

OPTIMISASI PEMBENTUKAN SEL DIINTEGRASIKAN DENGAN PENEMPATAN MESIN DAN PENJADWALAN DI DALAM SELULAR MANUFAKTUR MENGGUNAKAN ALGORITMA GENETIKA

Moh Khoiron¹, Imam Mukhlash², Soetrisno³

^{1,2,3}Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111

³imamm@matematika.its.ac.id

Abstrak

Perindustrian di Indonesia belakangan ini semakin berkembang, hal ini mengantarkannya pada persaingan global, sehingga mendorong banyak perusahaan untuk semakin memperbaiki dan meningkatkan usahanya agar lebih efektif dan efisien. Salah satu yang berdampak signifikan pada keefektifan dan keefesian suatu perusahaan adalah perencanaan fasilitas. *Cellular Manufacturing* (CM) adalah salah satu metode yang telah terbukti mampu menambah efisiensi serta fleksibilitas dalam lingkungan produksi manufaktur. Diantara faktor-faktor yang diperlukan dalam selular manufaktur adalah *Cell Formation* (CF), *Group Layout* (GL), dan *Group Scheduling* (GS). Biasanya, tiga faktor ini diselesaikan dalam keadaan terpisah atau diselesaikan secara berurutan. Tugas akhir ini membahas tentang metode untuk penyelesaian CF, GL, GS secara bersamaan. A *hierarchical genetic algorithm* (HGA) digunakan untuk menyelesaikan permasalahan tersebut.

Kata kunci: Algoritma Genetika; *Cellular Manufacturing*; *Cell Formation* (CF); *Group Layout* (GL); *Group Scheduling* (GS); *Heuristic*

1. Pendahuluan

Perencanaan fasilitas merupakan rancangan dari fasilitas-fasilitas industri yang akan dibangun. Di dunia industri, perencanaan fasilitas dimaksudkan sebagai sarana untuk perbaikan layout fasilitas yang digunakan dalam penanganan material (*material handling*), penentuan peralatan dalam proses produksi, dan perencanaan fasilitas secara keseluruhan. Ada 2 hal pokok dalam perencanaan fasilitas yaitu, berkaitan dengan perencanaan lokasi

pabrik (*plant location*) dan perancangan fasilitas produksi yang meliputi perancangan struktur pabrik, perancangan tata letak fasilitas dan perancangan sistem penanganan material [2].

Penempatan fasilitas di area pabrik, sering disebut juga sebagai *Facility Layout Problem* (FLP), dikenal memiliki dampak yang signifikan terhadap biaya produksi, proses kerja, penjadwalan dan produktivitas. Sebuah penempatan fasilitas yang baik mampu meningkatkan efisiensi keseluruhan operasi dan dapat mengurangi sampai 50% jumlah beban usaha[3]. Sebagian besar prosedur dalam pemecahan masalah FLP mengadopsi rumusan masalah yang dikenal sebagai *Quadratic Assignment Problem* (QAP). Pada QAP, lokasi tiap *site* ditentukan di depan. Solusi akhir adalah menempatkan fasilitas untuk setiap *site* sehingga jarak total antara fasilitas diminimalkan. *Weights* dapat diukur baik oleh indeks *adjacency* atau dengan volume aliran material antara fasilitas. Itu menunjukkan bahwa QAP merupakan masalah NP-lengkap[4]. Sebelumnya telah banyak penelitian yang telah dilakukan untuk menemukan solusi terbaik untuk menyelesaikan masalah FLP. Algoritma genetika adalah salah satu metode yang mampu menemukan solusi optimal.

Meskipun sama-sama menggunakan GA, tapi kebanyakan peneliti berbeda dalam pendekatan permasalahannya, sehingga *objective function* yang diambil juga berbeda. diantaranya adalah yang dilakukan Koopmans dan Beckmann (1957) yang mendefinisikan FLP sebagai konfigurasi fasilitas, sehingga dapat meminimalkan biaya *material handling*. Pada kajian yang dilakukan oleh Azadivar dan Wang (2000), didefinisikan bahwa FLP adalah penentuan lokasi relatif dan alokasi untuk beberapa fasilitas pada ruangan yang telah diberikan. Lee dan Lee (2002) mendefinisikan bahwa FLP adalah pengaturan n fasilitas dengan ukuran yang berbeda di dalam total ruang yang telah ditentukan untuk meminimalkan biaya *material handling* dan total *slack area cost*. Shayan dan Chittilappilly (2004) mendefinisikan FLP sebagai masalah optimasi yang membuat layout lebih efisien dengan memperhatikan berbagai interaksi antara fasilitas dan sistem material handling saat merancang tata letak.

Dalam usulan tugas akhir ini, metode penyelesaian masalah FLP akan menggunakan metode Algoritma Genetika (GA) untuk mengintegrasikan *cell formation* dengan *machine layout* dan *scheduling* dalam pendekatan *cellular manufacturing*. Pertama kali yang dilakukan adalah menganalisa karakteristik *workshop* dan juga *production flow* sebuah departemen yang akan diteliti, selanjutnya adalah menformulasikan fungsi tujuan, yang mana dalam penelitian ini fungsi tujuannya adalah meminimalkan *makespan*, yang terakhir adalah menyelesaikan permasalahan dengan Algoritma Genetika. Mengingat prinsip GA adalah teknik pencarian stokastik menggunakan prinsip Darwin "*survival of the fittest*", GA mampu menemukan penempatan posisi fasilitas yang optimal. Sehingga terjadinya proses alur produk yang rapi dan total waktu tempuh *part* dalam alur produksi diminimalkan. Hal tersebut sangat penting untuk diselesaikan agar dapat menjaga kualitas dan kelancaran produksi.

2. Tinjauan Pustaka

2.1. Cellular Manufacturing

Cellular Manufacturing System (CMS) adalah sistem produksi yang mengklasifikasikan *parts* yang mirip menjadi bagian *part families* dan mesin yang berkaitan ditempatkan ke dalam sel untuk meningkatkan efektivitas biaya produksi dan fleksibilitas *job shop* manufaktur.

Pada dasarnya, sistem manufaktur dipecah menjadi beberapa subsistem, dinamakan manufaktur sel. Desain CMS meliputi;

- 1) Pembentukan Sel(PS) - pengelompokan *part* yang didasarkan pada fitur desain atau proses yang serupa menjadi *part families* dan mesin yang terkait ke dalam sel mesin,
- 2) *group layout* - meletakkan mesin dalam setiap sel (tata letak intra-sel) dan sel-sel terhadap satu sama lain (tata letak antar-sel),
- 3) *group scheduling* - bagian penjadwalan
- 4) alokasi sumber daya – penugasan alat, sumber daya manusia dan material.

2.2. Formulasi Masalah

Untuk membantu lebih memahami masalah dan membangun dasar untuk mengembangkan prosedur heuristik, sebuah model matematika yang komprehensif, yang mempertimbangkan faktor-faktor yang paling penting yang menyangkut CM dan mengintegrasikan CF, GL dan GS, dikembangkan beberapa variabel dan notasi yang digunakan dalam model

- Pengindekan
 - i : indek untuk *part*; $i = 1, \dots, n$
 - j : indek untuk mesin; $j = 1, \dots, m$
 - k : indek untuk *part*; $k = 1, \dots, c$
 - o : indek untuk operasi; $o = 1, \dots, n_j$
 - p : indek untuk nomer posisi mesin $p = 1, \dots, M$
- Parameter

S_p	set pasang (i,j) seperti $a_{ij} = 1$	N_i	set operasi yang terjadi pada mesin i , $i = 1, \dots, m$ ($Z_{joi} > 0, j = 1, \dots, n, o = 1, \dots, n_j$)
M	bilangan sembarang, bilangan yang besar	t_{jo}	waktu proses mesin untuk operasi o dari <i>part</i> j , $o = 1, \dots, n_j, j = 1, \dots, n$
IC_k	1, jika sel k dibentuk; 0, selainnya	g_{jo}	waktu selesai operasi o untuk <i>part</i> j , $h = 1, \dots, n_j, j = 1, \dots, n$
X_{ik}	1, jika mesin i berada di sel k ; 0, selainnya	$g_{[j]}$	waktu selesai operasi o (urutan terakhir) dari <i>part</i> $j, j=1, \dots, n$
Y_{jk}	1, jika <i>part</i> j berada di sel k ; 0, selainnya	$T_{j(o'o')}$	waktu tempuh <i>part</i> j antara o dan o'

M_{ip}	p , mesin i ditaruh pada posisi p	T_{jA}	waktu tempuh intra-cell $part$ j
Z_{joi}	1, jika operasi o dari $part$ j diproses pada mesin i ; 0, selainnya	T_{jE}	waktu tempuh inter-cell $part$ j
R_j	set pasang dari operasi $[o, o']$ untuk $part$ j , dimana operasi o mendahului o' , $j = 1, \dots, n$ ($Z_{jo'i'} - Z_{joi} = 1$)	$z_{oo'}$	1, jika operasi o mendahului operasi o' ; 0, selain itu. $\forall [o, o'] \in N_i, i = 1, \dots, m$
I_j	set operasi tanpa ada batasan mendahului (operasi yang satu dengan yang lain), $j=1, \dots, n$		

- Fungsi objektif

$$\text{Min} \sum_{j=1}^n g_{[j]}$$

- Kendala

$$\sum_{k=1}^c X_{ik} = 1, \quad i = 1, \dots, m$$

$$\sum_{k=1}^c Y_{jk} = 1, \quad j = 1, \dots, n$$

$$\sum_{i=1}^m X_{ik} \leq NM \times IC_k, \quad k = 1, \dots, c$$

$$\sum_{i=1}^m Y_{jk} \geq NM \times IC_k, \quad k = 1, \dots, c$$

$$\sum_{i=1}^m M_{ip} = 1, \quad p = 1, \dots, M$$

$$\sum_{p=1}^M M_{ip} = 1, \quad i = 1, \dots, m$$

$$\sum_{k=1}^{n_j} Z_{joi} X_{ik} Y_{jk} \geq 1, \quad j = 1, \dots, m; o = 1, \dots, n_j$$

$$g_{jo'} - g_{j'o} + M(1 - z_{oo'}) \geq t_{jo'} \quad \forall [o, o'] \in N_i, \quad i = 1, \dots, m$$

$$g_{j'o} - g_{jo'} + Mz_{oo'} \geq t_{jo}, \quad j \neq j'$$

$$g_{jo} \geq t_{jo}, \quad \forall j, o$$

$$X_{ik}, Y_{jk}, Z_{joi} = 0 \text{ atau } 1$$

$$z_{oo'} = 0 \text{ atau } 1 \quad \forall [o, o'] \in N_i, i = 1, \dots, m$$

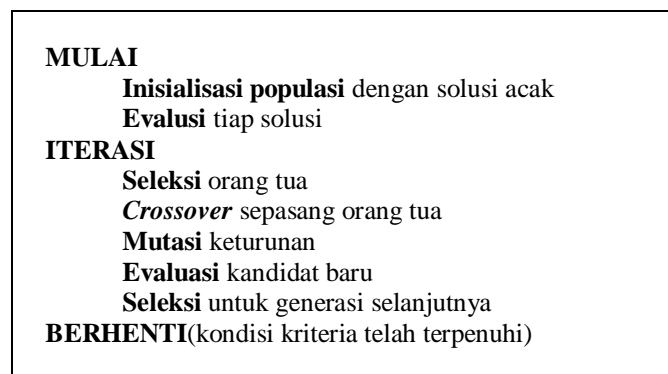
2.3. Algoritma Genetik

GA adalah teknik pencarian stokastik menggunakan prinsip-prinsip Darwin yaitu “*survival of the fittest*” untuk menemukan solusi terbaik. Semakin tinggi *fittest*-nya, maka semakin tinggi pula kemungkinannya untuk bertahan hidup dan begitu juga sebaliknya. Aspek penting dari GA adalah bahwa, jika diberikan permasalahan yang jelas, GA mampu memberikan solusi yang optimal atau mendekati, bahkan dalam ruang pencarian yang besar dan rumit. GA telah digunakan dalam berbagai bidang praktis termasuk desain, penjadwalan, konfigurasi sistem, manajemen.

GA adalah metode yang mengintegrasikan model stokastik dan pencarian langsung untuk menemukan solusi optimal di dalam waktu komputasi yang terhitung cepat. GA membentuk sebuah populasi solusi yang memenuhi persyaratan dan bertahan dari uji fungsi kelayakan, tapi juga tetap membiarkan solusi yang mempunyai kualitas rendah tetap bertahan agar menghasilkan populasi solusi yang beraneka ragam. Proses ini mampu membuat GA memberikan solusi yang baik juga menghindari *Premature Convergence*.

Tiap kandidat solusi dalam populasi dikodekan dengan string digit yang disebut kromosom. *Offspring* atau keturunan dihasilkan dari operator probabilistik, yaitu *crossover* dan mutasi. Kandidat baru(anak) dan lama (orang tua) dibandingkan berdasarkan fungsi kelayakan supaya mengasilkan kandidat yang lebih bagus pada generasi selanjutnya. Pada proses GA ini, karakteristik tiap kandidat solusi diturunkan pada tiap generasai melalui seleksi, *crossover* dan mutasi.

Rangka GA secara umum ditampilkan dibawah ini.



3. Implementasi

3.1. Penyelesaian Pembentukan Sel, Lay-Outing dan Penjadwalan dengan Algoritma Genetika

Pembentukan Sel (PS) adalah mengelompokkan *part* yang memiliki kemiripan proses dan desain, serta memasukkan mesin yang bersangkutan ke dalam sel mesin. Pada banyak literatur, dalam pembentukan sel, peneliti mengacu pada analisis aliran produksi untuk membentuk grup *part* dan mesin yang mempunyai beberapa kriteria, seperti *grouping efficiency*, *grouping*

efficacy, group technology efficiency, dll. PS dimodelkan berdasarkan pada matrik indikator, misal matrik indikator $A = [a_{ij}]$. $a_{ij} = 1$, jika *part* diproses pada mesin j , selain itu $a_{ij} = 0$. Ide utama dalam Pembentukan Sel adalah meng-*cluster* semua ‘satu’ ke dalam tiap sel/*diagonal form*.

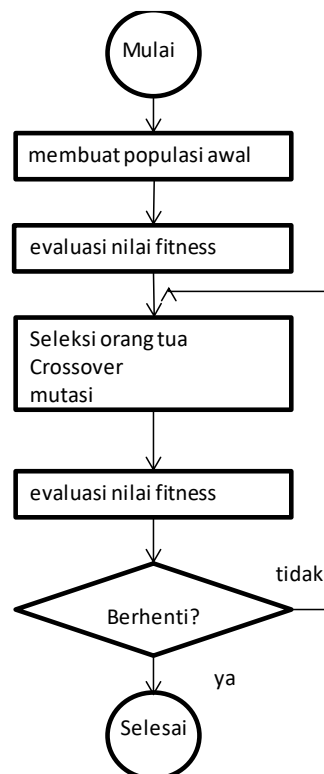
Setelah PS, isu selanjutnya adalah mengurutkan mesin-mesin yang sudah dikelompokkan ke dalam sel-sel tersebut, pada masuk penelitian, fokus utama dalam pengurutan adalah total jarak tempuh proses. Aturan yang digunakan dalam penelitian ini adalah *Positional Weight*(PB).

Di sisi lain, penjadwalan tergambar oleh p pekerjaan dan m mesin akan memiliki $(n!)^m$ kemungkinan jadwal yang akan terbentuk. Tingkat keberhasilan sebuah jadwal dihitung berdasarkan *makespan*-nya, yaitu waktu total seluruh proses dikerjakan, menuruti aturan sesuai dengan *flow process* tiap *part*.

Dalam penelitian tugas akhir ini, penyelesaian ketiga isu tersebut menggunakan GA berlangsung dalam beberapa tahap; tergambar dalam flow chart dibawah.

1. Representasi Kromosom

Untuk setiap penerapan GA, tahap pertama adalah memetakan solusi karakteristik dalam format string kromosom. Setiap kromosom terdiri dari urutan gen tertentu dari alfabet. Alfabet bisa menjadi satu set bilangan biner, bilangan real, bilangan bulat, simbol, atau matriks. Representasi Skema tidak hanya menentukan seberapa efektif masalahnya terstruktur, tetapi juga bagaimana operator genetik dapat digunakan secara efisien.



Gambar 1 flow chart Algoritma Genetika

Dalam studi ini, skema yang diusulkan adalah dua-lapis hirarkis. Kromosom terdiri dari dua lapisan gen dengan panjang yang sama. Lapisan pertama digunakan untuk mengkodekan hasil PS dan lapisan kedua digunakan untuk mewakili tata letak dan informasi penjadwalan. Setiap lapisan gen dibagi menjadi dua zona. Zona pertama terdiri dari gen m , terkait dengan mesin dan zona kedua, yang terdiri dari gen n , terkait dengan *parts*. Oleh karena itu, setiap lapisan memiliki kromosom $m + n$ gen.\

kromosom	mesin									
	1	2	3	4	5	6	7	...	72	
<i>locus</i>	1	2	3	4	5	6	7	...	72	
lapis 1	2	1	1	4	1	2	2	...	3	
lapis 2	3	72	16	5	1	25	9	...	2	

<i>part</i>										
1	2	3	4	5	6	7	8	...	14	
73	74	75	76	77	78	79	80	...	86	
1	4	4	1	2	1	2	3	...	3	
5	8	13	14	2	7	1	6	...	10	

Di layer 1, skema *direct encoding* digunakan, nilai dari setiap gen mewakili nomer sel dimana mesin atau *part* berada. Misalnya, di contoh menunjukkan bahwa mesin 1 berada pada sel # 2, mesin 2 berada pada sel # 1, dan seterusnya, begitu juga untuk tiap *part*. *Part* 1 berada pada sel#1, *part* berada pada sel#4, dan seterusnya. Pada layer 2, skema encoding tidak langsung digunakan, dimana nilai dari setiap gen mewakili Posisi Berat (PB) yang mewakili informasi tata letak dan penjadwalan. Besarnya PB ini sendiri sama dengan banyaknya mesin atau *part* yang diangkat sebagai permasalahan. Semakin besar nilai PB-nya, maka urutannya dalam sel akan didahulukan. Misalnya, PB untuk mesin 1 adalah "3", dan "72" untuk mesin 2, dan seterusnya. PB untuk *part* 1 adalah "5" dan "8" untuk *part* 2. Dengan peringkat PB dari semua mesin, kita bisa menyimpulkan bahwa mesin akan diletakkan dengan urutan pertama untuk *layouting* adalah mesin 2 dan *part* 4 untuk *scheduling*.

2. Inisialisasi Populasi

Tahap kedua GA adalah untuk menghasilkan satu set solusi awal, yang disebut populasi. banyaknya solusi awal yang dimasukkan dalam populasi disebut *population size*. Solusi awal bisa didapat dengan meng-*generate* secara acak atau mengkombinasikan antara *generate* secara acak dan solusi heuristik. Menentukan ukuran populasi yang tepat adalah keputusan yang penting dalam GA. Jika *size* yang dipilih terlalu kecil, tidak mungkin bisa mendapatkan solusi yang baik. Sebaliknya, jika *size* tersebut terlalu besar, CPU membutuhkan waktu yang lama untuk menemukan solusi yang baik. Inisialisasi populasi yang diberikan dalam penelitian TA ini adalah 40.

3. Fitness Function

Fitness Funtion digunakan untuk mengevaluasi dan menentukan apakah suatu kromosom akan bertahan dan digunakan untuk mereproduksi kromosom baru, yang biasa disebut *offspring*. *Fitness Function* ini digunakan untuk menghitung nilai *fitness* setiap kromosom. Nilai *fitness*

tidak memerlukan suatu nilai absolut tetapi yang digunakan adalah nilai relatif terhadap populasi tertentu. Kromosom mendapat peringkat sesuai dengan nilai *fitness*-nya. Pada Tugas Akhir ini, penelitian berfokus pada pencarian *makespan* terpendek. Dengan memisalkan $g(x)$ adalah fungsi tujuannya, C_{max} didefinisikan sebagai *boundary makespan* yang diharapkan dan $f(x)$ adalah fungsi *fitness*-nya didapatkan model matematika;

$$f(x) = \begin{cases} C_{max} - g(x) & \text{when } g(x) < C_{max} \\ 0 & \text{selainnya} \end{cases}$$

Ada beberapa cara untuk menentukan C_{max} . Bisa saja diambil dari nilai terbesar $g(x)$ sejauh ini atau bisa dari nilai terbesar populasi saat ini.

4. Prosedur Seleksi

Tujuan dari prosedur seleksi adalah untuk memberikan kepada kandidat solusi yang ‘terlayak’ mempunyai kemungkinan yang lebih besar untuk terpilih menghasilkan keturunan, pada penelitian tugas ahir ini prosedur seleksi yang digunakan adalah *Roulette Wheel*

5. Operator Genetika

a. Kawin Silang

Reproduksi dilakukan dengan menggunakan operator kawin silang (*crossover*) yang dilakukan pada orang tua terpilih untuk menghasilkan anak (*offspring*). *Crossover* menggabungkan informasi dari dua orang tua tersebut, sehingga kedua anak memiliki kemiripan dari tiap orang tua. Dalam penelitian ini digunakan *crossover* dengan menggunakan *one-point crossover* dan *partially matched crossover* (PMX). Dengan probabilitas *crossover*-nya adalah 0.65.

b. Mutasi

Operasi mutasi diterapkan pada segmen kedua untuk anak ke-2 yang dihasilkan dari operasi *crossover*, kemungkinan terjadinya biasanya ditentukan sangat kecil. Mutasi dapat menyebabkan perubahan nilai gen secara acak. Probabilitas mutasi pada penelitian Tugas Akhir ini adalah 0.1.

6. Stopping Criteria

Stopping Criteria digunakan untuk menghentikan proses, di penelitian ini *stopping criteria* yang digunakan adalah banyaknya generasi yang sudah ditentukan sebelumnya. Maksimal generasi yang digunakan adalah 70.

4. Pengujian Sistem dan Hasil

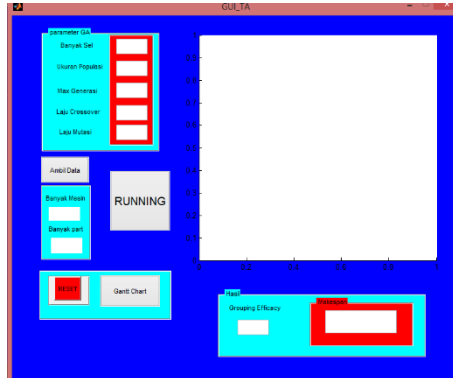
Pengujian sistem dilakukan dengan tujuan mengevaluasi apakah penerapan suatu algoritma memenuhi kondisi yang telah ditetapkan atau tidak. Pada penelitian kali ini, data uji yang digunakan adalah data TA dari penelitian sebelumnya [8], banyaknya mesin adalah 72 dan banyaknya *part* adalah 14 jenis., dan data yang di ambil dan di manipulasi dari jurnal, dengan jumlah mesin adalah 7 dan *part* adalah 14.

Nilai parameter yang digunakan untuk implementasi GA adalah

1. Jumlah sel: 4
2. Ukuran populasi: 40

3. probabilitas *crossover*: 0.65
4. probabilitas mutasi: 0.1
5. Max Generasi: 70

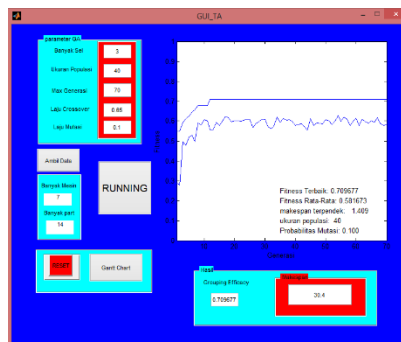
Dari pengembangan perangkat lunak dan permasalahan yang diangkat, dihasilkan solusi untuk masing-masing data sebagai berikut:



Gambar 2 tampilan interface program yang dikembangkan

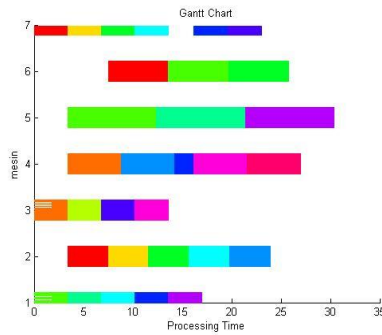
1. Data Jurnal

Pada data jurnal, fungsi *fitness* mulai *stagnan* pada generasi kurang dari 20, ini dikarenakan ruang lingkup permasalahan masih pada lingkup yang kecil. Untuk lebih jelasnya, hasil untuk solusi data jurnal ditampilkan pada gambar dibawah.



Gambar 3 hasil untuk permasalahan pada data jurnal

Dan tampilan untuk penjadwalan, ditampilkan pada table Gantt chart dibawah. Setiap warna mewakili masing-masing *part*



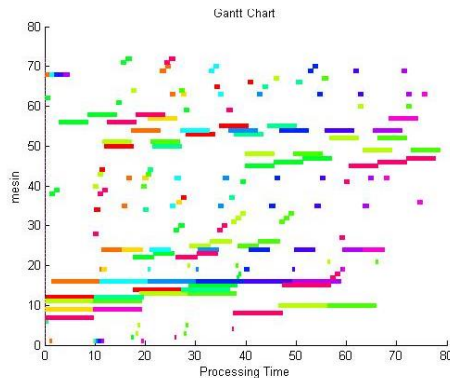
Gambar 4 Gantt chart untuk penjadwalan pada data jurnal

2. Data TA

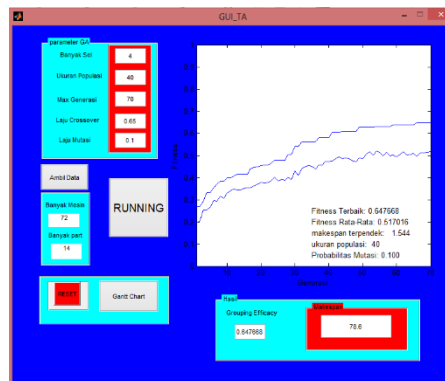
Pada data TA, fungsi *fitness* terus menanjak sampai dan mulai stagnan ketika mendekati pada generasi terakhir, ini dikarenakan ruang lingkup permasalahan termasuk pada lingkup yang besar. Untuk lebih jelasnya, hasil untuk solusi data jurnal ditampilkan pada gambar dibawah.

5. Kesimpulan

Berdasarkan keseluruhan hasil analisis diperoleh kesimpulan sebagai berikut:



Gambar 5 Gantt chart untuk penjadwalan pada data jurnal



Gambar 6 hasil untuk permasalahan pada data TA

6. Daftar Pustaka

- [1] Purnomo, H., Kusumadewi, S. *Aplikasi Algoritma Genetikav Untuk Penentuan Tata Letak Mesin*. Yogyakarta. Universitas Islam Indonesia.
- [2] Apple, JM. *Tata Letak Pabrik dan Pemindahan Bahan*. Bandung: ITB. 1990.
- [3] Drira, A., Pierreval, H., Hajri-Gabouj, S.2007. *Facility layout problems: A survey*. Annual Reviews in Control, Volume 31, Issue 2, Pages 255-267.
- [4] Tarn, K.Y.1991.*Genetic algorithms, function optimization, and facility layout design*. European Journal of Operational Research, Volume 63, Issue 2, Pages 322-346.
- [5] Mahdavi, I, Paydar, M.M., Solimampur, M., Heidarzade, A. 2008. *Genetic Algorithm Approach for Solving a Cell Formation Problem in Cellular Manufacturing*. Expert Systems with Applications, Volume 36, Issue 3, Part 2, Pages 6598-6604
- [6] Wu, X., Chu, C. H., Wang, Y., Yan, W.2007. *A genetic algorithm for cellular manufacturing design and layout*. European Journal of Operational Research, Volume 181, Issue 1, Pages 156-167
- [7] Wu, X, Chu, C., Wang, Y., Yue, D.2007.*Genetic algorithms for integrating cell formation with machine layout and scheduling*. Computers & Industrial Engineering, Volume 53, Issue 2, Pages 277-289.
- [8] Shururi, A. 2008.*Optimisasi Tata Letak Fasilitas Dalam Sistem manufacturing Cellular Dengan Menggunakan Pendekatan Genetic Algorithm*. Tugas Akhir Teknik Industri FTI ITS.