

LAMPIRAN 1

PENGOLAHAN DATA KASUS PERUSAHAAN

Berikut ini merupakan contoh kasus penjadwalan pembuatan komponen dudukan rol mesin *Bamboo Crusher* dengan menggunakan Algoritma Genetika sebagai metode pemecahan masalahnya. Data-data yang diberikan perusahaan antara lain :

1. Jenis dan jumlah mesin, terdapat enam jenis mesin dan masing- masing berjumlah satu mesin, yaitu:
 - a. Mesin 1 = mesin *Plasma Cutter*
 - b. Mesin 2 = mesin *Frais*
 - c. Mesin 3 = mesin Skrap
 - d. Mesin 4 = mesin Bor
 - e. Mesin 5 = mesin Gergaji
 - f. Mesin 6 = mesin Bubut
2. Jumlah *job* : 5 *job*
3. Tabel Matriks *Routing*

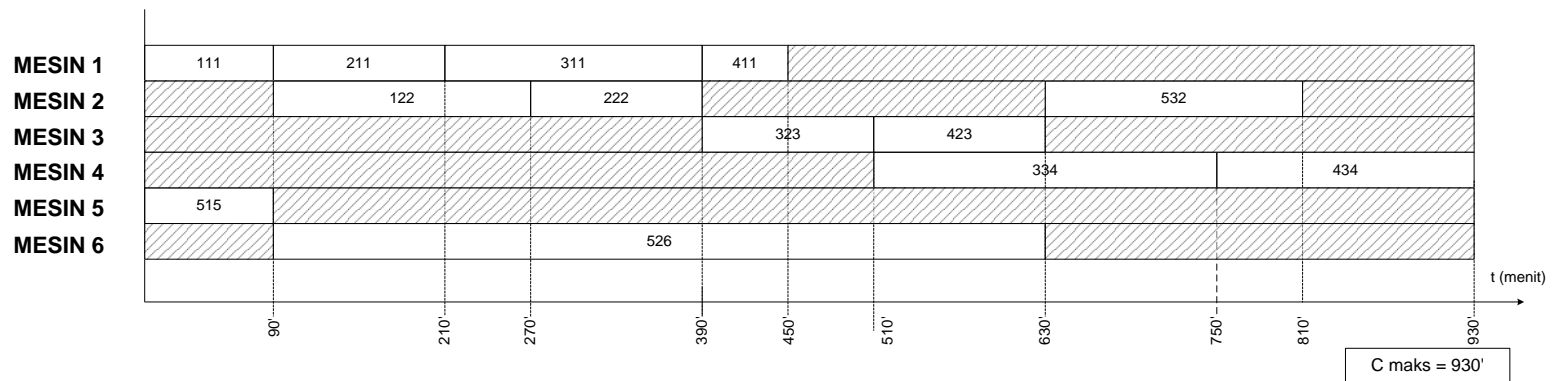
Tabel Matriks routing kasus perusahaan

JOB	KET	OPERASI ke -			
		1	2	3	
1	Mesin	m-1	m-2	-	
	t	90'	180'	-	
2	Mesin	m-1	m-2	-	
	t	120'	120'	-	
3	Mesin	m-1	m-3	m-4	
	t	180'	120'	240'	
4	Mesin	m-1	m-3	m-4	
	t	60'	120'	180'	
5	Mesin	m-5	m-6	m-2	
	t	90'	540'	180'	

Dimana : *Job* 1 = PLAT 270x180
Job 2 = PLAT 180x65
Job 3 = PLAT 270x75
Job 4 = DUDUKAN MUR
Job 5 = BATANG ULIR

❖ **Penjadwalan dengan Metode Perusahaan**

Urutan penjadwalan yang digunakan perusahaan ialah berturut-turut: *job 1, job 2, job 3, job 4* dan *job 5*. Urutan penjadwalan ini berdasarkan pada prioritas perakitan, dimana *job 1* aka dirakit terlebih dahulu dengan *job 2, 3* dan seterusnya. Berdasarkan hasil penjadwalan dari metode perusahaan, maka *gant chart* yang dihasilkan ialah sebagai berikut:



Gantt Chart Metode Perusahaan

Dari *Gantt Chart* di atas, terlihat bahwa *makespan* yang diperoleh dengan metode perusahaan ialah sebesar 930 menit.

❖ **Penjadwalan dengan Metode Algoritma Genetik**

Data- data (parameter) yang diperlukan dalam Algoritma Genetika, yaitu :

1. Ukuran Populasi = 4
2. Jumlah Generasi = 1
3. Probabilitas *crossover* (Pc) = 0.95
4. Probabilitas mutasi (Pm)= 0.01

➤ ***Encoding (Preference List Based Representation)***

Sesuai dengan ukuran populasinya, proses pembentukan kromosom awal akan dibentuk sebanyak 4 buah kromosom. Kromosom tersebut antara lain :

1. Kromosom 1
[(1234)-(125)-(34)-(34)-(5)-(5)]
2. Kromosom 2
[(1324)-(512)-(43)-(43)-(5)-(5)]
3. Kromosom 3
[(2314)-(152)-(43)-(34)-(5)-(5)]
4. Kromosom 4
[(3241)-(512)-(34)-(34)-(5)-(5)]

➤ ***Decoding***

Proses *decoding* ini, diawali dengan melihat *gen/job* yang muncul pertama kali di setiap sub kromosom. Lalu *gen/job* tersebut dijadwalkan sesuai dengan *routing*-nya.

❖ Kromosom 1

1. Iterasi 1 [(1234)-(125)-(34)-(34)-(5)-(5)]

Job yang muncul kali pertama pada sub kromosom :

- *Job* 1 di mesin 1
- *Job* 3 di mesin 3
- *Job* 1 di mesin 2
- *Job* 3 di mesin 4

- *Job* 5 di mesin 5
- *Job* 5 di mesin 6

Job yang bisa dijadwalkan berdasarkan urutan prosesnya (*routing*), yaitu:

- *Job* 1 di mesin 1
- *Job* 5 di mesin 5

2. Iterasi 2 [(234)-(125)-(34)-(34)-(-)-(5)]

Job yang muncul kali pertama pada sub kromosom :

- *Job* 2 di mesin 1
- *Job* 3 di mesin 4
- *Job* 1 di mesin 2
- *Job* 5 di mesin 6
- *Job* 3 di mesin 3

Job yang bisa dijadwalkan berdasarkan urutan prosesnya (*routing*), yaitu:

- *Job* 2 di mesin 1
- *Job* 1 di mesin 2
- *Job* 5 di mesin 6

3. Iterasi 3 [(34)-(25)-(34)-(34)-(-)-(-)]

Job yang muncul kali pertama pada sub kromosom :

- *Job* 3 di mesin 1
- *Job* 3 di mesin 3
- *Job* 2 di mesin 2
- *Job* 3 di mesin 4

Job yang bisa dijadwalkan berdasarkan urutan prosesnya (*routing*), yaitu:

- *Job* 3 di mesin 1
- *Job* 2 di mesin 2

4. Iterasi 4 [(4)-(5)-(34)-(34)-(-)-(-)]

Job yang muncul kali pertama pada sub kromosom :

- *Job* 4 di mesin 1
- *Job* 3 di mesin 3
- *Job* 5 di mesin 2
- *Job* 3 di mesin 4

Job yang bisa dijadwalkan berdasarkan urutan prosesnya (*routing*), yaitu:

- *Job* 4 di mesin 1
- *Job* 5 di mesin 2
- *Job* 3 di mesin 3

5. Iterasi 5 [()-()-(4)-(34)-(-)-(-)]

Job yang muncul kali pertama pada sub kromosom :

- *Job* 4 di mesin 3
- *Job* 3 di mesin 4

Job yang bisa dijadwalkan berdasarkan urutan prosesnya (*routing*), yaitu:

- *Job* 4 di mesin 3
- *Job* 3 di mesin 4

6. Iterasi 6 [()-()-()-(4)-()-()]

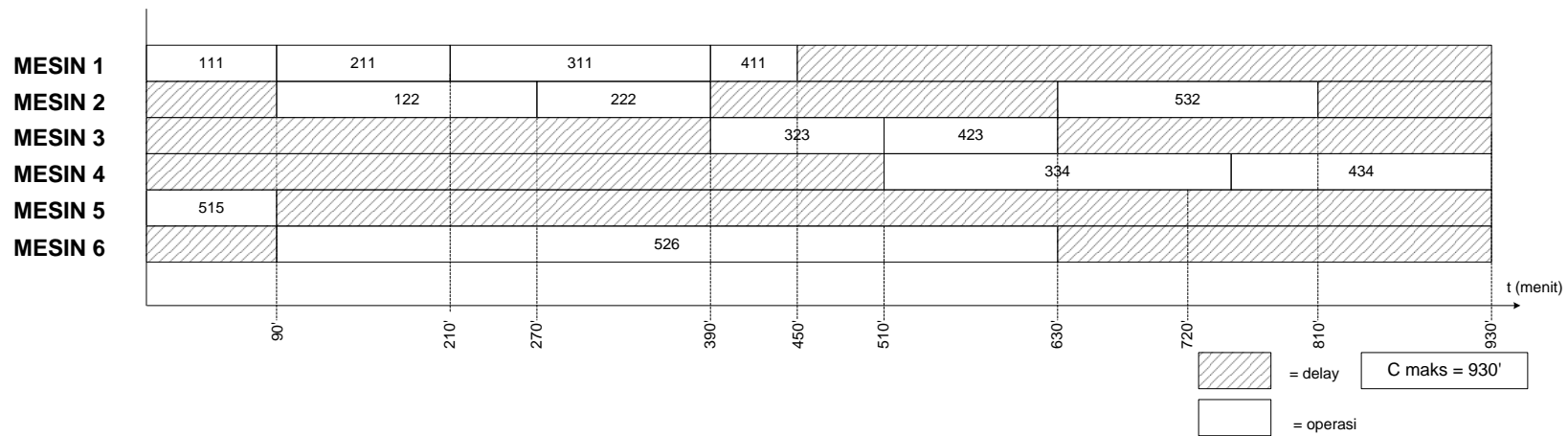
Job yang muncul kali pertama pada sub kromosom :

- *Job* 4 di mesin 4

Job yang bisa dijadwalkan berdasarkan urutan prosesnya (*routing*), yaitu:

- *Job* 4 di mesin 4

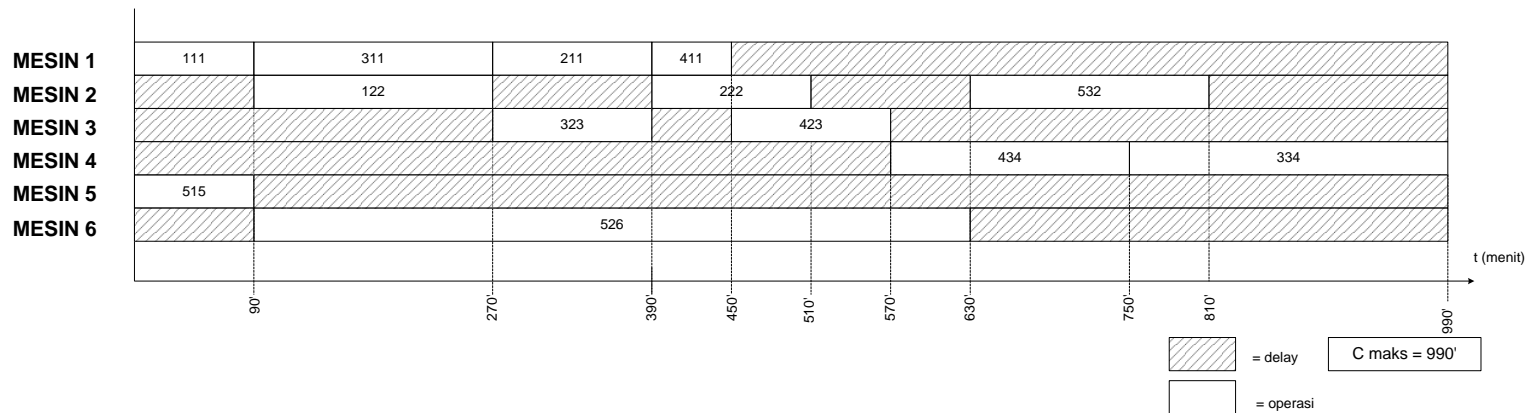
Berikut ialah *Gantt Chart* dari proses *decoding* kromosom 1:



❖ Kromosom 2 [(1324)-(512)-(43)-(43)-(5)-(5)]

1. Iterasi 1 [(1324)-(512)-(43)-(43)-(5)-(5)]
2. Iterasi 2 [(324)-(512)-(43)-(43)-(0)-(5)]
3. Iterasi 3 [(24)-(512)-(43)-(43)-(0)-(0)]
4. Iterasi 4 [(4)-(12)-(43)-(43)-(0)-(0)]
5. Iterasi 5 [(0)-(2)-(43)-(43)-(0)-(0)]
6. Iterasi 6 [(0)-(0)-(3)-(43)-(0)-(0)]
7. Iterasi 7 [(0)-(0)-(0)-(3)-(0)-(0)]

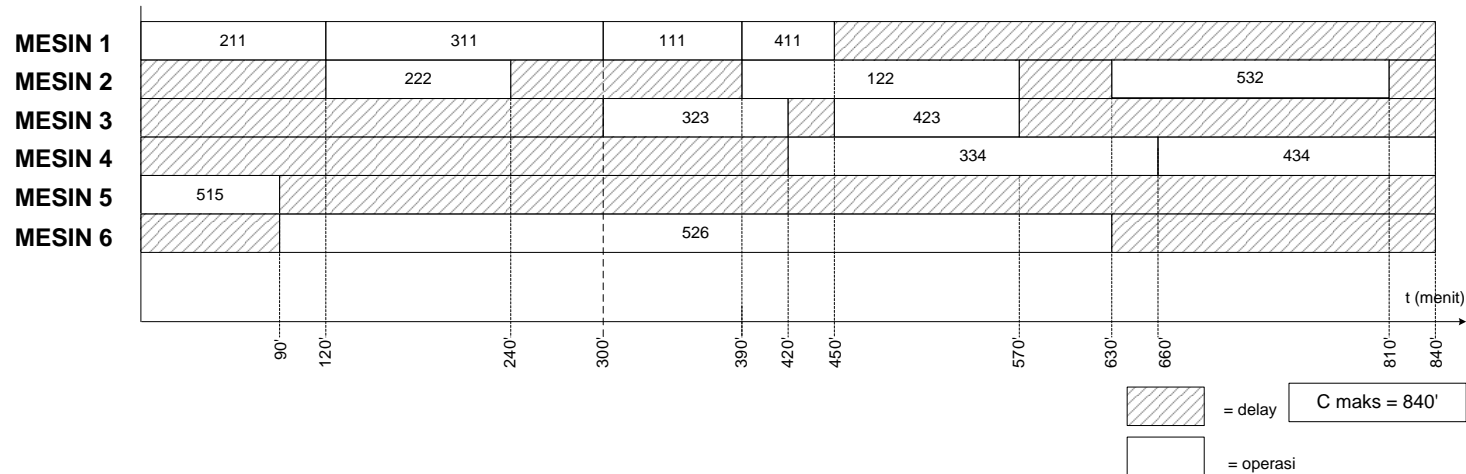
Berikut ialah *Gantt Chart* dari proses *decoding* kromosom 2:



❖ Kromosom 3 [(2314)-(152)-(43)-(34)-(5)-(5)]

1. Iterasi 1 [(2314)-(152)-(43)-(34)-(5)-(5)]
2. Iterasi 2 [(314)-(152)-(43)-(34)-()-(-5)]
3. Iterasi 3 [(14)-(152)-(43)-(34)-()-(-)]
4. Iterasi 4 [(4)-(152)-(43)-(34)-()-(-)]
5. Iterasi 5 [()-(-52)-(43)-(34)-()-(-)]
6. Iterasi 6 [()-(-2)-(-3)-(34)-()-(-)]
7. Iterasi 7 [()-()-(-)-(-34)-()-(-)]
8. Iterasi 8 [()-()-(-)-(-4)-()-(-)]

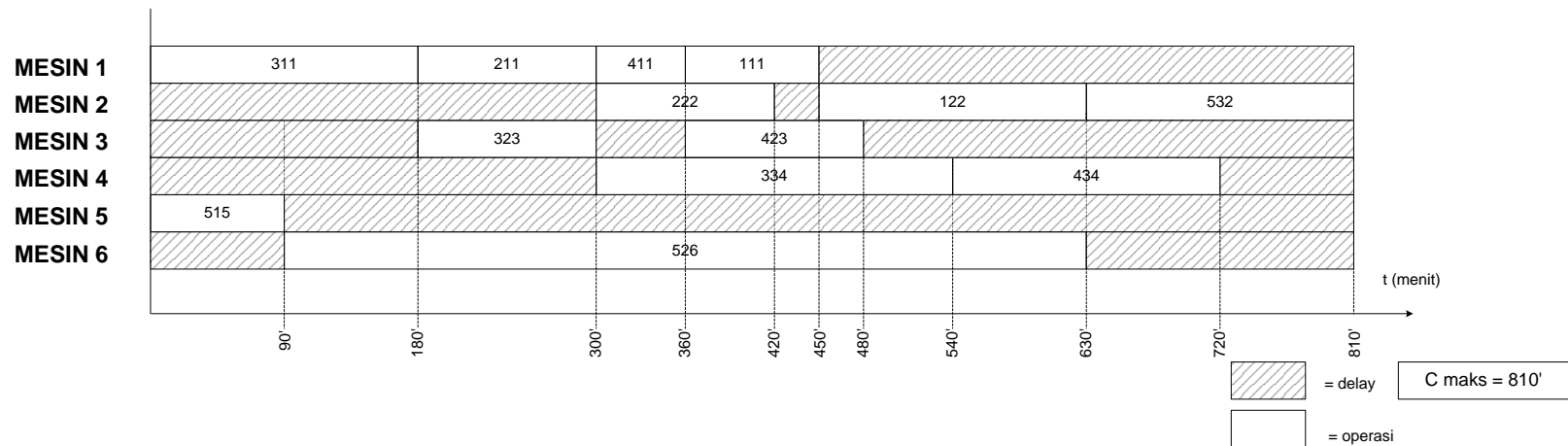
Berikut ialah *Gantt Chart* dari proses *decoding* kromosom 3:



❖ Kromosom 4 [(3241)-(512)-(34)-(34)-(5)-(5)]

1. Iterasi 1 [(3241)-(512)-(34)-(34)-(5)-(5)]
2. Iterasi 2 [(241)-(512)-(34)-(34)-(0)-(5)]
3. Iterasi 3 [(41)-(512)-(4)-(34)-(0)-(5)]
4. Iterasi 4 [(1)-(12)-(4)-(4)-(0)-(0)]
5. Iterasi 5 [(0)-(12)-(0)-(4)-(0)-(0)]
6. Iterasi 6 [(0)-(2)-(0)-(0)-(0)-(0)]

Berikut ialah *Gantt Chart* dari proses *decoding* kromosom 4:



Evaluasi kromosom dilakukan terhadap keempat kromosom berdasarkan nilai *fitness*nya. Nilai *fitness* didapat dari $1/C_{\text{maks}}$ ($1/\text{nilai makespan maksimum}$).

Berikut ialah tabel rangkuman nilai *fitness* dari keempat kromosom tersebut :

Tabel Rangkuman Nilai *Fitness*

No.	Kromosom	<i>makespan</i>	nilai <i>fitness</i>
1	[(1234)-(125)-(34)-(34)-(5)-(5)]	930'	0.00108
2	[(1324)-(512)-(43)-(43)-(5)-(5)]	990'	0.00101
3	[(2314)-(152)-(43)-(34)-(5)-(5)]	840'	0.00119
4	[(3241)-(512)-(34)-(34)-(5)-(5)]	810'	0.00123

➤ **Crossover (*Partially Mapped Crossover*)**

Setelah mengalami *decoding*, maka semua kromosom menjadi anggota untuk menjadi *parent crossover*. Probabilitas *crossover* (P_c) ditentukan sebesar 0.95. Sebelum mengalami proses *crossover*, setiap kromosom akan memiliki nilai probabilitasnya sendiri yang di dapat dari pembangkitan bilangan *random* (dari 0 s/d 1).

Tabel . Pembangkitan bilangan *random Crossover*

No.	Kromosom	bil. <i>Random</i>	Pasangan
1	[(1234)-(125)-(34)-(34)-(5)-(5)]	0.47	1
2	[(1324)-(512)-(43)-(43)-(5)-(5)]	0.08	2
3	[(2314)-(152)-(43)-(34)-(5)-(5)]	0.70	3
4	[(3241)-(512)-(34)-(34)-(5)-(5)]	0.99	*

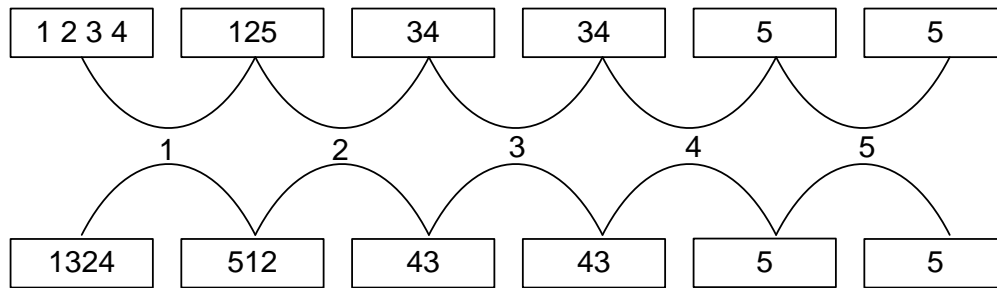
Berdasarkan pembangkitan bilangan *random* pada setiap kromosom, maka kromosom yang terpilih menjadi *parent crossover* ialah kromosom yang memiliki bilangan *random* lebih kecil dari P_c (0.95), yaitu kromosom nomor 1,2 dan 3. Kromosom nomor 4 tidak masuk menjadi *parent crossover* karena memiliki bilangan *random* lebih besar dari probabilitas *crossover*. Setelah pemilihan jumlah *parent crossover*, maka kromosom akan dipasangkan secara *random*. Karena jumlah *parent* kromosom berjumlah ganjil, maka kromosom 3 tidak memiliki pasangan untuk menjadi *parent crossover*. Maka kromosom 3 tidak ikut mengalami proses *crossover*.

Proses selanjutnya ialah melakukan proses penyilangan antar subkromoson dari dua *parent crossover* menggunakan metode *Partially Mapped Crossover*. Metode ini memiliki aturan-aturan sebagai berikut :

1. Menentukan *crossing site*

Penentuan *crossing site* ini bertujuan untuk menentukan posisi di sub kromosom mana yang akan mengalami pertukaran. Proses ini dilakukan dengan membangkitkan bilangan *random* antara 1 sampai dengan $m-1$ (m = jumlah sub kromosom).

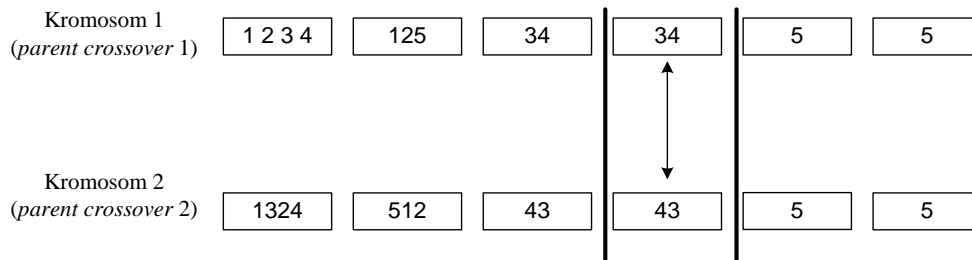
No.	Kromosom	<i>crossing site</i>
1	[(1234)-(125)-(34)-(34)-(5)-(5)]	3 dan 4
2	[(1324)-(512)-(43)-(43)-(5)-(5)]	



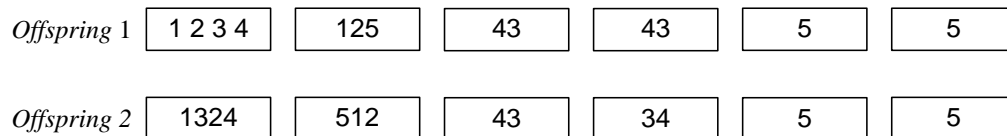
Gambar . *crossing site*

Dari gambar di atas dapat terlihat bahwa ada 5 posisi untuk melakukan *crossing site*. Kedua kromosom di atas memiliki nilai *crossing site* 3 dan 4, yang berarti pertukaran akan dilakukan pada posisi antara sub kromosom 3-4 dan 4-5.

- Melakukan pertukaran antara sub kromosom yang terletak diantara *crossing site* terhadap kedua *parent crossover*.



- Setelah melalui proses pertukaran tersebut, maka terbentuklah kromosom baru yang selanjutnya akan disebut *offspring crossover*.

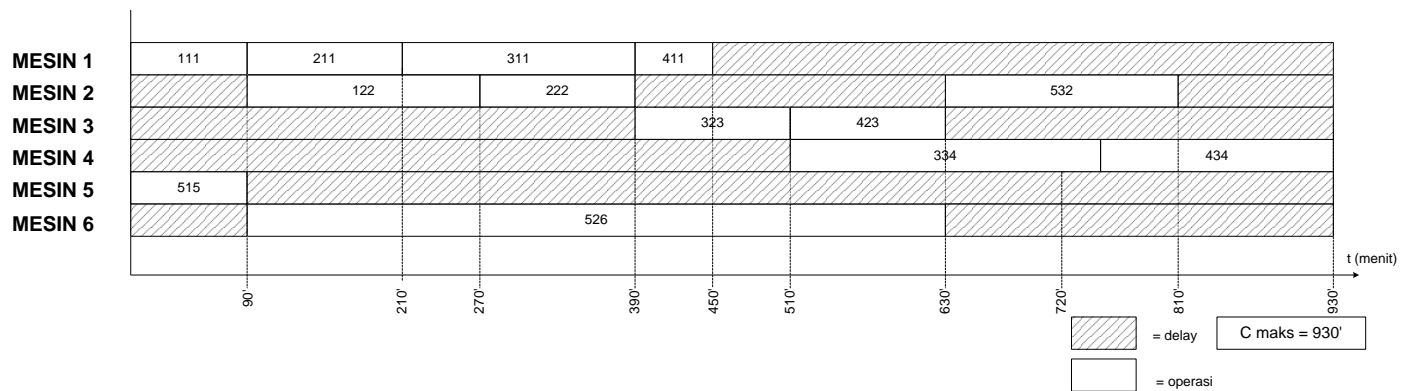


Kedua kromosom baru (*offspring*) ini akan mengalami proses *decoding* juga untuk diketahui nilai *fitness*nya masing-masing.

Proses *decoding* untuk *offspring crossover* 1 :

- ❖ [(1234)-(125)-(34)-(43)-(5)-(5)]
- 1. Iterasi 1 [(1234)-(125)-(34)-(43)-(5)-(5)]
- 2. Iterasi 2 [(234)-(125)-(34)-(43)-()-(5)]
- 3. Iterasi 3 [(34)-(25)-(34)-(43)-()-(0)]
- 4. Iterasi 4 [(4)-(5)-(34)-(43)-()-(0)]
- 5. Iterasi 5 [()-(5)-(4)-(43)-()-(0)]
- 6. Iterasi 6 [()-(5)-()-(43)-()-(0)]
- 7. Iterasi 7 [()-(5)-()-(3)-()-(0)]
- 8. Iterasi 8 [()-(5)-()-()-()-(0)]

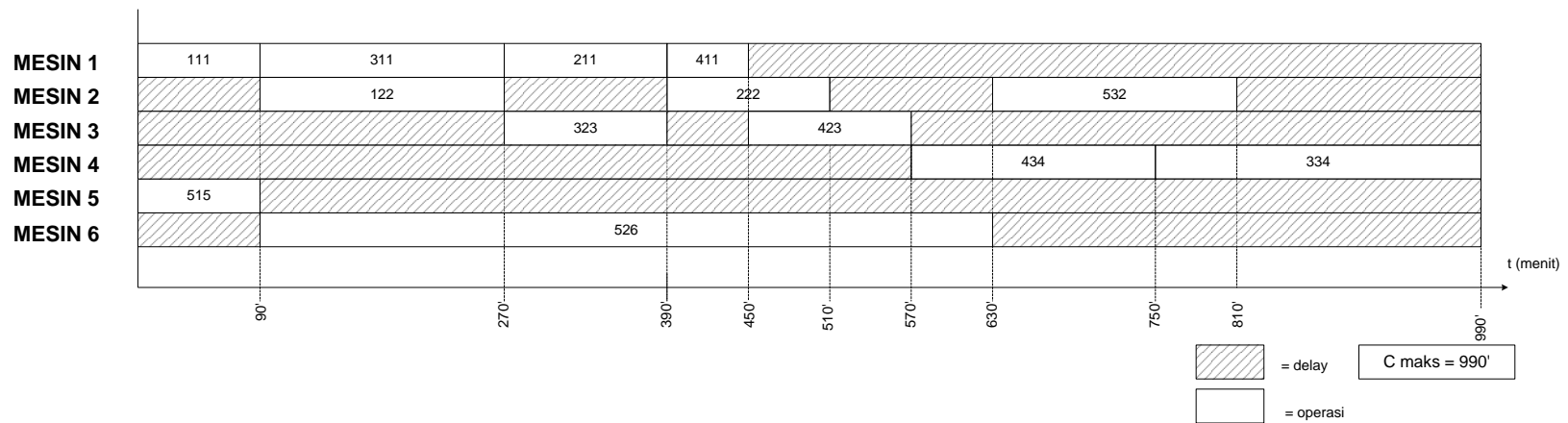
Berikut ialah *Gantt Chart* dari proses *decoding offspring crossover* 1:



Proses *decoding* untuk *offspring crossover 2* :

- ❖ [(1324)-(512)-(43)-(34)-(5)-(5)]
- 1. Iterasi 1 [(324)-(512)-(43)-(34)-(0)-(5)]
- 2. Iterasi 2 [(24)-(512)-(43)-(34)-(0)-(0)]
- 3. Iterasi 3 [(4)-(12)-(43)-(34)-(0)-(0)]
- 4. Iterasi 4 [(0)-(2)-(43)-(34)-(0)-(0)]
- 5. Iterasi 5 [(0)-(0)-(3)-(34)-(0)-(0)]
- 6. Iterasi 6 [(0)-(0)-(0)-(34)-(0)-(0)]
- 7. Iterasi 7 [(0)-(0)-(0)-(4)-(0)-(0)]

Berikut ialah *Gantt Chart* dari proses *decoding offspring crossover 2*:



➤ **Mutasi (*Order Based Mutation*)**

Proses selanjutnya ialah mutasi, yang bertujuan untuk menukar posisi gen yang ada di dalam sub kromosom. Kromosom yang akan mengalami mutasi ialah berjumlah 6 buah, yaitu 4 buah berasal dari populasi awal dan 2 buah dari *offspring crossover*.

Tabel. Kromosom yang akan mengalami Mutasi

No.	Kromosom
1	[(1234)-(125)-(34)-(34)-(5)-(5)]
2	[(1324)-(512)-(43)-(43)-(5)-(5)]
3	[(2314)-(152)-(43)-(34)-(5)-(5)]
4	[(3241)-(512)-(34)-(34)-(5)-(5)]
5	[(1234)-(125)-(34)-(34)-(5)-(5)]
6	[(1324)-(512)-(43)-(43)-(5)-(5)]

Prosedur *Order Based Mutation* ini ialah dengan membangkitkan bilangan *random* dari 0 s/d 1 kepada setiap gen yang ada di dalam setiap kromosom. Gen yang memiliki bilangan *random* lebih kecil dari P_m (0.01), maka kromosom tersebut akan mengalami mutasi. Apabila semua gen dalam satu kromosom memiliki bilangan *random* lebih besar dari P_m maka kromosom tersebut tidak akan mengalami mutasi. Proses mutasinya yaitu gen yang bermutasi akan ditukarkan dengan gen yang ada di sebelah kanannya. Apabila tidak terdapat gen di sebelah kanannya, maka gen yang terpilih akan ditukarkan dengan gen yang memiliki posisi di awal dalam sub kromosom tersebut.

Tabel Pembangkitan bilangan *random* untuk tiap-tiap gen di dalam kromosom

Error! Not a valid link.

no	kromosom																									
1	1	2	3	4	1	2	5	3	4	3	4	5	5	0.743	0.050	0.693	0.853	0.325	0.654	0.188	0.624	0.769	0.135	0.517	0.152	0.602
2	1	3	2	4	5	1	2	4	3	4	3	5	5	0.982	0.658	0.029	0.542	0.538	0.078	0.350	0.808	0.647	0.746	0.708	0.686	0.842
3	2	3	1	4	1	5	2	4	3	3	4	5	5	0.428	0.613	0.295	0.206	0.985	0.836	0.917	0.061	0.977	0.534	0.271	0.750	0.873
4	3	2	4	1	5	1	2	3	4	3	4	5	5	0.8	0.577	0.116	0.643	0.228	0.552	0.505	0.754	0.912	0.416	0.781	0.892	0.287
5	1	2	3	4	1	2	5	3	4	4	3	5	5	0.707	0.64	0.113	0.696	0.357	0.279	0.278	0.931	0.059	0.666	0.979	0.712	0.158
6	1	3	2	4	5	1	2	4	3	3	4	5	5	0.269	0.312	0.255	0.498	0.058	0.869	0.282	0.421	0.501	0.562	0.408	0.783	0.623

Dari tabel di atas dapat terlihat bahwa tidak ada kromosom yang akan mengalami mutasi karena tidak ada gen yang memiliki $P_m < 0.01$.

Tahap selanjutnya ialah pembentukan ruang sampling yang terdiri atas kromosom-kromosom populasi awal ditambah *offspring* dari *crossover* dan mutasi. Hal ini sesuai dengan metode seleksi *enlarge sampling space* yang mengikutkan *offspring* dan *parents* untuk menjadi bagian dalam seleksi populasi awal di generasi selanjutnya.

Tabel . Ruang sampling

No.	Kromosom
1	[(1234)-(125)-(34)-(34)-(5)-(5)]
2	[(1324)-(512)-(43)-(43)-(5)-(5)]
3	[(2314)-(152)-(43)-(34)-(5)-(5)]
4	[(3241)-(512)-(34)-(34)-(5)-(5)]
5	[(1234)-(125)-(34)-(34)-(5)-(5)]
6	[(1324)-(512)-(43)-(43)-(5)-(5)]

➤ **Evaluasi Kromosom (seleksi)**

Proses selanjutnya ialah pemilihan kromosom terbaik berdasarkan evaluasi nilai *fitness* untuk semua kromosom dari populasi baru. Kromosom terbaik ialah yang memiliki nilai *fitness* terbesar.

Tabel Evaluasi kromosom

No.	Kromosom	<i>makespan</i>	nilai <i>fitness</i>
1	[(1234)-(125)-(34)-(34)-(5)-(5)]	930'	0.00108
2	[(1324)-(512)-(43)-(43)-(5)-(5)]	990'	0.00101
3	[(2314)-(152)-(43)-(34)-(5)-(5)]	840'	0.00119
4	[(3241)-(512)-(34)-(34)-(5)-(5)]	810'	0.00123
5	[(1234)-(125)-(34)-(34)-(5)-(5)]	930'	0.00108
6	[(1324)-(512)-(43)-(43)-(5)-(5)]	990'	0.00101

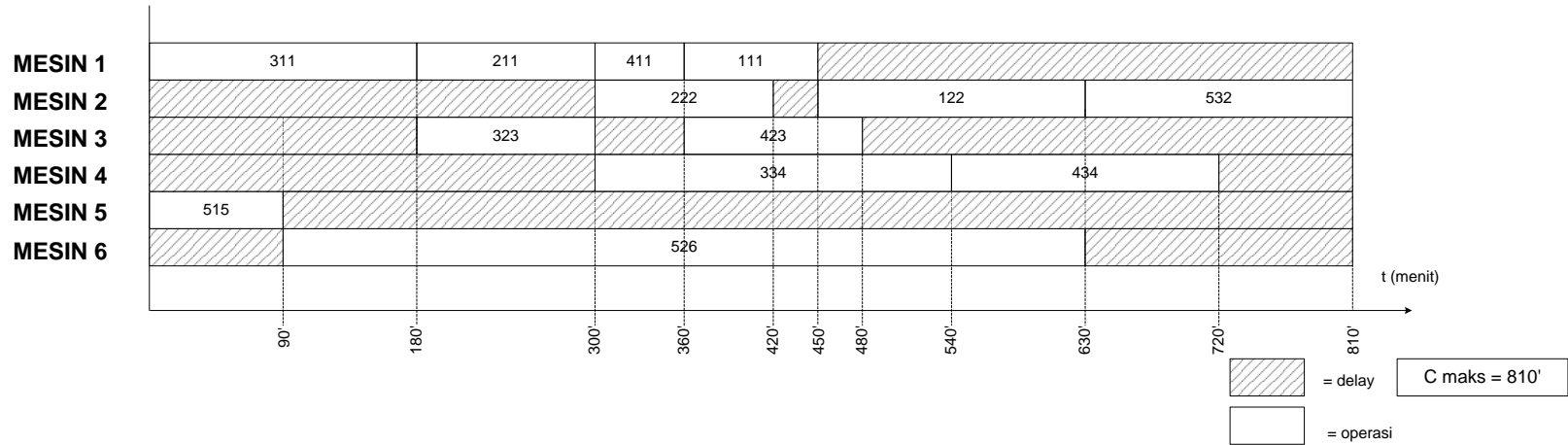
Untuk memilih kromosom yang dapat digunakan untuk menjadi populasi awal di generasi selanjutnya ialah dengan cara memilih empat buah kromosom, sesuai

jumlah ukuran populasi, secara *random* dari kromosom yang memiliki nilai *fitness* terbesar.

Tabel . Kromosom terbaik (populasi baru)

No.	Kromosom	<i>makespan</i>	nilai <i>fitness</i>
4	[(3241)-(512)-(34)-(34)-(5)-(5)]	810'	0.00123
3	[(2314)-(152)-(43)-(34)-(5)-(5)]	840'	0.00119
1	[(1234)-(125)-(34)-(34)-(5)-(5)]	930'	0.00108
5	[(1234)-(125)-(34)-(34)-(5)-(5)]	930'	0.00108

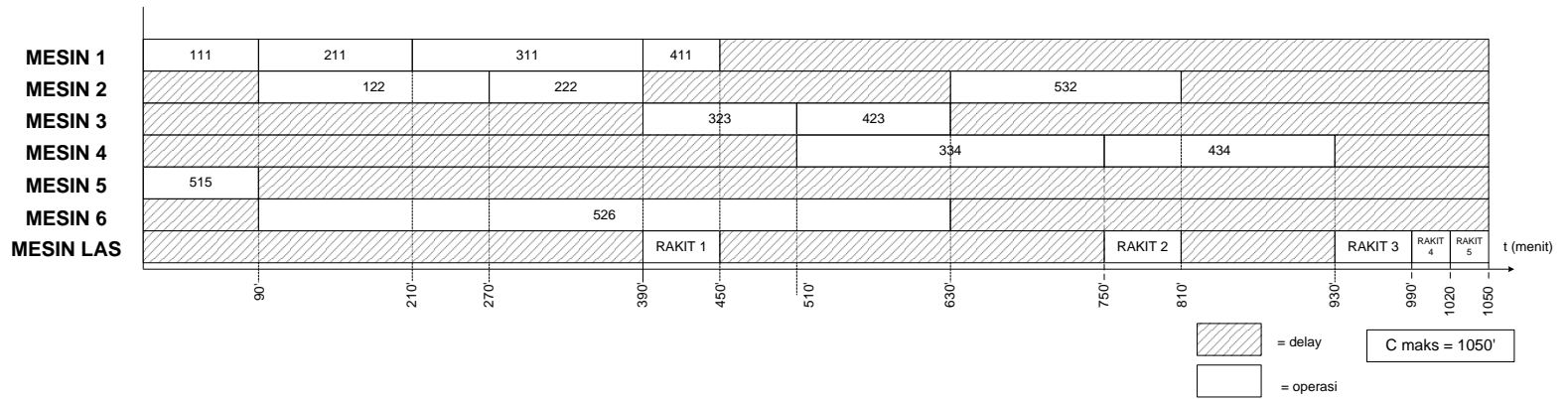
Dari sekumpulan solusi di atas, diambil salah satu solusi yang memiliki nilai fitness yang terbesar secara random. Pada Tugas Akhir ini solusi yang diambil ialah kromosom 1 yaitu : [(3241)-(512)-(34)-(34)-(5)-(5)], bila kromosom tersebut diterjemahkan ke dalam *Gantt Chart* ialah sebagai berikut:



Gantt Chart Algoritma Genetik

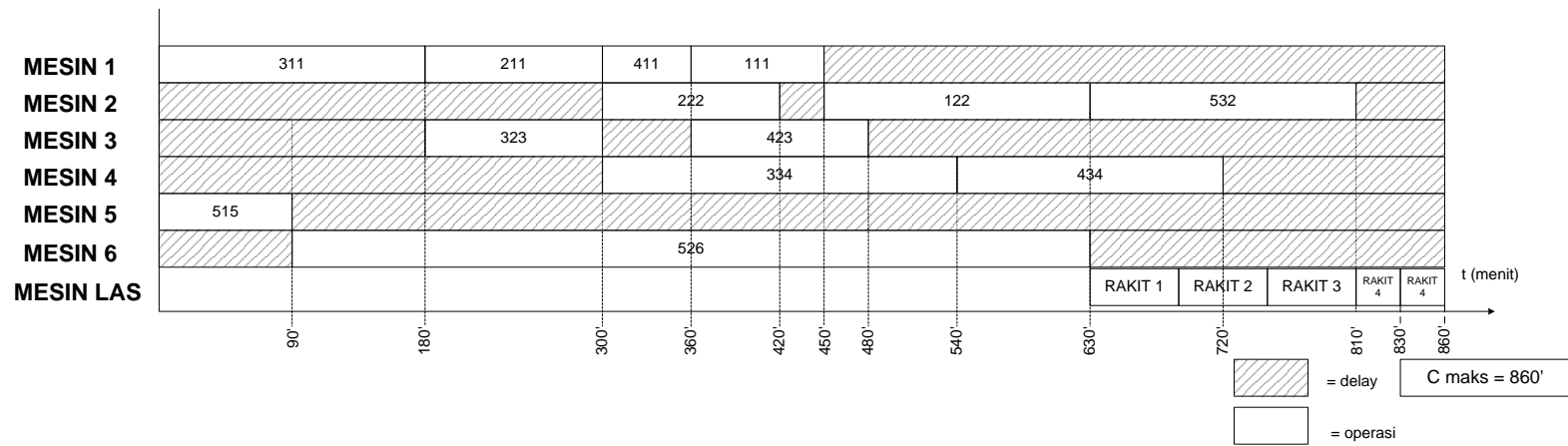
Setelah menghitung *makespan*, langkah selanjutnya ialah menghitung *completion time* keseluruhan dengan cara menambahkan waktu perakitan secara manual (langkah perakitan tidak dibahas dalam tugas akhir ini dan masuk kedalam pembatasan masalah).

❖ Metode Perusahaan



Gantt Chart Metode Perusahaan Setelah Perakitan

❖ Algoritma Genetik



Gantt Chart Algoritma Genetik Setelah Perakitan

Dari *Gantt Chart* di atas, dapat ditarik kesimpulan bahwa waktu penyelesaian total atau *completion time* dari komponen dudukan rol mesin *Bamboo Crush* yang dihasilkan dari kedua metode ialah sebagai berikut:

Tabel Perhitungan *Completion Time* Kedua Metode

Metode Perusahaan	Algoritma Genetik
1050 menit	930 menit

❖ **Penjadwalan dengan Metode Penjadwalan Aktif**

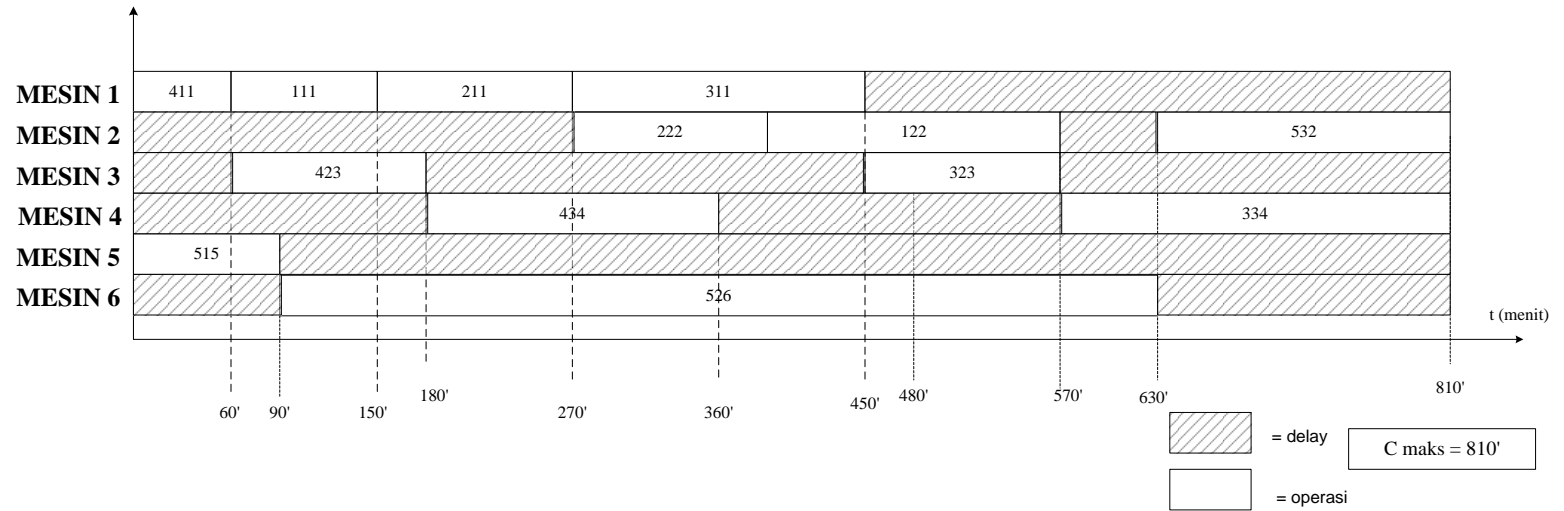
Aturan :1. SPT

2. FCFS

3. Random

Perhitungan Penjadwalan Aktif													
Tahap	Mesin						St	Cj	tij	rj	r*	m*	PSt
	1	2	3	4	5	6							
0	0	0	0	0	0	0	111	0	90	90	60	1	411 111 211 311
							211	0	120	120			
							311	0	180	180			
							411	0	60	60			
							515	0	90	90			
1	450	0	0	0	0	0	122	150	180	330	90	5	515
							222	270	120	390			
							323	450	120	570			
							423	60	120	180			
							515	0	90	90			
2	450	0	0	0	90	0	122	150	180	330	180	3	423
							222	270	120	390			
							323	450	120	570			
							423	60	120	180			
							526	90	540	630			
3	450	0	180	0	90	0	122	150	180	330	330	2	222 122
							222	270	120	390			
							323	450	120	570			
							434	180	180	360			
							526	90	540	630			
4	450	570	180	0	90	0	323	450	120	570	360	4	434
							434	180	180	360			
							526	90	540	630			
5	450	570	180	360	90	0	323	450	120	570	570	3	323
							526	90	540	630			
6	450	570	180	360	90	0	334	570	240	810	630	6	526
							526	90	540	630			
7	450	570	570	360	90	630	334	570	240	810	810	2	532
							532	630	180	810			
8	450	810	570	360	90	630	334	570	240	810	810	4	334

Gantt Chart penjadwalan aktif



❖ **Penjadwalan dengan Metode Penjadwalan Non Delay**

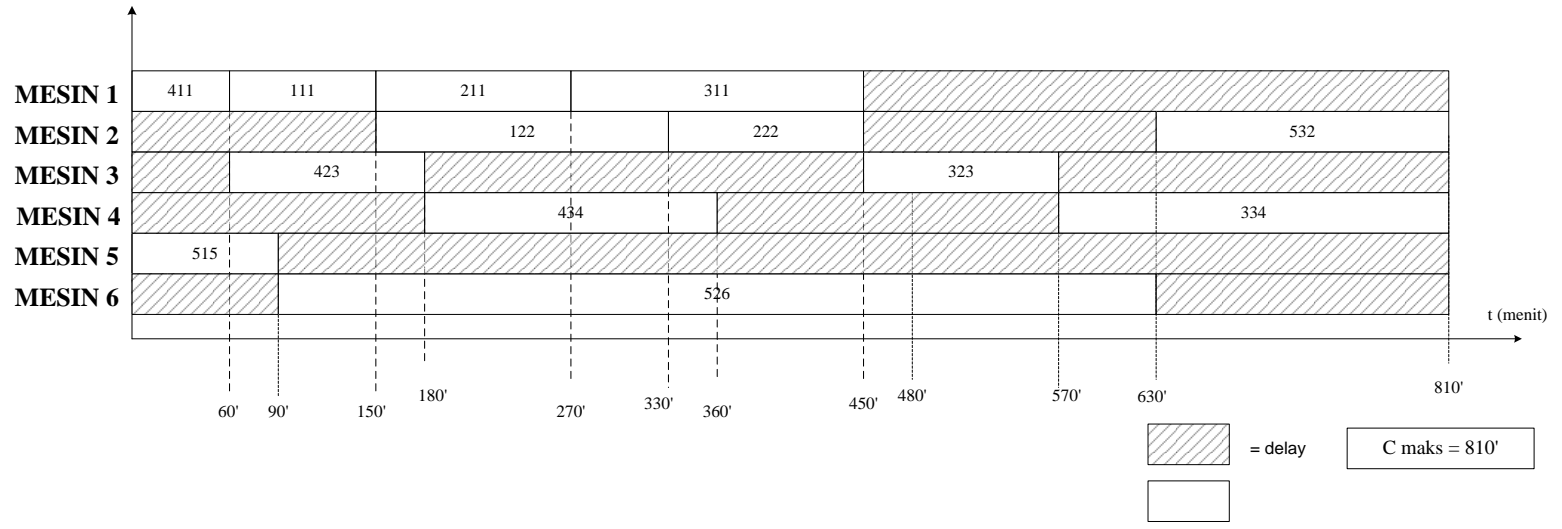
Aturan :1. SPT

2. FCFS

3. Random

Tahap	Mesin						St	Cj	tij	rj	c*	m*	PSt
	1	2	3	4	5	6							
0	0	0	0	0	0	0	111	0	90	90	0	1,5	411
							211	0	120	120			515
							311	0	180	180			111
							411	0	60	60			211
							515	0	90	90			311
1	450	0	0	0	90	0	122	150	180	330	60	3	423
							222	270	120	390			
							323	450	120	570			
							423	60	120	180			
							526	90	540	630			
2	450	0	180	0	90	0	122	150	180	330	90	6	526
							222	270	120	390			
							323	450	120	570			
							434	180	180	360			
							526	90	540	630			
3	450	0	180	0	90	630	122	150	180	330	150	2	122
							222	270	120	390			
							323	450	120	570			
							434	180	180	360			
							532	630	180	810			
4	450	330	180	0	90	630	222	330	120	450	180	4	434
							323	450	120	570			
							434	180	180	360			
							532	630	180	810			
5	450	330	180	360	90	630	222	330	120	450	330	2	222
							323	450	120	570			
							532	630	180	810			
6	450	450	180	360	90	630	323	450	120	570	450	3	323
							532	630	180	810			
7	450	450	570	360	90	630	334	570	240	810	570	4	334
							532	630	180	810			
8	450	450	570	810	90	630	532	630	180	810	630	2	532

Gantt Chart penjadwalan non delay



■ Kasus 2

Tabel *Routing* Proses

Pesanan	Operasi (mesin)		
	1	2	3
1	3	4	
2	3	4	5
3	2	4	5
4	1	4	

Tabel *Routing* Waktu (menit)

Pesanan	Operasi		
	1	2	3
1	510.78	17.58	
2	1694.82	58.98	197.46
3	1363.26	22.14	196.98
4	1570.98	298.14	

Penjadwalan Aktif

Stage	Mesin					S _t	C _j	t _j	r _j	r*	m*	P _{st}
	1	2	3	4	5							
0	0	0	0	0	0	1.1.3	0	510,78	510,78	510,78	3	1.1.3 2.1.3
						2.1.3	0	1694,8	1694,82			
						3.1.2	0	1363,3	1363,26			
						4.1.1	0	1571	1570,98			
1	0	0	2205,6	0	0	1.2.4	510,78	528,36	1039,14	1039,14	4	1.2.4
						2.2.4	2205,6	2264,6	4470,18			
						3.1.2	0	1363,3	1363,26			
						4.1.1	0	1571	1570,98			
2	0	0	2205,6	528,36	0	2.2.4	2205,6	58,98	2264,58	1363,26	2	3.1.2
						3.1.2	0	1363,3	1363,26			
3	0	1363,26	2205,6	528,36	0	2.2.4	2205,6	58,98	2264,58	1385,4	4	3.2.4
						3.2.4	1363,26	22,14	1385,4			
						4.1.1	0	1571	1570,98			
4	0	1363,26	2205,6	528,36	0	2.2.4	2205,6	58,98	2264,58	1582,38	1	4.1.1
						3.3.5	1385,4	196,98	1582,38			
						4.1.1	1570,98	1571	3141,96			
5	1570,98	1363,26	2205,6	528,36	0	2.2.4	2205,6	58,98	2264,58	1582,38	5	3.3.5
						3.3.5	1385,4	196,98	1582,38			
						4.2.4	1570,98	298,14	1869,12			
6	1570,98	1363,26	2205,6	528,36	1582,38	2.2.4	2205,6	58,98	2264,58	1869,12	4	4.2.4
						4.2.4	1570,98	298,14	1869,12			
7	1570,98	1363,26	2205,6	528,36	1582,38	2.2.4	2205,6	58,98	2264,58	2264,58	4	2.2.4
8	1570,98	1363,26	2205,6	528,36	1582,38	2.3.5	2264,58	197,46	2462,04	2462,04	5	2.3.5

Kesimpulan :

Makespan = 2462.04 menit

Penjadwalan *Non Delay*

Stage	Mesin					S _i	C _j	t _j	r _j	c*	m*	P _{st}
	1	2	3	4	5							
0	0	0	0	0	0	1.1.3	0	510,78	510,78	0	1 2 3	1.1.3
						2.1.3	0	1694,82	1694,82			2.1.3
						3.1.2	0	1363,26	1363,26			3.1.2
						4.1.1	0	1570,98	1570,98			4.1.1
1	1570,98	1363,26	2205,6	0	0	1.2.4	510,78	17,58	528,36	510,78	4	1.2.4
						2.2.4	2205,6	58,98	2264,58			
						3.2.4	1363,26	22,14	1385,4			
						4.2.4	1570,98	298,14	1869,12			
2	1570,98	1363,26	2205,6	528,36	0	2.2.4	2205,6	58,98	2264,58	1363,26	4	3.2.4
						3.2.4	1363,26	22,14	1385,4			
						4.2.4	1570,98	298,14	1869,12			
3	1570,98	1363,26	2205,6	1385,4	0	2.2.4	2205,6	58,98	2264,58	1385,4	5	3.3.5
						3.3.5	1385,4	196,98	1582,38			
						4.2.4	1570,98	298,14	1869,12			
4	1570,98	1363,26	2205,6	1385,4	1582,38	2.2.4	2205,6	58,98	2264,58	1582,38	4	4.2.4
						4.2.4	1582,38	298,14	1880,52			
5	1570,98	1363,26	2205,6	1880,52	1582,38	2.2.4	2205,6	58,98	2264,58	2205,6	4	2.2.4
6	1570,98	1363,26	2205,6	2264,58	1582,38	2.3.5	2264,58	197,46	2462,04	2264,58	5	2.3.5

Kesimpulan :

Makespan = 2462.04 menit

■ Kasus 3

Tabel Matriks *Routing*

Job	Operasi Ke-			
	1	2	3	4
1	M-4	M-6		
2	M-1	M-6		
3	M-4	M-6		
4	M-4	M-2		
5	M-1	M-5	M-3	
6	M-4	M-2		
7	M-4	M-2		
8	M-1	M-5		
9	M-1	M-5	M-3	
10	M-1	M-2		
11	M-1	M-2	M-3	M-3

Tabel Matriks Waktu

Job	Operasi Ke-			
	1	2	3	4
1	72	95		
2	120	190		
3	30	36		
4	10	20		
5	20	80	80	
6	20	20		
7	20	20		
8	30	30		
9	30	40	120	
10	20	20		
11	20	20	240	120

Penjadwalan Aktif

Stage	Mesin						S _t	C _j	t _j	r _j	r*	m*	P _{st}
	1	2	3	4	5	6							
0	0	0	0	0	0	0	1.1.4	0	72	72	10	4	4.1.4 6.1.4 7.1.4 3.1.4 1.1.4
							2.1.1	0	120	120			
							3.1.4	0	30	30			
							4.1.4	0	10	10			
							5.1.1	0	20	20			
							6.1.4	0	20	20			
							7.1.4	0	20	20			
							8.1.1	0	30	30			
							9.1.1	0	30	30			
							10.1.1	0	20	20			
							11.1.1	0	20	20			
1	0	0	0	152	0	0	1.2.6	152	95	247	20	1	10.1.1 11.1.1 5.1.1 8.1.1 9.1.1 2.1.1
							2.1.1	0	120	120			
							3.2.6	80	36	116			
							4.2.2	10	20	30			
							5.1.1	0	20	20			
							6.2.2	30	20	50			
							7.1.2	50	20	70			
							8.1.1	0	30	30			
							9.1.1	0	30	30			
							10.1.1	0	20	20			
							11.1.1	0	20	20			

Stage	Mesin						S _t	C _j	t _j	r _j	r*	m*	P _{st}
	1	2	3	4	5	6							
2	240	0	0	152	0	0	1.2.6	152	95	247	30	2	4.2.2 10.1.2 6.2.2
							2.2.6	240	190	430			
							3.2.6	80	36	116			
							4.2.2	10	20	30			
							5.2.5	60	80	140			
							6.2.2	30	20	50			
							7.1.2	50	20	70			
							8.2.5	90	30	120			
							9.2.5	120	40	160			
							10.1.2	20	20	40			
							11.2.2	40	20	60			
3	240	80	0	152	0	0	1.2.6	152	95	247	90	2	7.1.2 11.2.2
							2.2.6	240	190	430			
							3.2.6	80	36	116			
							5.2.5	60	80	140			
							7.1.2	70	20	90			
							8.2.5	90	30	120			
							9.2.5	120	40	160			
							11.2.2	70	20	90			
4	240	120	0	152	0	0	1.2.6	152	95	247	116	6	3.2.6
							2.2.6	240	190	430			
							3.2.6	80	36	116			
							5.2.5	60	80	140			
							8.2.5	90	30	120			
							9.2.5	120	40	160			
							11.3.3	110	240	350			
5	240	120	0	152	0	116	1.2.6	152	95	247	120	5	8.2.5 9.2.5
							2.2.6	240	190	430			
							5.2.5	60	80	140			
							8.2.5	90	30	120			
							9.2.5	120	40	160			
							11.3.3	110	240	350			
6	240	120	0	152	160	116	1.2.6	152	95	247	240	5	5.2.5
							2.2.6	240	190	430			
							5.2.5	160	80	240			
							9.3.3	160	120	280			
							11.3.3	110	240	350			

Stage	Mesin						S _t	C _j	t _j	r _j	r*	m*	P _{st}
	1	2	3	4	5	6							
7	240	120	0	152	240	116	1.2.6	152	95	247	247	6	1.2.6 2.2.6
							2.2.6	240	190	430			
							5.3.3	240	80	320			
							9.3.3	160	120	280			
							11.3.3	110	240	350			
8	240	120	0	152	240	437	5.3.3	240	80	320	280	3	5.3.3 9.3.3 11.3.3
							9.3.3	160	120	280			
							11.3.3	110	240	350			
9	240	120	0	680	240	437	11.4.3	680	120	800	800	3	11.4.3

Kesimpulan :

Makespan = 800 menit

Penjadwalan Non Delay

Stage	Mesin						S _i	C _j	t _j	r _j	c*	m*	P _{st}
	1	2	3	4	5	6							
0	0	0	0	0	0	0	1.1.4	0	72	72	0	4 1	4.1.4
							2.1.1	0	120	120			6.1.4
							3.1.4	0	30	30			7.1.4
							4.1.4	0	10	10			3.1.4
							5.1.1	0	20	20			1.1.4
							6.1.4	0	20	20			11.1.1
							7.1.4	0	20	20			10.1.1
							8.1.1	0	30	30			5.1.1
							9.1.1	0	30	30			8.1.1
							10.1.1	0	20	20			9.1.1
							11.1.1	0	20	20			2.1.1
1	250	0	0	152	0	0	1.2.6	152	95	247	10	2	4.2.2
							2.2.6	240	190	430			
							3.2.6	80	36	116			
							4.2.2	10	20	30			
							5.2.5	60	80	140			
							6.2.2	30	20	50			
							7.1.2	50	20	70			
							8.2.5	90	30	120			
							9.2.5	120	40	160			
							10.1.2	40	20	60			
							11.2.2	20	20	40			
2	250	30	0	152	0	0	1.2.6	152	95	247	30	2	11.2.2 6.2.2
							2.2.6	240	190	430			
							3.2.6	80	36	116			
							5.2.5	60	80	140			
							6.2.2	30	20	50			
							7.1.2	50	20	70			
							8.2.5	90	30	120			
							9.2.5	120	40	160			
							10.1.2	40	20	60			
							11.2.2	30	20	50			
	3	250	70	0	152	0	0	1.2.6	152	95			
							2.2.6	240	190	430			
							3.2.6	80	36	116			
							5.2.5	60	80	140			
							7.1.2	70	20	90			
							8.2.5	90	30	120			
							9.2.5	120	40	160			
							10.1.2	70	20	90			
							11.3.3	50	240	290			

Stage	Mesin						S _t	C _j	t _j	r _j	c*	m*	P _{st}
	1	2	3	4	5	6							
4	250	70	290	152	0	0	1.2.6	152	95	247	60	5	5.2.5
							2.2.6	240	190	430			
							3.2.6	80	36	116			
							5.2.5	60	80	140			
							7.1.2	70	20	90			
							8.2.5	90	30	120			
							9.2.5	120	40	160			
							10.1.2	70	20	90			
							11.4.3	290	120	410			
5	250	70	290	152	140	0	1.2.6	152	95	247	70	2	7.1.2 10.1.2
							2.2.6	240	190	430			
							3.2.6	80	36	116			
							5.3.3	290	80	370			
							7.1.2	70	20	90			
							8.2.5	90	30	120			
							9.2.5	120	40	160			
							10.1.2	70	20	90			
							11.4.3	290	120	410			
6	250	110	290	152	140	0	1.2.6	152	95	247	80	6	3.2.6
							2.2.6	240	190	430			
							3.2.6	80	36	116			
							5.3.3	290	80	370			
							8.2.5	140	30	170			
							9.2.5	120	40	160			
7	250	110	290	152	140	116	1.2.6	152	95	247	140	5	8.2.5 9.2.5
							2.2.6	240	190	430			
							5.3.3	290	80	370			
							8.2.5	140	30	170			
							9.2.5	140	40	180			
8	250	110	290	152	210	116	1.2.6	152	95	247	240	6	1.2.6 2.2.6
							2.2.6	240	190	430			
							5.3.3	290	80	370			
							9.3.3	290	120	410			
							11.4.3	290	120	410			
9	250	110	610	152	210	437	5.3.3	290	80	370	290	3	5.3.3 9.3.3 11.4.3
							9.3.3	290	120	410			
							11.4.3	290	120	410			

Kesimpulan :

Makespan = 610 menit

Rangkuman nilai *makespan* dari ketiga kasus yang di ujikan.

Kasus	Permutasi	Algoritma Genetik	Jadwal Aktif	Jadwal Non Delay
Kasus 1	576	810	810	810
Kasus 2	96	2223.18	2462.04	2462.04
Kasus 3	8957952000	600	800	610

Berdasarkan tabel di atas dapat terbukti bahwa dari 3 kasus yang diuji, Algoritma Genetik mampu menghasilkan *makespan* terkecil sebanyak 2 kasus jika dibandingkan dengan metode penjadwalan Aktif dan *Non Delay*. Maka Algoritma Genetika terbukti berpeluang lebih besar untuk mampu menghasilkan *makespan* yang kecil.

LAMPIRAN 2

LIST PROGRAM JOB SHOP GA

```
unit Proses;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, ExtCtrls;
```

```
type
```

```
TFormProses = class(TForm)
```

```
Panel2: TPanel;
```

```
Label1: TLabel;
```

```
Label2: TLabel;
```

```
Label3: TLabel;
```

```
Label4: TLabel;
```

```
Label5: TLabel;
```

```
TGenerasi: TEdit;
```

```
TPopulasi: TEdit;
```

```
TPc: TEdit;
```

```
TPm: TEdit;
```

```
TParameter: TEdit;
```

```
PButton: TButton;
```

```
Panel1: TPanel;
```

```
Label6: TLabel;
```

```
Label7: TLabel;
```

```
Label10: TLabel;
```

```
TMesin: TEdit;
```

```
TJob: TEdit;
```

```
TCase: TEdit;
```

```
CButton: TButton;
```

```
Label8: TLabel;
```

```
LCase: TLabel;
```

```
BProses: TButton;
```

```
BExit: TButton;
```

```
Splitter1: TSplitter;
```

```
Panel3: TPanel;
```

```
Label9: TLabel;
```

```
Label11: TLabel;
```

```
TMakespan: TEdit;
```

```
TGenBest: TEdit;
```

```
Shape1: TShape;
```

```
Label12: TLabel;
```

```
Shape2: TShape;
```

```
Shape3: TShape;
```

```
Label13: TLabel;
```

```
Label14: TLabel;
```

```
No1: TEdit;
```

```
No2: TEdit;
```

```
No3: TEdit;
```

```
No4: TEdit;
```

```

Job1: TEdit;
Job2: TEdit;
Job3: TEdit;
Job4: TEdit;
Operasi1: TEdit;
Operasi2: TEdit;
Operasi3: TEdit;
Operasi4: TEdit;
SB: TScrollBar;
CB: TComboBox;
Label15: TLabel;
Shape4: TShape;
Shape5: TShape;
Shape6: TShape;
Label16: TLabel;
Label17: TLabel;
Label18: TLabel;
Waktu1: TEdit;
Waktu2: TEdit;
Waktu3: TEdit;
Waktu4: TEdit;
Mulai1: TEdit;
Mulai2: TEdit;
Mulai3: TEdit;
Mulai4: TEdit;
Akhir1: TEdit;
Akhir2: TEdit;
Akhir3: TEdit;
Akhir4: TEdit;
Permutasi: TEdit;
Label19: TLabel;
Label20: TLabel;
procedure TCaseChange(Sender: TObject);
procedure TParameterChange(Sender: TObject);
procedure CButtonClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure TGenerasiChange(Sender: TObject);
procedure TPopulasiChange(Sender: TObject);
procedure TPcChange(Sender: TObject);
procedure TPmChange(Sender: TObject);
procedure PButtonClick(Sender: TObject);
procedure BExitClick(Sender: TObject);
procedure BProsesClick(Sender: TObject);
procedure CBChange(Sender: TObject);
procedure SBChange(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

```

Type

```
RJobGenetic = Record
  Job : String[30];
  Operasi : String[30];
  Urutan : Integer;
  Waktu : Real;
  WaktuMulai : Real;
  WaktuAkhir : Real;
  ProbMut : Real;
  Assign : Real;
End;
RMesinGenetic = Record
  Nama : String[30];
  AssignJob : Integer;
  WaktuTotal : Real;
  ProbCross : Real;
  JumJob : Integer;
  JobOperasi : Array[1..250] of RJobGenetic;
End;
RKasusGenetic = Record
  MakesPan : Real;
  Mesin : Array[1..100] of RMesinGenetic;
End;

RParameter = Record
  Populasi : Integer;
  Generasi : Integer;
  Pc : Real;
  Pm : Real;
End;
ROperasi = Record
  Nama : String[30];
  Waktu : Real;
  WaktuMulai : Real;
  WaktuAkhir : Real;
  Mesin : String[30];
  Assign : Integer;
End;
RJob = Record
  Nama : String[30];
  JumOperasi : Integer;
  Operasi : Array[1..250] of ROperasi;
End;
RCase = Record
  JumJob : Integer;
  Job : Array[1..150] of RJob;
End;
RMesin = Record
```

```

    JumMesin : Integer;
    Mesin : Array[1..100] of String[30];
End;

RParent = Record
    Job : Array[1..1000] of String[30];
    Pm : Array[1..1000] of Real;
End;

RSeleksi = Record
    Populasi : Integer;
    Makespan : Real;
End;

var
    FormProses: TFormProses;
    Parameter : RParameter;
    FParameter : File Of RParameter;
    Kasus,KasusTemp : RCase;
    FKasus : File of RCase;
    Mesin : RMesin;
    FMesin : File of RMesin;
    pilih : Integer;

    KasusGenetic,KasusGenetic2,KasusGeneticKsg,KasusGeneticDuo :
RKasusGenetic;
    FKasusGenetic,FKasusGeneticBest : File of RKasusGenetic;
    SummaryJob,SummaryJobTemp : Array[1..100] of RMesinGenetic;

    RandomJob : Integer;
    JumOperasiTotal : Integer;
    OperasiAssign: Integer;

    RandomCross : Real;
    ParentAssign,RandomParent : Integer;
    CParent,Parent1,Parent2,OSpring1,OSpring2 : RKasusGenetic;
    FCParent,FParent : File Of RKasusGenetic;
    OfSpringIteration : Integer;
    Random1,Random2,Min,Max : Integer;
    CekSama1,CekSama2, Sama1, Sama2 : Integer;
    Ubah1,Ubah2 : Array[1..1000] of String[30];
    DumyJob : String[30];
    MasukMutasi: Integer;

    Seleksi : Array[1..100000] of RSeleksi;
    SeleksiTemp : RSeleksi;

    i,j,k,l,m : integer;
    DumyJumJob : Integer;
    DumyString : String;

```



```
n : integer;
PopulasiDumy, DumyMasuk : Integer;
GenerasiTerbaik : Integer;
IterasiGenerasi : Integer;
Lolos : Integer;
OperasiAssignAwal : Integer;
z : integer;
readyjob, readymesin : real;
TempJob : RJobGenetic;
NormalAssign : Array[1..1000] of integer;
```

```
CBPilih : Integer;
```

```
Click1,Click2 : Integer;
Test : Text;
DumyText : String;
Permutasix : Real;
```

implementation

```
{ $R *.dfm }
```

```
procedure TFormProses.BExitClick(Sender: TObject);
begin
    Close;
end;
```

```
procedure TFormProses.BProsesClick(Sender: TObject);
Procedure MakesPan;
Begin
    OperasiAssign:=0;
    KasusTemp:=Kasus;
    Lolos :=1 ;
    KasusGeneticDuo:=KasusGenetic;
    while OperasiAssign<JumOperasiTotal do
    Begin
        OperasiAssignAwal:=OperasiAssign;
        for i := 1 to Mesin.JumMesin do
        Begin
            if KasusGeneticDuo.Mesin[i].AssignJob<KasusGeneticDuo.Mesin[i].JumJob
then
                for k := 1 to Kasus.JumJob do
                    if
KasusGeneticDuo.Mesin[i].JobOperasi[KasusGeneticDuo.Mesin[i].AssignJob+1].Job
=KasusTemp.Job[k].Nama then
                        Begin
                            for l := 1 to KasusTemp.Job[k].JumOperasi do
                                if
KasusGeneticDuo.Mesin[i].JobOperasi[KasusGeneticDuo.Mesin[i].AssignJob+1].Ope
rasi=KasusTemp.Job[k].Operasi[l].Nama then
```

```

Begin
  if l=1 then
    Begin
      readyjob:=0;
      readyMesin:=KasusGeneticDuo.Mesin[i].WaktuTotal;
      if Readyjob>ReadyMesin then ReadyMesin:=ReadyJob;

      for z := 1 to KasusGeneticDuo.Mesin[i].AssignJob do
        Begin
          if z=1 then
            if (KasusGeneticDuo.Mesin[i].JobOperasi[1].WaktuMulai-
0)>=KasusGeneticDuo.Mesin[i].JobOperasi[KasusGeneticDuo.Mesin[i].AssignJob+1].
Waktu then
              Begin
                ReadyMesin:=0;
                Break;
              End;
            if z>1 then
              if (KasusGeneticDuo.Mesin[i].JobOperasi[z].WaktuMulai-
KasusGeneticDuo.Mesin[i].JobOperasi[z-
1].WaktuAkhir)>=KasusGeneticDuo.Mesin[i].JobOperasi[KasusGeneticDuo.Mesin[i].
AssignJob+1].Waktu then
                Begin
                  ReadyMesin:=KasusGeneticDuo.Mesin[i].JobOperasi[z-
1].WaktuAkhir;
                Break;
              End;
            End;
          End;

          OperasiAssign:=OperasiAssign+1;

          KasusGeneticDuo.Mesin[i].JobOperasi[KasusGeneticDuo.Mesin[i].AssignJob+1].Wak
tuMulai:=ReadyMesin;

          KasusTemp.Job[k].Operasi[l].WaktuMulai:=KasusGeneticDuo.Mesin[i].JobOperasi[K
asusGeneticDuo.Mesin[i].AssignJob+1].WaktuMulai;

          KasusGeneticDuo.Mesin[i].JobOperasi[KasusGeneticDuo.Mesin[i].AssignJob+1].Wak
tuAkhir:=ReadyMesin+KasusGeneticDuo.Mesin[i].JobOperasi[KasusGeneticDuo.Mes
in[i].AssignJob+1].Waktu;

          KasusTemp.Job[k].Operasi[l].WaktuAkhir:=KasusGeneticDuo.Mesin[i].JobOperasi[K
asusGeneticDuo.Mesin[i].AssignJob+1].WaktuAkhir;

          KasusGeneticDuo.Mesin[i].JobOperasi[KasusGeneticDuo.Mesin[i].AssignJob+1].Assi
gn:=1;
          KasusTemp.Job[k].Operasi[l].Assign:=1;
          if ReadyMesin>=KasusGeneticDuo.Mesin[i].WaktuTotal then

```

```

KasusGeneticDuo.Mesin[i].WaktuTotal:=KasusGeneticDuo.Mesin[i].JobOperasi[Kas
usGeneticDuo.Mesin[i].AssignJob+1].WaktuAkhir;

KasusGeneticDuo.Mesin[i].AssignJob:=KasusGeneticDuo.Mesin[i].AssignJob+1;
    Break;
End;
if (l<>1) and (KasusTemp.Job[k].Operasi[l-1].Assign=1) Then
Begin
    readyjob:=KasusTemp.Job[k].Operasi[l-1].WaktuAkhir;
    readyMesin:=KasusGeneticDuo.Mesin[i].WaktuTotal;
    if Readyjob>ReadyMesin then ReadyMesin:=ReadyJob;

    for z := 1 to KasusGeneticDuo.Mesin[i].AssignJob do
    Begin
        if z=1 then
            if (KasusGeneticDuo.Mesin[i].JobOperasi[1].WaktuMulai-
0)>=KasusGeneticDuo.Mesin[i].JobOperasi[KasusGeneticDuo.Mesin[i].AssignJob+1].
Waktu then
                Begin
                    if 0>=readyjob then
                    Begin
                        ReadyMesin:=0;
                        Break;
                    End;
                    if (readyjob>0) and
((Readyjob+KasusGeneticDuo.Mesin[i].JobOperasi[KasusGeneticDuo.Mesin[i].Assign
Job+1].Waktu)<=KasusGeneticDuo.Mesin[i].JobOperasi[1].WaktuMulai) then
                        Begin
                            ReadyMesin:=ReadyJob;
                            Break;
                        End;
                    End;
                    if z>1 then
                        if (KasusGeneticDuo.Mesin[i].JobOperasi[z].WaktuMulai-
KasusGeneticDuo.Mesin[i].JobOperasi[z-
1].WaktuAkhir)>=KasusGeneticDuo.Mesin[i].JobOperasi[KasusGeneticDuo.Mesin[i].
AssignJob+1].Waktu then
                            Begin
                                if KasusGeneticDuo.Mesin[i].JobOperasi[z-
1].WaktuAkhir>=ReadyJob then
                                    Begin
                                        ReadyMesin:=KasusGeneticDuo.Mesin[i].JobOperasi[z-
1].WaktuAkhir;
                                        Break;
                                    End;
                                if (readyjob>KasusGeneticDuo.Mesin[i].JobOperasi[z-
1].WaktuAkhir) and
((Readyjob+KasusGeneticDuo.Mesin[i].JobOperasi[KasusGeneticDuo.Mesin[i].Assign
Job+1].Waktu)<=KasusGeneticDuo.Mesin[i].JobOperasi[z].WaktuMulai) then

```

```

        Begin
            ReadyMesin:=ReadyJob;
            Break;
        End;
    End;
End;
OperasiAssign:=OperasiAssign+1;

```

```

KasusGeneticDuo.Mesin[i].JobOperasi[KasusGeneticDuo.Mesin[i].AssignJob+1].WaktuMulai:=ReadyMesin;

```

```

KasusTemp.Job[k].Operasi[l].WaktuMulai:=KasusGeneticDuo.Mesin[i].JobOperasi[KasusGeneticDuo.Mesin[i].AssignJob+1].WaktuMulai;

```

```

KasusGeneticDuo.Mesin[i].JobOperasi[KasusGeneticDuo.Mesin[i].AssignJob+1].WaktuAkhir:=ReadyMesin+KasusGeneticDuo.Mesin[i].JobOperasi[KasusGeneticDuo.Mesin[i].AssignJob+1].Waktu;

```

```

KasusTemp.Job[k].Operasi[l].WaktuAkhir:=KasusGeneticDuo.Mesin[i].JobOperasi[KasusGeneticDuo.Mesin[i].AssignJob+1].WaktuAkhir;

```

```

KasusGeneticDuo.Mesin[i].JobOperasi[KasusGeneticDuo.Mesin[i].AssignJob+1].Assign:=1;

```

```

        KasusTemp.Job[k].Operasi[l].Assign:=1;
        if ReadyMesin>=KasusGeneticDuo.Mesin[i].WaktuTotal then

```

```

KasusGeneticDuo.Mesin[i].WaktuTotal:=KasusGeneticDuo.Mesin[i].JobOperasi[KasusGeneticDuo.Mesin[i].AssignJob+1].WaktuAkhir;

```

```

KasusGeneticDuo.Mesin[i].AssignJob:=KasusGeneticDuo.Mesin[i].AssignJob+1;
        Break;

```

```

        End;
        End;
        Break;
        End;
        if (OperasiAssignAwal<>OperasiAssign) and (lolos=0) then Break;

```

```

End;

```

```

if (OperasiAssignAwal<>OperasiAssign) and (lolos=1) then

```

```

    Begin
        for z := 1 to Mesin.JumMesin do
            NormalAssign[z]:=KasusGeneticDuo.Mesin[z].AssignJob;

```

```

        End
        Else if (OperasiAssignAwal=OperasiAssign) and (lolos=1) then

```

```

            Begin
                for z := 1 to Mesin.JumMesin do
                    Begin
                        NormalAssign[z]:=KasusGeneticDuo.Mesin[z].AssignJob;

```

```

KasusGeneticDuo.Mesin[z].AssignJob:=KasusGeneticDuo.Mesin[z].AssignJob+1;
    End;
    lolos:=0;
End
Else if (OperasiAssignAwal=OperasiAssign) and (lolos=0) then
Begin
    for z := 1 to Mesin.JumMesin do

KasusGeneticDuo.Mesin[z].AssignJob:=KasusGeneticDuo.Mesin[z].AssignJob+1;
    End
    Else if (OperasiAssignAwal<>OperasiAssign) and (lolos=0) then
    Begin
        for z := 1 to Mesin.JumMesin do
            if
KasusGeneticDuo.Mesin[z].JobOperasi[KasusGeneticDuo.Mesin[z].AssignJob].Assign=1 then
                Begin
                    NormalAssign[z]:=NormalAssign[z]+1;
                    TempJob:=KasusGeneticDuo.Mesin[z].JobOperasi[NormalAssign[z]];

KasusGeneticDuo.Mesin[z].JobOperasi[NormalAssign[z]]:=KasusGeneticDuo.Mesin[z].JobOperasi[KasusGeneticDuo.Mesin[z].AssignJob];

KasusGeneticDuo.Mesin[z].JobOperasi[KasusGeneticDuo.Mesin[z].AssignJob]:=TempJob;
                End;
                for z := 1 to Mesin.JumMesin do
                    KasusGeneticDuo.Mesin[z].AssignJob:=NormalAssign[z];
                    lolos:=1;
                End;

            if OperasiAssignAwal<>OperasiAssign then
                for i := 1 to Mesin.JumMesin do
                    for k := 1 to KasusGeneticDuo.Mesin[i].AssignJob-1 do
                        for l := k+1 to KasusGeneticDuo.Mesin[i].AssignJob do
                            if
KasusGeneticDuo.Mesin[i].JobOperasi[l].WaktuMulai<KasusGeneticDuo.Mesin[i].JobOperasi[k].WaktuMulai then
                                Begin
                                    TempJob:=KasusGeneticDuo.Mesin[i].JobOperasi[k];

KasusGeneticDuo.Mesin[i].JobOperasi[k]:=KasusGeneticDuo.Mesin[i].JobOperasi[l];
                                    KasusGeneticDuo.Mesin[i].JobOperasi[l]:=TempJob;
                                End;

                            End;

                        End;

                    End;

                End;

            End;

        End;

    End;

```

```

for i := 1 to Mesin.JumMesin do
  if KasusGeneticDuo.MakesPan < KasusGeneticDuo.Mesin[i].WaktuTotal then
    KasusGeneticDuo.MakesPan := KasusGeneticDuo.Mesin[i].WaktuTotal;

    KasusGenetic.MakesPan := KasusGeneticDuo.MakesPan;
  for i := 1 to Mesin.JumMesin do
    for k := 1 to KasusGenetic.Mesin[i].JumJob do
      for l := 1 to KasusGeneticDuo.Mesin[i].JumJob do
        if
          KasusGenetic.Mesin[i].JobOperasi[k].Job = KasusGeneticDuo.Mesin[i].JobOperasi[l].Job then
            if
              KasusGenetic.Mesin[i].JobOperasi[k].Urutan = KasusGeneticDuo.Mesin[i].JobOperasi[l].Urutan then
                Begin

          KasusGenetic.Mesin[i].JobOperasi[k].WaktuMulai = KasusGeneticDuo.Mesin[i].JobOperasi[l].WaktuMulai;

          KasusGenetic.Mesin[i].JobOperasi[k].WaktuAkhir = KasusGeneticDuo.Mesin[i].JobOperasi[l].WaktuAkhir;
          Break;
          End;

        End;

      End;

    End;

  End;

begin
  CB.Items.Clear;
  CB.Text := "";
  SB.Position := 1;
  SB.Enabled := False;
  No1.Text := "";
  No2.Text := "";
  No3.Text := "";
  No4.Text := "";
  Job1.Text := "";
  Job2.Text := "";
  Job3.Text := "";
  Job4.Text := "";
  Operasi1.Text := "";
  Operasi2.Text := "";
  Operasi3.Text := "";
  Operasi4.Text := "";
  Waktu1.Text := "";
  Waktu2.Text := "";
  Waktu3.Text := "";
  Waktu4.Text := "";
  Mulai1.Text := "";
  Mulai2.Text := "";

```

```

Mulai3.Text:="";
Mulai4.Text:="";
Akhir1.Text:="";
Akhir2.Text:="";
Akhir3.Text:="";
Akhir4.Text:="";

JumOperasiTotal:=0;
for i := 1 to Kasus.JumJob do
JumOperasiTotal:=JumOperasiTotal+Kasus.Job[i].JumOperasi;

for i := 1 to Mesin.JumMesin do
Begin
SummaryJob[i].JumJob:=0;
SummaryJob[i].Nama:=Mesin.Mesin[i];
for j := 1 to Kasus.JumJob do
for k := 1 to Kasus.Job[j].JumOperasi do
Begin
if Kasus.Job[j].Operasi[k].Mesin=mesin.Mesin[i] then
Begin
SummaryJob[i].JumJob:=SummaryJob[i].JumJob+1;

SummaryJob[i].JobOperasi[SummaryJob[i].JumJob].Job:=Kasus.Job[j].Nama;

SummaryJob[i].JobOperasi[SummaryJob[i].JumJob].Operasi:=Kasus.Job[j].Operasi[
k].Nama;

SummaryJob[i].JobOperasi[SummaryJob[i].JumJob].Waktu:=Kasus.Job[j].Operasi[k
].Waktu;
SummaryJob[i].JobOperasi[SummaryJob[i].JumJob].Urutan:=k;
End;
End;
End;

System.Assign(FKasusGenetic,'Temp\Temp.POP');
Rewrite(FKasusGenetic);
PopulasiDumy:=0;

{Pembuatan Populasi Awal}
while PopulasiDumy<>Parameter.Populasi do
Begin
KasusGenetic := KasusGeneticKsg;
SummaryJobTemp:=SummaryJob;

for i := 1 to Mesin.JumMesin do
Begin
DumyJumJob:=SummaryJobTemp[i].JumJob;
KasusGenetic.Mesin[i].Nama:=SummaryJobTemp[i].Nama;
KasusGenetic.Mesin[i].JumJob:=SummaryJobTemp[i].JumJob;
for j := 1 to SummaryJobTemp[i].JumJob do

```

```

    Begin
        RandomJob := Random(DummyJumJob)+1;

KasusGenetic.Mesin[i].JobOperasi[j].Job:=SummaryJobTemp[i].JobOperasi[Random
Job].Job;

KasusGenetic.Mesin[i].JobOperasi[j].Operasi:=SummaryJobTemp[i].JobOperasi[Ran
domJob].Operasi;

KasusGenetic.Mesin[i].JobOperasi[j].Waktu:=SummaryJobTemp[i].JobOperasi[Rand
omJob].Waktu;

KasusGenetic.Mesin[i].JobOperasi[j].Urutan:=SummaryJobTemp[i].JobOperasi[Rand
omJob].Urutan;

SummaryJobTemp[i].JobOperasi[RandomJob]:=SummaryJobTemp[i].JobOperasi[Dum
yJumJob];
    DummyJumJob:=DummyJumJob-1;
    End;
End;

for i := 1 to Mesin.JumMesin do
for j := 1 to KasusGenetic.Mesin[i].JumJob-1 do
for k := j+1 to KasusGenetic.Mesin[i].JumJob do
if
(KasusGenetic.Mesin[i].JobOperasi[j].Job=KasusGenetic.Mesin[i].JobOperasi[k].Job)
and
(KasusGenetic.Mesin[i].JobOperasi[j].Urutan>KasusGenetic.Mesin[i].JobOperasi[k].
Urutan) Then
    Begin

SummaryJobTemp[1].JobOperasi[1].Job:=KasusGenetic.Mesin[i].JobOperasi[j].Job;

SummaryJobTemp[1].JobOperasi[1].Operasi:=KasusGenetic.Mesin[i].JobOperasi[j].
Operasi;

SummaryJobTemp[1].JobOperasi[1].Waktu:=KasusGenetic.Mesin[i].JobOperasi[j].W
aktu;

SummaryJobTemp[1].JobOperasi[1].Urutan:=KasusGenetic.Mesin[i].JobOperasi[j].U
rutan;

KasusGenetic.Mesin[i].JobOperasi[j].Job:=KasusGenetic.Mesin[i].JobOperasi[k].Job;

KasusGenetic.Mesin[i].JobOperasi[j].Operasi:=KasusGenetic.Mesin[i].JobOperasi[k].
Operasi;

KasusGenetic.Mesin[i].JobOperasi[j].Waktu:=KasusGenetic.Mesin[i].JobOperasi[k].W
aktu;

```



```

KasusGenetic.Mesin[i].JobOperasi[j].Urutan:=KasusGenetic.Mesin[i].JobOperasi[k].
Urutan;

KasusGenetic.Mesin[i].JobOperasi[k].Job:=SummaryJobTemp[1].JobOperasi[1].Job;

KasusGenetic.Mesin[i].JobOperasi[k].Operasi:=SummaryJobTemp[1].JobOperasi[1].
Operasi;

KasusGenetic.Mesin[i].JobOperasi[k].Waktu:=SummaryJobTemp[1].JobOperasi[1].W
aktu;

KasusGenetic.Mesin[i].JobOperasi[k].Urutan:=SummaryJobTemp[1].JobOperasi[1].
Urutan;
    End;

    Makespan;
    for i := 1 to System.FileSize(FKasusGenetic) do
    Begin
        DumyMasuk:=JumOperasiTotal;
        System.Seek(FKasusGenetic,i-1); Read(FKasusGenetic,KasusGenetic2);
        for j:=1 to Mesin.JumMesin do
            for k := 1 to KasusGenetic.Mesin[j].JumJob do
                if
(kasusGenetic2.Mesin[j].JobOperasi[k].Job=kasusGenetic.Mesin[j].JobOperasi[k].Jo
b) and
(kasusGenetic2.Mesin[j].JobOperasi[k].Operasi=kasusGenetic.Mesin[j].JobOperasi[k]
.Operasi) then DumyMasuk:=DumyMasuk-1
                    else Break;
                if DumyMasuk=0 then Break;
            End;
        if (System.FileSize(FKasusGenetic)=0) or (DumyMasuk>0) then
        Begin
            PopulasiDumy:=PopulasiDumy+1;
            System.Seek(FKasusGenetic,PopulasiDumy-1);
            Write(FKasusGenetic,KasusGenetic);
        End;
    End;

    System.Assign(Test,'Solusi\Solusi.TXT');
    Rewrite(Test);
    for k := 1 to System.FileSize(FKasusGenetic) do
    Begin
        System.Seek(FKasusGenetic,k-1); Read(FKasusGenetic,KasusGenetic);
        Str(0,DumyString);
        DumyText:='Generasi ke-'+DumyString+' Populasi ke-';
        Str(k,DumyString);
        DumyText:=DumyText+DumyString+', Makespan = ';
        Str(KasusGenetic.Makespan:0:2,DumyString);
        DumyText:=DumyText+DumyString;
    End;

```

```

System.Append(Test);
Writeln(Test,DumyText);
for i := 1 to Mesin.JumMesin do
  for j := 1 to KasusGenetic.Mesin[i].JumJob do
    Begin
DumyText:=KasusGenetic.Mesin[i].Nama+';'+KasusGenetic.Mesin[i].JobOperasi[j].Job
b+';'+KasusGenetic.Mesin[i].JobOperasi[j].Operasi+';';
      Str(KasusGenetic.Mesin[i].JobOperasi[j].Waktu:0:2,DumyString);
      DumyText:=DumyText + DumyString+';';
      Str(KasusGenetic.Mesin[i].JobOperasi[j].WaktuMulai:0:2,DumyString);
      DumyText:=DumyText + DumyString+';';
      Str(KasusGenetic.Mesin[i].JobOperasi[j].WaktuAkhir:0:2,DumyString);
      DumyText:=DumyText + DumyString;
      System.Append(Test);
      Writeln(Test,DumyText);
    End;
  Writeln(Test);
  Writeln(Test);
End;

IterasiGenerasi:=0;
GenerasiTerbaik:=0;
while IterasiGenerasi<>Parameter.Generasi do
  Begin
  {Proses CrossOver}
  if Mesin.JumMesin>2 then
    Begin
    System.Assign(FCParent,'Temp\Temp.CPR');
    Rewrite(FCParent);
    for i := 1 to System.FileSize(FKasusGenetic) do
      Begin
      System.Seek(FKasusGenetic,i-1); Read(FKasusGenetic,KasusGenetic);
      RandomCross:=Random(1001)/1000;
      if RandomCross<=Parameter.Pc then
        Begin
          for j := 1 to Mesin.JumMesin do
            Begin
              CParent.Mesin[j].Nama:=KasusGenetic.Mesin[j].Nama;
              CParent.Mesin[j].JumJob:=KasusGenetic.Mesin[j].JumJob;
              for k := 1 to KasusGenetic.Mesin[j].JumJob do
                Begin
CParent.Mesin[j].JobOperasi[k].Job:=KasusGenetic.Mesin[j].JobOperasi[k].Job;

CParent.Mesin[j].JobOperasi[k].Operasi:=KasusGenetic.Mesin[j].JobOperasi[k].Oper
asi;
                End;
              End;
            End;
          End;
        End;
      End;
    End;
  End;
End;

```

```

CParent.Mesin[j].JobOperasi[k].Urutan:=KasusGenetic.Mesin[j].JobOperasi[k].Urutan;

CParent.Mesin[j].JobOperasi[k].Waktu:=KasusGenetic.Mesin[j].JobOperasi[k].Waktu;
    End;
        End;
            System.Seek(FCParent, System.FileSize(FCParent));
Write(FCParent, CParent);
    End;
End;

ParentAsign:=(System.FileSize(FCParent) Div 2)*2;

System.Assign(FParent, 'Temp\Temp.PRT');
Rewrite(FParent);
while ParentAsign<>0 do
Begin
    RandomParent:=Random(ParentAsign)+1;
    System.Seek(FCParent, RandomParent-1); Read(FCParent, CParent);
    System.Seek(FParent, System.FileSize(FParent)); Write(FParent, CParent);
    System.Seek(FCParent, ParentAsign-1); Read(FCParent, CParent);
    System.Seek(FCParent, RandomParent-1); Write(FCParent, CParent);
    ParentAsign:=ParentAsign-1;
End;
System.Close(FCParent);

for OfSpringIteration := 1 to System.FileSize(FParent) Div 2 do
Begin
    System.Seek(FParent, 2*(OfSpringIteration-1)); Read(FParent, Parent1);
    System.Seek(FParent, (2*(OfSpringIteration-1)+1)); Read(FParent, Parent2);
    Random1:=Random(Mesin.JumMesin-2)+2;
    Random2:=Random(Mesin.JumMesin-2)+2;
    if Random1<=Random2 then
    Begin
        Min:=Random1;
        Max:=Random2;
    End;
    if Random2<Random1 then
    Begin
        Min:=Random2;
        Max:=Random1;
    End;
    for i := 1 to Min-1 do
    Begin
        OSpring1.Mesin[i]:=Parent1.Mesin[i];
        OSpring2.Mesin[i]:=Parent2.Mesin[i];
    End;

```

```

for i := Min to Max do
Begin
    OSpring1.Mesin[i]:=Parent2.Mesin[i];
    OSpring2.Mesin[i]:=Parent1.Mesin[i];
End;
for i := Max+1 to Mesin.JumMesin do
Begin
    OSpring1.Mesin[i]:=Parent1.Mesin[i];
    OSpring2.Mesin[i]:=Parent2.Mesin[i];
End;

KasusGenetic:=OSpring1;
MakesPan;
System.Seek(FKasusGenetic,System.FileSize(FKasusGenetic));
Write(FKasusGenetic,KasusGenetic);

    KasusGenetic:=OSpring2;
    MakesPan;
    System.Seek(FKasusGenetic,System.FileSize(FKasusGenetic));
Write(FKasusGenetic,KasusGenetic);
End;
System.Close(FParent);
End;

{Proses Mutasi}
System.Assign(FParent,'Temp\Temp.PRT');
Rewrite(FParent);
for i := 1 to System.FileSize(FKasusGenetic) do
Begin
    System.Seek(FKasusGenetic,i-1); Read(FKasusGenetic,KasusGenetic);
    MasukMutasi:=0;
    for j := 1 to Mesin.JumMesin do
    Begin
        CParent.Mesin[j].Nama:=KasusGenetic.Mesin[j].Nama;
        CParent.Mesin[j].JumJob:=KasusGenetic.Mesin[j].JumJob;
        for k := 1 to KasusGenetic.Mesin[j].JumJob do
            Begin

CParent.Mesin[j].JobOperasi[k].Job:=KasusGenetic.Mesin[j].JobOperasi[k].Job;

CParent.Mesin[j].JobOperasi[k].Operasi:=KasusGenetic.Mesin[j].JobOperasi[k].Operasi;

CParent.Mesin[j].JobOperasi[k].Urutan:=KasusGenetic.Mesin[j].JobOperasi[k].Urutan;

CParent.Mesin[j].JobOperasi[k].Waktu:=KasusGenetic.Mesin[j].JobOperasi[k].Waktu;
;
                CParent.Mesin[j].JobOperasi[k].ProbMut:=Random(1001)/1000;
            End;
        End;
    End;
End;

```



```

    {if
CParent.Mesin[j].JobOperasi[k].Job<>CParent.Mesin[j].JobOperasi[1].Job then}
    Begin

OSpring1.Mesin[1].JobOperasi[1].Job:=CParent.Mesin[j].JobOperasi[k].Job;
OSpring1.Mesin[1].JobOperasi[1].Operasi:=CParent.Mesin[j].JobOperasi[k].Operasi;
OSpring1.Mesin[1].JobOperasi[1].Waktu:=CParent.Mesin[j].JobOperasi[k].Waktu;
OSpring1.Mesin[1].JobOperasi[1].Urutan:=CParent.Mesin[j].JobOperasi[k].Urutan;
CParent.Mesin[j].JobOperasi[k].Job:=CParent.Mesin[j].JobOperasi[1].Job;
CParent.Mesin[j].JobOperasi[k].Operasi:=CParent.Mesin[j].JobOperasi[1].Operasi;
CParent.Mesin[j].JobOperasi[k].Waktu:=CParent.Mesin[j].JobOperasi[1].Waktu;
CParent.Mesin[j].JobOperasi[k].Urutan:=CParent.Mesin[j].JobOperasi[1].Urutan;
CParent.Mesin[j].JobOperasi[1].Job:=OSpring1.Mesin[1].JobOperasi[1].Job;
CParent.Mesin[j].JobOperasi[1].Operasi:=OSpring1.Mesin[1].JobOperasi[1].Operasi;
CParent.Mesin[j].JobOperasi[1].Waktu:=OSpring1.Mesin[1].JobOperasi[1].Waktu;
CParent.Mesin[j].JobOperasi[1].Urutan:=OSpring1.Mesin[1].JobOperasi[1].Urutan;
    End;
    End;

    KasusGenetic:=CParent;

    for i := 1 to Mesin.JumMesin do
    for j := 1 to KasusGenetic.Mesin[i].JumJob-1 do
    for k := j+1 to KasusGenetic.Mesin[i].JumJob do
    if
(KasusGenetic.Mesin[i].JobOperasi[j].Job=KasusGenetic.Mesin[i].JobOperasi[k].Job)
and
(KasusGenetic.Mesin[i].JobOperasi[j].Urutan>KasusGenetic.Mesin[i].JobOperasi[k].
Urutan) Then
    Begin

SummaryJobTemp[1].JobOperasi[1].Job:=KasusGenetic.Mesin[i].JobOperasi[j].Job;

SummaryJobTemp[1].JobOperasi[1].Operasi:=KasusGenetic.Mesin[i].JobOperasi[j].
Operasi;

SummaryJobTemp[1].JobOperasi[1].Waktu:=KasusGenetic.Mesin[i].JobOperasi[j].W
aktu;

```

```

SummaryJobTemp[1].JobOperasi[1].Urutan:=KasusGenetic.Mesin[i].JobOperasi[j].U
rutan;

KasusGenetic.Mesin[i].JobOperasi[j].Job:=KasusGenetic.Mesin[i].JobOperasi[k].Job;

KasusGenetic.Mesin[i].JobOperasi[j].Operasi:=KasusGenetic.Mesin[i].JobOperasi[k].
Operasi;

KasusGenetic.Mesin[i].JobOperasi[j].Waktu:=KasusGenetic.Mesin[i].JobOperasi[k].W
aktu;

KasusGenetic.Mesin[i].JobOperasi[j].Urutan:=KasusGenetic.Mesin[i].JobOperasi[k].
Urutan;

KasusGenetic.Mesin[i].JobOperasi[k].Job:=SummaryJobTemp[1].JobOperasi[1].Job;

KasusGenetic.Mesin[i].JobOperasi[k].Operasi:=SummaryJobTemp[1].JobOperasi[1].
Operasi;

KasusGenetic.Mesin[i].JobOperasi[k].Waktu:=SummaryJobTemp[1].JobOperasi[1].W
aktu;

KasusGenetic.Mesin[i].JobOperasi[k].Urutan:=SummaryJobTemp[1].JobOperasi[1].
Urutan;
    End;

    MakesPan;
    System.Seek(FKasusGenetic, System.FileSize(FKasusGenetic));
Write(FKasusGenetic, KasusGenetic);
    End;
    System.Close(FParent);

    {Proses Seleksi}
    for i := 1 to System.FileSize(FKasusGenetic) do
    Begin
        System.Seek(FKasusGenetic, i-1); Read(FKasusGenetic, KasusGenetic);
        Seleksi[i].Populasi:=i;
        Seleksi[i].Makespan:=KasusGenetic.MakesPan;
    End;

    for i := 1 to System.FileSize(FKasusGenetic) - 1 do
    for j := i+1 to System.FileSize(FKasusGenetic) do
    if Seleksi[i].Makespan>Seleksi[j].Makespan then
    Begin
        SeleksiTemp:=Seleksi[i];
        Seleksi[i]:=Seleksi[j];
        Seleksi[j]:=SeleksiTemp;
    End;

```

```

    if Seleksi[1].Populasi>Parameter.Populasi then
GenerasiTerbaik:=IterasiGenerasi+1;

    System.Assign(FKasusGeneticBest,'Temp\Temp.SLK');
    Rewrite(FKasusGeneticBest);
    for i := 1 to Parameter.Populasi do
    Begin
        System.Seek(FKasusGenetic,Seleksi[i].Populasi-1);
    Read(FKasusGenetic,KasusGenetic);
        System.Seek(FKasusGeneticBest,i-1);
    Write(FKasusGeneticBest,KasusGenetic);
    End;
    System.Close(FKasusGenetic);

    System.Assign(FKasusGenetic,'Temp\Temp.POP');
    Rewrite(FKasusGenetic);
    for i := 1 to Parameter.Populasi do
    Begin
        System.Seek(FKasusGeneticBest,i-1);
    Read(FKasusGeneticBest,KasusGenetic);
        System.Seek(FKasusGenetic,i-1); Write(FKasusGenetic,KasusGenetic);
    End;
    System.Close(FKasusGeneticBest);

    for k := 1 to System.FileSize(FKasusGenetic) do
    Begin
    System.Seek(FKasusGenetic,k-1); Read(FKasusGenetic,KasusGenetic);
    Str(IterasiGenerasi+1,DummyString);
    DummyText:='Generasi ke-'+DummyString+' Populasi ke-';
    Str(k,DummyString);
    DummyText:=DummyText+DummyString+', Makespan = ';
    Str(KasusGenetic.Makespan:0:2,DummyString);
    DummyText:=DummyText+DummyString;
    System.Append(Test);
    Writeln(Test,DummyText);
    for i := 1 to Mesin.JumMesin do
        for j := 1 to KasusGenetic.Mesin[i].JumJob do
        Begin
            DummyText:=KasusGenetic.Mesin[i].Nama+';'+KasusGenetic.Mesin[i].JobOperasi[j].Job
            b+';'+KasusGenetic.Mesin[i].JobOperasi[j].Operasi+';';
            Str(KasusGenetic.Mesin[i].JobOperasi[j].Waktu:0:2,DummyString);
            DummyText:=DummyText + DummyString+';';
            Str(KasusGenetic.Mesin[i].JobOperasi[j].WaktuMulai:0:2,DummyString);
            DummyText:=DummyText + DummyString+';';
            Str(KasusGenetic.Mesin[i].JobOperasi[j].WaktuAkhir:0:2,DummyString);
            DummyText:=DummyText + DummyString;
            System.Append(Test);
            Writeln(Test,DummyText);
        End;
    End;
    End;

```



```

End;
Writeln(Test);
Writeln(Test);
End;

IterasiGenerasi:=IterasiGenerasi+1;
End;

System.Close(Test);

{Penulisan Solusi Terbaik}
System.Assign(Test,'Solusi\SolusiTerbaik.TXT');
Rewrite(Test);
System.Seek(FKasusGenetic,0); Read(FKasusGenetic,KasusGenetic);
Str(GenerasiTerbaik,DumyString);
DumyText:='Generasi Terbaik = '+DumyString+', Makespan = ';
Str(KasusGenetic.Makespan:0:2,DumyString);
DumyText:=DumyText+DumyString;
System.Append(Test);
Writeln(Test,DumyText);
for i := 1 to Mesin.JumMesin do
  for j := 1 to KasusGenetic.Mesin[i].JumJob do
    Begin
DumyText:=KasusGenetic.Mesin[i].Nama+';'+KasusGenetic.Mesin[i].JobOperasi[j].Job
b+';'+KasusGenetic.Mesin[i].JobOperasi[j].Operasi+';';
      Str(KasusGenetic.Mesin[i].JobOperasi[j].Waktu:0:2,DumyString);
      DumyText:=DumyText + DumyString+';';
      Str(KasusGenetic.Mesin[i].JobOperasi[j].WaktuMulai:0:2,DumyString);
      DumyText:=DumyText + DumyString+';';
      Str(KasusGenetic.Mesin[i].JobOperasi[j].WaktuAkhir:0:2,DumyString);
      DumyText:=DumyText + DumyString;
      System.Append(Test);
      Writeln(Test,DumyText);
    End;
System.Close(Test);
System.Close(FKasusGenetic);

Str(KasusGenetic.MakesPan:0:2,DumyString);
TMakesPan.Text:=DumyString;
Str(GenerasiTerbaik,DumyString);
TGenBest.Text:=DumyString;
for i := 1 to Mesin.JumMesin do CB.Items.Append(KasusGenetic.Mesin[i].Nama);
CB.Enabled:=True;
end;

procedure TFormProses.CBChange(Sender: TObject);
begin
  for i := 1 to Mesin.JumMesin do
    if CB.Text=KasusGenetic.Mesin[i].Nama then CBPilih:=i;

```

```

if KasusGenetic.Mesin[CBPilih].JumJob=1 then
Begin
  SB.Position:=1;
  SB.Enabled:=False;
  No1.Text:='1';
  No2.Text:="";
  No3.Text:="";
  No4.Text:="";
  Job1.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[1].Job;
  Job2.Text:="";
  Job3.Text:="";
  Job4.Text:="";
  Operasi1.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[1].Operasi;
  Operasi2.Text:="";
  Operasi3.Text:="";
  Operasi4.Text:="";
  Str(KasusGenetic.Mesin[CBPilih].JobOperasi[1].Waktu:0:2,DumyString);
  Waktu1.Text:=DumyString;
  Waktu2.Text:="";
  Waktu3.Text:="";
  Waktu4.Text:="";

Str(KasusGenetic.Mesin[CBPilih].JobOperasi[1].WaktuMulai:0:2,DumyString);
  Mulai1.Text:=DumyString;
  Mulai2.Text:="";
  Mulai3.Text:="";
  Mulai4.Text:="";
  Str(KasusGenetic.Mesin[CBPilih].JobOperasi[1].WaktuAkhir:0:2,DumyString);
  Akhir1.Text:=DumyString;
  Akhir2.Text:="";
  Akhir3.Text:="";
  Akhir4.Text:="";
End;
if KasusGenetic.Mesin[CBPilih].JumJob=2 then
Begin
  SB.Position:=1;
  SB.Enabled:=False;
  No1.Text:='1';
  No2.Text:='2';
  No3.Text:="";
  No4.Text:="";
  Job1.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[1].Job;
  Job2.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[2].Job;
  Job3.Text:="";
  Job4.Text:="";
  Operasi1.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[1].Operasi;
  Operasi2.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[2].Operasi;
  Operasi3.Text:="";
  Operasi4.Text:="";
  Str(KasusGenetic.Mesin[CBPilih].JobOperasi[1].Waktu:0:2,DumyString);

```

```

    Waktu1.Text:=DumyString;
    Str(KasusGenetic.Mesin[CBPilih].JobOperasi[2].Waktu:0:2,DumyString);
    Waktu2.Text:=DumyString;
    Waktu3.Text:="";
    Waktu4.Text:="";

Str(KasusGenetic.Mesin[CBPilih].JobOperasi[1].WaktuMulai:0:2,DumyString);
    Mulai1.Text:=DumyString;

Str(KasusGenetic.Mesin[CBPilih].JobOperasi[2].WaktuMulai:0:2,DumyString);
    Mulai2.Text:=DumyString;
    Mulai3.Text:="";
    Mulai4.Text:="";
    Str(KasusGenetic.Mesin[CBPilih].JobOperasi[1].WaktuAkhir:0:2,DumyString);
    Akhir1.Text:=DumyString;
    Str(KasusGenetic.Mesin[CBPilih].JobOperasi[2].WaktuAkhir:0:2,DumyString);
    Akhir2.Text:=DumyString;
    Akhir3.Text:="";
    Akhir4.Text:="";
End;
if KasusGenetic.Mesin[CBPilih].JumJob=3 then
Begin
    SB.Position:=1;
    SB.Enabled:=False;
    No1.Text:='1';
    No2.Text:='2';
    No3.Text:='3';
    No4.Text:="";
    Job1.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[1].Job;
    Job2.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[2].Job;
    Job3.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[3].Job;
    Job4.Text:="";
    Operasi1.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[1].Operasi;
    Operasi2.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[2].Operasi;
    Operasi3.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[3].Operasi;
    Operasi4.Text:="";
    Str(KasusGenetic.Mesin[CBPilih].JobOperasi[1].Waktu:0:2,DumyString);
    Waktu1.Text:=DumyString;
    Str(KasusGenetic.Mesin[CBPilih].JobOperasi[2].Waktu:0:2,DumyString);
    Waktu2.Text:=DumyString;
    Str(KasusGenetic.Mesin[CBPilih].JobOperasi[3].Waktu:0:2,DumyString);
    Waktu3.Text:=DumyString;
    Waktu4.Text:="";

Str(KasusGenetic.Mesin[CBPilih].JobOperasi[1].WaktuMulai:0:2,DumyString);
    Mulai1.Text:=DumyString;

Str(KasusGenetic.Mesin[CBPilih].JobOperasi[2].WaktuMulai:0:2,DumyString);
    Mulai2.Text:=DumyString;

```

```

Str(KasusGenetic.Mesin[CBPilih].JobOperasi[3].WaktuMulai:0:2,DumyString);
    Mulai3.Text:=DumyString;
    Mulai4.Text:="";
    Str(KasusGenetic.Mesin[CBPilih].JobOperasi[1].WaktuAkhir:0:2,DumyString);
    Akhir1.Text:=DumyString;
    Str(KasusGenetic.Mesin[CBPilih].JobOperasi[2].WaktuAkhir:0:2,DumyString);
    Akhir2.Text:=DumyString;
    Str(KasusGenetic.Mesin[CBPilih].JobOperasi[3].WaktuAkhir:0:2,DumyString);
    Akhir3.Text:=DumyString;
    Akhir4.Text:="";
End;
if KasusGenetic.Mesin[CBPilih].JumJob>=4 then
Begin
    SB.Position:=1;
    SB.Enabled:=False;
    No1.Text:='1';
    No2.Text:='2';
    No3.Text:='3';
    No4.Text:='4';
    Job1.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[1].Job;
    Job2.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[2].Job;
    Job3.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[3].Job;
    Job4.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[4].Job;
    Operasi1.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[1].Operasi;
    Operasi2.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[2].Operasi;
    Operasi3.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[3].Operasi;
    Operasi4.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[4].Operasi;
    Str(KasusGenetic.Mesin[CBPilih].JobOperasi[1].Waktu:0:2,DumyString);
    Waktu1.Text:=DumyString;
    Str(KasusGenetic.Mesin[CBPilih].JobOperasi[2].Waktu:0:2,DumyString);
    Waktu2.Text:=DumyString;
    Str(KasusGenetic.Mesin[CBPilih].JobOperasi[3].Waktu:0:2,DumyString);
    Waktu3.Text:=DumyString;
    Str(KasusGenetic.Mesin[CBPilih].JobOperasi[4].Waktu:0:2,DumyString);
    Waktu4.Text:=DumyString;

Str(KasusGenetic.Mesin[CBPilih].JobOperasi[1].WaktuMulai:0:2,DumyString);
    Mulai1.Text:=DumyString;

Str(KasusGenetic.Mesin[CBPilih].JobOperasi[2].WaktuMulai:0:2,DumyString);
    Mulai2.Text:=DumyString;

Str(KasusGenetic.Mesin[CBPilih].JobOperasi[3].WaktuMulai:0:2,DumyString);
    Mulai3.Text:=DumyString;

Str(KasusGenetic.Mesin[CBPilih].JobOperasi[4].WaktuMulai:0:2,DumyString);
    Mulai4.Text:=DumyString;
    Str(KasusGenetic.Mesin[CBPilih].JobOperasi[1].WaktuAkhir:0:2,DumyString);
    Akhir1.Text:=DumyString;

```

```

Str(KasusGenetic.Mesin[CBPilih].JobOperasi[2].WaktuAkhir:0:2,DumyString);
Akhir2.Text:=DumyString;
Str(KasusGenetic.Mesin[CBPilih].JobOperasi[3].WaktuAkhir:0:2,DumyString);
Akhir3.Text:=DumyString;
Str(KasusGenetic.Mesin[CBPilih].JobOperasi[4].WaktuAkhir:0:2,DumyString);
Akhir4.Text:=DumyString;
End;
if KasusGenetic.Mesin[CBPilih].JumJob>4 then
Begin
  SB.Enabled:=True;
  SB.Position:=1;
  SB.Min:=1;
  SB.Max:=KasusGenetic.Mesin[CBPilih].JumJob-3;
  SB.SmallChange:=1;
  SB.LargeChange:=3;
End;
end;

procedure TFormProses.CButtonClick(Sender: TObject);
begin

  System.Assign(FKasus,'Case\' +TCase.Text+'.JOB');
  {$I-}
  Reset(FKasus);
  {$I+}
  n:=IORESULT;
  Reset(FKasus);
  System.Seek(FKasus,0);
  Read(FKasus,Kasus);
  System.Close(FKasus);

  System.Assign(FMesin,'Case\' +TCase.Text+'.MSN');
  Reset(FMesin);
  System.Seek(FMesin,0);
  Read(FMesin,Mesin);
  System.Close(FMesin);

  if n=0 then
  Begin
    LCase.Caption:=TCase.Text;
    Str(Mesin.JumMesin,DumyString);
    TMesin.Text:=DumyString;
    Str(Kasus.JumJob,DumyString);
    TJob.Text:=DumyString;
    Click1:=1;
    for i := 1 to Mesin.JumMesin do
    Begin
      SummaryJob[i].JumJob:=0;
      SummaryJob[i].Nama:=Mesin.Mesin[i];
      for j := 1 to Kasus.JumJob do

```

```

    for k := 1 to Kasus.Job[j].JumOperasi do
    Begin
        if Kasus.Job[j].Operasi[k].Mesin=mesin.Mesin[i] then
            Begin
                SummaryJob[i].JumJob:=SummaryJob[i].JumJob+1;
            End;
        End;
    End;
    Permutasix:=1;
    for i := 1 to Mesin.JumMesin do
        for j := 1 to SummaryJob[i].JumJob do Permutasix:=Permutasix*j;
        Str(Permutasix:0:0,DumyString);
        Permutasi.Text:=DumyString;
    End;
    TCase.Text:="";

    if (Click1+Click2)= 2 then BProses.Enabled:=True;
    if (Click1+Click2)< 2 then BProses.Enabled:=False;
end;

procedure TFormProses.FormCreate(Sender: TObject);
begin
    Randomize;
    Click1:=0;
    Click2:=0;
end;

procedure TFormProses.PButtonClick(Sender: TObject);
begin
    System.Assign(FParameter,'Parameter\'+TParameter.Text+'.PRM');
    {$I-}
    Reset(FParameter);
    {$I+}
    n:=IORESULT;
    Reset(FParameter);
    System.Seek(FParameter,0);
    Read(FParameter,Parameter);
    System.Close(FParameter);

    if n=0 then
    Begin
        Click2:=1;
        Str(Parameter.Populasi,DumyString);
        TPopulasi.Text:=DumyString;
        Str(Parameter.Generasi,DumyString);
        TGenerasi.Text:=DumyString;
        Str(Parameter.Pc:0:2,DumyString);
        TPc.Text:=DumyString;
        Str(Parameter.Pm:0:2,DumyString);
        TPm.Text:=DumyString;
    End;
end;

```

```

End;
TParameter.Text:="";

if (Click1+Click2)= 2 then BProses.Enabled:=True;
if (Click1+Click2)< 2 then BProses.Enabled:=False;
end;

procedure TFormProses.SBChange(Sender: TObject);
begin
  str(SB.Position+1-1,DumyString);
  No1.Text:=DumyString;
  str(SB.Position+2-1,DumyString);
  No2.Text:=DumyString;
  str(SB.Position+3-1,DumyString);
  No3.Text:=DumyString;
  str(SB.Position+4-1,DumyString);
  No4.Text:=DumyString;
  Job1.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[SB.Position+1-1].Job;
  Job2.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[SB.Position+2-1].Job;
  Job3.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[SB.Position+3-1].Job;
  Job4.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[SB.Position+4-1].Job;
  Operasi1.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[SB.Position+1-1].Operasi;
  Operasi2.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[SB.Position+2-1].Operasi;
  Operasi3.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[SB.Position+3-1].Operasi;
  Operasi4.Text:=KasusGenetic.Mesin[CBPilih].JobOperasi[SB.Position+4-1].Operasi;
  str(KasusGenetic.Mesin[CBPilih].JobOperasi[SB.Position+1-1].waktu:0:2,DumyString);
  Waktu1.Text:=DumyString;
  str(KasusGenetic.Mesin[CBPilih].JobOperasi[SB.Position+2-1].waktu:0:2,DumyString);
  Waktu2.Text:=DumyString;
  str(KasusGenetic.Mesin[CBPilih].JobOperasi[SB.Position+3-1].waktu:0:2,DumyString);
  Waktu3.Text:=DumyString;
  str(KasusGenetic.Mesin[CBPilih].JobOperasi[SB.Position+4-1].waktu:0:2,DumyString);
  Waktu4.Text:=DumyString;
  str(KasusGenetic.Mesin[CBPilih].JobOperasi[SB.Position+1-1].WaktuMulai:0:2,DumyString);
  Mulai1.Text:=DumyString;
  str(KasusGenetic.Mesin[CBPilih].JobOperasi[SB.Position+2-1].WaktuMulai:0:2,DumyString);
  Mulai2.Text:=DumyString;
  str(KasusGenetic.Mesin[CBPilih].JobOperasi[SB.Position+3-1].WaktuMulai:0:2,DumyString);
  Mulai3.Text:=DumyString;

```

```

    str(KasusGenetic.Mesin[CBPilih].JobOperasi[SB.Position+4-
1].WaktuMulai:0:2,DumyString);
    Mulai4.Text:=DumyString;
    str(KasusGenetic.Mesin[CBPilih].JobOperasi[SB.Position+1-
1].WaktuAkhir:0:2,DumyString);
    Akhir1.Text:=DumyString;
    str(KasusGenetic.Mesin[CBPilih].JobOperasi[SB.Position+2-
1].WaktuAkhir:0:2,DumyString);
    Akhir2.Text:=DumyString;
    str(KasusGenetic.Mesin[CBPilih].JobOperasi[SB.Position+3-
1].WaktuAkhir:0:2,DumyString);
    Akhir3.Text:=DumyString;
    str(KasusGenetic.Mesin[CBPilih].JobOperasi[SB.Position+4-
1].WaktuAkhir:0:2,DumyString);
    Akhir4.Text:=DumyString;
end;

```

```

procedure TFormProses.TCaseChange(Sender: TObject);
begin
    if TCase.Text<>" then CButton.Enabled:=True;
    if TCase.Text="" then CButton.Enabled:=False;
end;

```

```

procedure TFormProses.TGenerasiChange(Sender: TObject);
begin
    Val(TGenerasi.Text,Parameter.Generasi,i);
    if (TGenerasi.Text<>" ) and (TPopulasi.Text<>" ) and (TPC.Text<>" ) and
(TPm.Text<>" ) then CLick2:=1;
    if (TGenerasi.Text="" ) or (TPopulasi.Text="" ) or (TPC.Text="" ) or (TPm.Text="" ) then
CLick2:=0;
    if (Click1+Click2)= 2 then BProses.Enabled:=True;
    if (Click1+Click2)< 2 then BProses.Enabled:=False;
end;

```

```

procedure TFormProses.TParameterChange(Sender: TObject);
begin
    if TParameter.Text<>" then PButton.Enabled:=True;
    if TParameter.Text="" then PButton.Enabled:=False;
end;

```

```

procedure TFormProses.TPcChange(Sender: TObject);
begin
    Val(TPc.Text,Parameter.Pc,i);
    if (TGenerasi.Text<>" ) and (TPopulasi.Text<>" ) and (TPC.Text<>" ) and
(TPm.Text<>" ) then CLick2:=1;
    if (TGenerasi.Text="" ) or (TPopulasi.Text="" ) or (TPC.Text="" ) or (TPm.Text="" ) then
CLick2:=0;
    if (Click1+Click2)= 2 then BProses.Enabled:=True;
    if (Click1+Click2)< 2 then BProses.Enabled:=False;
end;

```



```
procedure TFormProses.TPmChange(Sender: TObject);
begin
    Val(TPm.Text,Parameter.Pm,i);
    if (TGenerasi.Text<>") and (TPopulasi.Text<>") and (TPC.Text<>") and
(TPm.Text<>") then CLick2:=1;
    if (TGenerasi.Text=") or (TPopulasi.Text=") or (TPC.Text=") or (TPm.Text=") then
CLick2:=0;
    if (Click1+Click2)= 2 then BProses.Enabled:=True;
    if (Click1+Click2)< 2 then BProses.Enabled:=False;
end;
```

```
procedure TFormProses.TPopulasiChange(Sender: TObject);
begin
    Val(TPopulasi.Text,Parameter.Populasi,i);
    if (TGenerasi.Text<>") and (TPopulasi.Text<>") and (TPC.Text<>") and
(TPm.Text<>") then CLick2:=1;
    if (TGenerasi.Text=") or (TPopulasi.Text=") or (TPC.Text=") or (TPm.Text=") then
CLick2:=0;
    if (Click1+Click2)= 2 then BProses.Enabled:=True;
    if (Click1+Click2)< 2 then BProses.Enabled:=False;
end;
```

```
end.
```

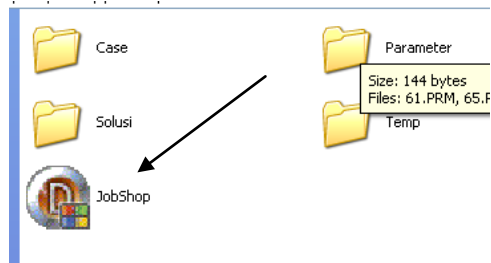
LAMPIRAN 3

PANDUAN PENGGUNAAN PROGRAM *JOB SHOP GA*

⊕ Input Data

❖ Data Kasus dan Parameter

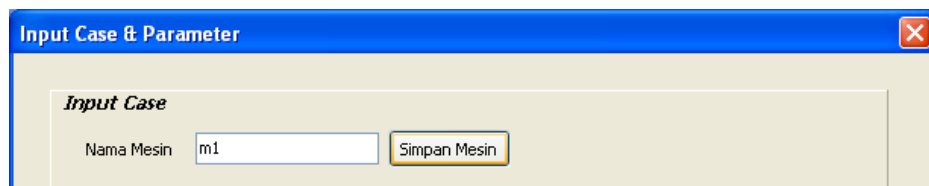
1. Jalankan program *Job Shop GA.exe*



2. Klik *Input*



3. Masukkan jenis mesin, satu per satu, lalu klik Simpan Mesin.



4. Masukkan nama pekerjaan (*job*), lalu masukkan berapa jumlah operasi untuk job tersebut. Masukkan Job dan jumlah operasinya satu simpan satu per satu.

Input Case & Parameter

Input Case

Nama Mesin:

Input Job

Nama Job: Jumlah Operasi:

5. Untuk memasukkan data mesin yang digunakan dan waktu operasi pada job yang bersangkutan, pilih job yang diinginkan lalu isi satu persatu data tersebut.

Input Operasi **Save Case**

Pilih Job: Nama File:

No	Nama Operasi	Nama Mesin	Waktu
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

6. Masukkan nama operasi dan mesin yang digunakan beserta waktunya pada kolom yang disediakan.

Input Operasi **Save Case**

Pilih Job: Nama File:

No	Nama Operasi	Nama Mesin	Waktu
1	potong	m1	10
2	bubut	m2	30
3	las	m3	15
4	<input type="text"/>	<input type="text"/>	0.00

7. Simpan Kasus ! klik *save case*.

Input Operasi

Pilih Job: roda Nama File: pembuatan motor

No	Nama Operasi	Nama Mesin	Waktu
1	potong	m1	10
2	bubut	m2	30
3	las	m3	15
4			0.00

Save Case

❖ Data Parameter

Input Parameter

Generasi: Pc: Populasi: Pm: Nama File: Save Parameter Exit

Masukkan besar parameter yang diinginkan, literatur menyarankan generasi yang besar (penulis menyarankan sekitar 100), dan populasi sebesar 20-100. Bila ingin menyimpan set parameter tersebut, klik *save parameter*.

⊕ Menjalankan Program

1. Klik tombol **Proses** pada menu utama.



2. Masukkan nama file kasus dan nama file set parameter untuk me-load file yang telah disimpan sebelumnya. Lalu klik load case atau load parameter
3. lalu klik Proses

Form Proses

Load Case

5x6

Jumlah Mesin: 6

Permutasi: 12541132800

Jumlah Job: 5

Nama File:

Load Parameter

Generasi: 100 Pc: 0.90

Populasi: 10 Pm: 0.01

Nama File:

Proses Exit

⊕ Melihat Solusi Penjadwalan

Lihat *makespan* yang di hasilkan, untuk melihat urutan penjadwalan di mesin yang diinginkan, klik jenis mesin.

Result

Makespan: 260.00

Generasi Terbaik: 1

Mesin: M1

No	Job	Operasi	Waktu	Mulai	Akhir
1	A	1	20.00	0.00	20.00
2	E	2	25.00	45.00	70.00
3	C	4	40.00	110.00	150.00
4	B	4	35.00	155.00	190.00

Ver 1.2

LAMPIRAN 4

PERHITUNGAN ANALISIS PARAMETER

PERCOBAAN ENAM KASUS

❖ Jumlah Generasi dalam perhitungan ditetapkan sebesar 100 generasi.

❖ Matriks Routing

○ Kasus 1 (5 job 6 mesin)

JOB	KET	OPERASI ke -					
		1	2	3	4	5	6
1	Mesin	1	2	3	4	6	1
	t(menit)	20	15	25	35	30	10
2	Mesin	5	6	2	1	6	
	t(menit)	30	15	25	35	30	
3	Mesin	2	4	3	1	4	6
	t(menit)	25	35	40	40	30	10
4	Mesin	4	2	6	2	1	
	t(menit)	35	40	25	25	30	
5	Mesin	5	1	2	4	1	6
	t(menit)	15	25	40	25	30	10

○ Kasus 2 (5 job 6 mesin)

JOB	KET	OPERASI ke -		
		1	2	3
1	Mesin	1	2	-
	t	90'	180'	-
2	Mesin	1	2	-
	t	120'	120'	-
3	Mesin	1	3	4
	t	180'	120'	240'
4	Mesin	1	3	4
	t	60'	120'	180'
5	Mesin	5	6	2
	t	90'	540'	180'

o Kasus 3 (8 job 6 mesin)

JOB	KET	OPERASI ke -					
		1	2	3	4	5	6
1	Mesin	1	2	3	4	3	6
	t(menit)	21	23	24	25	25	30
2	Mesin	3	4	5	6	4	3
	t(menit)	35	34	30	12	32	34
3	Mesin	3	2	1	4	5	2
	t(menit)	23	24	25	32	24	45
4	Mesin	4	2	4	2	3	4
	t(menit)	25	30	32	31	33	35
5	Mesin	2	3	4	2	3	4
	t(menit)	35	30	25	31	30	25
6	Mesin	1	2	3	6	5	4
	t(menit)	30	35	20	25	21	20
7	Mesin	3	5	6	2	1	3
	t(menit)	21	25	31	35	30	25
8	Mesin	3	5	6	1	3	2
	t(menit)	24	27	30	31	40	25

o Kasus 4

JOB	KET	OPERASI ke -			
		1	2	3	4
1	Mesin	1	2	3	2
	t(menit)	20	30	60	35
2	Mesin	1	3	4	2
	t(menit)	30	60	35	40
3	Mesin	4	3	2	3
	t(menit)	35	65	25	80
4	Mesin	1	4	2	4
	t(menit)	15	45	40	70

○ Kasus 5

<i>JOB</i>	KET	OPERASI ke -			
		1	2	3	4
1	Mesin	2	3	4	3
	t(menit)	20	32	60	40
2	Mesin	1	4		
	t(menit)	75	50		
3	Mesin	1	4	3	
	t(menit)	40	80	40	
4	Mesin	2	4	3	
	t(menit)	30	50	35	
5	Mesin	2	3	2	4
	t(menit)	60	70	30	40
6	Mesin	1	3	4	
	t(menit)	30	50	75	
7	Mesin	1	4	3	3
	t(menit)	20	40	40	25

o Kasus 6

JOB	KET	OPERASI ke -			
		1	2	3	4
1	Mesin	4	6		
	t(menit)	72	95		
2	Mesin	1	6		
	t(menit)	120	190		
3	Mesin	4	6		
	t(menit)	30	36		
4	Mesin	4	2		
	t(menit)	10	20		
5	Mesin	1	5	3	
	t(menit)	20	80	80	
6	Mesin	4	2		
	t(menit)	20	20		
7	Mesin	4	2		
	t(menit)	20	20		
8	Mesin	1	5		
	t(menit)	30	30		
9	Mesin	1	5	3	
	t(menit)	30	40	120	
10	Mesin	1	2		
	t(menit)	20	20		
11	Mesin	1	2	3	3
	t(menit)	20	20	240	120

❖ Tabel *makespan* hasil perhitungan.

Untuk mencari nilai Pc, nilai Pm di-set tetap sebesar 0.01. dan untuk mencari Pm, nilai Pc di set yang menghasilkan solusi terbaik.

○ Kasus 1

Pc	makespan
0.1	230
0.2	230
0.3	230
0.4	230
0.5	230
0.6	230
0.7	230
0.8	230
0.9	230
0.95	230

Pm	makespan
0.001	230
0.005	230
0.01	230
0.015	230
0.02	230
0.025	230
0.03	230
0.035	230
0.04	230
0.045	230

populasi	makespan
1	290
5	230
10	230
15	230
20	230
25	230
30	230
35	230

○ Kasus 2

Pc	makespan
0.1	810
0.2	810
0.3	810
0.4	810
0.5	810
0.6	810
0.7	810
0.8	810
0.9	810
0.95	810

Pm	makespan
0.001	810
0.005	810
0.01	810
0.015	810
0.02	810
0.025	810
0.03	810
0.035	810
0.04	810
0.045	810

populasi	makespan
1	810
5	810
10	810
15	810
20	810
25	810
30	810
35	810

○ Kasus 3

Pc	makespan
0.1	361
0.2	347
0.3	334
0.4	343
0.5	335
0.6	331
0.7	342
0.8	331
0.9	331
0.95	331

Pm	makespan
0.001	331
0.005	331
0.01	331
0.015	331
0.02	331
0.025	331
0.03	331
0.035	331
0.04	331
0.045	331

populasi	makespan
1	335
5	331
10	331
15	331
20	331
25	331
30	331
35	331

○ Kasus 4

Pc	makespan
0.1	415
0.2	415
0.3	430
0.4	425
0.5	435
0.6	415
0.7	420
0.8	415
0.9	417
0.95	415

Pm	makespan
0.001	415
0.005	417
0.01	415
0.015	415
0.02	415
0.025	417
0.03	415
0.035	425
0.04	415
0.045	415

populasi	makespan
1	482
5	447
10	427
15	425
20	420
25	415
30	415
35	415

○ Kasus 5

Pc	makespan
0.1	295
0.2	300
0.3	295
0.4	295
0.5	295
0.6	295
0.7	300
0.8	295
0.9	295
0.95	295

Pm	makespan
0.001	295
0.005	295
0.01	295
0.015	295
0.02	295
0.025	295
0.03	295
0.035	295
0.04	295
0.045	295

populasi	makespan
1	390
5	295
10	295
15	295
20	295
25	295
30	295
35	295

○ Kasus 6

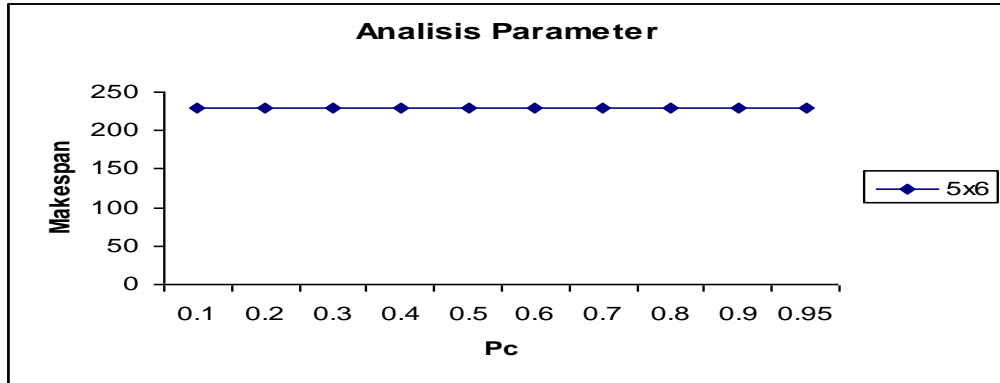
Pc	makespan
0.1	600
0.2	600
0.3	600
0.4	600
0.5	600
0.6	600
0.7	630
0.8	600
0.9	600
0.95	600

Pm	makespan
0.001	600
0.005	600
0.01	600
0.015	600
0.02	600
0.025	600
0.03	600
0.035	600
0.04	600
0.045	600

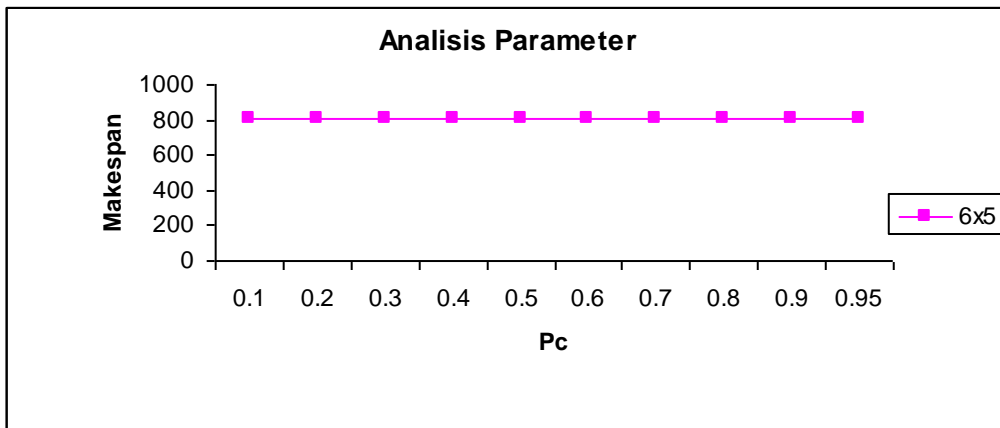
populasi	makespan
1	710
5	600
10	600
15	600
20	600
25	600
30	600
35	600

❖ Grafik variasi Pc

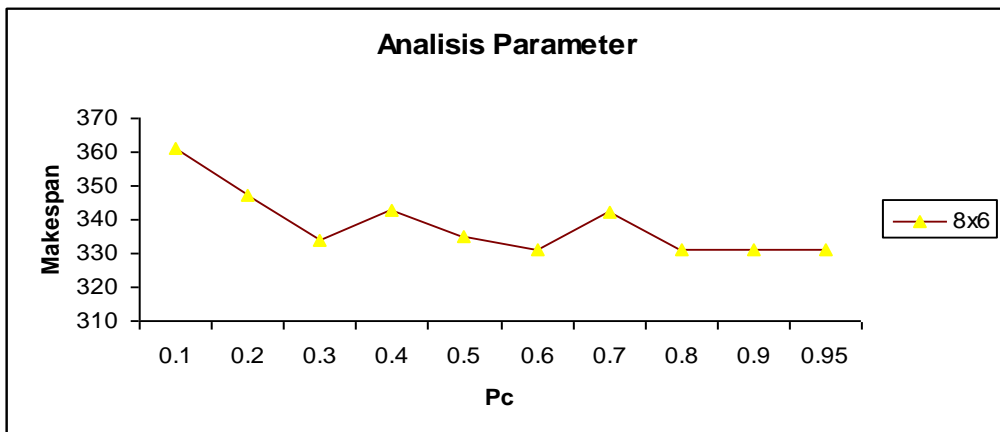
○ Kasus 1



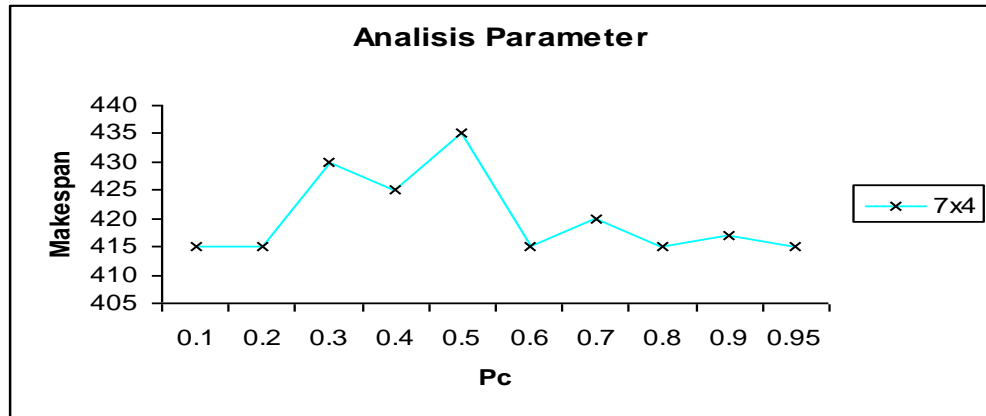
○ Kasus 2



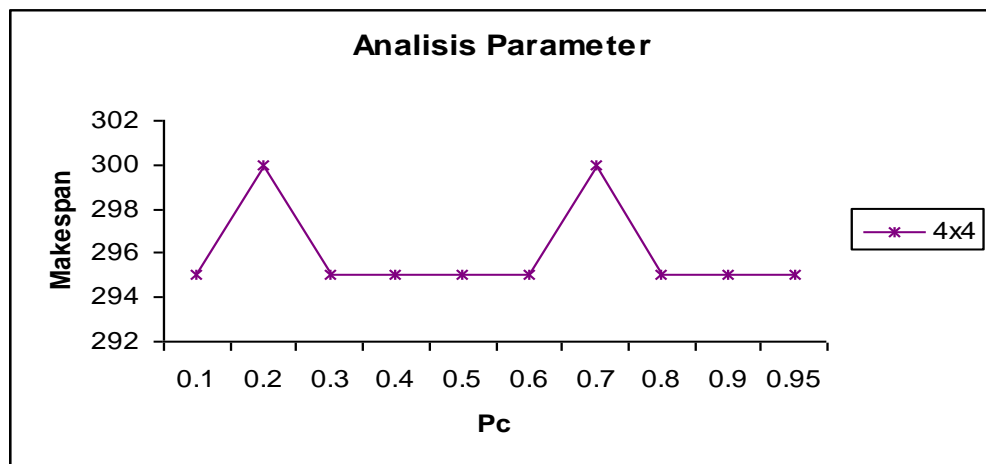
○ Kasus 3



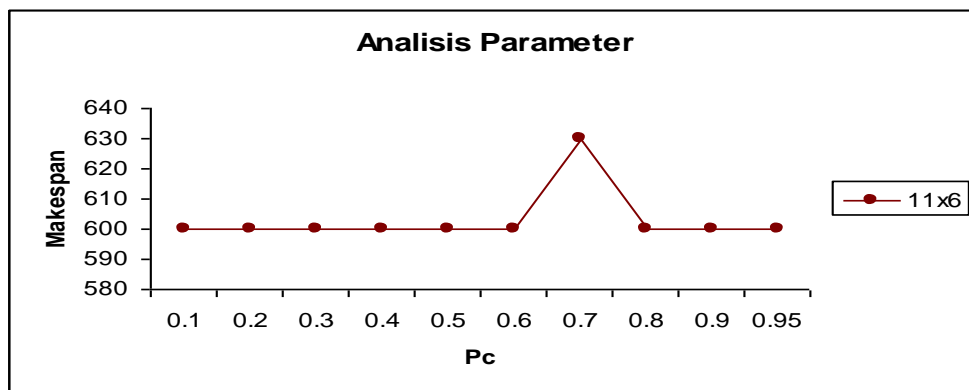
o Kasus 4



o Kasus 5

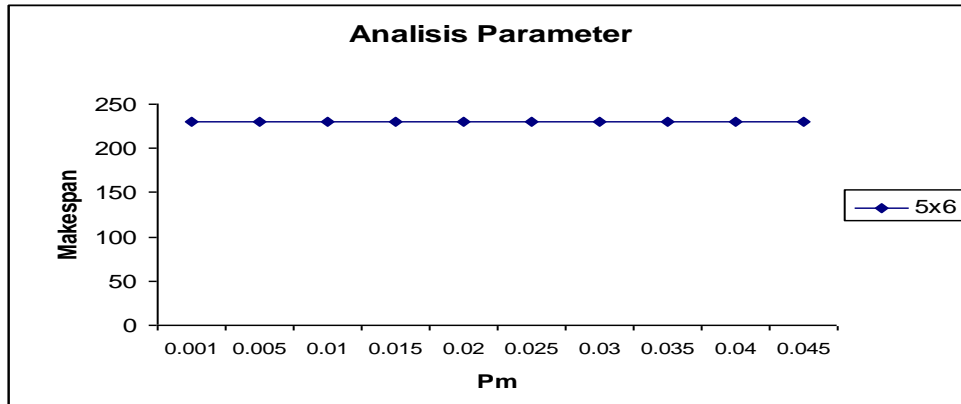


o Kasus 6

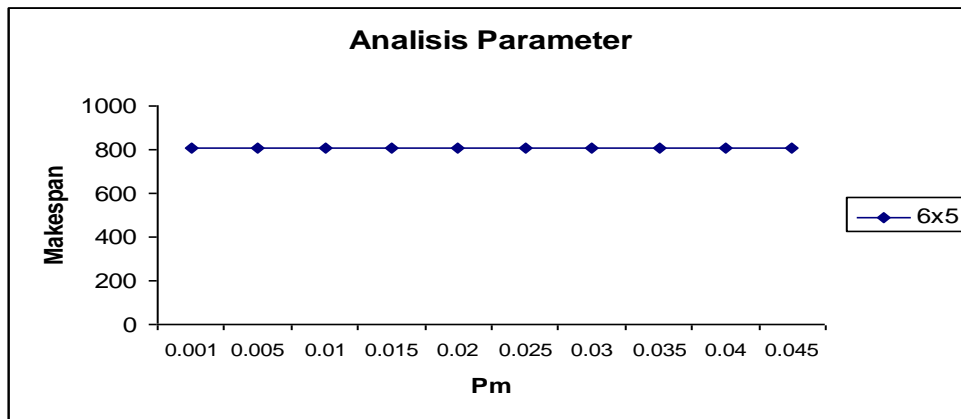


❖ Grafik variasi Pm

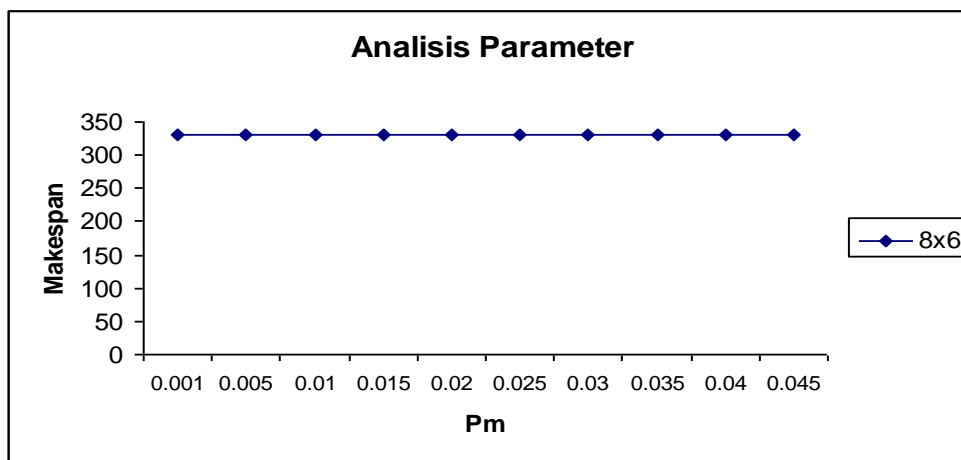
○ Kasus 1



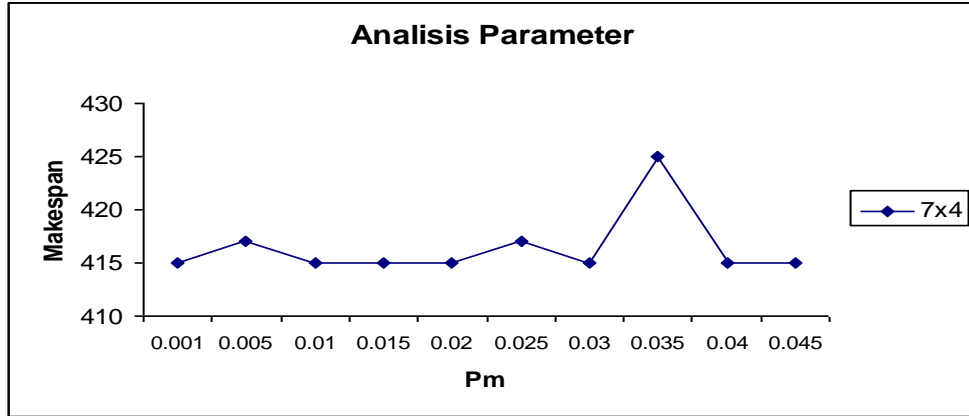
○ Kasus 2



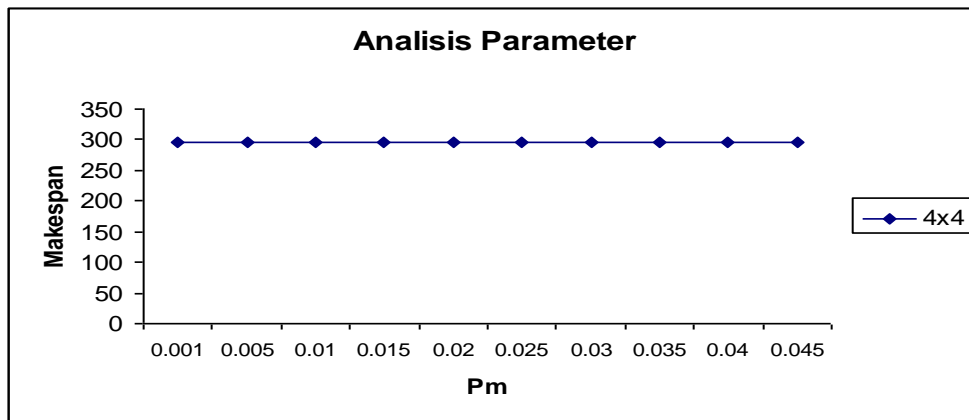
○ Kasus 3



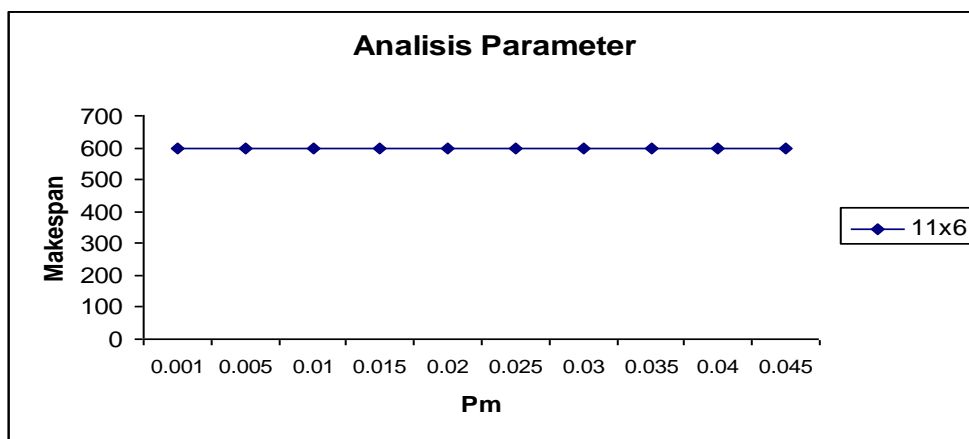
○ Kasus 4



○ Kasus 5

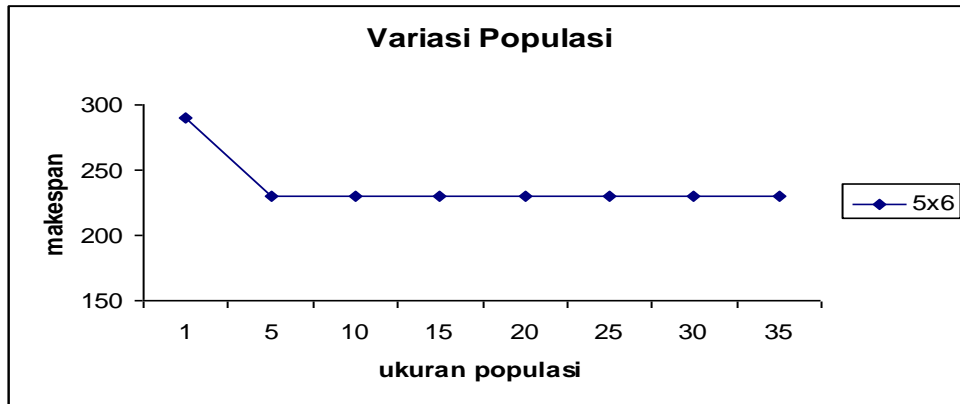


○ Kasus 6

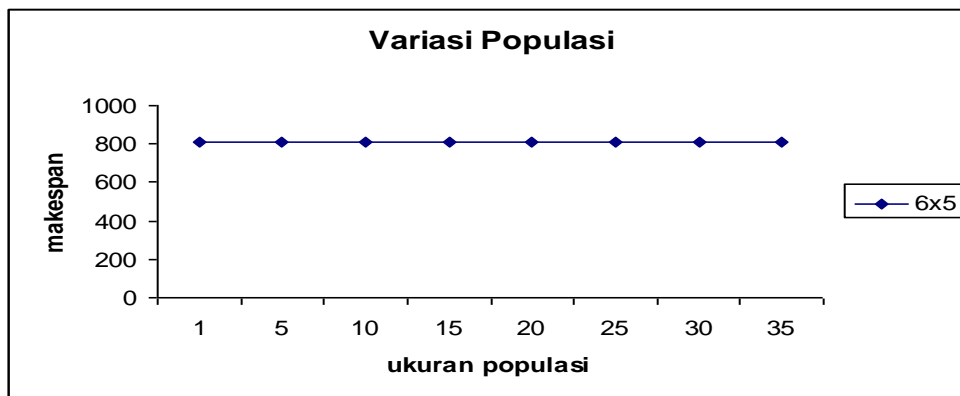


❖ Grafik variasi parameter ukuran populasi

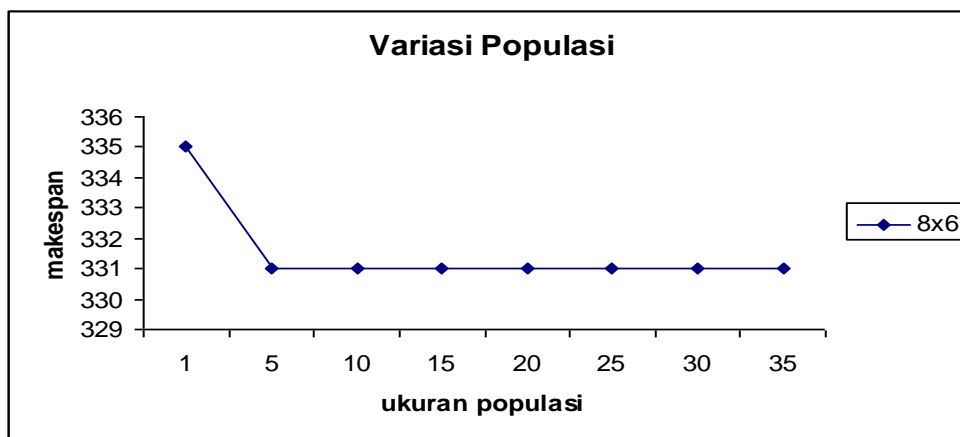
○ Kasus 1



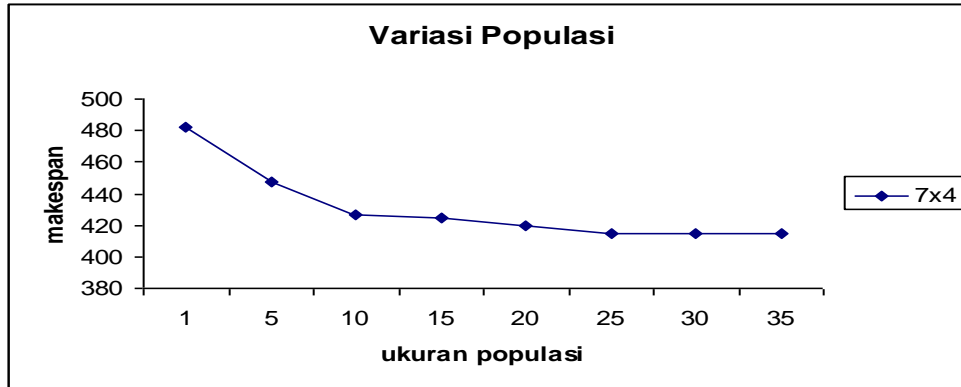
○ Kasus 2



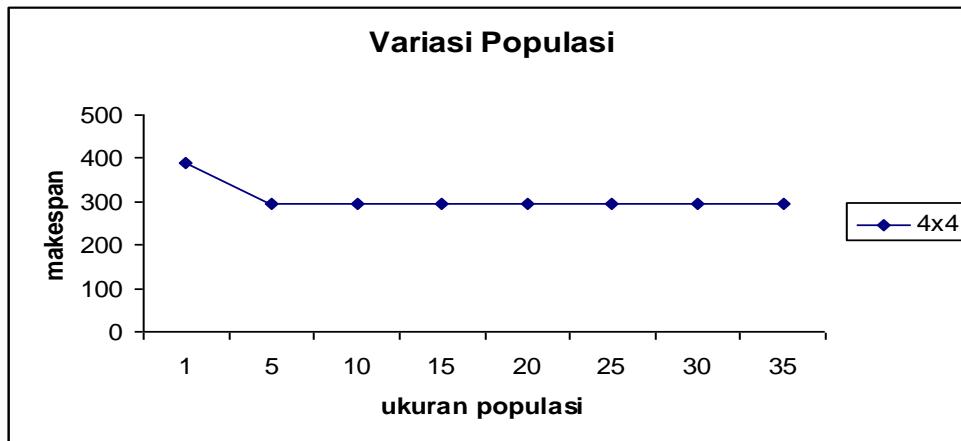
○ Kasus 3



○ Kasus 4



○ Kasus 5



○ Kasus 6

