap Total Biaya

Pengaruh perubahan biaya waktu siklus terhadap total biaya dapat dilihat pada gambar 6.

Dari gambar tersebut, dapat disimpulkan bahwa semakin besar biaya waktu siklus, maka fungsi *objective*-nya juga akan semakin besar.

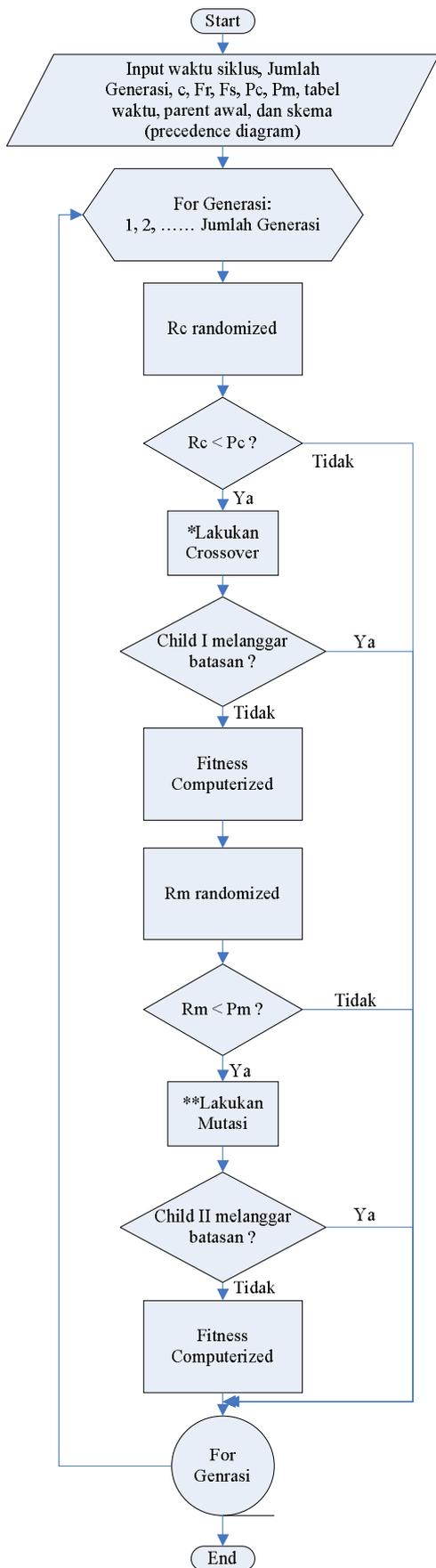
IV.6.3. Analisis Pengaruh Perubahan Gaji Operator Terhadap Total Biaya

Pengaruh perubahan gaji operator terhadap total biayadapat dilihat pada gambar 7.

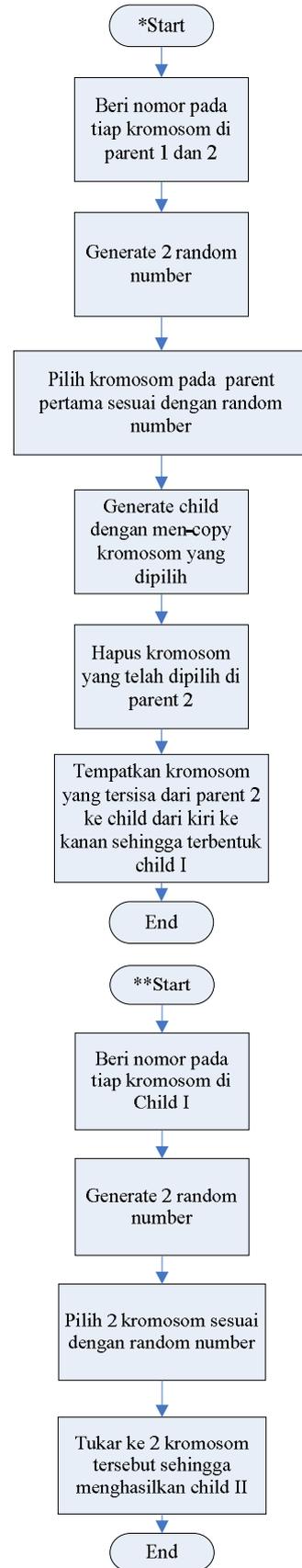
Dari gambar tersebut, dapat disimpulkan bahwa semakin besar biaya waktu siklus, maka fungsi *objective*-nya juga akan semakin besar.

IV.6.4. Analisis Pengaruh Perubahan Gaji Operator Terhadap Total Biaya

Pengaruh perubahan gaji operator terhadap total biaya dapat dilihat pada gambar 7.



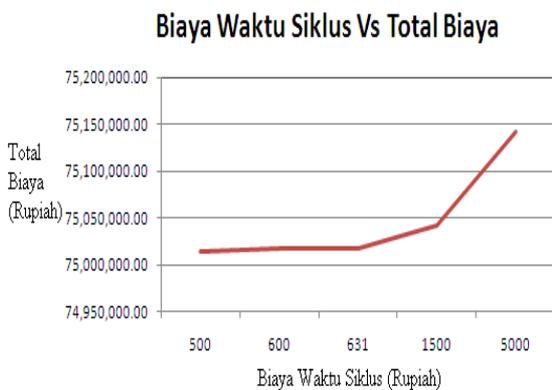
Gambar 2. Flowchart program Algoritma Genetika



Gambar 3. Flowchart program untuk Crossover (Lanjutan gambar 2)



Gambar 5. Pengaruh perubahan biaya pengadaan stasiun kerja terhadap total biaya.



Gambar 6. Pengaruh perubahan biaya waktu siklus terhadap total biaya.

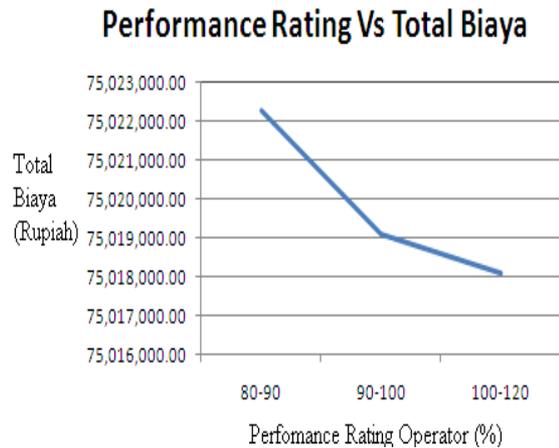


Gambar 7. Pengaruh perubahan gaji operator terhadap total biaya.

Dari gambar diatas, dapat disimpulkan bahwa semakin besar gaji operator, maka total biayanya juga akan semakin besar.

IV.6.5. Analisis Pengaruh Perubahan Performance Rating Operator Terhadap Total Biaya

Pengaruh perubahan *performance rating* operator terhadap total biaya dapat dilihat pada gambar 8.

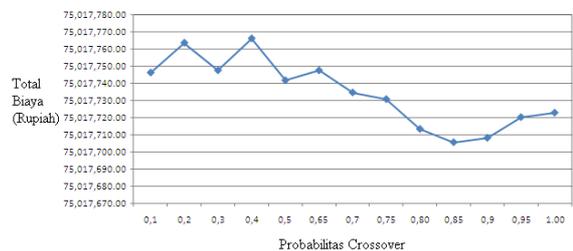


Gambar 8. Pengaruh perubahan *performance rating* terhadap total biaya

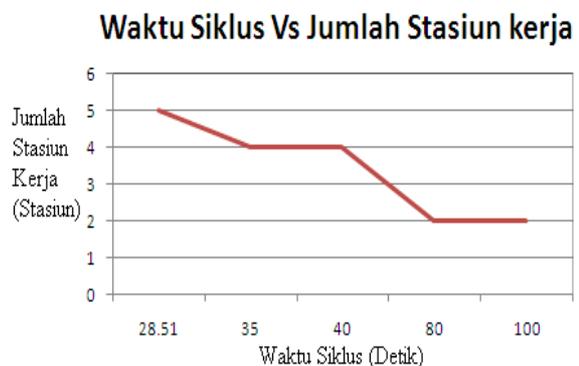
Dari gambar diatas, dapat disimpulkan bahwa semakin besar *Performance Rating* operator, maka total biayanya juga akan semakin kecil.

IV.6.5. Analisis Pengaruh Perubahan Waktu Siklus Operator Terhadap Jumlah Stasiun Kerja

Pengaruh perubahan waktu siklus terhadap jumlah stasiun kerja dapat dilihat pada gambar 9.



Gambar 10. Grafik Hasil Rangkuman Percobaan 1



Gambar 9. Pengaruh perubahan Waktu Siklus terhadap Jumlah Stasiun Kerja

IV.7. Analisis Parameter Model Algoritma Genetik

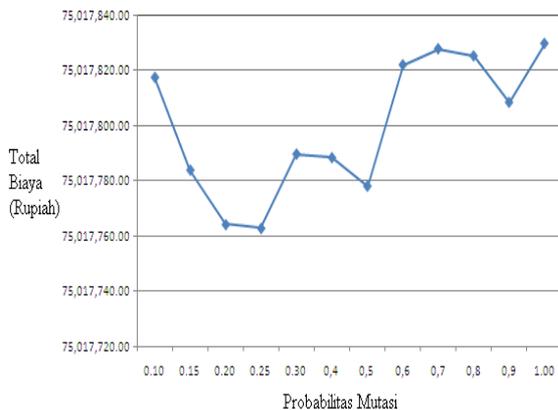
IV.7.1. Analisis Pengaruh Perubahan Probabilitas Crossover Terhadap Total Biaya

Dengan menggunakan *software* Algoritma Genetik, dilakukan beberapa percobaan dengan mengganti probabilitas *crossover* (Pc) dengan nilai antara 0,1 sampai 1 dan dengan probabilitas mutasi (Pm) yang nilainya tetap (0,25). Percobaan dilakukan sebanyak 100 kali dengan 10 replikasi tiap percobaan (Gambar 10).

Dari gambar diatas, rata-rata nilai *fitness* terkecil didapat dengan nilai Pc = 0,85. Probabilitas *crossover* yang terlalu besar akan mengakibatkan struktur dari nilai fungsi objektif yang baik akan hilang dengan lebih cepat dari seleksi. Namun sebaliknya, apabila probabilitas *crossover* terlalu kecil maka akan menghalangi proses seleksi dalam proses Algoritma genetik. Jadi apabila hanya dilihat dari sudut pandang *crossover*, nilai *fitness* terbaik dapat diperoleh dengan menggunakan probabilitas *crossover* (Pc) sebesar 0,85.

IV.7.2. Analisis Pengaruh Perubahan Probabilitas Mutasi Terhadap Total Biaya

Dengan menggunakan *software* Algoritma Genetik, dilakukan beberapa percobaan dengan mengganti probabilitas mutasi (Pm) dengan nilai antara 0,1 sampai 1 dan dengan probabilitas *crossover* (Pc) yang nilainya tetap (0,75). Percobaan dilakukan sebanyak 100 kali dengan 10 replikasi tiap percobaan. Berikut merupakan hasil rangkuman percobaan yang dilakukan:



Gambar 11. Grafik Hasil Rangkuman Percobaan 2

Dari tabel dan gambar diatas, rata-rata nilai *fitness* terkecil didapat dengan nilai Pm = 0,25. Probabilitas mutasi yang terlalu kecil akan menyebabkan gen-gen yang berpotensi tidak dicoba. Namun sebaliknya, apabila probabilitas mutasi terlalu tinggi maka akan menyebabkan keturunan akan semakin mirip dengan induknya. Jadi apabila hanya dilihat dari sudut pandang mutasi, nilai *fitness* terbaik dapat diperoleh

dengan menggunakan probabilitas mutasi (Pm) sebesar 0,25.

IV.7.3. Analisis Pengaruh Perubahan Probabilitas Crossover dan Mutasi Terhadap Total Biaya

Dengan menggunakan *software* Algoritma Genetik, dilakukan beberapa percobaan dengan mengganti probabilitas *crossover* (Pc) dan probabilitas mutasi (Pm) secara bersamaan. Percobaan dilakukan sebanyak 100 kali dengan 10 replikasi tiap percobaan. Berikut merupakan hasil rangkuman percobaan yang dilakukan :

Tabel 4. Hasil Rangkuman Percobaan 2

Pc	Pm	Rata-rata
0,1	0,5	75,017,818.60
0,5	0,4	75,017,763.94
0,8	0,5	75,017,765.23
0,5	0,5	75,017,769.73
0,65	0,30	75,017,821.17
0,80	0,20	75,017,720.48
0,65	0,1	75,017,804.45
0,95	0,20	75,017,743.37
1,00	0,10	75,017,786.45
1,00	0,30	75,017,720.86

Dari tabel diatas, rata-rata nilai *fitness* terkecil didapat dengan nilai Pc = 0,80 dan Pm = 0,20. Jadi apabila dilihat dari sudut pandang *crossover* dan mutasi, nilai *fitness* terbaik dapat diperoleh dengan menggunakan probabilitas *crossover* (Pc) sebesar 0,80 dan probabilitas mutasi (Pm) sebesar 0,20.

IV.8. Analisis Hasil Model Algoritma Genetik

Dari hasil *output* terbaik model Algoritma Genetik, nilai *fitness* terkecil yang diperoleh adalah adalah Rp 75.017.690,98 dengan Pc = 0,80 dan Pm = 0,20. Terbentuk 5 stasiun kerja, stasiun kerja 1 berisi elemen kerja 1,2,10,11,12,13,14,15,dan 16. Stasiun kerja 2 berisi elemen kerja 3,4,5,6,17,18,19,20,21,22,23. Stasiun kerja 3 berisi elemen kerja 7,8,24,25,26,27,28,29,30. Stasiun kerja 4 berisikan elemen kerja 9,31,32,33,34,35. Stasiun kerja 5 berisikan elemen kerja 36,37,38,39, 40,41 dan 42.

V. Kesimpulan

Dari hasil pengolahan dan **Analisis** data, dapat disimpulkan beberapa hal sebagai berikut:

1. Algoritma genetik hanya menyimpang 0,0001 % dari hasil model optimasi.
2. Dengan Algoritma Genetik terbentuk 5 stasiun kerja dengan total biaya sebesar Rp. 75.017.690,98.

Daftar Pustaka

- [1] Elsayed, Elsayed A., Boucher Thomas O., “*Analysis and Control of Production Systems*“, Hlm. 344-346, Prentice-Hall International Inc., New Jersey, 1994.
- [2] Santosa, Budi dan Paul Willy, “*Metoda Metaheuristik: Konsep dan Implementasi*“, Hlm. 87-93, Guna Widya, Indonesia, 2011.
- [3] Anderson, E.J. and M.C. Ferris, A genetic algorithm for the assembly line balancing problem, *In Proceeding of the Integer Programming / Combinatorial Optimization Conference, Waterloo, Ontario, Canada, May 28-30*, Hlm. 1-23, University of Waterloo Press, 1993.
- [4] Chong, Kuan Eng, Mohamed K. Omar, dan Nooh Abu Bakar, Solving assembly line balancing problem using genetic algorithm with heuristics treated initial population, *Proceeding of the World Congress on Engineering 2008 Vol II*, Hlm. 3-7, London, July 2-4, 2008.
- [5] Subiantoro, G., *Pengembangan Model Line Balancing Untuk Mixed Model Line Balancing dengan mempertimbangkan Kapasitas*, Hlm. 12-18. Surabaya: Universitas Katolik Widya Mandala, 2007.

Suplemen 1. Waktu Standar Elemen Kerja

Elemen	Waktu Standar (detik)
1	0,68
2	1,20
3	0,85
4	0,91
5	0,93
6	0,63
7	1,44
8	1,53
9	1,78
10	2,24
11	1,85
12	2,89
13	3,08
14	4,48
15	2,58
16	2,57
17	2,63
18	3,41
19	2,99
20	1,77
21	3,13
22	3,99
23	3,81
24	2,27
25	2,46
26	5,61
27	1,44
28	5,37
29	3,40
30	2,70
31	11,79
32	6,72
33	5,14
34	1,27
35	1,75
36	3,15
37	1,47
38	5,43
39	2,48
40	2,65
41	1,71
42	4,09

Suplemen 2. Waktu Elemen Kerja dan *Direct Predecessor*

Elemen	<i>Task Time</i>	<i>Predecessor</i>	Ei	Li	<i>Assignment of Zero-One Variables</i>
1	0,68	-	1	2	X11,X12
2	1,20	-	1	2	X21,X22
3	0,85	-	1	2	X31,X32
4	0,91	-	1	2	X41,X42
5	0,93	-	1	2	X51,X52

6	0,63	-	1	2	X61,X62
7	1,44	-	1	2	X71,X72
8	1,53	-	1	2	X81,X82
9	1,78	-	1	2	X91,X92
10	2,24	-	1	2	X101,X102,
11	1,85	1,10	1	2	X111,X112
12	2,89	2,11	1	2	X121,X122
13	3,08	12	1	2	X131,X132
14	4,48	13	1	2	X141,X142
15	2,58	14	1	2	X151,152
16	2,57	15	2	3	X162,X163
17	2,63	16	2	3	X172,X173,
18	3,41	3,17	2	3	X182,X183
19	2,99	18	2	3	X192,X193
20	1,77	4	2	3	X202,X203
21	3,13	19,20	2	3	X212,X213
22	3,99	5,21	2	3	X222,X223
23	3,81	22	2	3	X232,X233
24	2,27	23	2	3	X242,X243
25	2,46	6	2	4	X252,X253,X254
26	5,61	24,25	2	4	X262,X263,X264
27	1,44	26	2	4	X272,X273,X274
28	5,37	7,8	2	4	X282,X283,X284
29	3,40	27	2	4	X292,X293,X294
30	2,70	28,29	2	4	X302,X303,X304
31	11,79	30	3	5	X313,X314,X315
32	6,72	31	3	5	X323,X324,X325
33	5,14	32	3	5	X333,X334,X335
34	1,27	33	3	5	X343,X344,X345
35	1,75	9,34	3	5	X353,X354,X355
36	3,15	35	4	5	X364,X365
37	1,47	36	4	5	X374,X375
38	5,43	37	4	5	X384,X385
39	2,48	38	4	5	X394,X395
40	2,65	39	4	5	X404,X405
41	1,71	40	4	5	X414,X415
42	4,09	41	4	5	X424,X425
No	Elemen Kerja				
1	Mensesep Toe Box				
2	Menyablon Vamp				
3	Menyablon Back Counter				
4	Menyablon Side Rime				
5	Menyablon Quarter Lining				

6	Mensesep Counter Box
7	Menyablon Tongue Atas
8	Menyablon Tongue bawah
9	Menyablon Sock Lining
10	Vamp Lining Dijahit Zigzag
11	Elemen 10 dijahit dengan Elemen 1
12	Elemen 11 dijahit dengan Elemen 2
13	Elemen 12 dijahit tepinya supaya kuat
14	Elemen 13 dijahit membentuk pola tertentu
15	Elemen 14 dijahit dengan Var. Quarter 1-3
16	Elemen 15 dijahit dengan Var. Quarter 2-4
17	Elemen 16 dijahit dan disatukan bagian belakangnya
18	Elemen 17 dijahit dengan Vamp
19	Elemen 18 dijahit dengan Eyestay
20	Menjahit Elemen 4
21	Elemen 19 dijahit dengan elemen 20
22	Elemen 21 dijahit dengan elemen 5
23	Elemen 22 dijahit dengan Collar Lining
24	Elemen 23 Colar Liningnya dilatex luar dalam
25	Melatex Elemen 6
26	Elemen 24 digabung dengan elemen 25 kemudian dijahit dengan Collar Filler dan dimasukkan oven
27	Elemen 26 Colar Liningnya dibalik
28	Menjahit Pita Tongue, Tongue Lining Atas, Tongue Lining, Elemen 7, dan Elemen 8
29	Elemen 27 diplong untuk membuat lubang tali sepatu
30	Elemen 29 dijahit dengan elemen 28
31	Elemen 30 di Injection untuk membuat Outsole
32	Elemen 31 di cat bagian bawahnya
33	Elemen 32 diberi Insole dan di lem
34	Elemen 33 dipaku Outsolenya depan dan belakang
35	Elemen 34 diberi Elemen 9
36	Elemen 35 Di Inspeksi
37	Elemen 36 diberi kertas bagian dalamnya
38	Elemen 37 diberi tali sepatu
39	Elemen 38 dibersihkan
40	Elemen 39 diberi string pin dan label
41	Elemen 40 diberi nomor seri
42	Elemen 41 dimasukkan ke dalam box