

BAB IV
PENGOLAHAN DATA DAN ANALISIS

4.1. MODEL OPTIMASI PENJADWALAN

Penjadwalan dari kegiatan bongkar muat barang ekspor kedalam kontainer dimodelkan dalam bentuk model matematika seperti di bawah ini :

Variabel : $a_{ij}^t = 0$ atau 1 ; dibongkar-muat atau tidaknya barang kontainer i dari truk ke j oleh tim t

b_{ij} =waktu mulai dibongkar-muatnya truk j ke kontainer i

c_{ij} =waktu selesai dibongkar-muatnya truk j ke kontainer i

Konstanta : WK_{ij} =waktu kedatangan truk j ke kontainer i

WBM_{ij} =waktu bongkar barang kontainer i dari truk j

BL = Biaya lembur

Kendala:

1. $b_{ij} \geq WK_{ij}$

$c_{ij} \leq b_{ij+1} \Rightarrow$ Waktu pembongkar-muatan truk berikutnya harus \geq waktu selesainya truk sebelumnya

$c_{ij} = b_{ij} + WBM_{ij} \sum_t a_{ij}^t \Rightarrow$ untuk semua t

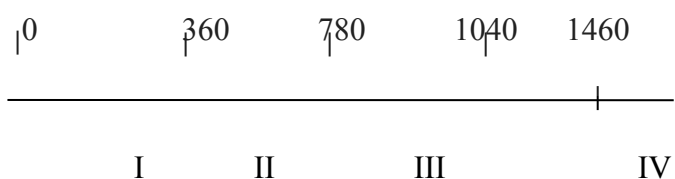
$c_{ij} \leq 1460 \rightarrow$ menunjukkan bahwa batas waktu lembur adalah pukul 24.00

2. Suatu tim tidak dapat bekerja pada waktu yang bersamaan

$b_{pq} \geq b_{ij} + WBM_{ij} a_{ij}^t \Rightarrow$ untuk semua i, j, t, dan p, q atau

$c_{pq} \leq c_{ij} - WBM_{ij} a_{ij}^t \Rightarrow$ untuk semua i, j, t, dan p, q

3.



Wilayah I dan II merupakan waktu kerja pada hari ke-1 dan Wilayah III dan IV merupakan wilayah kerja pada hari ke-2.

Wilayah II dan IV merupakan wilayah waktu lembur dimana batas maksimal dari waktu lembur adalah pukul 24.00

p_{ij}^t = waktu lembur bongkar-muat pada wilayah II dan IV

jika $p_{ij}^t \leq 780$ maka $p_{ij}^t = b_{ij} + WBM_{ij} \sum_t a_{ij}^t - 360$

jika $p_{ij}^t \geq 780$ maka $p_{ij}^t = b_{ij} + WBM_{ij} \sum_t a_{ij}^t - 1040$

Jika $p_{ij}^t \geq 420$ maka $p_{ij}^t = p_{ij}^t$ atau $p_{ij}^t = 0$

Fungsi Tujuan :

a. Lembur wilayah II

Adalah p_{ij}^t yang terbesar dinotasikan sebagai U^t

Dibandingkan p_{ij}^t dengan U^t

$$p_{ij}^t \leq U^t$$

b. Lembur wilayah IV

Adalah q_{ij}^t yang terbesar dinotasikan sebagai V^t

Dibandingkan q_{ij}^t dengan V^t

Di mana fungsi tujuan dari model optimasi ini adalah meminimumkan Biaya buruh baik itu biaya buruh bekerja maupun biaya lembur dari kegiatan yang terjadi, yaitu :

$$\text{Tujuan : } f \min Z = BL / \text{jam} \sum_t U^t + BL / \text{jam} \sum_t V^t$$

4.2. PENYUSUNAN ALGORITMA

Data penelitian ini diolah dengan menggunakan bahasa pemrograman Matlab. Bahasa pemrograman ini dipilih karena banyak memberi kemudahan untuk mengimplementasikan algoritma-algoritma yang banyak menggunakan operasi matriks.

Matlab adalah suatu bahasa pemrograman tingkat tinggi yang diperuntukkan untuk komputasi teknis. Matlab mengintegrasikan aspek komputasi, visualisasi, dan pemrograman dalam suatu lingkungan yang mudah dilakukan²³. Dalam memvisualisasikan sebuah objek, Matlab memiliki kemampuan merotasi obyek tanpa mengubah programnya. Konstruksi penyelesaian komputasi teknis dengan Matlab dapat dilakukan lebih cepat dibandingkan dengan bahasa pemrograman tradisional, seperti C, C++, dan Fortran. Matlab menyediakan fungsi-fungsi matematis untuk aljabar linear, statistik, optimasi, dan lainnya. Selain itu, Matlab juga menyediakan fitur-fitur dokumentasi dan integrasi algoritma berbasis Matlab dengan bahasa dan aplikasi lain, seperti C, C++, Fortran, Java, COM, dan Microsoft Excel. Bahasa Matlab memudahkan operasi-operasi vektor dan matriks yang merupakan dasar bagi permasalahan di bidang teknik dan ilmiah²⁴

4.2.1. Langkah-Langkah Penyusunan Algoritma

Diagram alir penyusunan program algoritma *Differential Evolution* (DE) untuk menyelesaikan penjadwalan *job shop* pada dapat dilihat pada gambar 2.2 (flowchart kerja) dan program yang telah dibuat dapat dilihat pada Lampiran 2. (lampiran M script file) Prosedur penyusunan program tersebut dapat diuraikan sebagai berikut:

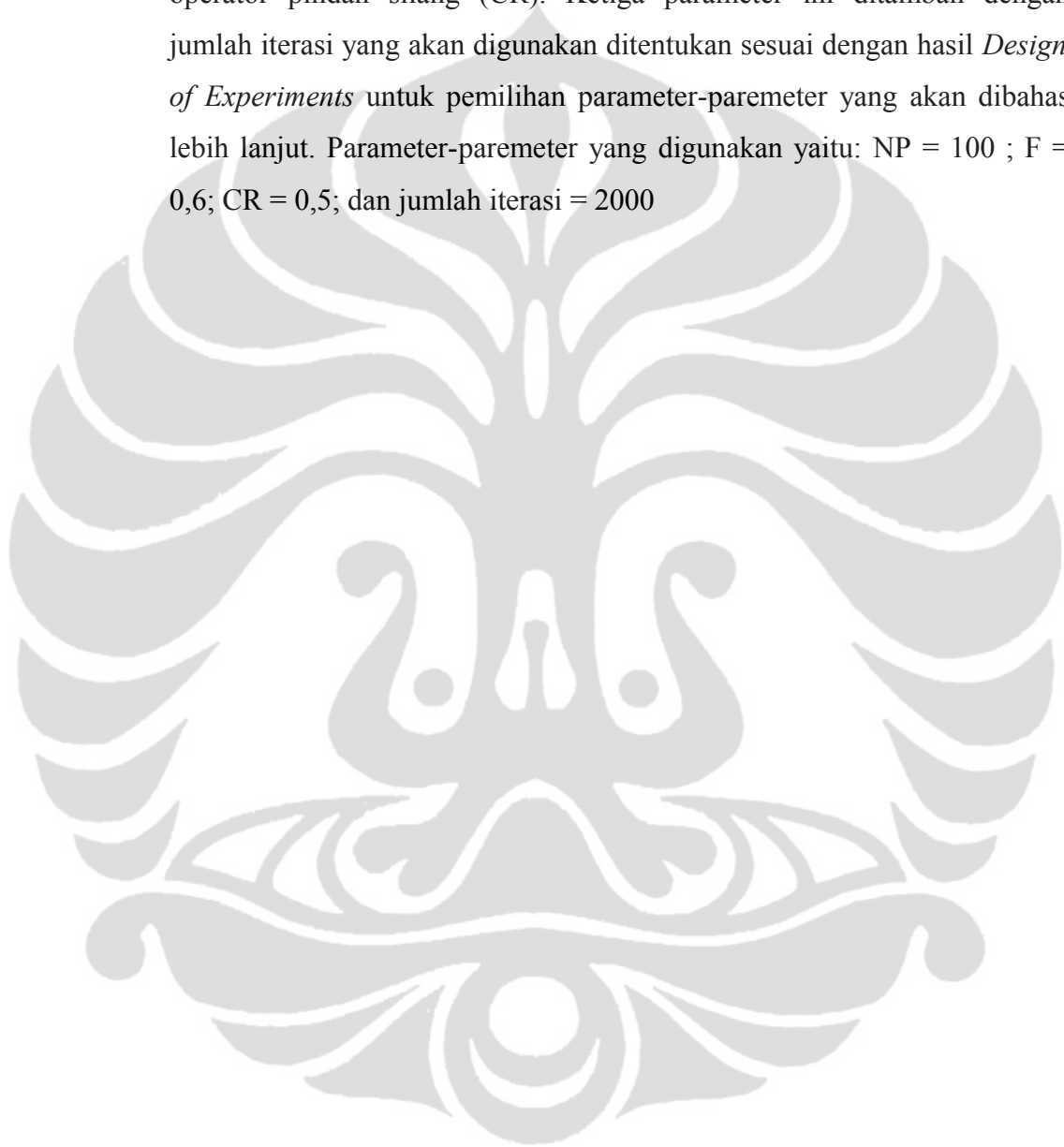
²³ Budi Santosa, *Matlab untuk Statistika & Teknik Optimasi - Aplikasi untuk Rekayasa & Bisnis*, Graha Ilmu, Yogyakarta, 2008, p.1.

²⁴ <http://www.mathworks.com/products/matlab/description1.html>

1. Inisialisasi populasi awal

- Melakukan penyetelan pada generasi ke-0.

Pada awal perhitungan sebelum generasi (iterasi) dimulai, dilakukan input parameter-parameter, yaitu ukuran populasi (NP), operator mutasi (F), dan operator pindah silang (CR). Ketiga parameter ini ditambah dengan jumlah iterasi yang akan digunakan ditentukan sesuai dengan hasil *Design of Experiments* untuk pemilihan parameter-parameter yang akan dibahas lebih lanjut. Parameter-parameter yang digunakan yaitu: NP = 100 ; F = 0,6; CR = 0,5; dan jumlah iterasi = 2000



- Menentukan populasi awal (populasi target awal)

Ukuran populasi menyatakan jumlah individu dalam populasi. Satu individu dinyatakan dengan satu kolom. Tiap individu memiliki 120 gen/dimensi (jumlah *job*), dinyatakan dengan jumlah baris. Populasi awal dibentuk dengan setiap dimensi untuk setiap individu dicari secara acak dengan rumus:

$\text{batas_bawah} + (\text{batas_atas} - \text{batas_bawah}) \times \text{bilangan acak}$

Nilai-nilai input dari rumus tersebut yaitu -1 (batas bawah), 1 (batas atas), dan bilangan acak antara 0 dan 1. Karena populasi terdiri dari 100 individu (nilai NP) dan setiap individu terdiri dari 120 dimensi, maka populasi awal merupakan matriks berukuran 120 x 100. Pada algoritma DE ini, definisi individu dan vektor adalah sama. Setiap individu awal pada populasi awal ini diurutkan menjadi vektor permutasi sehingga dapat mempresentasikan urutan *job*.
- Menentukan vektor permutasi (urutan)

Nilai dimensi pertama hingga ke-120 setiap individu awal memiliki nilai yang berbeda-beda. Untuk setiap individu awal ini dilakukan pengurutan nilai dimensi dari yang terkecil hingga yang terbesar. Pengurutan tersebut akan menghasilkan vektor berdimensi 120 dengan nilai setiap dimensinya berupa indeks hasil pengurutan. Indeks-indeks tiap individu inilah yang nantinya akan menjadi urutan-urutan pengerjaan 120 *job* sesuai data penelitian. Sebagai contoh, jika ada tiga *job* yang dikerjakan dan terdapat tiga dimensi pada suatu individu, yaitu 0,26; -0,54; dan 0,78; yang berturut-turut terdapat pada dimensi ke-1, dimensi ke-2, dan dimensi ke-3, maka nilai dimensi tersebut pada vektor permutasi pada individu itu berturut-turut adalah 2, 1, dan 3. Artinya, urutan pengerjaan *job* yaitu 2-1-3. Proses ini berlaku untuk seluruh individu.
- Mengevaluasi setiap individu

Setelah mendapatkan vektor-vektor urutan *job* setiap individu, kemudian dibuat fungsi objektifnya (Total Biaya Lembur). Pengevaluasian dilakukan dengan cara menghubungkan vektor tersebut dengan data waktu operasi sehingga fungsi objektif dapat dihitung. Pengevaluasian ini dimaksudkan untuk mengetahui individu pada generasi (iterasi) awal yang memiliki Total Biaya Lembur terkecil yang selanjutnya akan diturunkan pada generasi selanjutnya.

- Memperbaharui generasi (iterasi)

Populasi individu pada iterasi awal akan berevolusi membentuk populasi individu iterasi baru. Individu-individu mengalami evolusi melalui serangkaian proses, yang dimulai dengan proses mutasi, proses pindah silang, dan proses penyeleksian. Jika generasi awal disimbolkan sebagai $t = 0$, maka iterasi baru disimbolkan sebagai $t = t + 1$.

2. Proses Mutasi

Proses mutasi merupakan proses utama dari algoritma DE karena merupakan langkah pertama individu dalam berevolusi. Sesuai nama algoritma (Differential Evolution), proses ini menekankan perbedaan nilai atau selisih dari dua vektor acak (vektor acak 1 dan vektor acak 2) yang memunculkan *difference vector*. Selanjutnya, *difference vector* tersebut akan dikalikan dengan operasi mutasi (F), yang hasilnya kemudian dijumlahkan dengan vektor target. Proses mutasi dianggap sangat penting sehingga penentuan parameter untuk mutasi pun sangat menentukan hasil yang diperoleh.

3. Proses pindah silang

Proses pindah silang merupakan rekombinasi antara individu target dengan individu mutan yang menghasilkan individu *trial*. Nilai dimensi individu *trial* ini sebagian berasal dari individu target dan sebagian lagi berasal dari individu mutan, dengan mempertimbangkan operator pindah silang (CR) dan bilangan acak. Jika bilangan acak r (antara 0 sampai 1) yang dihasilkan lebih kecil atau sama nilainya dengan CR maka yang berpeluang menjadi nilai dimensi ke- k individu *trial* adalah nilai dimensi ke- k individu mutan, begitu pun sebaliknya. Nilai CR berada pada rentang 0 sampai 1.

4. Proses penyeleksian

Penyeleksian dilakukan antara individu target dan individu *trial*. Proses tersebut bertujuan untuk menentukan individu yang layak menjadi anggota pada generasi berikutnya. Proses penyeleksian dilakukan dengan membandingkan nilai fungsi objektif individu target dengan individu *trial*. Individu yang memiliki nilai total Biaya Lembur akan menjadi individu anggota populasi dari generasi berikutnya. Dengan adanya proses penyeleksian ini, maka populasi individu generasi berikutnya akan menjadi lebih baik.

5. Terminasi

Pada penelitian ini, kriteria terminasi yang digunakan adalah jumlah iterasi (generasi). Proses pembentukan iterasi baru akan terus berulang sampai jumlah iterasi yang telah ditentukan tercapai. Jumlah iterasi yang ditentukan adalah 2000 iterasi berdasarkan percobaan yang telah dilakukan. Jadi, jika iterasi sudah mencapai 2000 kali, maka program komputer secara otomatis berhenti melakukan perhitungan. Penentuan jumlah iterasi juga dipengaruhi oleh lamanya waktu perhitungan. Jumlah iterasi yang sangat besar memiliki kemungkinan untuk mencapai hasil yang optimal, tetapi waktu perhitungan yang dibutuhkan akan sangat lama.

4.2.2. Verifikasi dan Validasi Program

Sebelum data penelitian diolah ke dalam program yang telah disusun, verifikasi dan validasi program harus dilakukan. Model program dikatakan telah terverifikasi apabila telah berjalan sesuai konseptual model, dimana ada perubahan output, yaitu total biaya lembur atas nilai kelebihan waktu operasi positif. Untuk bagian-bagian selanjutnya, Lembur yang dimaksud adalah bila kelebihan waktu operasi terjadi. Artinya, lembur hanya dihitung bila waktu penyelesaian lebih lama daripada waktu operasi yang semestinya. Oleh karena itu, pada fungsi tujuan, Jika lembur negatif, maka biaya lembur yang negatif tersebut akan berubah menjadi bernilai nol sebelum dikalikan biaya lembur per jamnya.

Selanjutnya, dilakukan validasi program. Validasi terhadap program dilakukan dengan memasukkan data *dummy*. Hasil run program dengan data *dummy* kemudian dibandingkan dengan perhitungan manual. Jika hasil keduanya sama, maka program telah tervalidasi.

Data *dummy* yang digunakan adalah 5 buah *job* dan setiap *job* memiliki 2 buah operasi dengan 2 tim buruh yang mengerjakan. Waktu operasi setiap rute untuk setiap jenis dapat dilihat pada tabel 4.1. Waktu operasi yang bernilai nol menunjukkan bahwa rute tersebut tidak dilewati oleh jenis yang bersangkutan. Jumlah operasi dan jumlah mesin ini sesuai dengan data penelitian. Parameter algoritma DE untuk validasi ini juga ditentukan dan dapat dilihat pada tabel 4.2. parameter yang digunakan adalah dengan

nilai CR : 0,8 dan F : 0,5 karena pada level tersebut nilai yang dihasilkan mencapai angka maksimal.

Tabel 4.1. Data *dummy* untuk Validasi

No	ij	BARANG DATANG		
		Kendaraan	Total Kubikasi	Waktu Kedatangan Kendaraan
1	10 1	Kendaraan 1	40	160
	10 2	Kendaraan 2	5	170
2	10 3	Kendaraan 3	2	170
	10 4	Kendaraan 4	5	180
	11 1	Kendaraan 5	13	190

Tabel 4.2. Parameter yang Digunakan dalam Validasi

Parameter	Nilai
Ukuran Populasi	5
Jumlah Iterasi	1
Operator Mutasi	0.5
Operator Pindah Silang	0.8

4.2.2.1. Hasil Perhitungan Manual

Langkah-langkah perhitungan manual untuk validasi program adalah sebagai berikut:

1. Melihat populasi target yang didapat dari program.

Populasi target, yang merupakan populasi awal, berisi bilangan acak antara -1 dan 1 sesuai dengan rumus yang telah dibuat pada program. Matriks populasi target untuk data *dummy* ini berukuran 5 x 8 (ukuran populasi x jumlah *job*). Matriks populasi target ini adalah matriks yang memiliki populasi sebanyak 5 individu atau 5 vektor. Jadi, 1 kolom merepresentasikan 1 individu. Tiap kolom memiliki 5 baris, berarti 1 individu terdiri dari 5 gen (*job*).

Tabel 4.3. Populasi Target (Hasil *Run* Program)

Individu				
1	2	3	4	5
0.2423	0.5073	0.319	0.2468	-0.3792
0.1204	0.3193	-0.6333	0.3718	0.5582
-0.5119	-0.5719	0.2731	0.3547	-0.3854
0.644	0.2042	-0.6594	0.7537	0.8534
-0.4736	0.2099	0.0792	-0.9742	0.3574

2. Melakukan pengurutan (permutasi) pada setiap individu dari populasi target. Urutan setiap individu didapatkan dengan cara mengurutkan bilangan acak setiap kolom pada tabel 4.3 dari yang terkecil hingga terbesar. Sebagai contoh, pada kolom pertama, urutan bilangan acak dari yang terkecil hingga terbesar adalah -0,5119; -0,4736; 0,1204; 0,2423; 0,644. Bilangan-bilangan tersebut berturut-turut terdapat pada kolom ke 3, 5, 2, 1, dan 4. Jadi, urutan pengerjaan *job* pada individu (kolom) tersebut adalah 3-5-2-1-4. Setiap individu memiliki urutan *job* masing-masing.
3. Setelah didapat urutan *job* tersebut, maka untuk setiap individu dicari fungsi objektifnya, yaitu nilai total biaya lembur
 Prosesnya adalah sebagai berikut:
 - Untuk setiap individu, sesuai urutan *job* masing-masing, dicari waktu penyelesaian untuk setiap *job* dengan terus menambahkan waktu proses di setiap proses yang bersangkutan dengan waktu penyelesaian.
 - Kemudian waktu penyelesaian setiap *job* akan dikurangi dengan batas waktu dimulainya lembur pada setiap harinya (dimulai dari menit ke 250 dan hari II). apabila nilai waktu lembur bernilai negatif (*earliness*), nilai tersebut akan berubah menjadi nol.
 - Selanjutnya, waktu lembur setiap tim buruh dikalikan dengan biaya lembur per jam per tim. Lalu biaya lembur seluruh tim dijumlahkan sehingga didapatlah total biaya lembur.

Tabel-tabel di bawah ini menunjukkan proses perhitungan total biaya lembur untuk setiap tim buruh dalam populasi. Dari 5 Individu yang ditampilkan pada penulisan ini adalah sebanyak 3 individu.

Tabel 4.4. Perhitungan Waktu Operasi setiap *Job* Individu 1 Populasi Target

Mesin	1	2	3	Waktu akhir
1	Kend.3	Kend.1	Kend.4	348
2	Kend.5	Kend.2		379

Tabel 4.5. Perhitungan Total Biaya Lembur Individu 1 Populasi Target

mesin	1	2
Waktu Selesai	348	379
Due date	250	205
Penalti	98	174
Biaya Lembur	196.000	348.000
Total	544.000	

Tabel 4.6. Perhitungan Waktu Operasi setiap *Job* Individu 2 Populasi Target

Mesin	1	2	3	Waktu akhir
1	Kend.1	Kend.4		414
2	Kend.3	Kend.2	Kend.5	305

Tabel 4.7. Perhitungan Total Biaya Lembur Individu 2 Populasi Target

mesin	1	2
Waktu Selesai	414	305
Due date	250	205
Penalti	164	100
Biaya Lembur	328.000	200.000
Total	528.000	

Tabel 4.8. Perhitungan Waktu operasi setiap *Job* Individu 3 Populasi Target

Mesin	1	2	3	Waktu akhir
1	Kend.1	Kend.5		367
2	Kend.3	Kend.2	Kend.4	379

Tabel 4.9. Perhitungan Total Biaya lembur Individu 3 Populasi Target

mesin	1	2
Waktu Selesai	367	379
Due date	250	205
Penalti	117	174
Biaya Lembur	234.000	348.000
Total	582.000	

4. Setelah didapat nilai total biaya lembur seluruh individu, maka dipilih individu yang memiliki total biaya lembur terkecil, yaitu individu 1 Individu 1 memiliki nilai total Biaya lembur sebesar Rp. 528.000 Jadi, individu 1 pada populasi target akan menjadi vektor target untuk tahap selanjutnya.
5. Lalu dibuatlah populasi mutan dengan input vektor target dan dua vektor acak. Penjelasannya adalah sebagai berikut:
 - Seperti telah dibahas sebelumnya, vektor target didapat dari individu populasi target yang memiliki total waktu lembur terkecil, yaitu individu 1 Semua kolom (individu) pada vektor target berisi bilangan-bilangan yang sama, yaitu individu 1 pada populasi target.

Tabel 4.10. Vektor Target

Individu				
1	2	3	4	5
0.3190	0.3190	0.3190	0.3190	0.3190
-0.6333	-0.6333	-0.6333	-0.6333	-0.6333
0.2731	0.2731	0.2731	0.2731	0.2731
-0.6594	-0.6594	-0.6594	-0.6594	-0.6594
0.0792	0.0792	0.0792	0.0792	0.0792

- Kemudian dipilih dua vektor acak yang setiap kolomnya (individu) berasal dari populasi target dengan suatu aturan yaitu antara vektor target, vektor acak 1, dan vektor acak 2 pada kolom tertentu tidak boleh sama. Sebagai contoh, pada kolom pertama vektor target, vektor acak 1, dan vektor acak 2, angkanya berturut-turut berasal dari populasi target pada kolom 3,1, dan 2. Ketiga kolom tersebut berbeda, tidak boleh ada yang sama.

Tabel 4.11. Vektor Acak 1 (Hasil *Run* Program)

Individu				
1	2	3	4	5
0.2423	0.2423	0.5073	0.5073	-0.3792
0.1204	0.1204	0.3193	0.3193	0.5582
-0.5119	-0.5119	-0.5719	-0.5719	-0.3854
0.6440	0.6440	0.2042	0.2042	0.8534
-0.4736	-0.4736	0.2099	0.2099	0.3574

Tabel 4.12. Vektor Acak 2 (Hasil *Run* Program)

Individu				
1	2	3	4	5
0.5073	0.5073	0.2468	0.2468	0.2423
0.3193	0.3193	0.3718	0.3718	0.1204
-0.5719	-0.5719	0.3547	0.3547	-0.5119
0.2042	0.2042	0.7537	0.7537	0.6440
0.2099	0.2099	-0.9742	-0.9742	-0.4736

- Setelah didapat vektor target, vektor acak 1, dan vektor acak 2, kemudian terbentuk populasi mutan (tabel 4.13.). Populasi ini didapatkan dengan menggunakan rumus:

populasi mutan = vektor target + (vektor acak 1 – vektor acak 2)*konstanta mutasi (F).

Rumus ini berlaku untuk setiap gen. Jadi, sebagai contoh, pada baris pertama kolom pertama populasi mutan, yaitu 0,1070 didapatkan dari $0,3190 + (0,2423 - 0,5073) \cdot 0,6$. Angka-angka 0,3190; 0,2423; dan 0,5073 secara berurutan adalah angka-angka di baris pertama kolom pertama dari vektor target, vektor acak 1,

dan vektor acak 2. Sedangkan 0,6 adalah nilai dari operator mutasi (konstanta mutasi).

Tabel 4.13. Populasi Mutan

Individu				
1	2	3	4	5
0.1070	0.1070	0.5274	0.5274	-0.1782
-0.7923	-0.7923	-0.6753	-0.6753	-0.2831
0.3210	0.3210	-0.4682	-0.4682	0.3743
-0.3076	-0.3076	-1.0989	-1.0989	-0.4919
-0.4676	-0.4676	1.0265	1.0265	0.7440

6. Setelah populasi mutan didapat, maka dibentuklah populasi *trial*. Setiap gen individu populasi *trial* ini berasal dari gen individu populasi target atau gen individu populasi mutan. Untuk memilih antara gen individu populasi target atau populasi mutan, digunakan bilangan acak antara 0 dan 1, dan operator pindah silang (CR) sebesar 0,5. Jika bilangan acak yang keluar bernilai kurang dari atau sama dengan 0,5; maka nilai suatu gen pada populasi *trial* berasal dari gen yang bersangkutan dari populasi mutan. Begitu pun sebaliknya, jika lebih dari 0,5; maka berasal dari populasi target. Tabel 4.14 menunjukkan populasi *trial*. Gen-gen yang bercetak tebal pada tabel tersebut adalah gen-gen yang berasal dari populasi mutan.

Tabel 4.14. Populasi *Trial*

Individu				
1	2	3	4	5
0.2423	0.5073	0.5274	0.5274	-0.3792
0.1204	0.3193	-0.6333	0.3718	0.5582
0.3210	0.3210	-0.4682	-0.4682	0.3743
0.6440	-0.3076	-1.0989	0.7537	-0.4919
-0.4736	0.2099	0.0792	1.0265	0.3574

7. Melakukan pengurutan (permutasi) pada setiap individu dari populasi *trial*. Caranya sama dengan pengurutan pada setiap individu dari populasi target.

8. Setelah didapatkan urutan *job* dari setiap individu populasi *trial*, maka dari setiap individu tadi, dicari total biaya Lembur masing-masing (tabel 4.15. sampai tabel 4.30.) diman akan ditampilkan dari perhitungan 3 individu.

Tabel 4.15. Perhitungan Waktu operasi setiap *Job* Individu 1 Populasi *Trial*

Mesin	1	2	3	Waktu akhir
1	Kend.3	Kend.4	Kend.5	354
2	Kend.1	Kend.2		379

Tabel 4.16. Perhitungan Total Biaya lembur Individu 1 Populasi *Trial*

mesin	1	2
Waktu Selesai	354	379
Due date	250	205
Penalti	104	174
Biaya Lembur	208.000	348.000
Total	556.000	

Tabel 4.17. Perhitungan Waktu operasi setiap *Job* Individu 2 Populasi *Trial*

Mesin	1	2	3	Waktu akhir
1	Kend.2	Kend.4	Kend.5	407
2	Kend.1	Kend.3		325

Tabel 4.18. Perhitungan Total Biaya lembur Individu 2 Populasi *Trial*

mesin	1	2
Waktu Selesai	407	325
Due date	250	205
Penalti	157	120
Biaya Lembur	314.000	240.000
Total	554.000	

Tabel 4.19. Perhitungan Waktu operasi setiap *Job* Individu 3 Populasi *Trial*

Mesin	1	2	3	Waktu akhir
1	Kend.1	Kend.3	Kend.5	416
2	Kend.2	Kend.4		302

Tabel 4.20. Perhitungan Total Biaya lembur Individu 3 Populasi *Trial*

mesin	1	2
Waktu Selesai	416	302
Due date	250	205
Penalti	166	97
Biaya Lembur	332.000	194.000
Total	526.000	

9. Kemudian dilakukan tahap seleksi. Setelah didapat nilai total biaya lembur setiap individu pada populasi *trial*, nilai-nilai tersebut dibandingkan dengan nilai total biaya lembur setiap individu populasi target. Perbandingan tersebut dapat dilihat pada tabel 4.21. Sebagai contoh, nilai total biaya lembur individu 1 populasi target dibandingkan dengan nilai total biaya lembur individu 1 populasi *trial*. Jika individu 1 populasi target memiliki nilai total biaya lembur yang lebih kecil dari individu 1 populasi *trial*, maka nilai populasi target, total biaya lembur, dan urutan (permutasi) untuk iterasi selanjutnya (iterasi 2) berturut-turut diambil dari individu 1 populasi target, total biaya lembur individu 1 populasi target, dan permutasi individu 1 populasi target, begitupun sebaliknya jika nilai total biaya lembur populasi target lebih besar atau sama dengan nilai total biaya lembur populasi *trial*.

Tabel 4.21. Perbandingan antara Total Biaya lembur Populasi Target dan Populasi *Trial*

Total Biaya Lembur Populasi Target (Rp)	Total Biaya Lembur Populasi Trial (Rp)	Asal Populasi Target Iterasi Selanjutnya
528000	556000	Pop Target
582000	554000	Pop Trial
544000	526000	Pop Trial

Dari tabel terlihat bahwa dari setiap perbandingan total biaya lembur populasi target dan populasi *trial*, nilai total biaya lembur yang terkecil adalah Rp. 526.000, nilai tersebut terletak pada individu 3 pada nilai-nilai total biaya lembur populasi target yang baru.

Maka didapatkan nilai total biaya lembur terbaik, yaitu Rp. 526.000 dengan urutan pengerjaan Mesin 1 Kendaraan 1-3-5 dan Mesin 2. Kendaraan 2-4

4.3. PENGOLAHAN DATA DAN HASIL

Program untuk pengolahan data penelitian ini dijalankan dengan spesifikasi komputer, yaitu Intel (R) Pentium(R) CPU 1.50GHz 1.5GHz, 512 MB of RAM. *Script M File* program untuk perhitungan jadwal dapat dilihat pada lampiran 1.

Data yang digunakan dalam simulasi adalah digunakan data dengan jumlah job sebanyak 122 yang terdiri 63 job hari jumat dan 59 job hari sabtu.

4.3.1. Hasil Penjadwalan

Perhitungan dilakukan pada kondisi dengan jumlah tim 4 dan 5 tim. Setelah diperlohe hasil akhir, maka hasil anatar kedua kondisi akan dibandingkan manakah hasil yang terbaik.

Perhitungan yang pertama dilakukan pada kondisi jumlah tim yang bekerja adalah 4 tim.

Tabel 4.22. Hasil Perhitungan Jadwal (4 Tim)

Hari ke-1

	Urutan Pekerjaan																Waktu Akhir (menit ke)	Penalti (menit)	Biaya Lembur (Rp)	
Tim 1	1	2	10	14	15	26	27	28	35	36	44	45	46	57	58	59	60	756	396	1.188.000
Tim 2	3	8	9	16	18	24	25	33	34	40	47	48	49	50	61			766	406	1.218.000
Tim 3	4	7	11	13	19	23	29	30	37	38	39	51	52	56	62			740	380	1.140.000
Tim 4	5	6	12	17	20	21	22	31	32	41	42	43	53	54	55			698	338	1.014.000
TOTAL BIAYA LEMBUR																			4.560.000	

Hari ke-2

	Urutan Pekerjaan																Waktu Akhir (menit ke)	Penalti (menit)	Biaya Lembur (Rp)	
Tim 1	63	72	73	74	85	86	87	92	93	94	95	112	117	119	120	121		1402	362	1.086.000
Tim 2	64	65	75	76	88	89	90	96	97	98	113	114	115	116	122			1455	415	1.245.000
Tim 3	66	67	68	77	78	79	91	99	100	102	103	104	105	118				1443	403	1.209.000
Tim 4	69	70	71	80	81	82	83	84	101	106	107	108	109	110	111			1458	418	1.254.000
TOTAL BIAYA LEMBUR																			4.794.000	

Total biaya lembur Hari ke-1 dan 2 adalah Rp. 9.354.000

4.3.2. Hasil Penjadwalan dengan Algoritma DE

Setelah parameter-parameter dari permasalahan penjadwalan ini ditentukan, maka program yang telah dibuat di-*run* sebanyak tiga kali. Hasil yang terbaik diambil dari kelima hasil *run* ini, yaitu hasil *run* yang memiliki nilai total biaya keterlambatan terkecil. Tabel 4.23. sampai tabel 4.26. berikut ini menunjukkan hasil *run* sebanyak tiga kali dengan jumlah 4 tim pekerja sedangkan tabel 4.27.

Sampai 4.29. menunjukkan penjadwalan dengan jumlah 5 tim.

Tabel 4.23. Hasil Run 1 (4 Tim)

Hari Ke -1

	Urutan Pekerjaan																Waktu Akhir (menit ke)	Penalti (menit)	Biaya Lembur (Rp)
Tim 1	1	10	11	15	16	17	25	26	35	36	41	42	53	54	55	56	699	339	1.017.000
Tim 2	2	3	4	18	19	23	27	28	29	43	44	45	57	58	59	690	330	990.000	
Tim 3	5	6	12	13	14	24	30	32	34	46	47	48	49	50	60	61	780	420	1.260.000
Tim 4	7	8	9	20	21	22	31	33	37	38	39	40	51	52	62	760	400	1.200.000	
TOTAL BIAYA LEMBUR																		4.467.000	

Hari Ke - 2

	Urutan Pekerjaan																Waktu Akhir (menit ke)	Penalti (menit)	Biaya Lembur (Rp)
Tim 1	63	64	65	78	79	90	91	92	93	101	109	110	111	118	119	1409	369	1.107.000	
Tim 2	66	71	72	80	85	86	87	94	95	96	104	105	106	120	121	1456	416	1.248.000	
Tim 3	67	68	69	81	82	83	84	97	98	99	107	108	116	117	122	1422	382	1.146.000	
Tim 4	70	73	74	75	76	77	88	89	100	102	103	112	113	114	115	1415	375	1.125.000	
TOTAL BIAYA LEMBUR																		4.626.000	

Total biaya lembur Hari ke-1 dan 2 pada hasil run ke-1 adalah Rp. 9.093.000

Tabel 4.25. Hasil Run 2 (4 Tim)

Hari Ke - 1

	Urutan Pekerjaan																Waktu Akhir (menit ke)	Penalti (menit)	Biaya Lembur (Rp)
Tim 1	4	10	11	13	21	22	28	29	30	44	45	46	47	60	61	632	272	816.000	
Tim 2	2	3	12	18	19	23	31	36	37	38	48	49	50	51	52	53	689	329	987.000
Tim 3	1	7	8	9	20	24	25	26	39	40	41	42	54	55	62	755	395	1.185.000	
Tim 4	5	6	14	15	16	17	27	32	33	34	35	43	56	57	58	59	767	407	1.221.000
TOTAL BIAYA LEMBUR																		4.209.000	

Hari Ke - 2

	Urutan Pekerjaan																Waktu Akhir (menit ke)	Penalti (menit)	Biaya Lembur (Rp)
Tim 1	66	69	70	76	77	84	96	97	102	103	104	116	117	118	119	1427	387	1.161.000	
Tim 2	63	71	72	78	79	88	89	90	98	99	105	114	115	120	121	1432	392	1.176.000	
Tim 3	64	65	73	80	81	82	83	91	100	101	106	107	108	109	1421	381	1.143.000		
Tim 4	67	68	74	75	85	86	87	92	93	94	95	110	111	112	113	122	1454	414	1.242.000
TOTAL BIAYA LEMBUR																		4.722.000	

Total biaya lembur Hari ke-1 dan 2 pada hasil run ke-2 adalah Rp. 8.931.000

Tabel 4.25. Hasil Run 3 (4 Tim)

Hari Ke - 1

	Urutan Pekerjaan																Waktu Akhir (menit ke)	Penalti (menit)	Biaya Lembur (Rp)	
Tim 1	66	69	70	76	77	84	96	97	102	103	104	116	117	118	119			1427	387	1.161.000
Tim 2	63	71	72	78	79	88	89	90	98	99	105	114	115	120	121			1432	377	1.131.000
Tim 3	64	65	73	80	81	82	83	91	100	101	106	107	108	109				1421	381	1.143.000
Tim 4	67	68	74	75	85	86	87	92	93	94	95	110	111	112	113	122		1454	414	1.242.000
TOTAL BIAYA LEMBUR																			4.677.000	

Hari Ke -2

	Urutan Pekerjaan																Waktu Akhir (menit ke)	Penalti (menit)	Biaya Lembur (Rp)	
Tim 1	63	69	70	81	82	83	88	93	84	103	104	111	112	117	118	121		1454	414	1.242.000
Tim 2	64	65	75	76	77	86	85	87	99	100	101	110	109	116	119	122		1453	403	1.209.000
Tim 3	67	68	74	78	89	90	94	95	96	102	105	106	115	113				1411	371	1.113.000
Tim 4	66	71	72	73	79	80	92	91	98	97	107	108	114	120				1407	367	1.101.000
TOTAL BIAYA LEMBUR																			4.665.000	

Total biaya lembur Hari ke-1 dan 2 pada hasil run ke-3 adalah Rp. 9.342.000

Dari hasil perhitungan, diperoleh bahwa pada hasil run ke-2 merupakan hasil yang paling baik dalam kondisi jumlah pekerja 4 tim. Setelah diperoleh hasil perhitungan dengan jumlah pekerja 4 tim, maka dilakukan juga perhitungan dengan jumlah pekerja 5 tim.

Tabel 4.23. Hasil Run 1 (5Tim)

Hari Ke -1

	Urutan Pekerjaan																Waktu Akhir (menit ke)	Penalti	Biaya Lembur (Rp)	
Tim 1	1	11	9	19	26	7	28	37	47	48	58	59	62					754	394	1.182.000
Tim 2	3	2	16	17	18	29	38	39	43	49	55	56	57					670	310	930.000
Tim 3	4	10	12	13	25	30	31	32	45	50	51	60						598	238	714.000
Tim 4	5	8	14	21	22	33	36	40	41	42	55	61						578	218	654.000
Tim 5	6	7	15	20	23	24	34	35	46	52	53	54						590	230	690.000
TOTAL BIAYA LEMBUR																			4.170.000	

Hari Ke - 2

	Urutan Pekerjaan													Waktu Akhir (menit ke)	Penalti	Biaya Lembur (Rp)			
Tim 1	63	70	75	76	85	86	87	103	108	115	116	117					1309	269	807.000
Tim 2	64	65	77	82	83	84	101	102	109	110	111	121					1309	269	807.000
Tim 3	66	73	74	88	89	96	97	98	104	112	113	120					1368	328	984.000
Tim 4	67	71	72	81	90	91	92	99	100	114	118	119					1302	262	786.000
Tim 5	68	69	78	79	80	93	94	95	105	106	107	122					1432	392	1.176.000
TOTAL BIAYA LEMBUR																			4.560.000

Total biaya lembur Hari ke-1 dan 2 pada hasil run ke-1 adalah Rp. 8.730.000

Tabel 4.25. Hasil *Run 2* (5 Tim)

Hari Ke - 1

	Urutan Pekerjaan													Waktu Akhir (menit ke)	Penalti	Biaya Lembur (Rp)			
Tim 1	1	2	16	17	18	31	32	33	44	45	46	61					543	183	549.000
Tim 2	3	4	5	19	25	26	27	42	43	51	52	57					533	173	519.000
Tim 3	6	11	12	13	28	29	34	41	47	48	53	54	55	60			567	207	621.000
Tim 4	8	9	14	20	21	30	35	36	37	49	56	58					543	183	549.000
Tim 5	7	10	15	22	23	24	38	39	40	50	59	62					658	298	894.000
TOTAL BIAYA LEMBUR																			3.132.000

Hari Ke - 2

	Urutan Pekerjaan													Waktu Akhir (menit ke)	Penalti	Biaya Lembur (Rp)			
Tim 1	63	71	80	81	86	93	94	100	106	104	116	122					1365	325	975.000
Tim 2	64	65	75	76	87	91	95	101	103	102	117						1320	280	840.000
Tim 3	66	72	77	78	79	92	96	97	107	112	113	120	121				1422	382	1.146.000
Tim 4	67	68	73	82	88	89	90	98	108	109	118	119					1398	358	1.074.000
Tim 5	69	70	74	83	84	85	99	105	110	111	115	114					1399	359	1.077.000
TOTAL BIAYA LEMBUR																			5.112.000

Total biaya lembur Hari ke-1 dan 2 pada hasil run ke-2 adalah Rp. 8.244.000

Tabel 4.25. Hasil *Run 3* (5 Tim)

Hari Ke - 1

	Urutan Pekerjaan													Waktu Akhir (menit ke)	Penalti	Biaya Lembur (Rp)			
Tim 1	5	10	16	20	26	27	36	37	49	50	58	59					545	185	555.000
Tim 2	2	8	15	17	21	28	35	38	39	51	52	53					566	206	618.000
Tim 3	1	11	9	18	22	23	33	34	46	47	48	60	61				502	142	426.000
Tim 4	3	4	12	14	24	29	30	40	44	45	54	62					737	377	1.131.000
Tim 5	6	7	13	19	25	31	32	41	42	43	55	56	57				575	215	645.000
TOTAL BIAYA LEMBUR																			3.375.000

Hari Ke -2

	Urutan Pekerjaan													Waktu Akhir (menit ke)	Penalti	Biaya Lembur (Rp)				
	63	70	76	77	89	90	91	107	108	109	110	121								
Tim 1	63	70	76	77	89	90	91	107	108	109	110	121						1355	315	945.000
Tim 2	64	71	72	84	85	86	99	103	104	111	119	120						1322	282	846.000
Tim 3	65	66	78	79	80	87	100	101	102	112	113	114	122					1455	415	1.245.000
Tim 4	67	73	74	81	82	92	96	97	98	115	116							1376	336	1.008.000
Tim 5	68	69	75	83	88	93	94	95	105	106	117	118						1398	358	1.074.000
TOTAL BIAYA LEMBUR																				5.118.000

Total biaya lembur Hari ke-1 dan 2 pada hasil run ke-3 adalah Rp. 8.493.000

Setelah di-run sebanyak tiga kali, terlihat bahwa hasil yang terbaik bila dibanding (tabel4.26) adalah hasil run ke 2 dengan jumlah tim 5 .

Tabel 4.26. Perbandingan Hasil Akhir

	Manual (Rp)	Biaya Tim (Rp)	Biaya Lembur (Rp)			Total Manual (Rp)	Total Terkecil (Rp)	Penguran gan Biaya Lembur
			1	2	3			
4 Tim	9.354.000	2.000.000	9.093.000	8.931.000	9.342.000	11.354.000	10.931.000	4%
5 Tim		2.500.000	8.730.000	8.244.000	8.493.000	11.854.000	10.744.000	9%

Jadi, solusi terbaik dari masalah penjadwalan ini adalah dengan urutan jadwal seperti tertera pada hasil *run* 2. Nilai fungsi objektif pada kondisi dengan jumlah truk yang datang sebanyak 122 kedatangan dan jumlah kubikasi 2.528 cbm, yaitu total biaya lembur terkecil adalah sebesar Rp. 8.244.000 dengan nilai pengurangan biaya lembur dibandingkan jadwal perhitungan manual tanpa menggunakan Algoritma DE adalah sebesar 9%. Waktu *run* (waktu komputasi) untuk memperoleh hasil ini adalah selama kurang lebih 8,45 menit.