



TUGAS AKHIR – TI 091324

**PENGEMBANGAN SIMPLE ITERATIVE MUTATION
ALGORITHM (SIM-A) UNTUK MENYELESAIKAN
PERMASALAHAN SHIP SCHEDULING AND ASSIGNMENT
(STUDI KASUS : DISTRIBUSI SEMEN CURAH PADA PT. X)**

KETUT HENDRA HARIANTO

NRP 2510 100 090

Dosen Pembimbing

Prof. Ir. I Nyoman Pujawan, M.Eng., Ph.D., CSCP.

Prof. Ir. Budi Santosa, MS., PhD.

JURUSAN TEKNIK INDUSTRI

Fakultas Teknologi Industri

Institut Teknologi Sepuluh Nopember

Surabaya 2014



FINAL PROJECT – TI 091324

**DEVELOPMENT OF SIMPLE ITERATIVE MUTATION
ALGORITHM (SIM-A) TO SOLVE SHIP SCHEDULING AND
ASSIGNMENT PROBLEM (CASE STUDY: BULK CEMENT
DISTRIBUTION AT PT. X)**

KETUT HENDRA HARIANTO

NRP 2510 100 190

Supervisor

Prof. Ir. I Nyoman Pujawan, M.Eng., Ph.D., CSCP.

Prof. Ir. Budi Santosa, MS., Ph.D.

DEPARTMENT OF INDUSTRIAL ENGINEERING

Faculty of Industrial Technology

Institut Teknologi Sepuluh Nopember

Surabaya 2014

LEMBAR PENGESAHAN

**PENGEMBANGAN *SIMPLE ITERATIVE MUTATION*
ALGORITHM (SIM-A) UNTUK MENYELESAIKAN
PERMASALAHAN *SHIP SCHEDULING AND ASSIGNMENT*
(STUDI KASUS : DISTRIBUSI SEMEN CURAH PADA PT. X)**

TUGAS AKHIR

Diajukan untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana Teknik
pada
Program Studi S-1 Jurusan Teknik Industri
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Oleh:

KETUT HENDRA HARIANTO
NRP. 2510 100 090

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Prof. Ir. I Nyoman Pujawan, M.Eng., Ph.D., CSCP (Pembimbing)
NIP. 196901071994121001

Prof. Ir. Budi Santosa, MS., PhD. (Ko – Pembimbing)
NIP. 196905121994021001



**PENGEMBANGAN *SIMPLE ITERATIVE MUTATION*
ALGORITHM (SIM-A) UNTUK MENYELESAIKAN
PERMASALAHAN *SHIP SCHEDULING AND ASSIGNMENT*
(STUDI KASUS : DISTRIBUSI SEMEN CURAH PADA PT. X)**

Nama Mahasiswa : Ketut Hendra Harianto
NRP : 2510100090
Pembimbing : Prof. Ir. I Nyoman Pujawan, M. Eng., Ph. D., CSCP.
Prof. Ir. Budi Santosa, MS., Ph. D.

ABSTRAK

PT. X merupakan perusahaan penghasil semen dengan pangsa pasar terbesar di Indonesia. Pada tahun 2010, PT. X mampu merebut 43.3% dari total *market share* dalam Negeri. 20% dari volume barang yang dihasilkan oleh PT. X didistribusikan melalui jalur laut. Pada tahun 2013, PT. X menghasilkan biaya distribusi sebesar 923 Milyar atau setara dengan 39.8% dari total beban usaha. Pengiriman semen curah pada PT. X dilakukan dengan menggunakan kapal sewa dengan sistem *time charter*. Salah satu penyebab biaya distribusi yang besar adalah adanya mekanisme penjadwalan dan penugasan yang kurang baik. Saat ini PT. X menggunakan 13 kapal sewa untuk digunakan mengirimkan semen curah ke 7 *packing plant* dengan lokasi yang berbeda. Dengan adanya kondisi seperti itu, maka diperlukan adanya suatu penelitian yang membahas mengenai penjadwalan dan penugasan kapal (*ship scheduling and assignment*). Penjadwalan dan penugasan kapal merupakan permasalahan yang tergolong ke dalam *NP-Hard Problem* yaitu permasalahan yang membutuhkan waktu penyelesaian yang lebih lama seiring bertambahnya ukuran permasalahan. Permasalahan *ship scheduling and assignment* (SSA) tergolong ke dalam permasalahan NP-Hard. Oleh karena itu diperlukan adanya suatu metode yang mampu menyelesaikan permasalahan SSA dalam waktu yang relatif singkat. Salah satu metode yang bisa digunakan adalah metode metaheuristik. Pada penelitian ini dikembangkan metode *Simple Iterative Mutation Algorithm* (SIM-A) untuk menyelesaikan permasalahan SSA dengan mempertimbangkan *minimum requirement draft* kapal untuk bisa berlabuh di pelabuhan. SIM-A merupakan kombinasi dari dua metode metaheuristik yaitu Cross Entropy dan Genetic Algorithm. Pada penelitian ini dikembangkan dua skenario berbeda dimana perbedaannya terletak pada durasi lama waktu mulai *loading* maupun *unloading*.

Kata Kunci : *Ship Scheduling and Assignment*, *NP - Hard Problem*, metode meta-heuristik, *Simple Iterative Mutation Algorithm*, *Minimum requirement draft*

(halaman ini sengaja dikosongkan)

DEVELOPMENT OF SIMPLE ITERATIVE MUTATION ALGORITHM (SIM-A) TO SOLVE SHIP SCHEDULING AND ASSIGNMENT PROBLEM (CASE STUDY: BULK CEMENT DISTRIBUTION AT PT. X)

Name : Ketut Hendra Harianto
Student ID : 2510100090
Supervisor : Prof. Ir. I Nyoman Pujawan, M. Eng., Ph. D., CSCP.
Prof. Ir. Budi Santosa, MS., Ph. D.

ABSTRACT

PT. X is a cement producer companies with the largest market share in Indonesia. In 2010, PT. X could seize about 43.3% from national total market share. 20% of goods produced by PT. X have distribution cost about 923 Billion or equal with 39.8% from total operating expenses. Bulk cement shipment at PT. X is conducted by using ship charter with time charter system. One of the major causes of distribution cost increase is unorganized scheduling and shipping. Currently, PT. X uses 13 ship charters to ship bulk cement to 7 packing plant with the different location. Due to this situation, we need a research to discuss about ship scheduling and assignment (SSA). Ship scheduling and assignment is included in NP-Hard Problem which is the problem that require a longer completion time over increasing the size of the problem. So we need a method to solve this problem in a short time by using Meta-heuristic. This research develops Simple Iterative Mutation Algorithm (SIM-A) to solve the SSA problem considering ship minimum requirement draft to dock in the port. SIM-A is combination of two Meta-heuristic methods which are Cross Entropy and Genetic Algorithm. This research develops two different scenarios; the changing parameter is time duration of loading and unloading starting time.

Keywords: Ship Scheduling and Assignment, NP - Hard Problem, Meta-heuristic Method, Simple Iterative Mutation Algorithm, Minimum requirement draft

(halaman ini sengaja dikosongkan)

KATA PENGANTAR

Om Dewa Suksma Parama Acintya Ya Namah Swaha, Sarwa Karya Prasadhantam. Puja dan puji syukur penulis panjatkan kehadirat Ida Sang Hyang Widhi Wasa, Tuhan Yang Maha Esa atas berkat dan rahmat-Nya penulis dapat menyelesaikan penelitian Tugas Akhir yang berjudul “Pengembangan Simple Iterative Mutation Algorithm (SIM-A) untuk Menyelesaikan Permasalahan Ship Scheduling and Assignment. Studi Kasus : Distribusi Semen Curah Pada PT. X). Laporan penelitian Tugas Akhir ini ditulis untuk memenuhi syarat kelulusan program strata 1 (S1) di Jurusan Teknik Industri – Institut Teknologi Sepuluh Nopember (ITS – Surabaya). Selama pengerjaan Tugas Akhir ini, penulis seringkali mendapatkan bimbingan, arahan, motivasi, semangat serta bantuan dari banyak pihak. Oleh karena itu, penulis ingin menyampaikan ucapan terimakasih kepada pihak – pihak yang sangat membantu penulis dalam menyelesaikan Tugas Akhir ini, antara lain :

1. Kedua orang tua penulis, I Made Yama dan Ketut Suartini atas segala doa dan semangat yang tiada hentinya diberikan kepada penulis.
2. Saudara – saudara penulis, Putu Eka, Ketut Dika Permana, Putu Tika Oliviani, dan adik – adik yang selama ini selalu memberikan canda tawa ketika penulis sedang mengerjakan laporan Tugas Akhir ini.
3. Bapak Prof. I Nyoman Pujawan, M. Eng., Ph. D., CSCP. Dan Bapak Prof. Ir. Budi Santosa, MS., Ph. D. selaku dosen pembimbing dan dosen ko-pembimbing yang telah banyak memberikan wawasan, arahan, kritik, dan saran yang membuat penulis dapat menyelesaikan laporan Tugas Akhir ini dengan tepat waktu.
4. Bapak Iwan Vanany dan Bapak Imam Baihaqi selalu dosen penguji Tugas Akhir atas masukan yang diberikan untuk perbaikan laporan Tugas Akhir.

5. Ibu Putu Dana Karningsih, S.T., M. Eng. Sc., Ph.D., yang selalu memberikan semangat dan arahan selama penulis sebagai Mahasiswa Teknik Industri – ITS.
6. Ibu Naning Aranti Wessiani, S.T., M. M. selaku dosen wali penulis yang selalu memberikan arahan kepada penulis terkait masalah akademik selama kuliah di TI – ITS.
7. Bapak / Ibu dosen Jurusan Teknik Industri – ITS yang telah memberikan banyak ilmu selama penulis menjalani proses akademik. Ilmu yang sangat membantu penulis dalam menyelesaikan laporan Tugas Akhir ini.
8. Sahabat – sahabat penulis : Hafizh Rifky Novaldi (Tekkim 2013), Ammar Farhan (MaMet ITK 2013), Tegar Aditya Cahyadi (Desain Interior 2013), Kadek Ria Restina, Reza Furqan Arrachman (Tekpal 2012), Bima Ero (Tekla 2013), Fajar Ade Putra (TC 2013), NaFa (TMJ 2013), Bangker (Ardian) (Tekpal 2013), Made Dedi Ariantina (T. Geologi ITB 2010), M. Fikri Aulia (T. Sipil 2013), Hadits Shofar F. (Siskal 2012), Kadek Astia Gunawan, Yuri Syahwirawan (Geofisika 2013) yang selalu ‘care’ sama penulis dan atas saran – saran yang diberikan serta kebersamaannya selama ini.
9. Kadek Sri Purnama Dewi, yang telah memberikan semangat selama ini, yang selalu memberikan warna kepada penulis selama 3,5 tahun belakangan dan terimakasih atas ‘ujian’ yang telah diberikan belakangan ini.
10. Teman – teman terbaik penulis, Angga Rizky Pratama, Aulia Muhammad, Dewi Saragih, Mas Wahyudi, Mbak Eki, Andrian Henry, Haga, Raka Ayuda, Angger, Ary Siwastika, Ivan, Igo (Argoemerta), M. Ihsan (TI UII 2011), Achmad Rusvito (TI Telkom 2011), Afiffuddin Thoif (TI UII 2011), Yoga S. W. (TI UII 2011), Yogi (TI UII 2011), Komang Rina Tristiyanti (TI 2009), Mbak Nilla (TI 2009), dan Harimurti Adly yang seringkali memberikan semangat kepada penulis baik dalam urusan akademik maupun dalam pengerjaan Tugas Akhir ini.

11. Mansur dan Mbak Mimin yang telah banyak membantu penulis dalam menyelesaikan Tugas Akhir ini.
12. Evi, Mansur, Hendy, Ratri, dan Marcel selaku teman seperjuangan Tugas Akhir dibawah bimbingan Prof. I Nyoman Pujawan atas kebersamaannya selama mengerjakan tugas akhir.
13. Dewie Saktia, Risal Arsyad, Arrad Ramadhan, Aditya Yanuar, Issam Chalibi, Andre T. Nugroho atas kebersamaannya untuk begadang selama mengerjakan Tugas Akhir.
14. Primawati Hayuningtyas dan Shafira Saravina, terimakasih atas ucapan selamat yang telah diberikan kepada penulis.
15. Asisten LSCM : Aulia Muhammad, Mansur, Evi, Hendy, Ratri, Reika, Giovanni, Reby, Gane, Kelvin, Troy, Willy, Putek, Ivana atas kerja sama selama penulis mengabdikan menjadi asisten laboratorium LSCM.
16. Muhammad Aulia Nurrahim, Abdan Sakur, Reza Firsada, Samir Joesoef, Adi Yudanto, Riko Hendri Kurniawan atas perlakuan baik yang telah diberikan kepada penulis selama ini. Terimakasih atas kebersamaannya.
17. Pihak – pihak lain yang telah berkontribusi dalam pengerjaan tugas akhir ini, yang penulis tidak bisa sebutkan satu per satu.

Penulis menyadari bahwa Tugas Akhir ini tidak sempurna, oleh karena itu kritik dan saran yang membangun sangat diharapkan untuk perbaikan ke depannya. Semoga laporan Tugas Akhir ini bermanfaat bagi para pembaca. Akhir kata, penulis ucapkan terima kasih.

Surabaya, Juli 2014

Penulis

(halaman ini sengaja dikosongkan)

DAFTAR ISI

ABSTRAK	i
ABSTRACT.....	iii
KATA PENGANTAR.....	v
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	5
1.3 Tujuan Penelitian.....	5
1.4 Manfaat Penelitian.....	5
1.5 Ruang Lingkup Penelitian	6
1.5.1 Batasan	6
1.5.2 Asumsi	6
1.6 Sistematika Penulisan.....	7
BAB 2 TINJAUAN PUSTAKA	9
2.1 Transportasi Laut.....	9
2.2 Karakteristik Transportasi Laut.....	11
2.3 <i>Ship Scheduling and Assignment</i>	13
2.4 <i>Simple Iterative Mutation Algorithm (SIM-A)</i>	14
2.4.1 Algoritma <i>Cross Entropy</i>	15
2.4.2 <i>Genetic Algorithm</i>	18
BAB 3 METODOLOGI PENELITIAN.....	27
3.1 Studi Literatur.....	27
3.2 Tahap Pengembangan Algoritma	29
3.2.1 Tahap Pengembangan Algoritma SIM-A.....	29

3.2.1.1	Skenario 1 (Kondisi Eksisting) : SIM-A dengan Time Windows Depot dan <i>Port</i> 08:00 – 18:00.....	30
3.2.1.2	Skenario 2 : SIM-A dengan Time Windows Depot 0:00- 24:00 dan <i>Port</i> 08:00 – 18:00	30
3.2.2	Verifikasi dan Validasi Algoritma	31
3.2.2.1	Verifikasi	31
3.2.2.2	Validasi.....	31
3.3	Tahap Aplikasi Algoritma Untuk Sistem Riil	31
3.3.1	Pengumpulan Data	32
3.3.2	Pengolahan Data.....	32
3.4	Tahap Analisis Hasil dan Rekomendasi	33
BAB 4 PENGEMBANGAN ALGORITMA.....		34
4.1	Gambaran Umum Permasalahan	35
4.2	Pengembangan Algoritma	37
4.3	Verifikasi	57
4.4	Validasi Algoritma SIM-A.....	61
4.4.1	Penyelesaian Dengan Enumerasi	61
4.4.2	Analisis Uji Validasi	68
BAB 5 APLIKASI ALGORITMA UNTUK SISTEM RIIL.....		69
5.1	Pengumpulan Data	69
5.2	Pengolahan Data.....	73
5.2.1	Distribusi Waktu Tunggu Sebelum <i>Loading</i> (WTSL).....	73
5.2.2	Distribusi Waktu Tunggu Sebelum <i>Unloading</i> (WTSU).....	74
5.2.3	<i>Running</i> Algoritma SIM – A.....	76
5.2.3.1	Hasil <i>Running</i> Skenario 1	77
5.2.3.2	Hasil <i>Running</i> Skenario 2	95
5.2.3.3	Perbandingan Total Biaya Skenario 1 dan 2.....	110
BAB 6 ANALISIS HASIL DAN REKOMENDASI		112
6.1	Analisis Hasil Algoritma SIM – A	113
6.1.1	Analisis Hasil Algoritma SIM – A Skenario 1	113
6.1.2	Analisis Hasil Algoritma SIM – A Skenario 2	115

6.1.3	Perbandingan Hasil Skenario 1 dan Skenario 2	117
6.2	Rekomendasi	118
BAB 7 KESIMPULAN DAN SARAN.....		119
6.1	Kesimpulan.....	119
6.2	Saran.....	120
DAFTAR PUSTAKA.....		121
BIOGRAFI PENULIS		123
LAMPIRAN.....		124

(halaman ini sengaja dikosongkan)

DAFTAR TABEL

Tabel 2. 1 Prinsip dalam Transportasi Laut	10
Tabel 4. 1 Skema Pemilihan Kapal untuk Diberangkatkan	49
Tabel 4. 2 Parameter Pelabuhan.....	61
Tabel 4. 3 Parameter Kapal (1)	62
Tabel 4. 4 Parameter Kapal (2)	62
Tabel 4. 5 Parameter Pelayaran.....	63
Tabel 4. 6 Fleksibilitas Kapal	64
Tabel 4. 7 <i>Scheduled</i> (Penjadwalan dan Penugasan Kapal).....	65
Tabel 4. 8 Informasi untuk <i>Port 1</i>	66
Tabel 4. 9 Informasi untuk <i>Port 2</i>	66
Tabel 4. 10 Informasi untuk <i>Port 3</i>	67
Tabel 4. 11 Informasi untuk <i>Port 4</i>	67
Tabel 4. 12 Informasi untuk <i>Port 5</i>	68
Tabel 5. 1 Parameter Pelabuhan <i>Loading</i> dan <i>Unloading</i> PT. X.....	70
Tabel 5. 2 Parameter Kapal yang Digunakan Pada PT. X	71
Tabel 5. 3 Parameter Biaya yang Digunakan Pada PT. X	71
Tabel 5. 4 Parameter Pelayaran.....	72
Tabel 5. 5 Rekap Data Penjadwalan dan Penugasan Kapal Skenario 1	77
Tabel 5. 6 Status Kapal Berdasarkan Solusi Optimum Skenario 1	79
Tabel 5. 7 Informasi Silo Port Tujuan 1	82
Tabel 5. 8 Informasi Silo Port Tujuan 2.....	83
Tabel 5. 9 Informasi Silo <i>Packing Plant 3</i>	84
Tabel 5. 10 Informasi Silo <i>Packing Plant 4</i>	85
Tabel 5. 11 Informasi Silo <i>Packing Plant 5</i>	86
Tabel 5. 12 Informasi Silo <i>Packing Plant 6</i>	87
Tabel 5. 13 Informasi Silo <i>Packing Plant 7</i>	88
Tabel 5. 14 Total Biaya 10 Kali Percobaan Skenario 1	94
Tabel 5. 15 Solusi Optimum Penjadwalan dan Penugasan Kapal Pada Skenario 2 ...	95

Tabel 5. 16 Status Kapal Berdasarkan Solusi Optimum Solusi 2.....	97
Tabel 5. 17 Informasi Silo <i>Packing Plant</i> 1 (Skenario 2).....	99
Tabel 5. 18 Informasi Silo <i>Packing Plant</i> 2 (Skenario 2).....	100
Tabel 5. 19 Informasi Silo <i>Packing Plant</i> 3 (Skenario 2).....	101
Tabel 5. 20 Informasi Silo <i>Packing Plant</i> 4 (Skenario 2).....	102
Tabel 5. 21 Informasi Silo <i>Packing Plant</i> 5 (Skenario 2).....	103
Tabel 5. 22 Informasi Silo <i>Packing Plant</i> 6 (Skenario 2).....	104
Tabel 5. 23 Informasi Silo <i>Packing Plant</i> 7 (Skenario 2).....	105
Tabel 5. 24 Total Biaya 10 Kali Percobaan Skenario 2.....	109

DAFTAR GAMBAR

Gambar 1. 1 Ilustrasi Pengiriman Kapal Mempertimbangkan MRD	4
Gambar 2. 1 <i>Flowchart</i> SIM-A (Budiman, 2012).....	17
Gambar 2. 2 <i>One-point</i> dan <i>Two-point Crossover</i> (Sastry et al., 2005)	21
Gambar 2. 3 <i>Uniform Crossover</i> (Sastry et al., 2005)	22
Gambar 2. 4 <i>Parents</i> dan <i>Template</i> dalam <i>Uniform Order-based crossover</i> (Sastry et al., 2005).....	22
Gambar 2. 5 Hasil <i>Uniform Order-based Crossover</i> (Sastry et al., 2005).....	23
Gambar 2. 6 <i>Order-based Crossover</i> (Sastry et al., 2005).....	23
Gambar 2. 7 <i>Partially Matched Crossover</i> (Sastry et al., 2013).....	24
Gambar 2. 8 <i>Parents</i> dalam <i>Cycle Crossover</i> (Sastry et al., 2013).....	24
Gambar 2. 9 Gambaran Proses <i>Cycle Crossover</i> (Sastry et al., 2005).....	25
Gambar 2. 10 Hasil Metode <i>Cycle Crossover</i> (Sastry et al., 2005)	25
Gambar 2. 11 <i>Swap Mutation</i> (Ursani et al., 2011)	26
Gambar 2. 12 <i>Insertion Mutation</i> (Ursani et al., 2011).....	26
Gambar 2. 13 <i>Inversion Mutation</i> (Ursani et al., 2011).....	26
Gambar 3. 1 Alur Pengerjaan Penelitian Tugas Akhir.....	28
Gambar 4. 1 <i>Flowchart</i> Pengembangan Algoritma SIM – A	38
Gambar 4. 2 <i>Flowchart</i> Penjadwalan dan Penugasan Kapal (1).....	44
Gambar 4. 3 <i>Flowchart</i> Penjadwalan dan Penugasan Kapal (2).....	45
Gambar 4. 4 <i>Flowchart</i> Penjadwalan dan Penugasan Kapal (3).....	46
Gambar 4. 5 <i>Flowchart</i> Penjadwalan dan Penugasan Kapal (4).....	47
Gambar 4. 6 <i>Flowchart</i> Penjadwalan dan Penugasan Kapal (5).....	48
Gambar 4. 7 Hasil <i>Running Software</i> MATLAB untuk Kasus Sederhana (1).....	59
Gambar 4. 8 Hasil <i>Running Software</i> MATLAB untuk Kasus Sederhana (2).....	60
Gambar 4. 9 Hasil <i>Running Software</i> MATLAB untuk Kasus Sederhana (3).....	60
Gambar 5. 1 <i>Fitting</i> Distribusi WTSU.....	74
Gambar 5. 2 Hasil <i>Fitting</i> Distribusi WTSU	76
Gambar 5. 3 Utilisasi Kapal Berdasarkan Solusi Optimum Skenario 1	80
Gambar 5. 4 Grafik Perbandingan OH dan CR di <i>Port</i> 1	89

Gambar 5. 5 Grafik Perbandingan OH dan CR di <i>Port 2</i>	90
Gambar 5. 6 Grafik Perbandingan OH dan CR <i>Port 3</i>	90
Gambar 5. 7 Grafik Perbandingan OH dan CR <i>Port 4</i>	91
Gambar 5. 8 Grafik Perbandingan OH dan CR <i>Port 5</i>	91
Gambar 5. 9 Grafik Perbandingan OH dan CR <i>Port 6</i>	92
Gambar 5. 10 Grafik Perbandingan OH dan CR <i>Port 7</i>	93
Gambar 5. 11 Rekap Total Biaya Selama 10 Kali Percobaan.....	94
Gambar 5. 12 Utilisasi Kapal Berdasarkan Solusi Optimum Skenario 2	98
Gambar 5. 13 Grafik Perbandingan OH dan CR <i>Port 1</i> (Skenario 2).....	106
Gambar 5. 14 Grafik Perbandingan OH dan CR <i>Port 2</i> (Skenario 2).....	106
Gambar 5. 15 Grafik Perbandingan OH dan CR di <i>Port 3</i> (Skenario 2)	107
Gambar 5. 16 Grafik Perbandingan OH dan CR <i>Port 4</i> (Skenario 2).....	107
Gambar 5. 17 Grafik Perbandingan OH dan CR <i>Port 5</i> (Skenario 2).....	108
Gambar 5. 18 Grafik Perbandingan OH dan CR di <i>Port 6</i> (Skenario 2)	108
Gambar 5. 19 Grafik Perbandingan OH dan CR di <i>Port 7</i> (Skenario 2)	109
Gambar 5. 20 Grafik Total Biaya 10 Kali Percobaan Skenario 2.....	110
Gambar 5. 21 Grafik Perbandingan Total Biaya Skenario 1 dan 2.....	110

BAB 1

PENDAHULUAN

Pada bab ini dibahas mengenai pendahuluan penelitian, yaitu latar belakang penelitian, perumusan masalah, tujuan, manfaat penelitian, dan ruang lingkup permasalahan yang berisi batasan dan asumsi yang digunakan dalam penelitian.

1.1 Latar Belakang

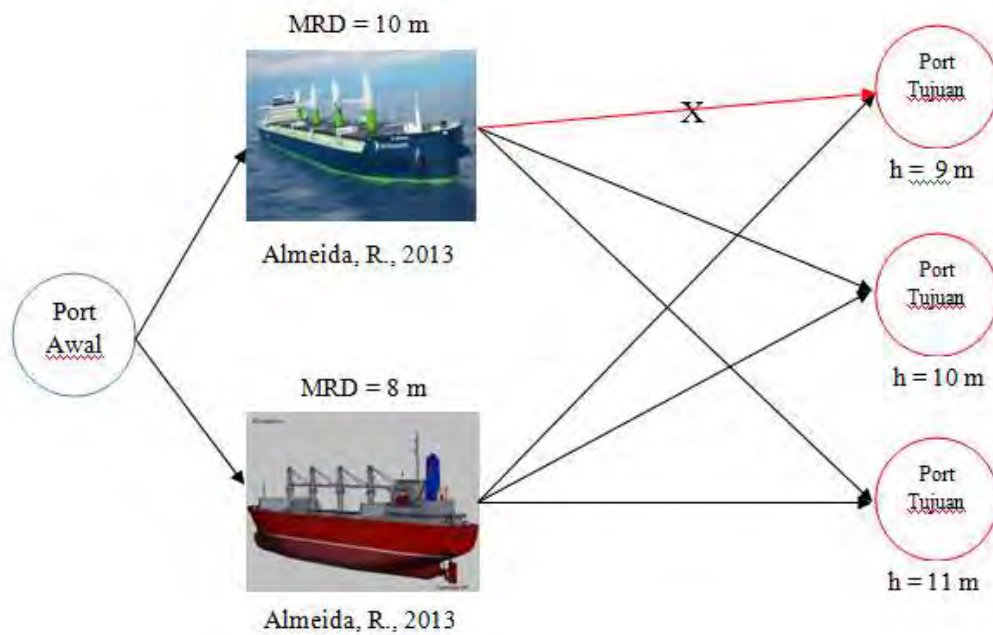
Transportasi laut atau transportasi maritim merupakan media transportasi yang paling sering digunakan untuk pengiriman barang dalam perdagangan internasional. Menurut UNCTAD (2007) dalam Andersson et al. (2011), menyatakan bahwa estimasi total nilai perdagangan yang dikirimkan melalui laut antara 65% dan 85% dari total seluruh nilai barang. Hal ini juga didukung oleh pernyataan Al-Khayyal & Hwang (2005) bahwa 90% dari total *volume* dan 70% dari nilai barang didistribusikan melalui jalur laut. Pengiriman produk dalam jumlah yang besar dan total biaya pengiriman yang paling murah merupakan kelebihan pengiriman barang melalui jalur laut dibandingkan dengan jenis transportasi yang lain (Chopra & Meindl, 2007). Biaya merupakan hal yang paling kritis untuk diperhatikan di dalam menjalankan bisnis usaha. Oleh karena itu, perlu dilakukan pengelolaan untuk dapat meminimalkan biaya operasional, salah satunya biaya operasional untuk pengiriman barang.

PT. X merupakan pabrik semen dengan pangsa pasar terbesar di Indonesia yang memiliki beberapa anak perusahaan penghasil semen di Indonesia. Pernyataan ini didukung oleh pernyataan Madasari (2012) yang menyatakan bahwa, berdasarkan laporan tahun 2010, PT. X mendominasi pangsa pasar semen nasional sebesar 43,3% dari total pangsa pasar dalam negeri. Di lain sisi, biaya angkut dan bongkar atau muat menjadi komponen biaya dengan proporsi terbesar yaitu sebesar Rp. 923 Miliar atau 39,8% dari total beban usaha. Saat ini, sistem distribusi PT. X menggunakan jalur darat dan laut. Dimana 80% melalui jalur

darat dan 20% melalui jalur laut (Renspandy, 2013). Berdasarkan jenis produk yang dikirimkan, saat ini PT. X mengirimkan semen dalam bentuk curah dan *cement bag* (zak). Pengeluaran yang sangat besar untuk biaya distribusi terjadi akibat tidak ada mekanisme penjadwalan yang baik. Saat ini, pengiriman semen khususnya semen curah dilakukan berdasarkan ketersediaan kapal. Jika kapal tersedia maka pengiriman semen segera dilakukan, jika kapal tidak tersedia maka pengiriman tidak akan dilakukan. Oleh karena itu, dapat dikatakan bahwa pengiriman atau distribusi semen pada PT. X tidak berdasarkan mekanisme penjadwalan pengiriman yang pasti. Tidak adanya mekanisme penjadwalan, mengakibatkan adanya pengeluaran biaya distribusi dalam jumlah yang besar. Biaya distribusi yang besar juga diakibatkan adanya aktivitas kapal yang tidak perlu seperti *waiting for stock*, menunggu akibat cuaca buruk, *waiting for port* untuk bisa melakukan *loading/unloading* selama proses pengiriman semen. Aktivitas-aktivitas tersebut berdampak pada bertambahnya waktu proses pengiriman yang berakibat pada bertambahnya biaya distribusi semen. Oleh karena itu, perlu dilakukan penelitian tentang transportasi maritim yang membahas mengenai penjadwalan dan penugasan kapal (*Ship Scheduling and Assignment*) untuk proses distribusi semen curah dengan tujuan meminimumkan total biaya distribusi.

Ship Scheduling and Assignment merupakan permasalahan yang membahas mengenai penentuan penjadwalan dan penugasan kapal, dimana keluarannya berupa kapan kapal ditugaskan dan jika ditugaskan, maka harus diberangkatkan kemana. *Ship Scheduling and Assignment* merupakan suatu permasalahan yang tergolong ke dalam *Non Polynomial - Hard Problem (NP-Hard Problem)*. *NP-Hard Problem* merupakan permasalahan yang membutuhkan waktu komputasi yang lama seiring bertambahnya ukuran permasalahan (Damayanti, 2013). Jika *exact method* yang digunakan untuk menyelesaikan permasalahan ini, maka akan dibutuhkan waktu komputasi yang sangat lama meskipun hasilnya pasti optimum. Oleh karena itu, diperlukan metoda metaheuristik untuk menyelesaikan permasalahan ini dengan waktu komputasi yang lebih cepat dan hasil yang mendekati nilai optimal bahkan bisa mencapai nilai optimal.

PT. X merupakan perusahaan yang menggunakan banyak kapal untuk mengirimkan semen ke banyak pelabuhan (*port*) tujuan, sehingga bisa dikatakan tingkat kompleksitas penjadwalan dan penugasan kapal pada PT. X sangat tinggi. Oleh karena itu, pada penelitian ini akan digunakan metode metaheuristik untuk menyelesaikan permasalahan penjadwalan dan penugasan kapal pada PT. X yaitu metode *Simple Iterative Mutation Algorithm* (SIM-A). SIM-A digunakan untuk menyelesaikan permasalahan *Ship Scheduling and Assignment* dengan waktu komputasi yang relatif singkat. Pada penelitian ini akan dikembangkan *Simple Iterative Mutation Algorithm* (SIM-A) untuk menyelesaikan permasalahan *Ship Scheduling and Assignment* pada distribusi laut semen curah PT. X dengan mempertimbangkan batasan *minimum requirement draft* kapal, yaitu kapal akan bisa singgah di suatu *port* yang memenuhi konstrain minimum kedalaman dermaga yang dibutuhkan kapal untuk bisa berlabuh. Dengan kata lain, suatu kapal akan bisa ditugaskan untuk mengirimkan semen ke port tujuan yang memenuhi *minimum requirement draft*. Adapun biaya-biaya yang dipertimbangkan dalam penelitian ini, yaitu biaya sewa kapal (*charter cost*), biaya *loading* dan *unloading*, biaya perjalanan menuju *port* tujuan dan biaya dari *port* tujuan menuju *port* awal, biaya keterlambatan (*penalty cost*). *Penalty cost* dikenakan jika kapal selesai melakukan proses *unloading* ketika *port* tujuan mengalami *stock out* dan atau pada saat tersebut *On-Hand* melebihi kapasitas maksimum silo. Dengan mempertimbangkan *minimum requirement draft constraint* kapal dan biaya-biaya yang telah disebutkan sebelumnya, akan ditentukan penjadwalan dan penugasan kapal pada kasus distribusi semen curah PT. X selama horison perencanaan tertentu dengan total biaya minimum yang didapatkan dari hasil *running Simple Iterative Mutation Algorithm*. Pada penelitian ini dipertimbangkan ketidakpastian untuk bisa melakukan proses *loading* di *production port* dan proses *unloading* di *port* tujuan, sedangkan faktor cuaca diabaikan. Dengan adanya faktor ketidakpastian tersebut, maka ketika kapal sampai pada depot atau *port* tujuan kapal tidak selalu bisa melakukan proses *loading/unloading* dan harus menunggu beberapa lama sampai proses bisa dilakukan. Berikut ini merupakan skema penjadwalan kapal dengan mempertimbangkan *minimum requirement draft* kapal:



Gambar 1. 1 Ilustrasi Pengiriman Kapal Mempertimbangkan MRD

Berdasarkan ilustrasi pengiriman kapal dengan mempertimbangkan *minimum requirement draft* (MRD) di atas, kapal 1 tidak bisa digunakan untuk mengirimkan semen curah ke *port* tujuan 1 karena kedalaman (*h*) *port* tujuan tidak bisa memenuhi *minimum requirement draft* kapal. Kapal 1 hanya bisa digunakan untuk mengirimkan semen curah ke *port* tujuan 2 dan 3. Sedangkan kapal 2 bisa digunakan untuk mengirimkan semen curah ke tiga *port* tujuan yang ada.

1.2 Perumusan Masalah

Berdasarkan uraian pada latar belakang yang dijelaskan sebelumnya, dapat dirumuskan permasalahan dalam penelitian ini, yaitu:

1. Bagaimana mengembangkan *Simple Iterative Mutation Algorithm* untuk menyelesaikan permasalahan *Ship Scheduling and Assignment* yang mempertimbangkan *minimum requirement draft constraint* supaya kapal bisa singgah di *port*.
2. Bagaimana menyelesaikan permasalahan *Ship Scheduling and Assignment* pada permasalahan distribusi semen curah PT. X dengan menggunakan *Simple Iterative Mutation Algorithm* yang dikembangkan.

1.3 Tujuan Penelitian

Adapun yang menjadi tujuan dari penelitian ini adalah sebagai berikut:

1. Memperoleh *Simple Iterative Mutation Algorithm* untuk menyelesaikan permasalahan *Ship Scheduling and Assignment* dengan mempertimbangkan *minimum requirement draft* kedalaman dermaga supaya kapal dapat singgah di *port*.
2. Menyelesaikan permasalahan *Ship Scheduling and Assignment* pada distribusi semen curah PT. X dengan menggunakan *Simple Iterative Mutation Algorithm* yang dikembangkan.

1.4 Manfaat Penelitian

Adapun manfaat yang ingin diperoleh dalam penelitian ini, yaitu:

1. Sebagai acuan dalam *Ship Scheduling and Assignment* yang memperhatikan *minimum requirement draft constraint*.
2. Memberikan kontribusi dalam bidang keilmuan optimasi, yaitu dengan adanya metode *Simple Iterative Mutation Algorithm* dalam menyelesaikan permasalahan *Ship Scheduling and Assignment*.

1.5 Ruang Lingkup Penelitian

Dalam ruang lingkup penelitian akan dijelaskan mengenai batasan dan asumsi yang digunakan dalam penelitian.

1.5.1 Batasan

Adapun yang menjadi batasan dalam penelitian ini, yaitu:

1. Algoritma SIM-A yang dikembangkan hanya mempertimbangkan faktor ketidakpastian saat akan melakukan proses *loading/unloading* sedangkan faktor ketidakpastian selama proses pengiriman diabaikan.
2. Dalam menentukan kapal yang bisa singgah di pelabuhan, hanya diperhatikan *minimum requirement draft constraint* yang bisa berlabuh.

1.5.2 Asumsi

Adapun asumsi yang digunakan dalam penelitian ini, yaitu:

1. Semen selalu tersedia di *production port* selama *planning horizon* yang telah ditentukan.
2. Semua kapal yang digunakan merupakan kapal sewa dengan sistem *time charter*.
3. *On – Hand* pada akhir periode 1 diasumsikan sama dengan nilai inisial *On – Hand*.
4. *Consumption rate* di setiap port tujuan tetap selama *planning horizon* yang ditentukan.
5. Waktu perjalanan dari depot menuju *port* tujuan dan sebaliknya bersifat deterministik dan tidak mengalami perubahan selama horizon perencanaan yang ditetapkan.
6. Distribusi lama waktu tunggu sebelum *unloading* diasumsikan sama untuk semua *port* tujuan.

1.6 Sistematika Penulisan

Berikut ini merupakan langkah sistematis yang dilakukan dalam penelitian ini, yaitu:

BAB 1 PENDAHULUAN

Pada bab pendahuluan dijelaskan mengenai tentang latar belakang penelitian, perumusan masalah, ruang lingkup permasalahan yang membahas mengenai batasan dan asumsi yang digunakan, tujuan dan manfaat penelitian, serta sistematika penulisan.

BAB 2 TINJAUAN PUSTAKA

Pada bab tinjauan pustaka dijelaskan mengenai dasar teori atau referensi yang terkait dengan penelitian yang dilakukan. Adapun referensi yang digunakan dalam penelitian, yaitu transportasi laut, karakteristik transportasi laut, *Ship Scheduling and Assignment, Simple Iterative Mutation Algorithm*.

BAB 3 METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan mengenai tahapan-tahapan yang dilakukan dalam penelitian pengembangan *Simple Iterative Mutation Algorithm* untuk menyelesaikan masalah *Ship Scheduling and Assignment* pada permasalahan distribusi semen curah PT. X yang memperhatikan *minimum requirement draft constraint* kapal yang bisa berlabuh di dermaga.

BAB 4 PENGEMBANGAN ALGORITMA

Pada bab ini akan dilakukan pengembangan *Simple Iterative Mutation Algorithm* (SIM-A) dengan dua skenario yang berbeda untuk menyelesaikan permasalahan *Ship Scheduling and Assignment* pada distribusi semen curah.

BAB 5 APLIKASI ALGORITMA UNTUK SISTEM RIIL

Pada bab ini akan dilakukan aplikasi dari algoritma SIM-A yang dikembangkan untuk sistem riil dengan mengacu pada dua skenario yang dikembangkan. Pada aplikasi algoritma ini dibagi menjadi dua tahap yaitu pengumpulan dan pengolahan data yang dibutuhkan dalam penyelesaian *Ship Scheduling and Assignment* pada permasalahan distribusi semen curah PT. X.

BAB 6 ANALISIS HASIL DAN REKOMENDASI

Pada bab ini akan dilakukan analisis hasil dari pengolahan data dan skenario yang telah dikembangkan pada bab sebelumnya. Adapun yang dibahas dalam bab analisis dan interpretasi hasil yaitu analisis hasil dari skenario SIM-A yang dikembangkan. Pada bab ini juga diberikan rekomendasi mengacu pada analisis hasil yang dilakukan.

BAB 7 KESIMPULAN DAN SARAN

Pada bab ini akan dilakukan pengambilan kesimpulan berdasarkan pengolahan data dan analisis hasil yang telah dilakukan. Kemudian dilakukan pemberian saran untuk penelitian ke depannya.

BAB 2

TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan mengenai dasar teori dan penelitian yang mendukung yang menjadi landasan dalam melakukan penelitian tugas akhir ini. Tinjauan pustakan ini terdiri dari transportasi laut, karakteristik transportasi laut, *Ship Scheduling and Assignment*, dan *Simple Iterative Mutation Algorithm*.

2.1 Transportasi Laut

Transportasi laut atau juga dikenal dengan istilah transportasi maritim merupakan moda transportasi yang sering digunakan untuk mengirimkan barang (produk) antar pulau maupun antar benua. Hwang (2005) menyatakan bahwa kurang lebih sekitar 90% dari volume barang dan 70% dari total nilai barang dikirimkan melalui jalur laut. Transportasi laut sering digunakan karena memiliki beberapa kelebihan, yaitu cocok digunakan untuk mengangkut barang dengan jumlah yang besar dan menghabiskan biaya distribusi yang lebih murah dibanding moda transportasi darat ataupun udara (Damayanti, 2013). Namun transportasi laut juga memiliki kelemahan, yaitu kecepatan pengirimannya paling rendah dibanding moda transportasi yang lain, dan juga sering mengalami *delay* pada *loading port* maupun *unloading port*.

Christiansen et al. (2007) mengklasifikasikan penelitian mengenai transportasi laut menjadi tiga bagian yaitu strategis, taktis, dan operasional. Penelitian yang termasuk ke dalam permasalahan strategis yaitu lokasi pelabuhan, desain kapal, ukuran armada transportasi, dan sebagainya. Permasalahan taktis terdiri dari penjadwalan kapal, *inventory ship routing*, *berth scheduling*, dan sebagainya. Penentuan kecepatan pelayaran, aktivitas pemuatan pada kapal dan sebagainya termasuk ke dalam permasalahan operasional.

Lawrence (1972) dalam Andersson et al. (2011) menyatakan bahwa ada tiga macam model transportasi laut antara lain *industrial shipping*, *liner shipping*, dan *tramp shipping*. *Industrial shipping*, moda transportasi yang

digunakan untuk mengirimkan barang merupakan milik perusahaan sehingga tujuan utamanya adalah minimasi biaya perjalanan. *Liner shipping*, moda transportasi yang digunakan sudah memiliki rute sesuai rencana dan jadwal perjalanan. Sedangkan *tramp shipping* moda transportasi yang digunakan menggunakan sistem kargo, sama halnya seperti taksi. Tujuan dari *tramp shipping* diterapkan untuk perusahaan yang memiliki tujuan untuk memaksimalkan keuntungan dengan membawa *optional spot cargoes*.

Pada transportasi laut, pemilihan kapal didasarkan pada muatan (barang) yang akan dikirimkan dan juga didasarkan pada pelabuhan yang bisa menerima kapal tersebut. Biasanya penentuan kapal yang bisa berlabuh pada dermaga didasarkan pada *deadweight tonnage* (DWT) dan juga pada minimal kedalaman dermaga untuk kapal bisa berlabuh. Hwang (2005) menjelaskan beberapa prinsip dalam transportasi laut, yaitu sebagai berikut:

Tabel 2. 1 Prinsip dalam Transportasi Laut

Kategori Utama	Komponen
Muatan	Jenis muatan (kargo)
	Jumlah muatan (kargo) yang harus dikirim
	<i>Loading port</i> untuk setiap tipe muatan
	<i>Delivery port</i> untuk setiap kargo
	<i>Time windows constraints</i> untuk <i>loading</i> dan <i>delivery</i>
Pelabuhan	Jumlah pelabuhan
	Kedalaman air tiap pelabuhan yang mempengaruhi kapal yang bisa bersandar
	Jarak antar pelabuhan
	Waktu <i>loading</i> dan <i>unloading</i> untuk setiap kargo
	Kapasitas penyimpanan setiap muatan
	Fasilitas yang dibutuhkan untuk <i>loading</i> atau <i>unloading</i> muatan
Kapal	Kapasitas

Kategori Utama	Komponen
	Kompartemen
	Jumlah kapal
	Tipe (heterogen atau homogen)
	Keterbatasan pelabuhan yang dapat menerima kapal berlabuh
	Kecepatan maksimum
	Lokasi kapal saat awal horison perencanaan
Biaya	Biaya sewa Kapal
	Biaya operasi:
	1. <i>Crew</i>
	2. Biaya bahan bakar
	3. Pencucian kompartemen
	4. Perawatan dan perbaikan
	5. <i>Port charges</i>

Sumber : Hwang (2005)

2.2 Karakteristik Transportasi Laut

Menurut Nurminarsih (2012), terdapat beberapa karakteristik dalam transportasi laut yang harus diperhatikan dalam melakukan penelitian. Berikut ini merupakan karakteristik – karakteristik yang dijadikan acuan dalam melakukan penelitian yang meliputi karakteristik kapal, karakteristik pelabuhan, dan karakteristik kargo.

1. Karakteristik Kapal

Biasanya ukuran kapal dinyatakan dalam kapasitas yang mampu dibawa ataupun berdasarkan volume kapal. Hwang (2005) membedakan kapal berdasarkan produk yang dibawa yaitu sebagai berikut:

a. *Container Ship*

Kapal yang mengangkut produk-produk yang dikemas seperti makanan, mebel, mesin, kendaraan bermotor, pakaian, dan lain sebagainya

b. Kapal *Tanker*

Kapal *tanker* biasanya digunakan untuk mengangkut *crude oil*, produk kimia, dan produk-produk yang berbentuk cair lainnya

c. *Bulk Carrier*

Kapal *bulk carrier* biasanya digunakan untuk mengangkut produk jenis curah seperti bijih besi, batubara, bauksit, aluminium, dan jenis mineral lainnya. Kapal *bulk carrier* dapat mengangkut beberapa kargo dalam waktu bersamaan.

d. *Liquefied Gas Carrier*

Kapal yang biasanya digunakan untuk mengangkut gas dengan tekanan tinggi, diantaranya *Liquefied Natural Gas* (LNG) dan *Liquefied Petroleum Gas* (LPG). Kapal ini biasanya memiliki kompartemen – kompartemen yang dapat menjaga tekanan tinggi dan temperatur yang rendah untuk dapat mengangkut produk dengan volume yang besar.

2. Karakteristik Pelabuhan

Karakteristik pelabuhan biasanya berhubungan dengan aktivitas *loading* dan *unloading*. Menurut Nurminarsih (2012) terdapat beberapa karakteristik pelabuhan yang perlu diperhatikan, antara lain:

- a. Dimensi kapal yang dapat berlabuh di *port* biasanya berkaitan dengan kedalaman laut di pelabuhan dan besarnya pelabuhan.
- b. Biaya yang terkait dengan *loading* dan *unloading* di masing-masing pelabuhan.
- c. Waktu yang dibutuhkan untuk melakukan proses *loading* dan *unloading* pada masing-masing *port*.
- d. Kapasitas penyimpanan pada *port* untuk masing-masing kargo atau produk.

3. Karakteristik Kargo

Produk yang dibawa melalui distribusi laut sangat beragam serta memiliki karakteristik yang berbeda-beda. Menurut Nurminarsih (2012)

terdapat beberapa hal yang harus diperhatikan terkait dengan karakteristik produk yang didistribusikan melalui distribusi laut, yaitu:

- a. Penguapan produk, dimana penguapan produk biasanya terjadi pada produk yang berwujud gas atau cair seperti minyak dan produk kimia.
- b. Kontainer yang digunakan untuk produk cair perlu didesain sedemikian rupa sebagai upaya untuk mencegah terjadinya pencampuran produk satu sama lain.

2.3 Ship Scheduling and Assignment

Salah satu penelitian taktis terkait dengan transportasi maritim adalah *ship scheduling and assignment*. *Ship scheduling and assignment* merupakan suatu permasalahan yang membahas mengenai permasalahan penjadwalan dan penugasan kapal, yaitu kapan kapal harus dijadwalkan untuk mengirimkan barang dan kemana kapal harus ditugaskan. Menurut Christiansen et al. (2003), tujuan atau objektif dari permasalahan *ship scheduling and assignment* adalah meminimasi total biaya keseluruhan kapal yang digunakan untuk mengirimkan semua kargo dari *loading port* ke *discharge port*. Model yang sering digunakan untuk menyelesaikan permasalahan *ship scheduling* adalah *set partitioning* (SP). Sebanyak 40% dari studi literatur yang diamati, menggunakan model SP untuk menyelesaikan permasalahan *ship scheduling* (Christiansen et al., 2003).

Bronmo et al. (2005), mengembangkan *multi-start search heuristic* untuk menyelesaikan permasalahan *ship scheduling*. Dimana heuristik yang dikembangkannya mempertimbangkan permasalahan penjadwalan jangka pendek dengan menggunakan *tramp shipping (short-term scheduling problems within tramp shipping)*. *Multi-start search heuristic* yang dikembangkan dibandingkan dengan *set partitioning approach* berdasarkan waktu komputasi dan kualitas solusi.

2.4 *Simple Iterative Mutation Algorithm (SIM-A)*

Simple Iterative Mutation Algorithm (SIM-A) merupakan algoritma yang dikembangkan oleh Budiman, yang mengkombinasikan konsep mutasi dari *Genetic Algorithm (GA)* dan konsep *elite sample* dari algoritma *Cross Entropy (CE)* (Budiman, 2012). *Simple Iterative Mutation Algorithm* yang dikembangkan oleh Budiman digunakan untuk memecahkan permasalahan *Truck Scheduling Problem with Temporary Storage*. *Temporary Storage* yang dimaksud adalah barang dapat ditempatkan pada suatu terminal (tempat) selama rentang waktu tertentu sambil menunggu barang tersebut siap di *load* pada *outbound truck*. Menurut Budiman (2012), algoritma SIM-A dibuat dengan mempertimbangkan kelebihan tiap algoritma GA dan CE yang dikombinasikan. Dimana GA cocok digunakan untuk menyelesaikan permasalahan dengan skala besar (*wide range of search sample*), sedangkan CE memastikan algoritma mendapatkan *best results* melalui mekanisme sampel elit (*elite sample mechanism*).

SIM-A merupakan algoritma modifikasi dari algoritma *hybrid Cross Entropy* dan *Genetic Algorithm*, yang dikenal dengan istilah CEGA. Pada *hybrid CEGA*, dilakukan proses *crossover* dan mutasi secara simultan sedangkan pada SIM-A hanya dilakukan proses mutasi saja tanpa melakukan *crossover* sebelumnya sehingga SIM-A memerlukan waktu komputasi yang lebih sedikit dibandingkan CEGA (Budiman, 2012). Budiman (2012) lebih memilih menggunakan proses mutasi dibanding *crossover* dalam melakukan *sample generation* karena mutasi memiliki kemampuan yang lebih baik dibandingkan *crossover* untuk keluar dari *local optimum* dengan menggunakan *random exchange technique* untuk masing-masing gen yang dimutasi. Jika menggunakan proses *crossover*, peluang untuk keluar dari *local optimum* sangat kecil. Gambar 2.1 menunjukkan langkah-langkah dalam SIM-A (Budiman, 2012) :

2.4.1 Algoritma *Cross Entropy*

Algoritma *Cross Entropy* termasuk ke dalam teknik yang cukup baru, yang awalnya diterapkan untuk simulasi kejadian langka (*rare event*), kemudian dikembangkan untuk menyelesaikan beberapa kasus optimasi seperti optimasi kombinatorial, optimasi kontinyu, *machine learning*, dan beberapa masalah lain (Santosa dan Willy, 2011). Rubinstein (1997) menyatakan bahwa *Cross Entropy* (CE) awalnya digunakan untuk melakukan estimasi probabilitas dari *rare event* dengan penerapan algoritma adaptif pada peristiwa stokastik yang kompleks dengan cara meminimasi variansi (*variance minimization*). Menurut Santosa dan Willy (2011), *Cross Entropy* termasuk ke dalam teknik Monte Carlo yang digunakan untuk menyelesaikan permasalahan estimasi maupun optimasi. Untuk permasalahan optimasi, *Cross Entropy* (CE) memberikan cara yang adaptif untuk menemukan distribusi *sampling* yang optimal untuk beberapa permasalahan yang cakupannya cukup luas. Di sisi lain, untuk masalah optimasi yang diformulasikan sebagai permasalahan estimasi (*estimation problem*), metode *Cross Entropy* akan sangat handal dan berlaku umum sebagai *search stochastic algorithm*.

Sama halnya dengan Monte Carlo, dalam algoritma *Cross Entropy* (CE) akan dibangkitkan sejumlah sampel, setelah itu dilakukan pembangkitan sampel *random* berikutnya yang lebih baik dari sampel sebelumnya dengan menggunakan parameter yang didapatkan dari sampel sebelumnya. Terdapat aturan penting untuk mengupdate parameter dalam algoritma *Cross Entropy* (Santosa dan Willy, 2011). Menurut Santosa dan Willy (2011), metoda *Cross Entropy* (CE) melibatkan prosedur iterasi, dimana setiap iterasi dapat dipecah menjadi dua fase:

1. Membangkitkan sampel *random* (x) dengan menggunakan mekanisme atau distribusi tertentu.
2. Melakukan *update* parameter (v) dari mekanisme *random* berdasarkan data sampel elite untuk menghasilkan sampel yang lebih baik pada iterasi berikutnya. Sampel elite merupakan berapa persen dari sampel yang dipilih untuk mengupdate parameter yang digunakan dalam permasalahan yang dipecahkan.

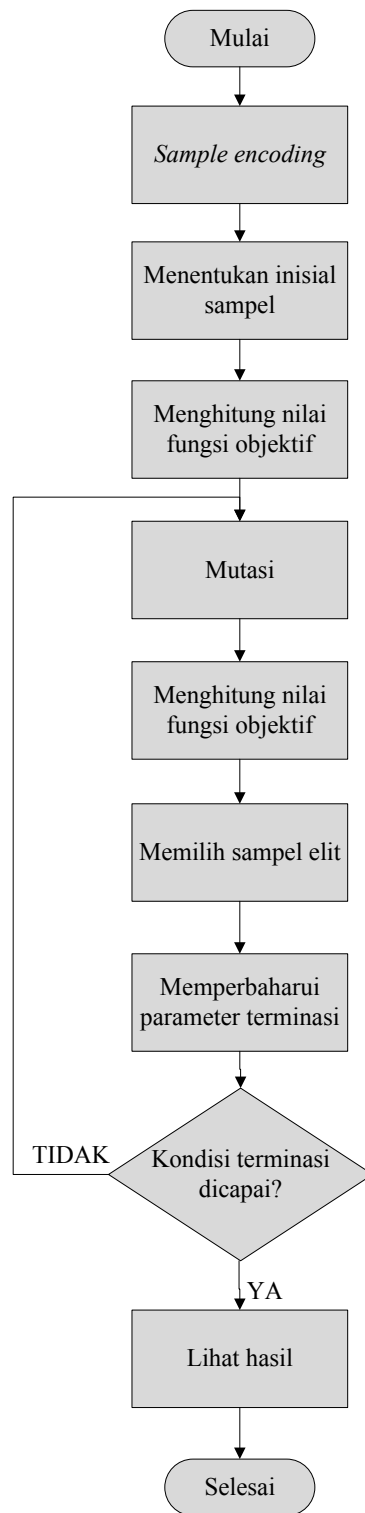
Menurut Damayanti (2013), ada beberapa parameter yang harus ditentukan di awal dalam metode *Cross Entropy*, diantaranya: vektor inisial \hat{v}_0 , ukuran sampel N , nilai *percentile* dari sampel elit ρ , dan parameter *smoothing* α serta *stopping criteria*. Menurut Santosa dan Willy (2011), algoritma utama CE untuk menyelesaikan permasalahan optimasi yaitu:

1. Tentukan parameter awal $\hat{v} = u, \alpha$, dan ρ . Tetapkan $it = 1$
2. Bangkitkan sampel *random* X_1, \dots, X_N dari fungsi probabilitas distribusi tertentu $f(-; u)$ dan pilih sampel $(1 - \rho)$ *quantile* dari performansi setelah diurutkan
3. Gunakan sampel yang sama untuk memperbarui nilai parameter
4. Untuk memuluskan vector $\hat{v} = u$ gunakan persamaan sebagai berikut:

$$\hat{v}_t = \alpha \tilde{w}_t + (1 - \alpha) \hat{v}_{t-1} \quad (2.1)$$

Setelah itu kembali ke langkah 2 dengan nilai parameter yang baru, dan *set it = it*
+ 1

5. Jika *stopping criteria* sudah dipenuhi, iterasi dihentikan.



Gambar 2. 1 *Flowchart* SIM-A (Budiman, 2012)

2.4.2 Genetic Algorithm

Genetic Algorithm atau algoritma genetika (GA) masuk ke dalam kelompok *Evolutionary Problem*, dimana GA didasarkan pada prinsip-prinsip genetika dan seleksi alam (Santosa dan Willy, 2011). Metoda GA terkenal setelah buku John Holland dirilis pada tahun 1970-an yang berjudul “*Adaption in Natural and Artificial System*” yang merupakan hasil penelitiannya di University of Michigan (Budiman, 2012). Menurut Damayanti (2013), penentuan populasi yang akan digunakan dalam pencarian solusi merupakan konsep terpenting dalam GA. Berbeda dengan metode pencarian tradisional, GA berdasar pada populasi yang menjadi kandidat solusi. Dimana ukuran populasi berupa parameter merupakan faktor penting yang mempengaruhi stabilitas dan performansi GA. Jika ukuran populasi kecil akan menyebabkan terjadinya konvergensi dalam waktu yang cepat dan solusi yang dihasilkan kurang optimal. Sedangkan ukuran populasi yang besar menyebabkan waktu komputasi menjadi lama. Reproduksi, *crossover*, dan mutasi merupakan elemen dasar dari algoritma GA. *Genetic Algorithm* banyak dipakai untuk menyelesaikan masalah kombinatorial seperti *Traveling Salesman Problem* (TSP), *Vehicle Routing Problem* (VRP), *crew scheduling* untuk *airline* hingga permasalahan kontrol (Santosa dan Willy, 2011). Prosedur pencarian dalam *Genetic Algorithm* hanya didasarkan pada nilai fungsi tujuan, tidak ada pemakaian *gradient* atau teknik kalkulus.

Menurut Santosa dan Willy (2011), bagian penting dari GA adalah kromosom, dimana satu kromosom atau individu mewakili satu vektor solusi. Terkadang bisa secara langsung menggunakan vektor solusi ini dalam implementasi GA atau juga bisa dilakukan *encoding* atau pengkodean. Dimana pengkodean dilakukan untuk mewakili suatu nilai solusi dengan menggunakan bilangan biner, tergantung pada permasalahan yang ingin diselesaikan. Biasanya nilai kontinyus akan diwakili dengan bilangan biner dalam menyelesaikan kasus optimasi. Jika pada pengerjaan algoritma diawali dengan pengkodean (enkoding) maka pada akhir *Genetic Algorithm* akan dilakukan proses decoding untuk mengembalikan nilai kontinyus asalnya.

Dalam *Genetic Algorithm* diperlukan ukuran untuk membedakan solusi yang baik dengan yang buruk sebagai upaya untuk mendapatkan solusi

yang baik dan untuk mengimplementasikan seleksi alam (Damayanti, 2013). Pengukuran solusi bisa dilihat dari fungsi objektif yang didapatkan dari permasalahan yang diselesaikan dengan menggunakan algoritma GA. Namun pada intinya, melalui mekanisme *fitness function* yang akan membawa solusi ke arah yang lebih optimal. Berikut ini merupakan langkah-langkah dalam algoritma GA menurut Santosa dan Willy (2011) :

1. Inisialisasi

Pada tahap ini dilakukan pembangkitan populasi awal atau kromosom-kromosom awal yang dilakukan secara *random*. Dimana kromosom-kromosom awal yang dibangkitkan merupakan kandidat solusi dari permasalahan yang diselesaikan. Pada tahap inisialisasi juga dilakukan penentuan ukuran populasi yang dibangkitkan, probabilitas *crossover* dan probabilitas mutasi. Jika pada tahap inisialisasi dibangkitkan sebanyak n kromosom awal dalam populasi yang dibangkitkan, maka pada setiap iterasi akan didapatkan sebanyak n alternatif solusi.

2. Evaluasi

Pada tahap ini dilakukan evaluasi terhadap setiap kromosom yang dibangkitkan di awal dengan menggunakan *fitness function*. *Fitness function* digunakan untuk mengukur tingkat kesesuaian suatu solusi dengan solusi yang dicari. Untuk permasalahan maksimasi, rumus *fitness function* - nya adalah sebagai berikut:

$$F(x) = f(x) \tag{2.2}$$

Sedangkan untuk permasalahan minimasi, rumus *fitness function* yang digunakan adalah sebagai berikut:

$$F(x) = \frac{1}{1 + f(x)} \tag{2.3}$$

Keterangan:

$F(x)$: *fitness function*

$f(x)$: fungsi tujuan

Sebelum melangkah pada tahap selanjutnya, perlu dilakukan penentuan iterasi $t = 1$

3. Elitisme

Pada tahap ini dilakukan pemilihan terbaik untuk disalin atau di-copy sejumlah tertentu untuk mengganti individu lain (elitism). Elitisme merupakan usaha yang dilakukan untuk mempertahankan individu terbaik yang telah diperoleh, sehingga individu yang dielitisme akan tetap muncul pada iterasi berikutnya.

4. Seleksi

Proses seleksi dilakukan untuk memilih anggota populasi (kromosom) sebagai induk untuk dilakukan kawin silang. Pemilihan induk didasarkan pada nilai *fitness* yang lebih tinggi, dimana kromosom dengan nilai *fitness* tertinggi akan di-copy. Berikut ini merupakan beberapa prosedur yang dapat dilakukan untuk proses seleksi (Damayanti, 2013) :

a. *Roulette – wheel selection*

Pada prosedur ini, setiap individu pada populasi akan dibagi ke dalam *roulette wheel* dengan proporsi sesuai nilai *fitness function* yang didapatkan. Dimana individu dengan nilai *fitness* tertinggi memiliki kesempatan besar untuk terpilih sebagai induk.

b. *Tournament selection*

Pada prosedur ini, s kromosom akan dipilih secara acak (*random*) dan akan masuk ke dalam *tournament* untuk saling melawan satu sama lain. Individu dengan nilai *fitness* (kesesuaian) tertinggi pada kelompok kromosom k akan memenangkan *tournament* dan dipilih sebagai induk.

c. *Truncation Selection*

Pada prosedur seleksi ini, urutan (1/s) teratas dari kumpulan individu akan disalin masing-masing sebanyak s pada *mating pool*.

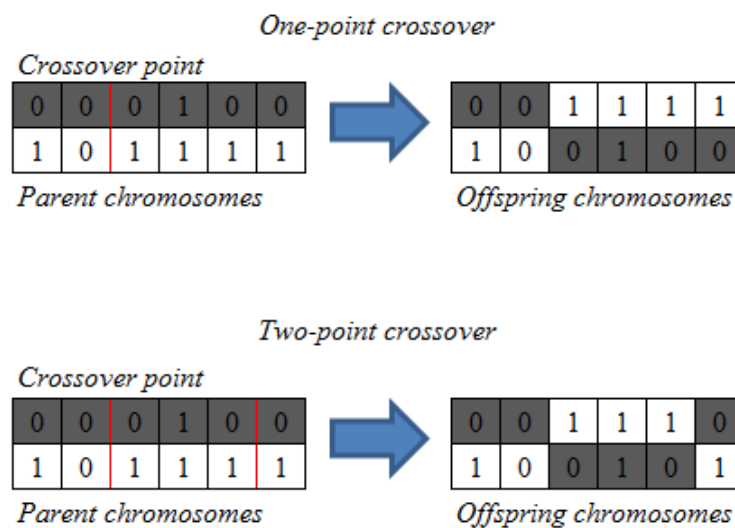
5. *Crossover*

Pada tahap ini dilakukan kawin silang antar induk yang terpilih pada tahap sebelumnya (seleksi) untuk menghasilkan anak atau kromosom baru yang memiliki kemungkinan untuk menghasilkan solusi yang lebih baik. Pada *crossover* sebageaian besar dilakukan dengan cara memilih dua individu (kromosom) secara *random* dan akan dikawin silang dengan probabilitas

crossover sebesar p_c . Kemudian probabilitas p_c dibandingkan dengan bilangan *random* yang dibangkitkan, r (0-1). Jika $r > p_c$ maka dua keturunan akan digantikan secara langsung dari induknya tanpa melalui kawin silang (induk langsung menjadi anak). Jika $r \leq p_c$, maka dua individu akan dipilih dari hasil kawin silang yang didapatkan. Berikut ini merupakan beberapa cara *crossover* yang bisa dilakukan`:

1. *k - point crossover*

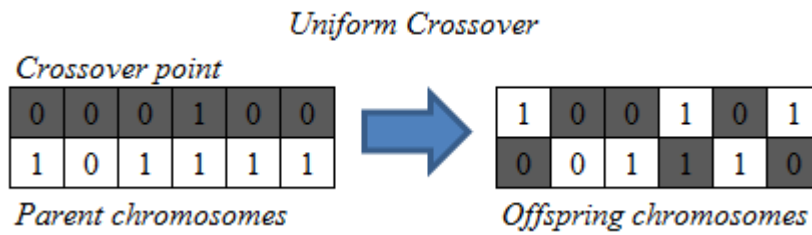
One-point dan *two-point crossover* merupakan metoda *crossover* yang mudah dan sering dilakukan dalam melakukan kawin silang pada GA (Sasty et al., 2005). Pada metode ini, tempat terjadinya kawin silang dipilih secara *random* dan tidak boleh melebihi panjang kromosom (Damayanti, 2013). Berikut ini merupakan ilustrasi dari *one-point* dan *two-point crossover*:



Gambar 2. 2 *One-point* dan *Two-point Crossover* (Sastry et al., 2005)

2. *Uniform crossover*

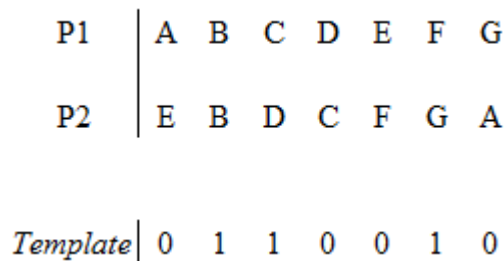
Pada *uniform crossover*, anak akan dihasilkan dengan menukarkan antar kromosom induk yang dipilih secara acak (*random*) dengan probabilitas p_e yang disebut probabilitas *swap*, dimana nilai p_e yang sering digunakan sebesar 0.5. Berikut ini merupakan ilustrasi dari *uniform crossover* (Sastry et al., 2005) :



Gambar 2. 3 *Uniform Crossover* (Sastry et al., 2005)

3. *Uniform order-based crossover*

Menurut Sastry et al. (2005), *k-point crossover* dan *uniform crossover*, tidak cocok digunakan untuk menyelesaikan permasalahan kombinatorial yang memiliki mekanisme permutasi seperti TSP, karena anak yang dihasilkan dari hasil *crossover* sering kali menghasilkan solusi yang tidak valid. Sehingga diperlukan mekanisme perbaikan keturunan supaya menghasilkan solusi yang valid. Misalkan induk P1 dan P2, pada metode *uniform order-based crossover* kedua induk tersebut (P1 dan P2) akan dipilih secara *random*.



Gambar 2. 4 *Parents* dan *Template* dalam *Uniform Order-based crossover* (Sastry et al., 2005)

Dalam *uniform order-based crossover*, keturunan C1 diambil dari P1 dengan nomor kolom yang sama dengan *template* yang bernilai 1, kemudian kolom C1 yang masih kosong diambil dari P2 dengan nomor kolom yang sama dengan *template* yang bernilai 0. Keturunan C2, diambil dari P1 dengan nomor kolom yang sama dengan *template* yang bernilai 0 dan kolom C2 yang masih kosong diambil dari P2 dengan nomor kolom yang sama dengan *template* yang bernilai 1. Gambar 2.5

merupakan ilustrasi hasil yang dihasilkan dari *parents* (P1 dan P2) serta *template* pada gambar 2.4:

C1	E	B	C	D	G	F	A
C2	A	B	D	C	E	G	F

Gambar 2. 5 Hasil *Uniform Order-based Crossover*

(Sastry et al., 2005)

4. *Order-based crossover*

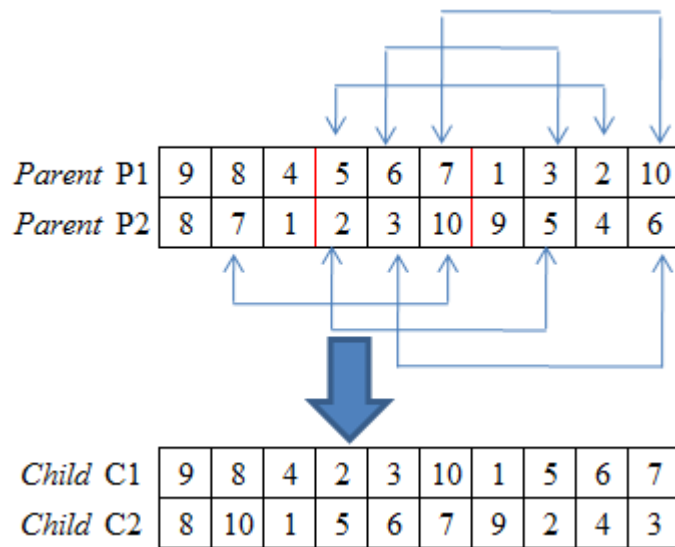
Dalam metode *order-based crossover*, dua induk dipilih secara *random* dan anak akan terbentuk melalui proses kawin silang yang dilakukan secara *random* (Sastry et al., 2005). Berikut ini merupakan ilustrasi dari *order-based crossover*:

<i>Parent P1</i>	A	B	C	D	E	F	G
<i>Parent P2</i>	C	B	G	E	F	D	A
<i>Child C1</i>	?	?	C	D	E	?	?
<i>Child C2</i>	?	?	G	E	F	?	?
<i>Child C1</i>	F	A	C	D	E	B	G
<i>Child C2</i>	C	D	G	E	F	A	B

Gambar 2. 6 *Order-based Crossover* (Sastry et al., 2005)

5. *Partially matched crossover*

Menurut Sastry et al. (2005), pada metode *partially matched crossover*, induk dan tempat terjadinya kawin silang dipilih secara *random*. Dimana gen – gen tempat terjadinya kawin silang diubah dengan gen milik induk, sehingga hasil kawin silang C1 (*child 1*) didapatkan dari P1 (*parent 1*) dan C2 (*child 2*) diambil dari P2 (*parent 2*). Berikut ini merupakan ilustrasi dari *partially matched crossover* :



Gambar 2. 7 *Partially Matched Crossover* (Sastry et al., 2013)

6. *Cycle crossover*

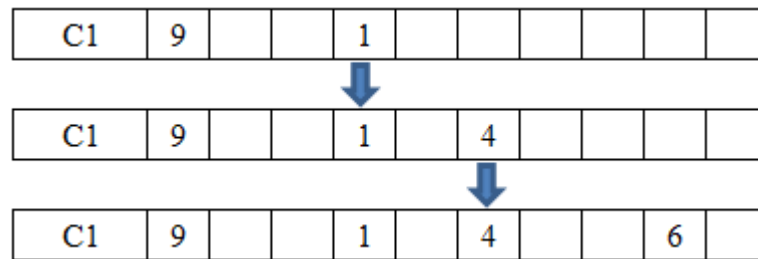
Penjelasan mengenai *cycle crossover* akan dilakukan melalui contoh berikut ini :

Misalkan terdapat dua induk, sebut saja P1 dan P2 sebagai berikut:

P1	9	8	2	1	7	4	5	10	6	3
P2	1	2	3	4	5	6	7	8	9	10

Gambar 2. 8 *Parents* dalam *Cycle Crossover* (Sastry et al., 2013)

Proses kawin silang akan menghasilkan dua anak, sebut saja C1 (*child 1*) dan C2 (*child 2*). C1 akan menerima variabel dari P1 yaitu angka 9. Kemudian dipilih angka pada P2 yang terletak pada kolom yang sama dengan angka 9 pada P1 yang dipilih tadi. Angka yang terletak satu kolom dengan angka 9 pada P1 adalah angka 1. Angka 1 ini akan diambil sebagai angka selanjutnya yang dipilih dari P1, begitu juga seterusnya. Berikut ini ditampilkan proses dari *cycle crossover* :



Gambar 2. 9 Gambaran Proses *Cycle Crossover* (Sastry et al., 2005)

Setelah angka 6 terpilih, angka selanjutnya yang seharusnya dipilih adalah angka 9, namun karena angka 9 sudah terpilih maka proses *cycle crossover* dihentikan. Bagian yang masih kosong dari C1 diisi dengan angka dari P2. Sedangkan C2 akan menyesuaikan dengan hasil C1.

P1	9	2	3	1	5	4	7	8	6	10
P2	1	8	2	4	7	6	5	10	9	3

Gambar 2. 10 Hasil Metode *Cycle Crossover* (Sastry et al., 2005)

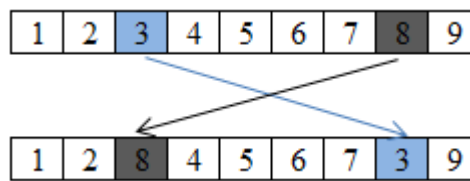
7. Mutasi

Menurut Santosa dan Willy (2011), mutasi dilakukan dengan tujuan untuk memunculkan individu baru yang bukan berasal dari proses *crossover* yang memungkinkan solusi baru bisa keluar dari *local optimum*. Mutasi memiliki probabilitas mutasi sebesar p_m . Jika p_m sebesar 0.05, maka akan ada sekitar 5% dari semua kromosom dalam populasi akan mengalami proses mutasi. Pada proses mutasi akan dibangkitkan bilangan *random* r (0-1), untuk menentukan proses mutasi yang akan dilakukan dengan cara membandingkan dengan probabilitas mutasi yang telah ditetapkan. Ursani et al. (2011) menyatakan bahwa terdapat tiga cara yang dapat dilakukan dalam melakukan mutasi. Berikut ini merupakan penjelasan dari 3 cara dalam melakukan mutasi dalam *Genetic Algorithm*:

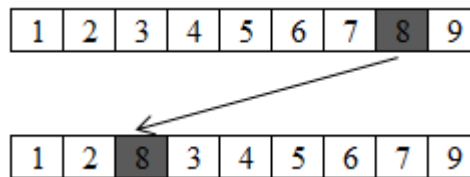
- a. *Swap* : Dua buah *gen* yang terpilih secara *random* akan ditukar posisinya.

- b. *Insertion* : *gen* yang terpilih secara *random* akan dikeluarkan dari posisi (lokasi) dan kemudian posisi tersebut akan digantikan oleh *gen* lain yang terpilih secara *random*.
- c. *Inversion* : Proses mutasi ini dilakukan dengan cara membalik urutan *gen* yang terpilih secara *random*.

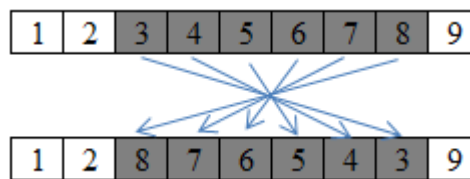
Berikut ini merupakan ilustrasi dari 3 cara proses mutasi tersebut di atas:



Gambar 2. 11 *Swap Mutation* (Ursani et al., 2011)



Gambar 2. 12 *Insertion Mutation* (Ursani et al., 2011)



Gambar 2. 13 *Inversion Mutation* (Ursani et al., 2011)

Menurut Damayanti (2013), ada beberapa jenis mutasi yang lain selain *swap*, *insertion*, dan *inversion mutation* yaitu *forward mutation* dan *backward mutation*. *Gen* yang terpilih secara *random* akan dipindahkan ke urutan pertama dikenal dengan istilah *forward mutation*. Sedangkan *gen* yang terpilih secara *random* dipindahkan ke urutan terakhir dikenal dengan istilah *backward mutation*.

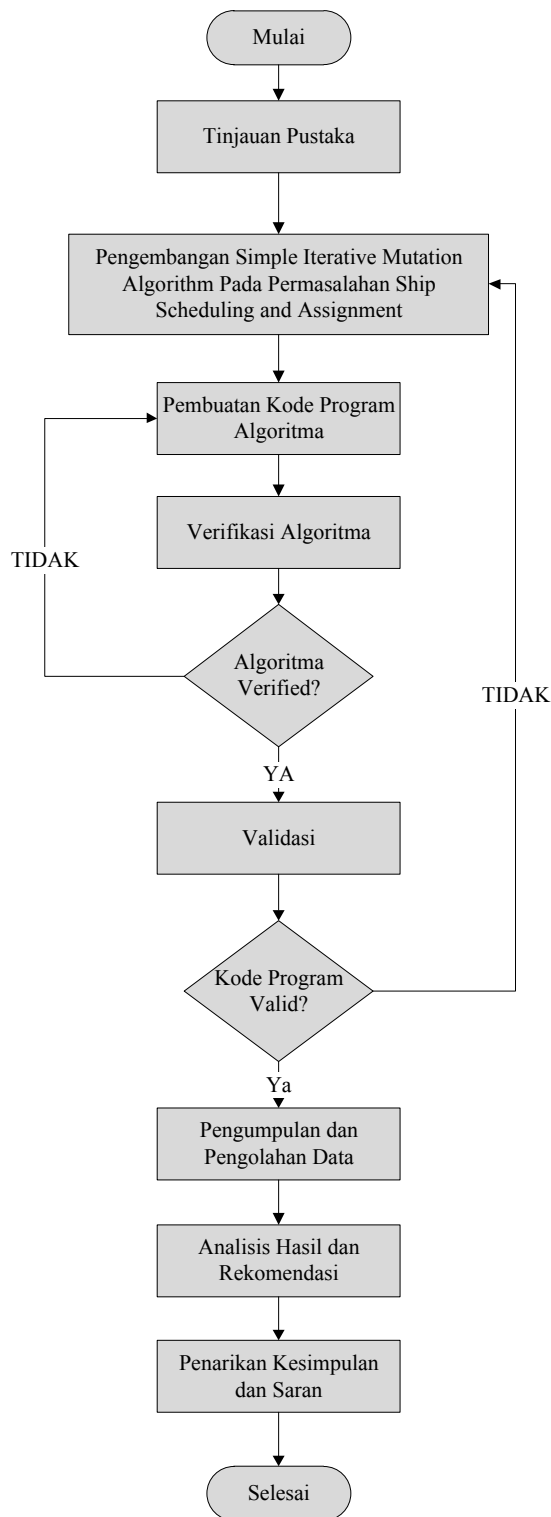
BAB 3

METODOLOGI PENELITIAN

Pada bagian ini dijelaskan mengenai langkah-langkah yang dilakukan dalam penelitian tugas akhir untuk mengembangkan *Simple Iterative Mutation Algorithm* dalam menyelesaikan permasalahan *Ship Scheduling and Assignment* yang mempertimbangkan *minimum requirement draft constraint*. Metodologi penelitian ini digunakan sebagai dasar atau acuan dalam melakukan penelitian sehingga penelitian dapat berjalan secara sistematis. Langkah – langkah yang akan dilakukan dalam penelitian tugas akhir ini dapat dilihat pada Gambar 3.1.

3.1 Studi Literatur

Pada tahap studi literatur dilakukan pencarian literatur yang berkaitan dengan pengembangan *Simple Iterative Mutation Algorithm* pada permasalahan *Ship Scheduling and Assignment*, diantaranya mengenai manajemen distribusi, *maritime logistics*, transportasi laut, *Ship Scheduling and Assignment*, dan *Simple Iterative Mutation Algorithm*.



Gambar 3. 1 Alur Pengerjaan Penelitian Tugas Akhir

3.2 Tahap Pengembangan Algoritma

Pada tahap pengembangan algoritma terbagi menjadi beberapa tahap yaitu tahap pengembangan algoritma *Simple Iterative Mutation Algorithm* pada permasalahan *Ship Scheduling and Assignment*, tahap validasi dan verifikasi algoritma yang dikembangkan.

Simple Iterative Mutation Algorithm dikembangkan dan pernah digunakan oleh Budiman (2012) untuk menyelesaikan permasalahan *truck scheduling* dengan sistem *cross docking*. *Simple Iterative Mutation Algorithm* merupakan kombinasi dari konsep mutasi dari *Genetic Algorithm* (GA) dan konsep sampel elit dari algoritma *Cross Entropy* (CE) (Budiman, 2012). *Simple Iterative Mutation Algorithm* mempertimbangkan kelebihan algoritma GA dan CE yang dikombinasikan. Dimana *Genetic Algorithm* dikembangkan untuk menyelesaikan masalah dengan *search sample* yang besar, sedangkan *Cross Entropy* memastikan algoritma yang dirancang menghasilkan nilai optimal melalui mekanisme sampel elit. Jika untuk menyelesaikan permasalahan dengan skala yang besar, digunakan algoritma *Cross Entropy* tanpa dikombinasikan dengan algoritma lain akan memerlukan waktu komputasi yang lama (Damayanti, 2013). Oleh karena itu, diperlukan algoritma lain untuk dikombinasikan dengan algoritma CE untuk mengurangi waktu komputasi. Menurut Santosa et al (2011) dalam Damayanti (2013), *Genetic Algorithm* merupakan salah satu algoritma yang dapat dikombinasikan dengan algoritma *Cross Entropy* untuk mengurangi waktu komputasi. Oleh karena itu, dengan melihat tingkat kompleksitas permasalahan penugasan dan penjadwalan (*Ship Scheduling and Assignment*) yang terdiri dari banyak kapal dan banyak *port* tujuan maka SIM-A merupakan algoritma yang tepat digunakan untuk menyelesaikan permasalahan tersebut karena memerlukan waktu komputasi yang relatif sedikit dan hasil yang didapatkan mendekati nilai optimal.

3.2.1 Tahap Pengembangan Algoritma SIM-A

Pada tahap ini dilakukan pengembangan *Simple Iterative Mutation Algorithm* pada permasalahan *Ship Scheduling and Assignment* yang mempertimbangkan *Minimum requirement draft constraint* supaya kapal bisa

berlabuh di *port*. *Simple Iterative Mutation Algorithm* (SIM-A) yang dikembangkan pada penelitian ini bersifat *single to many*, yang terdiri dari satu *production port* dan beberapa *port* tujuan. Dimana horison perencanaan untuk penjadwalan dan penugasan kapal dibatasi maksimum 1 (satu) bulan dengan periode waktu (t) dalam hari. Setiap kali penugasan, hanya diijinkan maksimal dua kapal yang bisa melakukan proses *loading* di *production port* dalam waktu yang bersamaan.

3.2.1.1 Skenario 1 (Kondisi Eksisting) : SIM-A dengan Time Windows Depot dan *Port* 08:00 – 18:00

Skenario 1 (kondisi eksisting) merupakan algoritma SIM-A untuk penjadwalan dan penugasan kapal, dimana waktu mulai proses *loading* di *production port* hanya bisa dilakukan dalam rentang waktu 12 jam, yaitu jam 08:00 – 18:00. Jika pada saat periode penugasan kapal datang sebelum jam 08:00, maka kapal tersebut akan menunggu sampai proses *loading* bisa dilakukan. Begitu juga untuk waktu mulai proses *unloading* bisa dilakukan dari jam 08:00 – 18:00.

3.2.1.2 Skenario 2 : SIM-A dengan Time Windows Depot 0:00- 24:00 dan *Port* 08:00 – 18:00

Skenario 2 merupakan algoritma SIM-A yang mengijinkan kapal memulai proses *loading* selama 24 jam penuh dari jam 0:00 – 24:00. Artinya kapanpun kapal sampai di *port* kapal bisa melakukan proses *unloading* tanpa harus menunggu waktu buka operasional depot. Pada skenario 2, proses *unloading* di *port* tujuan hanya boleh dimulai dari jam 08:00 – 18:00.

3.2.2 Verifikasi dan Validasi Algoritma

Pada tahap ini dibagi menjadi dua tahapan yaitu verifikasi dan validasi dari *Simple Iterative Mutation Algorithm* (SIM-A) yang dikembangkan.

3.2.2.1 Verifikasi

Verifikasi terhadap algoritma SIM-A yang dikembangkan dilakukan untuk mengecek kesesuaian antara logika operasional kode yang dibuat pada *software* MATLAB R2010a dengan *flowchart* pengembangan algoritma SIM-A. Dengan proses verifikasi ini, akan diketahui apakah kode algoritma yang dibuat pada MATLAB R2010a masih ada *error* atau tidak. Pengecekan kode pemrograman juga dilakukan dengan mengecek apakah kode pemrograman yang telah dibuat benar – benar telah memenuhi semua konstrain yang dipertimbangkan atau belum.

3.2.2.2 Validasi

Validasi dilakukan pada algoritma SIM-A yang dikembangkan dengan cara melakukan perhitungan enumerasi pada permasalahan sederhana. Kemudian hasil enumerasi dibandingkan dengan hasil *running* algoritma sesuai dengan *flowchart* yang telah dibuat. Pada tahap validasi, algoritma yang divalidasi adalah algoritma SIM-A pada skenario 1, karena skenario tersebut mewakili kondisi eksisting penjadwalan dan penugasan kapal pada PT. X.

3.3 Tahap Aplikasi Algoritma Untuk Sistem Riil

Pada tahap ini dilakukan aplikasi dari algoritma SIM-A untuk sistem riil pada permasalahan penjadwalan dan penugasan kapal untuk distribusi semen curah pada PT.X. Pada tahap ini dibagi menjadi dua tahapan yaitu tahap pengumpulan dan pengolahan data. Data-data yang diperlukan untuk penelitian ini terlebih dahulu dikumpulkan dan didapatkan secara langsung dari PT. X. Kemudian data tersebut diolah sesuai dengan algoritma yang dikembangkan untuk mendapatkan solusi permasalahan mengenai *Ship Scheduling and Assignment* pada distribusi semen curah PT. X dengan menggunakan algoritma SIM-A yang dikembangkan.

3.3.1 Pengumpulan Data

Pada penelitian ini, data yang didapatkan merupakan data sekunder yang didapatkan langsung dari PT. X. Adapun data yang dikumpulkan dari PT. X adalah sebagai berikut:

1. Data kapal

Data kapal yang dibutuhkan meliputi jumlah kapal, kecepatan *loading* dan *unloading* kapal, kapasitas kapal, *minimum requirement draft* masing-masing kapal beserta dengan parameter biayanya.

2. Data pelabuhan

Data pelabuhan yang dibutuhkan meliputi jumlah *port* tujuan yang harus dilayani, kedalaman dermaga, *consumption rate*, *on-hand* awal, kapasitas maksimum, dan *time windows*.

3. Data historis pengiriman semen curah

Data historis terkait pengiriman dan penugasan kapal yang dibutuhkan meliputi waktu tunggu sebelum proses *loading*, waktu perjalanan dari *production port* menuju *port* tujuan, dan waktu tunggu sebelum proses *unloading*.

3.3.2 Pengolahan Data

Pada tahap ini, dilakukan pengolahan terhadap data yang diperoleh pada tahap pengumpulan data. Pengolahan data dilakukan untuk mendapatkan informasi yang diperlukan pada tahap selanjutnya. Adapun informasi yang dibutuhkan adalah informasi detail mengenai *waiting time* untuk proses *loading* dan *unloading*, *loading time*, *unloading time*, *shipment time* dan *cycle time*.

3.3.2.1 Perhitungan Nilai Parameter

Di dalam algoritma penjadwalan yang dibangun, setiap kapal yang ditugaskan untuk proses pengiriman memiliki parameter proses. Berikut ini merupakan parameter proses untuk setiap kapal beserta dengan cara perhitungannya (jika ada).

1. Waktu Tunggu

Waktu tunggu merupakan waktu yang dibutuhkan kapal untuk bisa melakukan suatu proses tertentu. Pada penelitian ini, waktu tunggu dibagi menjadi 2 yaitu waktu tunggu sebelum *loading* (WTSL) dan waktu tunggu sebelum *unloading* (WTSU).

a. Waktu Tunggu Sebelum *Loading* (WTSL)

Waktu tunggu sebelum *loading* merupakan waktu yang dibutuhkan kapal untuk menunggu sampai bisa melakukan *loading* di *production port*. Besaran WTSL ditentukan dari data historis yang ada. Dari data historis tersebut kemudian dicari probabilitas terjadinya WTSL dan dilakukan *fitting distribution* terhadap data tersebut untuk mencari parameter lama waktu tunggu untuk proses *loading*. Probabilitas dan lama waktu tunggu *loading* inilah yang akan dijadikan parameter input dalam perhitungan algoritma yang dikembangkan.

b. Waktu Tunggu Sebelum *Loading* (WTSU)

WTSU merupakan waktu yang dibutuhkan kapal untuk menunggu sampai bisa melakukan *unloading* di *port* tujuan. Besaran WTSU ditentukan dari data historis yang ada. Dari data tersebut ditentukan besaran probabilitas terjadinya WTSU di *port* tujuan. Di samping itu juga dilakukan *fitting distribution* untuk mencari parameter lama waktu tunggu sebelum *unloading*. Probabilitas atau besar kemungkinan dan lama WTSU akan dijadikan parameter input dalam perhitungan algoritma yang dikembangkan.

3.4 Tahap Analisis Hasil dan Rekomendasi

Pada tahap ini dilakukan analisis dan interpretasi dari hasil pengolahan data yang dilakukan pada bagian sebelumnya. Analisis dan interpretasi hasil yang dilakukan antara lain mengenai hasil yang didapatkan dari dua skenario yang dikembangkan. Di samping itu juga, hasil dari dua skenario ini dibandingkan dilihat dari *total cost* yang dihabiskan untuk penjadwalan dan penugasan kapal selama horison perencanaan yang telah ditetapkan. Setelah itu, dilakukan penentuan rekomendasi dari analisis hasil dan perbandingan dari dua skenario yang dikembangkan.

(halaman ini sengaja dikosongkan)

BAB 4

PENGEMBANGAN ALGORITMA

Pada bab ini akan dilakukan pengembangan algoritma SIM-A yang akan dijelaskan tentang bagaimana cara mengembangkan *Simple Iterative Mutation Algorithm* (SIM-A) untuk menyelesaikan permasalahan *Ship Scheduling and Assignment* pada distribusi semen curah pada PT. X.

4.1 Gambaran Umum Permasalahan

Permasalahan *Ship Scheduling and Assignment* dengan menggunakan *Simple Iterative Mutation Algorithm* yang dikembangkan pada penelitian ini diterapkan untuk permasalahan distribusi semen curah dari *production port* (depot) menuju *packing plant*. Permasalahan ini mengintegrasikan antara permasalahan *inventory management* masing-masing *packing plant* dan distribusi produk (semen) dari *production port* menuju *packing plant* selama horison perencanaan tertentu. Permasalahan *inventory management* membahas mengenai bagaimana cara mengontrol persediaan di *packing plant* agar tidak terjadi kekurangan (*stock out*). Pada permasalahan ini, *stock out* diijinkan terjadi akan tetapi hal itu lebih baik untuk dihindari. Hal tersebut seringkali disebut sebagai *soft constraint*. Ketika terjadi *stock out* di *packing plant* maka akan dikenakan *penalty cost* per satuan kekurangan, dalam hal ini satuan yang digunakan adalah ton per periode. Dalam penelitian ini tidak dipertimbangkan konstrain yang bersifat *hard constraint* terkait dengan *On-Hand* yaitu di suatu periode tertentu jika *On-Hand* melebihi kapasitas maksimum silo hal tersebut diijinkan dalam penelitian ini meskipun dalam riil nya hal tersebut tidak mungkin terjadi, sehingga dalam penelitian ini juga dipertimbangkan *penalty cost* akibat *On-Hand* melebihi kapasitas. Hal yang dapat dilakukan untuk mengurangi *stock out* dalam jumlah yang besar pada masing-masing *packing plant* yaitu dengan merancang mekanisme penjadwalan dan penugasan kapal selama horison perencanaan tertentu untuk mengkover semua *port* tujuan yang ada yang menghasilkan total biaya paling minimum. Dalam permasalahan ini kapal yang digunakan v bersifat

undedicated, artinya kapal v tidak selalu digunakan untuk melayani *port* yang sama selama horison perencanaan yang ditetapkan. Misalnya, pada penjadwalan yang pertama kapal 1 ditugaskan untuk melayani *port* tujuan 2, ketika kapal tersebut telah kembali ke depot maka penjadwalan atau penugasan kedua untuk kapal bisa jadi digunakan untuk melayani *port* tujuan 5.

Kapal yang digunakan untuk mengirimkan semen dari *production port* (depot) menuju *port* tujuan merupakan kapal sewa dengan sistem *time charter*. Dalam satu kali penjadwalan, satu kapal hanya bisa melayani satu *port* tujuan karena kapasitas kapal yang digunakan relatif sama dengan kapasitas *packing plant* di *port* tujuan. Saat awal horison perencanaan semua kapal yang disewa berstatus *available* untuk dijadwalkan. Setiap periode t dalam horison perencanaan, *production port* hanya mampu menampung maksimal 2 kapal untuk melakukan proses *loading* dalam waktu yang bersamaan. Dengan waktu *loading* yang lama, maka setiap periode t hanya ada dua kapal yang mampu dijadwalkan untuk melakukan pengiriman semen ke dua *port* tujuan kritis sehingga dalam algoritma yang dikembangkan ditambahkan batasan yaitu maksimal jumlah kapal yang dijadwalkan pada periode t yaitu sebanyak 2 kapal. Pengiriman dan penjadwalan kapal setiap periode t didasarkan pada *stock criticality port* tujuan dan jumlah *inventory position* (IP) pada awal periode t yang dibandingkan dengan *Re-shipment Point* (RSP) *port* tujuan. Dua *port* dengan *stock criticality* paling rendah dan jika kondisi *inventory position* pada awal periode pada *port* tersebut kurang dari atau sama dengan RSP, maka dua *port* tersebut terpilih sebagai *port* (*packing plant*) yang akan dilayani terlebih dahulu untuk dikirimkan semen.

Production port (depot) dalam permasalahan ini diasumsikan tidak pernah mengalami kekurangan dan mampu melayani *port* tujuan berapapun jumlah yang diperlukan selama horison perencanaan yang ditetapkan. Dalam pengembangan algoritma batasan yang dipertimbangkan adalah *minimum requirement draft* kapal yang mampu singgah di *port*. Semua kapal yang disewa mampu berlabuh di dermaga depot karena kedalaman dermaga lebih besar dari pada *minimum requirement draft* (MRD) masing-masing kapal. Akan tetapi tidak semua kapal bisa berlabuh di *port* tujuan, karena tidak semua MRD kapal lebih kecil atau sama dengan kedalaman dermaga di *port* tujuan. Kapal akan bisa

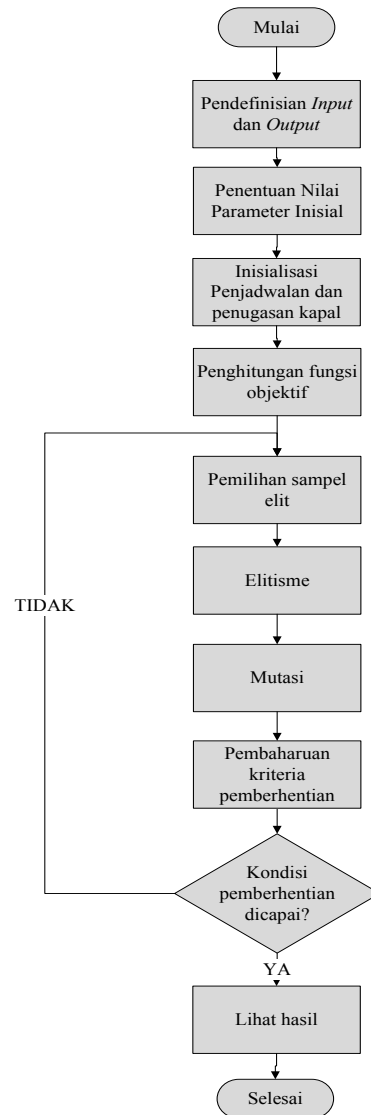
berlabuh di *port* tujuan jika *minimum requirement draft* kapal kurang dari atau sama dengan kedalaman dermaga. Akan tetapi, jika *minimum requirement draft* (MRD) kapal lebih besar dari kedalaman dermaga maka kapal tersebut tidak akan bisa berlabuh di *port* tersebut.

Production port dan *port* tujuan masing-masing memiliki *time windows* kedatangan. Artinya, kapal yang tiba di *port* diantara *time windows* kedatangan bisa memulai proses *loading* atau *unloading* tanpa melalui proses menunggu. Akan tetapi, jika kapal sampai pada *port* di luar *time windows* maka kapal akan menunggu sampai bisa melakukan proses *loading / unloading*. Jika kapal sampai sebelum *earliest time windows* maka kapal akan menunggu sampai *earliest time* operasional pelabuhan di mulai, sedangkan jika kapal sampai di pelabuhan melewati *latest time windows* maka kapal akan tidak bisa melakukan proses *loading / unloading* di periode t tersebut dan baru bisa dilakukan pada *earliest time windows* di periode $t + 1$. Permasalahan *Ship Scheduling and Assignment* di sini bersifat *single to many* yaitu terdiri dari 1 depot dan banyak *port* tujuan.

4.2 Pengembangan Algoritma

Algoritma yang dikembangkan dalam penelitian adalah *Simple Iterative Mutation Algorithm* (SIM-A) yang merupakan salah satu metoda yang tergolong ke dalam metoda metaheuristik. SIM-A merupakan metoda yang mengkombinasikan dua metode metaheuristik yaitu *Cross Entropy Algorithm* dan *Genetic Algorithm* tanpa melalui mekanisme *crossover*. SIM-A mengkombinasikan konsep mutasi dari *Genetic Algorithm* (GA) dan konsep *elite sample* dari *Cross Entropy Algorithm* (CE). SIM-A mempertimbangkan kelebihan dari algoritma GA dan CE, dimana GA bisa menyelesaikan permasalahan dengan skala yang besar (*population base*) sedangkan algoritma CE memastikan algoritma yang dirancang menghasilkan nilai optimal melalui mekanisme sampel elit (*elite sample*). Setelah dilakukan pengembangan algoritma SIM-A untuk menyelesaikan permasalahan *Ship Scheduling and Assignment*, kemudian dirancang kode pemrograman SIM-A dengan menggunakan *software* MATLAB R2010a.

Algoritma SIM-A yang dikembangkan digunakan untuk menyelesaikan permasalahan *Ship Scheduling and Assignment* dengan mempertimbangkan *Minimum requirement draft* kapal yang bisa berlabuh di suatu pelabuhan untuk mengirimkan semen curah ke beberapa pelabuhan tujuan (*packing plant*). Langkah – langkah dalam algoritma SIM - A dijelaskan pada gambar 4.1.



Gambar 4. 1 *Flowchart* Pengembangan Algoritma SIM – A

Berikut ini merupakan penjelasan dari tahapan-tahapan yang ada pada algoritma SIM – A berdasarkan *flowchart* yang ditunjukkan oleh gambar 4.1.

1. Pendefinisian *Input* dan *Output*

Pada bagian ini dijelaskan mengenai *input* apa saja yang diproses pada *Simple Iterative Mutation Algorithm* (SIM-A) untuk menyelesaikan permasalahan *Ship Scheduling and Assignment* pada kasus distribusi semen curah pada PT. X dan *output* apa saja yang akan didapatkan dari *running* algoritma SIM-A.

Input :

- a. Horison perencanaan dalam N periode (hari)
- b. Parameter biaya yang terdiri dari:
 - Biaya waktu tunggu sebelum melakukan *loading* (CWTSL)
 - Biaya *loading* di *production port* (LC)
 - Biaya perjalanan dari *production port* ke *port* tujuan i dengan menggunakan kapal v (DC)
 - Biaya perjalanan dari *port* tujuan i ke *production port* (TC)
 - Biaya akibat kapal menunggu sebelum melakukan *unloading* di *port* tujuan I (CWTSU)
 - Biaya akibat kapal menunggu di *port* tujuan karena kapasitas silo masih di atas *empty space standard* (CWTS)
 - Biaya *unloading* di *port* tujuan (ULC)
 - Biaya keterlambatan (*penalty cost*) per ton (PC)
 - *Penalty cost* akibat *On-Hand* melebihi kapasitas maksimum.
- c. Parameter kapal yang terdiri dari:
 - Jumlah kapal yang tersedia
 - *Minimum requirement draft* (MRD) kapal v
 - Kapasitas maksimum kapal v
 - Kecepatan *loading* dan *unloading* kapal
- d. Parameter pelabuhan
 - Jumlah pelabuhan yang harus dikunjungi.
 - Kecepatan *loading* dan *unloading* pelabuhan.
 - Kedalaman dermaga.

- *Inventory* awal di masing-masing *port* tujuan i (*inventory* awal di *silo packing plant*).
 - *Consumption rate* di masing-masing *port* tujuan i (CR).
 - Kapasitas maksimum *port* tujuan i (*silo packing plant*).
 - *Time windows* pelabuhan *port* tujuan i $[A_i, B_i]$
- e. Parameter pelayaran
- *Travelling time* dari *production port* ke *port* i .

Output :

- a. Total biaya selama horison perencanaan
- b. Informasi dan status kapal yang meliputi :
 - Pada periode t keberapa saja dan ke *port* tujuan yang mana saja kapal ditugaskan dalam bentuk matriks penjadwalan.
- c. Informasi mengenai *port* tujuan, yang meliputi:
 - *Inventory level (On-Hand)* di masing-masing *port* tujuan selama horison perencanaan.
 - *Intransit inventory* di masing-masing *port* tujuan selama horison perencanaan.
 - *Inventory position* di masing-masing *port* tujuan selama horison perencanaan.
 - *Scheduled receipt* di masing-masing *port* tujuan selama horison perencanaan.
 - *Consumption rate* di masing-masing *port* tujuan selama horison perencanaan.
 - *Re-shipment point* di masing-masing *port* tujuan selama horison perencanaan.
 - *Stock out* di masing-masing *port* tujuan selama horison perencanaan.
 - *Stock criticality* di masing-masing *port* tujuan selama horison perencanaan.

2. Penentuan Nilai Parameter Inisial

Nilai algoritma SIM-A yang dibutuhkan adalah jumlah sampel yang dibangkitkan dalam populasi (N), rasio sampel elit (ρ), lamanya horison perencanaan (n_{period}), fraksi *safety stock* (fss), dan besarnya nilai iterasi maksimum.

- Semakin banyak jumlah sampel yang dibangkitkan dalam populasi (N) akan didapatkan nilai solusi optimal dalam waktu komputasi yang relatif lebih singkat dikarenakan jumlah sampel yang dibangkitkan akan mengakibatkan kemungkinan solusi yang dihasilkan lebih banyak dalam satu kali iterasi sehingga waktu komputasi menjadi lebih singkat dibandingkan dengan jumlah sampel yang lebih kecil.
- Rasio sampel elit (ρ) digunakan untuk menentukan banyaknya sampel elit yang diambil dari populasi yang berukuran N .
- Lamanya horison perencanaan menentukan seberapa lama penjadwalan dan penugasan kapal dilakukan dalam satuan hari.
- Fraksi *safety stock* digunakan untuk menentukan besarnya jumlah *safety stock* yang diperlukan sebagai *stock* pengaman. Nilai fraksi *safety stock* sebesar f persen dari kapasitas maksimum silo di *port*.
- Nilai iterasi maksimum digunakan sebagai kriteria pemberhentian untuk menentukan seberapa lama proses pencarian solusi dilakukan. Iterasi akan berhenti jika nilai iterasi telah mencapai nilai iterasi maksimum. Setelah itu, akan didapatkan *output* dari hasil *running* algoritma SIM-A.

3. Inisialisasi Penjadwalan dan Penugasan Kapal

Solusi awal yang dibangkitkan merupakan solusi dari permasalahan penjadwalan dan penugasan kapal merupakan solusi yang *feasible* yang telah memenuhi konstrain – konstrain yang dipertimbangkan dalam permasalahan. Solusi yang dibangkitkan merupakan solusi yang menjelaskan mengenai penjadwalan dan penugasan kapal (*Ship Scheduling and Assignment*) selama horison perencanaan sesuai dengan urutan proses yang ada yang dijelaskan pada gambar 4.2 sampai 4.6.

Berikut penjelasan urutan proses untuk penjadwalan dan penugasan kapal, yaitu:

Langkah 1 : Penghitungan *Stock Criticality Port Tujuan*

Langkah pertama yang dilakukan untuk penjadwalan dan penugasan kapal (*ship scheduling and assignment*) adalah penghitungan *stock criticality* masing-masing *port* tujuan *i* untuk setiap periode *t*. Pengecekan *stock criticality* dilakukan dengan menggunakan persamaan 4.1.

$$\begin{aligned} & \textit{Stock Criticality Port} - i \\ & = \frac{(\textit{On Hand} + \textit{Intransit Inventory})}{(\textit{Consumption Rate} \times \textit{Lead Time})} \end{aligned} \quad (4.1)$$

Setelah dilakukan penghitungan *stock criticality* masing-masing *port* tujuan, langkah selanjutnya adalah mengurutkan *stock criticality* (SC) dari yang paling rendah sampai paling besar. *Port* dengan SC yang paling rendah maka *port* tersebut yang diutamakan untuk di-*supply* terlebih dahulu. Setiap periode *t*, maksimal ada 2 *port* kritis yang bisa di-*supply* karena terdapat batasan jumlah kapal yang bisa melakukan proses *loading* di depot (*production port*).

Kemudian juga dilakukan perhitungan *inventory position* di awal periode untuk masing-masing *port* tujuan. *Inventory position* (IP) di awal periode dihitung dengan menggunakan persamaan 4.2.

$$IP \textit{ awal periode} = IP \textit{ akhir periode}_{t-1} \quad (4.2)$$

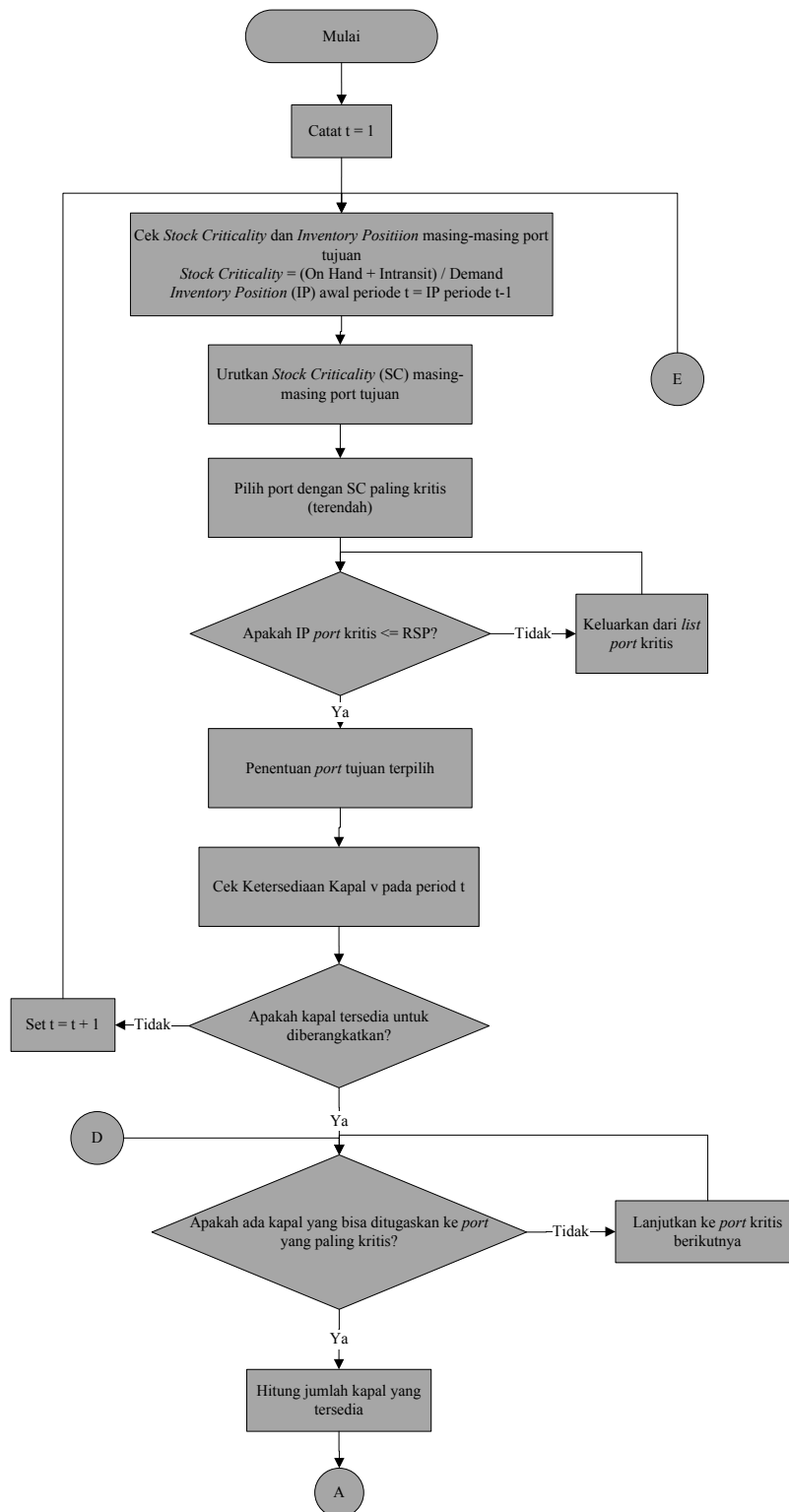
Setelah itu nilai IP untuk dua *port* dengan nilai *stock criticality* terendah dibandingkan dengan nilai *Re-shipment Point* (RSP) masing-masing *port*. Jika nilai $IP \leq RSP$ untuk kedua *port* tersebut, maka kedua *port* tersebut akan dipilih sebagai *port* yang akan dikunjungi oleh kapal untuk dikirim semen curah. Akan tetapi jika nilai $IP > RSP$ untuk kedua *port* maka pada periode tersebut tidak akan ada kapal yang ditugaskan ke *port* dengan nilai *stock criticality* terendah tersebut. Jika salah satu dari dua *port* tersebut memiliki nilai $IP \leq RSP$ maka *port* yang akan dikunjungi adalah *port* yang memiliki nilai *stock criticality* rendah (urutan 1 atau 2) dan yang memiliki nilai $IP \leq RSP$. Jadi intinya, *port* yang ditetapkan sebagai *port* tujuan adalah *port* yang memiliki SC kritis dan memenuhi syarat $IP \leq RSP$. Nilai RSP dihitung dengan menggunakan persamaan 4.3.

$$RSP = Demand \times Lead Time + Safety Stock \quad (4.3)$$

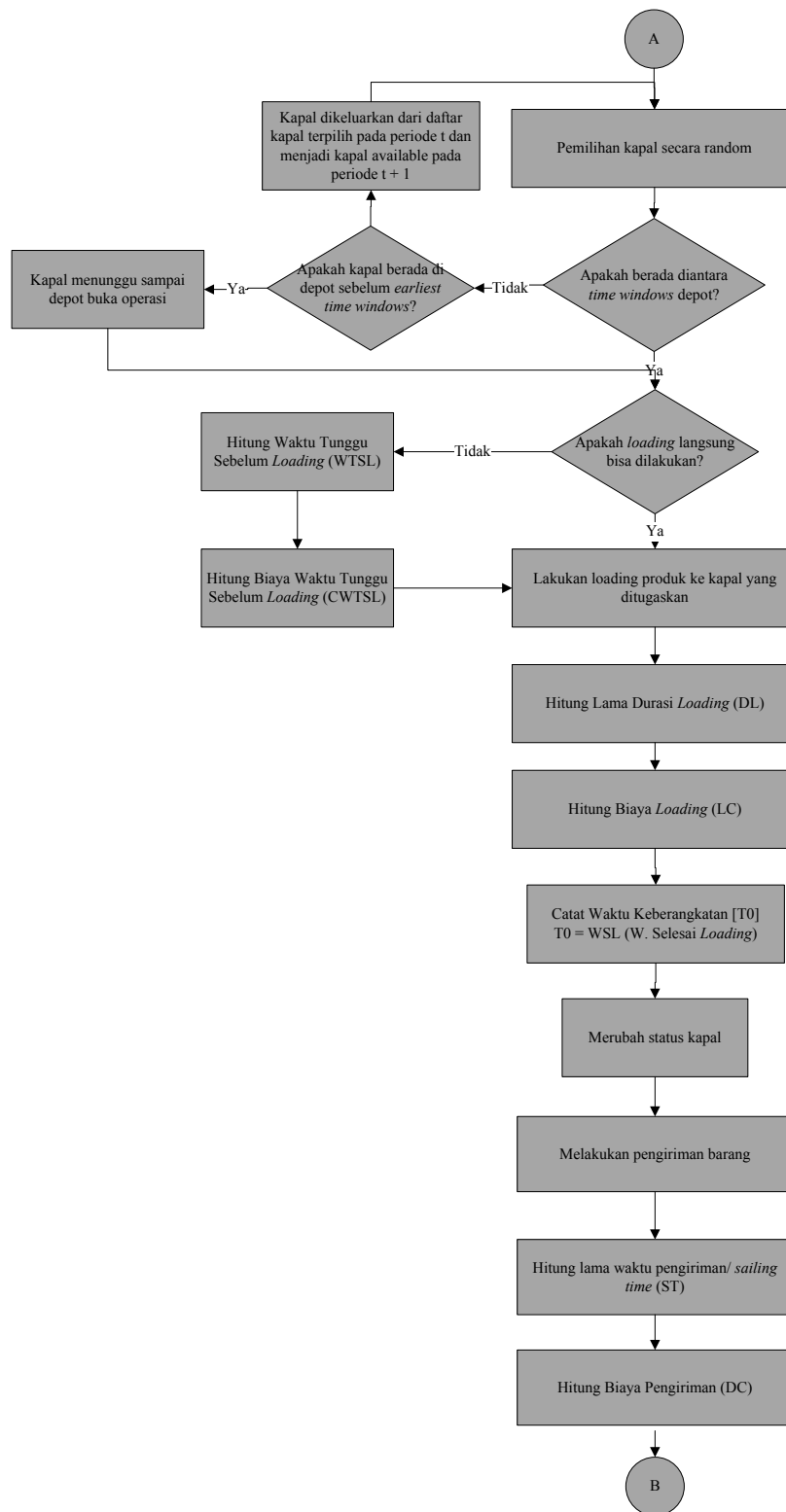
Dimana nilai *safety stock* didapatkan dengan menggunakan persamaan 4.4 berikut ini.

$$Safety Stock = fss \times Kapasitas \text{ maksimum silo } port \text{ tujuan} \quad (4.4)$$

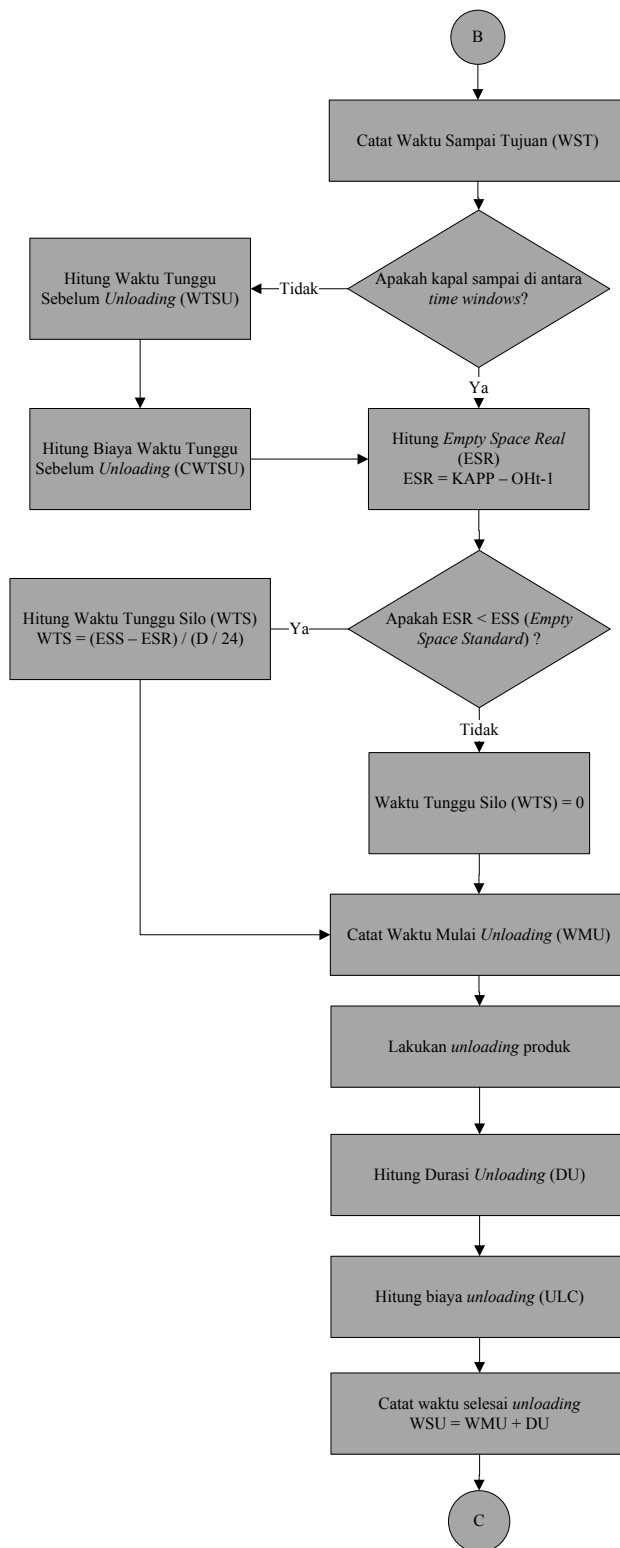
Nilai RSP merepresentasikan titik kritis persediaan dimana pengiriman semen curah ke *port* tersebut harus dilakukan. Secara konsep RSP sama dengan *Re-order Point* (titik dimana pemesanan harus dilakukan).



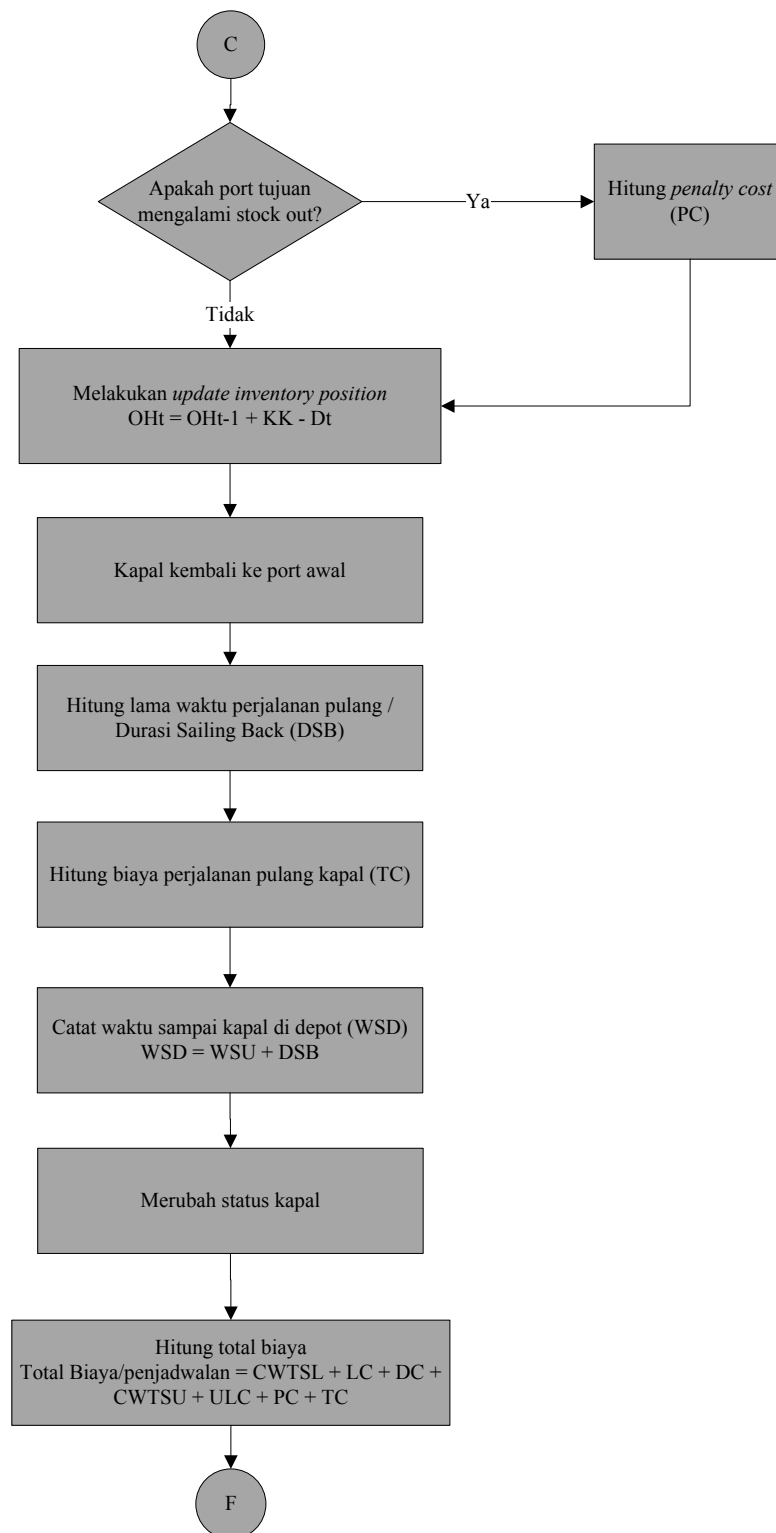
Gambar 4. 2 Flowchart Penjadwalan dan Penugasan Kapal (1)



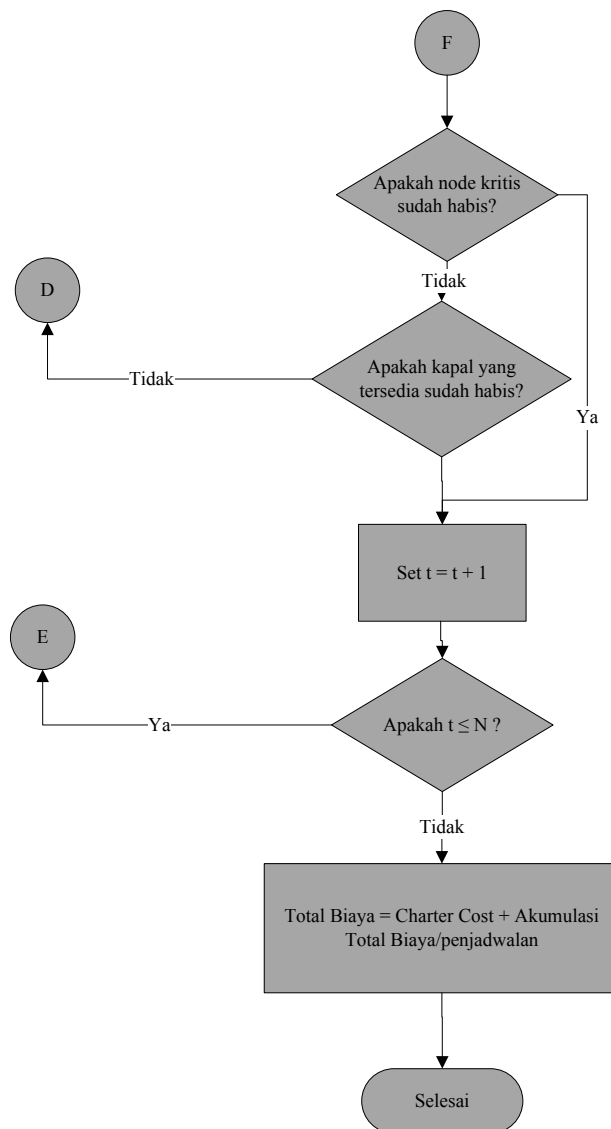
Gambar 4. 3 *Flowchart* Penjadwalan dan Penugasan Kapal (2)



Gambar 4. 4 *Flowchart* Penjadwalan dan Penugasan Kapal (3)



Gambar 4. 5 Flowchart Penjadwalan dan Penugasan Kapal (4)



Gambar 4. 6 *Flowchart* Penjadwalan dan Penugasan Kapal (5)

Langkah 2 : Pengecekan Ketersediaan dan Pemilihan Kapal

Pada langkah ini dilakukan proses pengecekan ketersediaan kapal untuk mengetahui jumlah kapal dan kapal yang mana saja tersedia untuk ditugaskan mengirimkan semen ke *port* tujuan. Setelah diketahui kapal mana saja yang tersedia untuk ditugaskan, langkah selanjutnya dilakukan pengecekan jumlah dan kapal mana yang *feasible* ditugaskan untuk mengirimkan semen ke *port* tujuan dengan SC paling kritis dan *port* yang memenuhi konstrain $IP \leq RSP$ yang didasarkan pada *minimum requirement draft* (MRD) kapal dan kedalaman *port* tujuan. Jika MRD kapal kurang dari atau sama dengan kedalaman *port* ($MRD \leq h$)

dan kapasitas kapal \leq kapasitas silo *port* tujuan maka kapal tersebut bisa (*feasible*) ditugaskan ke *port* tersebut. Akan tetapi, jika $MRD > h$ atau kapasitas kapal $>$ kapasitas *port* tujuan maka kapal tidak bisa ditugaskan ke *port* tersebut. Setelah diketahui banyak kapal yang tersedia dan *feasible* ditugaskan ke *port* paling kritis, maka akan dilakukan pemilihan kapal secara *random* untuk diberangkatkan ke *port* kritis. Setiap periode t , depot (*production port*) hanya mampu melakukan proses *loading* maksimal untuk dua kapal sehingga pada masing-masing periode hanya bisa dijadwalkan maksimal 2 kapal. Untuk *supply* ke *port* paling kritis (SC terendah pertama) pada periode t , misalkan pada periode t terdapat 5 kapal yang *available* untuk diberangkatkan ke *port* kritis. Kemudian dari 5 kapal tersebut dilakukan pengecekan jumlah kapal dan kapal yang mana saja yang *feasible* diberangkatkan ke *port* tujuan tersebut yang didasarkan pada *minimum requirement draft* (MRD) kapal dan kedalaman dermaga (h) dan atau kapasitas kapal \leq kapasitas silo *port* tujuan. Misalkan dari 5 kapal yang *available*, yang *feasible* untuk diberangkatkan hanya 4 kapal. Dari 4 kapal yang *feasible* kemudian dilakukan pemilihan kapal secara *random* untuk menentukan kapal mana yang ditugaskan ke *port* paling kritis. Dari 4 kapal yang tersedia dilakukan penentuan probabilitas masing-masing kapal untuk terpilih. Berikut ditunjukkan penentuan probabilitas masing-masing kapal untuk terpilih.

Tabel 4. 1 Skema Pemilihan Kapal untuk Diberangkatkan

Kapal	A	B	C	D
Probabilitas Terpilih	0.25	0.25	0.25	0.25
Probabilitas (Akumulasi)	0.25	0.50	0.75	1

Penentuan probabilitas masing-masing kapal untuk terpilih ditentukan dengan menggunakan rumus berikut

$$Probabilitas = \frac{1}{Jumlah\ Kapal\ feasible} \quad (4.5)$$

Sedangkan probabilitas akumulasi ditentukan dengan mengakumulasi atau menjumlahkan probabilitas masing-masing kapal untuk terpilih. Artinya, kapal A akan dipilih jika bilangan *random* (r) yang dibangkitkan ≤ 0.25 , kapal B terpilih jika $0.25 < r \leq 0.50$, kapal C terpilih jika $0.50 < r \leq 0.75$, dan kapal D terpilih jika $0.75 < r \leq 1$. Misalkan $r = 0.157$ maka kapal yang akan terpilih untuk ditugaskan ke *port* tujuan yang paling kritis adalah kapal A.

Pemilihan kapal kedua untuk dijadwalkan ke *port* tujuan dengan SC terendah kedua, melalui proses yang sama seperti pemilihan kapal pertama untuk *port* tujuan paling kritis. Hanya saja, dari jumlah kapal yang *available* lebih sedikit dibandingkan jumlah kapal yang *available* pada penjadwalan untuk kapal pertama. Ketika kapal pertama sudah dipilih untuk dijadwalkan ke *port* kritis, maka pada saat itu dilakukan perubahan status kapal yang terpilih dari *available* menjadi tidak tersedia dan pada saat yang sama dilakukan *update* kapal yang *available*. Jika pada penjadwalan kapal pertama tersedia 5 kapal yang *available*, maka pada penjadwalan kapal kedua terdapat 4 kapal yang tersedia karena 1 kapal sudah ditugaskan sebelumnya. Dari 4 kapal yang *available* pada penjadwalan kapal kedua, dilakukan pengecekan dan penghitungan jumlah kapal yang *feasible* untuk diberangkatkan ke *port* tujuan kritis kedua. Setelah itu dilakukan pemilihan kapal secara *random* untuk menentukan kapal mana yang akan ditugaskan ke *port* kritis kedua.

Langkah 3 : Pengecekan *Time Windows Depot*

Setelah dilakukan pemilihan kapal, dilakukan pengecekan mengenai waktu kedatangan kapal di depot (*production port*) apakah berada di *time windows* kedatangan atau tidak. Jika berada di antara *time windows* kedatangan maka kapal dapat melakukan proses *loading* secara langsung tanpa harus menunggu. Akan tetapi jika kapal datang di depot di luar *time windows* kedatangan maka kapal akan menunggu sampai proses *loading* bisa dilakukan. Jika kapal tiba di depot kurang dari *earliest time windows* maka kapal akan menunggu beberapa jam sampai depot dibuka untuk melakukan proses *loading*, sedangkan jika kapal sampai di depot melebihi dari *latest times windows* maka

kapal tidak bisa melakukan proses *loading* pada saat periode t tersebut dan *loading* baru bisa dilakukan pada saat $t + 1$ ketika *earliest time windows* di depot. Waktu tunggu sampai proses *loading* bisa dilakukan akan dicatat dan disimpan.

Langkah 4 : Pengecekan Kondisi Depot

Pada tahap ini dilakukan pengecekan kondisi *port* (depot) apakah di *port* melayani kapal lain atau tidak. Jika *port* sedang melayani kapal lain, maka kapal yang ditugaskan akan menunggu dengan lama waktu tertentu sebelum melakukan proses *loading*. Besarnya waktu tunggu sebelum *loading* (WTSL) didapatkan dengan cara melakukan *fitting* distribusi lama waktu tunggu berdasarkan data historis yang ada. Jika *port* sedang dalam keadaan kosong atau tidak sedang melayani kapal lain, maka kapal akan bisa melakukan proses *unloading* tanpa harus menunggu. Jika kapal menunggu sebelum melakukan *loading* maka akan dikenakan biaya sesuai dengan *rate* yang ditetapkan per satuan waktu.

Langkah 5 : Loading Produk ke Kapal Terpilih

Proses *loading* produk ke kapal terpilih dilakukan sampai kapasitas maksimal kapal tercapai. Lama waktu proses *loading* didapatkan dengan menggunakan rumus sebagai berikut.

$$\text{Waktu Loading} = \frac{\text{Kapasitas Kapal } v}{\text{Kecepatan Loading Kapal } v} \quad (4.6)$$

Setelah proses *loading* selesai, dilakukan penghitungan biaya yang dibutuhkan untuk melakukan *loading*, dan pada saat yang bersamaan dilakukan pencatatan dan penyimpanan waktu keberangkatan kapal. Kemudian dilanjutkan dengan pengubahan status kapal dari *available* menjadi tidak tersedia.

Langkah 6 : Melakukan Pengiriman Barang

Setelah itu dilakukan proses pengiriman barang dengan lama waktu pengiriman bersifat deterministik selama *sailing time* (ST) yang telah ditentukan

pada *input*. Setelah kapal sampai pada *port* tujuan kritis, akan dilakukan pencatatan dan penyimpanan waktu sampai kapal pada *port* tujuan serta dilakukan perhitungan biaya pengiriman barang dari depot sampai *port* tujuan.

Langkah 7 : Pengecekan *Time Windows Port Tujuan*

Ketika kapal sampai pada *port* tujuan, akan dilakukan pengecekan apakah kapal sampai pada *port* tujuan berada diantara *time windows* kedatangan atau tidak. Jika kapal sampai di *port* tujuan berada dalam *time windows* kedatangan maka kapal bisa melakukan proses *unloading* secara langsung tanpa harus menunggu. Akan tetapi jika kapal sampai di *port* tujuan berada di luar *time windows* maka proses *unloading* tidak bisa dilakukan secara langsung dan kapal harus menunggu dengan lama waktu tertentu sampai proses *unloading* bisa dilakukan. Jika kapal sampai di *port* tujuan sebelum *earliest time windows* maka kapal akan menunggu sampai waktu operasional di *port* tujuan dibuka. Sedangkan jika kapal sampai pada *port* tujuan melewati *latest time windows*, maka proses *unloading* tidak bisa dilakukan pada periode t tersebut dan proses *unloading* baru bisa dilakukan pada periode $t + 1$ ketika *earliest time windows port* dimulai. Misalnya *time windows* di suatu *port* kritis adalah [08:00 – 18:00], jika kapal sampai pada *port* tersebut periode 4 jam 04:00 maka proses *unloading* kapal baru bisa dilaksanakan pada periode tersebut jam 08:00. Artinya, kapal harus menunggu selama 4 jam. Jika kapal sampai pada *port* tersebut periode 4 jam 19:00 maka proses *unloading* pada periode tersebut tidak bisa dilakukan dan baru bisa dilakukan pada periode 5 jam 08:00 ketika *earliest time windows port* dimulai. Artinya, kapal akan menunggu selama 11 jam sampai proses *unloading* bisa dilakukan.

Langkah 8 : Pengecekan Kondisi Depot

Pada tahap ini dilakukan pengecekan kondisi *port* tujuan apakah di *port* melayani kapal lain atau tidak. Jika *port* sedang melayani kapal lain, maka kapal yang ditugaskan akan menunggu dengan lama waktu tertentu sebelum melakukan proses *unloading*. Besarnya waktu tunggu sebelum *unloading* (WTSU) didapatkan dengan cara melakukan *fitting* distribusi lama waktu tunggu

berdasarkan data historis yang ada. Jika *port* sedang dalam keadaan kosong atau tidak sedang melayani kapal lain, maka kapal akan bisa melakukan proses *unloading* tanpa harus menunggu. Jika kapal menunggu untuk melakukan *unloading* maka kapal akan dikenakan biaya tunggu sesuai dengan *rate* yang ditetapkan per satuan jam.

Langkah 9 : Pengecekan *Empty Space Real* (ESR)

Pada tahap ini dilakukan pengecekan *Empty Space Real* (ESR). ESR menyatakan seberapa besar dari kapasitas silo yang tidak terisi produk (semen curah). ESR dihitung dengan menggunakan persamaan 4.7.

$$ESR = \text{Kapasitas port tujuan} - OH_{t-1} \quad (4.7)$$

Dalam persamaan 4.7, seharusnya besarnya *On-Hand* yang digunakan adalah *On-Hand* (OH) pada jam yang bersangkutan di periode t. Akan tetapi karena *On-Hand* yang direkam dalam algoritma SIM-A dalam satuan hari (periode) bukan jam dan OH di-*update* di akhir periode, sehingga yang digunakan adalah *On-Hand* harian pada periode t-1 karena *On-Hand* pada akhir periode t-1 nilainya hampir sama dengan *On-Hand* pada awal periode t (dalam hal ini diasumsikan sama).

Langkah 10 : Perhitungan Waktu Tunggu Silo (WTS)

Jika $ESR < \text{Empty Space Standard (ESS)}$ maka kapal akan menunggu sebelum bisa melakukan *unloading*. *Unloading* bisa dilakukan jika nilai $ESR \geq ESS$. Nilai ESS menyatakan nilai standar kapasitas *port* yang kosong untuk bisa melakukan proses *unloading*. Besarnya *Empty Space Standard* (ESS) didapatkan dengan menggunakan persamaan 4.8 berikut ini.

$$ESS = \left(\frac{(UR_{kapal} - \frac{\text{Demand harian}}{24} \times \text{Kapasitas Kapal})}{UR_{kapal}} \right) \quad (4.8)$$

Jika kapal menunggu akibat $ESR < ESS$, maka lama waktu tunggu silo dihitung dengan menggunakan persamaan 4.9 di bawah ini.

$$WTS = \frac{(ESS - ESR)}{(Demand\ harian / 24)} \quad (4.9)$$

Jika kapal menunggu sampai ESR silo mencapai ESS , maka kapal akan dikenai biaya waktu tunggu silo sesuai dengan *rate* yang ditentukan.

Langkah 11 : Unloading Produk

Proses *unloading* produk di *port* tujuan akan dilakukan dengan lama waktu *unloading* yang didapatkan dari rumus 4.10 berikut ini.

$$Waktu\ Unloading = \frac{Kapasitas\ Kapal\ v}{Kecepatan\ Unloading\ Kapal\ v} \quad (4.10)$$

Setelah proses *unloading* selesai, dilakukan perhitungan biaya *unloading* dan pencatatan dan penyimpanan waktu selesai *unloading*. Waktu selesai *unloading* didapatkan dengan cara menjumlahkan antara waktu sampai kapal di *port*, waktu tunggu sebelum *unloading*, waktu tunggu silo, dan lama waktu proses *unloading*.

Langkah 12 : Pengecekan Keterlambatan

Setelah proses *unloading* selesai, dilakukan tahap pengecekan apakah terjadi keterlambatan atau tidak. Terjadinya keterlambatan dicek dengan cara melihat apakah terjadi *stock out* pada *port* di periode tersebut. Jika terjadi *stock out* di *port* tersebut maka akan dikenakan *penalty cost* dengan jumlah yang sangat besar per *ton* kekurangan. Dalam permasalahan ini, setiap kekurangan 1 ton akan dikenakan *penalty* sebesar Rp. 10.000.000,- per ton. *Penalty cost* akan berpengaruh terhadap nilai *objective function*. *Penalty cost* untuk kelebihan *On-Hand* sebesar Rp. 5.000. 000 per ton. Keterlambatan di sini berfungsi sebagai *soft constraint*, artinya keterlambatan diijinkan terjadi tapi hal tersebut harus dihindari.

Besarnya *penalty cost* per ton dengan nominal yang sangat besar sengaja di-*set* untuk menunjukkan perbedaan yang signifikan antara *objective function* pada sampel satu dengan sampel yang lainnya sehingga dapat dilakukan pemilihan solusi terbaik dari N solusi yang dibangkitkan yang dilihat dari total biaya yang dikeluarkan selama horison perencanaan yang ditetapkan.

Langkah 13 : Update Inventory Position (On-Hand)

Setelah *unloading* selesai dilakukan, langkah selanjutnya adalah dilakukan *update inventory position (On-Hand)*. *On-Hand* setiap periode dihitung dengan menggunakan rumus berikut 4.11

$$OH_t = OH_{t-1} + SR_t - CR_t \quad (4.5)$$

Langkah 14 : Perjalanan ke Depot

Setelah kapal selesai melakukan *unloading* (melayani *port* tujuan) maka kapal akan melakukan perjalanan untuk kembali ke depot. Waktu perjalanan yang dibutuhkan dari *port unloading (port* tujuan) bersifat deterministik yang besarnya sama dengan *sailing time* yang telah ditentukan sebagai *input*. *Rate* biaya perjalanan ke depot (*travelling cost*) sama dengan *rate* biaya pengiriman semen dari depot menuju *port* tujuan (*distribution cost*).

Langkah 15 : Pengecekan *Port* Kritis dan Jumlah Kapal

Pada periode *t* yang sama, ketika kapal 1 sudah ditugaskan untuk mengirimkan produk ke *port* paling kritis pada saat itu dilakukan pengecekan apakah *port* kritis sudah habis atau tidak (dalam 1 periode hanya bisa melayani 2 *port* kritis). Jika *port* kritis kedua belum dilayani maka akan dilakukan pengecekan kapal yang *available* yang bisa ditugaskan ke *port* kritis tersebut. Jika ada kapal yang tersedia maka akan dilakukan proses pemilihan kapal seperti pada pemilihan kapal pertama untuk *port* kritis sebelumnya. Setelah itu dilakukan proses yang sama seperti kapal pertama. Jika *port* kritis kedua belum dilayani tetapi tidak ada satupun kapal yang tersedia dan atau *port* kritis kedua sudah

selesai dilayani maka akan di *set* $t = t + 1$. Periode akan berhenti jika periode penjadwalan sudah mencapai lama horison perencanaan yang ditetapkan.

4. Penghitungan Nilai Fungsi Objektif

Setelah horison perencanaan telah selesai, maka dilakukan perhitungan nilai fungsi objektif yang dilihat dari total biaya yang dikeluarkan selama horison perencanaan yang ditetapkan. Total biaya selama horison perencanaan didapatkan dari akumulasi biaya yang dikeluarkan untuk masing-masing periode penjadwalan ditambah dengan biaya sewa kapal selama horison perencanaan. Setelah nilai fungsi objektif diurutkan, maka *set* iterasi = 1.

5. Pemilihan Sampel *Elite*

Pada proses ini dilakukan pemilihan sampel *elit* sebanyak $(p \times N)$ individu. Sampel *elit* yang dipilih merupakan sampel yang memberikan nilai fungsi objektif paling minimum. Sebelum dilakukan pemilihan sampel *elit*, terlebih dahulu dilakukan pengurutan sampel mulai dari sampel yang memberikan nilai sistem (fungsi objektif) yang paling minimum sampai terbesar.

6. Elitisme

Elitisme dilakukan untuk mempertahankan individu terbaik yang menghasilkan nilai fungsi objektif paling minimum agar tetap muncul pada iterasi berikutnya. Jumlah sampel yang akan dielitisme setiap iterasi sebanyak jumlah sampel elite yang terpilih, dimana sampel tersebut tidak akan mengalami proses mutasi dan akan tetap muncul pada iterasi selanjutnya.

7. Mutasi

Pada tahap ini dilakukan mutasi untuk membangkitkan solusi baru. Mutasi dilakukan dengan cara pembangkitan secara *random* yang telah memenuhi konstrain – konstrain yang dipertimbangkan dalam algoritma SIM-A. Mutasi dilakukan sebanyak $N - JSE$ (jumlah sampel elit).

8. Pembentukan Populasi Baru

Pada tahap ini dilakukan penggabungan hasil elitisme dengan hasil mutasi untuk membentuk solusi (populasi) baru. Populasi yang baru ini akan dikenai proses yang sama mulai dari elitism sampai mutasi.

9. Pembaharuan Kriteria Pemberhentian

Setelah dilakukan pembentukan populasi baru, dilakukan pembaharuan kriteria pemberhentian dengan cara $\text{set iterasi} = \text{iterasi} + 1$. Iterasi akan berhenti ketika iterasi telah mencapai jumlah iterasi maksimum.

4.3 Verifikasi

Pada tahap ini dilakukan verifikasi untuk mengecek apakah logika operasional yang dibuat dalam MATLAB R2010a sudah sesuai dengan logika pada *flowchart* pengembangan algoritma SIM-A. Di samping itu juga dilakukan pengecekan apakah tidak terjadi *error* pada koding MATLAB yang telah dibuat. Dari kode pemrograman SIM-A untuk *Ship Scheduling and Assignment* yang dirancang pada *software* MATLAB R2010a tidak ditemukan adanya *error*. Setelah tidak ditemukan adanya *error* pada kode pemrograman, untuk memastikan bahwa algoritma yang dirancang *verified* maka dilakukan *running* kode untuk mengetahui apakah logika operasional yang dibuat sudah sesuai dengan *flowchart* SIM-A yang dirancang atau tidak. Pengecekan logika operasional dilakukan dengan cara *me-running* kode yang telah dibuat untuk menyelesaikan masalah sederhana. Hasil *running* MATLAB ditunjukkan oleh gambar 4.7 – 4.9 di bawah ini. Berdasarkan hasil *running* MATLAB sesuai yang ditunjukkan oleh gambar 4.7 – 4.9 dapat diketahui bahwa kode yang dibuat *verified*. Dikatakan *verified* karena *output* yang dihasilkan dari hasil *running* telah sesuai dengan semua konstrain dan persamaan yang dipertimbangkan dalam algoritma. Misalnya *output* untuk *scheduled* yang ditunjukkan pada gambar 4.7. Berdasarkan *output scheduled* dapat diketahui bahwa pada setiap periode dalam horison perencanaan yang ditetapkan terdapat maksimal 2 kapal yang dijadwalkan ke 2 *port* yang kritis. Artinya, konstrain bahwa maksimal penugasan kapal setiap periode telah dipenuhi. Pada periode 1 dapat diketahui bahwa kapal 5 ditugaskan

ke *port* 1 dan kapal 4 ke *port* 5. Dimana kedua kapal yang ditugaskan tersebut telah memenuhi dua konstrain yang ditetapkan yaitu $MRD \leq$ kedalaman dermaga *port* tujuan dan kapasitas kapal \leq kapasitas silo *port* tujuan. Di samping itu juga *port* tujuan 1 dan 5 merupakan *port* dengan tingkat *stock criticality* terendah dan memenuhi konstrain tingkat *inventory position* \leq RSP. Kemudian pada periode 2, dapat diketahui bahwa kapal yang ditugaskan adalah kapal 1 ke *port* 2 dan kapal 3 ke *port* 4. Hal itu menunjukkan bahwa algoritma yang dirancang sudah *verified* karena pada periode 2, kapal 4 dan kapal 5 statusnya tidak *available* dan tidak mungkin dijadwalkan lagi pada periode 2 karena kedua kapal tersebut belum sampai pada periode 2. Kapal yang *available* pada periode 2 adalah kapal yang belum dijadwalkan pada periode sebelumnya (periode 1) yaitu kapal 1, 2 dan 3. Hal ini juga diperkuat dengan *port* yang dilayani pada periode 2 yaitu *port* 2 dan 4 yang merupakan *port* kritis pada periode 2. Selain itu, informasi masing-masing *port* yang meliputi *On-Hand* (OH), *Intransit Inventory* (IT), *Inventory Position* (IP), *Scheduled Receipt* (SR), *Consumption Rate* (CR), *Re-shipment Point* (RSP), *Stock Out* (SO), dan *Stock Criticality* (SC) di setiap periode dalam horison perencanaan yang ditentukan telah memenuhi logika operasional. Begitu juga untuk total biaya sistem telah memenuhi logika operasional. Oleh karena itu, dapat dikatakan bahwa model telah *verified*.

```

>> [Best_Cost, Best_Scheduled, Best_Solution, it_ttp, Waktu_Komputasi] = SIM_4(5, 5, 5, 30, 0.2, 0.05, 20)
OPTIMASI SHIP SCHEDULING AND ASSIGNMENT MENGGUNAKAN SIMPLE ITERATIVE MUTATION ALGORITHM

Best_Cost =

    12158865.5605905

Best_Scheduled =

     0     0     0     0     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     1     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0

Best_Solution =

      1000           0           0           0           0           1
     11000         11000         11000         11000         11000         1
     12000         11000         11000         11000         11000         1
           0           0           0           0           0           1
        1650         1650         1650         1650         1650         1
        2250         2250         2250         2250         2250         1
           0           650         1650         1650         1650         1
0.606060606060606  6.66666666666667  6.66666666666667  6.66666666666667  6.66666666666667  1

```

Gambar 4. 7 Hasil *Running Software* MATLAB untuk Kasus Sederhana (1)

```

1500      0      0      0      0      0
0      0      10000      10000      10000      0
1500      0      10000      10000      10000      9500
0      0      0      0      0      10000
1750      1750      1750      1750      1750      1750
3000      3000      3000      3000      3000      3000
0      150      1750      1750      0      0
0.1141289714285714      4.761509751380476      4.761509751380476      5.928571428571428
1000      600      0      0      0      0
0      0      0      0      5000      5000
-1000      600      0      0      5000      5000
0      0      0      0      0      0
1400      1400      1400      1400      1400      1400
1670      1670      1670      1670      1670      1670
0      0      800      1400      1400      1400
1.0989010989011      0.302670919670311      4.94505994305495      4.94505994305495
800      0      0      0      0      0
0      0      12000      12000      12000      12000
900      0      13000      13000      13000      13000
0      0      0      0      0      0
800      800      800      800      800      800
-1100      -1100      -1100      -1100      -1100      -1100
0      0      800      800      800      800
0.0000000000000007      0      10.83333333333333      10.83333333333333      10.83333333333333

```

Gambar 4. 8 Hasil *Running Software* MATLAB untuk Kasus Sederhana (2)

```

1000      50      0      0      0      0
12000      12000      12000      12000      12000      0
11000      12050      12000      12000      12000      12000
0      0      0      0      0      0
950      950      950      950      950      950
2850      2850      2850      2850      2850      2850
0      0      900      950      950      950
0.324315789473684      4.94210324315789      4.11578947368421      4.11578947368421      4.11578947368421

```

tic *tic* =
 10

Meaktu *Komputasi* =
 1.09200659089907

Gambar 4. 9 Hasil *Running Software* MATLAB untuk Kasus Sederhana (3)

4.4 Validasi Algoritma SIM-A

Pada bagian ini akan dilakukan tahap validasi algoritma SIM-A yang dikembangkan dengan menggunakan permasalahan sederhana untuk mendapatkan hasil penjadwalan dan penugasan kapal (*Ship Scheduling and Assignment*) yang memberikan nilai total biaya paling minimum.

4.4.1 Penyelesaian Dengan Enumerasi

Pada bagian ini ditunjukkan hasil enumerasi untuk menyelesaikan permasalahan *Ship Scheduling and Assignment* dengan menggunakan kasus sederhana. Enumerasi dilakukan dengan mengikuti urutan proses yang ditunjukkan pada gambar 4.1. Kasus sederhana ini digunakan untuk membuktikan bahwa perhitungan yang digunakan telah benar. Berikut ini merupakan informasi-informasi yang dijadikan sebagai *input* dalam melakukan enumerasi

Planning horizon = 5 hari

Parameter Kapal

Berikut ini merupakan parameter pelabuhan (*port*) yang digunakan sebagai *input* dalam melakukan enumerasi.

Tabel 4. 2 Parameter Pelabuhan

Port	OH Awal (ton)	Demand (ton)	Time Windows (TW)		Kedalaman (m)	Kapasitas (ton)
			<i>Earliest Time</i> [Ai]	<i>Latest Time</i> [Bi]		
Depot	-	-	8	18	-	-
1	1000	650	8	18	8	12000
2	1500	750	8	18	8	18000
3	2000	1100	8	18	9	17000
4	800	800	8	18	6	18000
5	1000	950	8	18	8	19000

Parameter Kapal

Berikut ini merupakan parameter kapal yang digunakan dalam melakukan enumerasi.

Tabel 4. 3 Parameter Kapal (1)

Kapal	Kapasitas (ton)	MRD (m)	<i>Loading Rate</i> (ton/jam)	<i>Unloading Rate</i> (ton/jam)	Rate Sewa Kapal (Rp/hari)	LC (Rp/jam)
1	10000	6	500	500	90	14
2	9000	6	400	300	110	22
3	13000	5	300	300	120	13
4	12000	6	200	400	130	34
5	11000	5	400	200	140	25

Tabel 4. 4 Parameter Kapal (2)

Kapal	ULC (Rp/jam)	RDC (Rp/jam)	RWTSL (Rp/jam)	RWTSU (Rp/jam)	RWTS (Rp/jam)	PC (Rp/ton)
1	2	14	1	5	8	1000
2	5	22	2	7	9	1000
3	8	13	3	9	11	1000
4	10	34	4	11	15	1000
5	12	25	5	13	20	1000

Parameter Pelayaran

Berikut ini merupakan parameter pelayaran yang digunakan dalam melakukan enumerasi.

Tabel 4. 5 Parameter Pelayaran

Depot	Parameter	Port 1	Port 2	Port 3	Port 4	Port 5
	LT (hari)	1	1.2	1.3	1.5	2
	ST (jam)	6	12	14	16	15

Keterangan :

MRD = *Minimum requirement draft*

LC = *Loading Cost*

ULC = *Unloading Cost*

RDC = *Rate Distribution Cost*

RWTSL = *Rate Waktu Tunggu Sebelum Loading*

RWTSU = *Rate Waktu Tunggu Sebelum Unloading*

RWTS = *Rate Waktu Tunggu Sebelum Silo*

PC = *Penalty Cost*

LT = *Lead Time*

Dari data parameter di atas, dapat diketahui bahwa kapal 3 tidak bisa singgah ke *port* 1 karena kapasitas kapal 3 > kapasitas *port* 1 meskipun konstrain kedalaman dermaga dipenuhi ($MRD \leq$ kedalaman dermaga). Untuk ke *port* yang lain, kapal 3 bisa singgah ke *port* tersebut karena telah memenuhi dua konstrain kedalaman yang ditetapkan. Selain kapal 3, semua kapal bisa ditugaskan/singgah ke semua *port*. Berikut ini diberikan fleksibilitas semua kapal ke masing-masing *port* berdasarkan konstrain yang ditetapkan yang ditunjukkan oleh gambar 4.6.

Tabel 4. 6 Fleksibilitas Kapal

	<i>Port</i> 1	<i>Port</i> 2	<i>Port</i> 3	<i>Port</i> 4	<i>Port</i> 5
Kapal 1	1	1	1	1	1
Kapal 2	1	1	1	1	1
Kapal 3	0	1	1	1	1
Kapal 4	1	1	1	1	1
Kapal 5	1	1	1	1	1

Dari table 4.6 dapat diketahui masing-masing kapal bisa ditugaskan ke *port* yang mana saja. Angka 1 menyatakan bahwa kapal bisa ditugaskan/singgah ke *port* dan angka 0 menyatakan bahwa kapal tidak bisa singgah ke *port* karena tidak memenuhi kedua konstrain yang ditetapkan.

Dengan mengikuti alur proses yang ditunjukkan pada gambar 4.1 didapatkan hasil optimal sebagai berikut. Tabel 4.7 di bawah ini merupakan penjadwalan dan penugasan kapal yang didapatkan melalui hasil enumerasi.

Tabel 4. 7 *Scheduled* (Penjadwalan dan Penugasan Kapal)

Port	Periode																													
	1					2					3					4					5									
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5					
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Berdasarkan table 4.7 dapat diketahui bahwa pada periode 1 kapal 5 ditugaskan ke *port* 1 dan kapal 4 ditugaskan ke *port* 5. Pada periode 2, kapal 1 ditugaskan ke *port* 2 dan kapal 3 ditugaskan ke *port* 4. Pada periode 3 hanya ada 1 kapal yang ditugaskan yaitu kapal 3 ke *port* 3 karena pada periode tersebut hanya ada 1 kapal yang *available* yaitu kapal 3. Periode 4 dan periode 5 tidak ada kapal yang ditugaskan karena pada periode tersebut tidak ada satupun kapal yang *available*.

Berikut ini ditunjukkan informasi mengenai masing – masing *port* berdasarkan penjadwalan yang didapatkan dari hasil enumerasi.

Tabel 4. 8 Informasi untuk *Port 1*

<i>Port 1</i>	Periode				
	1	2	3	4	5
OH	1000	0	0	0	0
IT	11000	11000	11000	11000	11000
IP	12000	11000	11000	11000	11000
SR	0	0	0	0	0
CR	1650	1650	1650	1650	1650
RSP	2250	2250	2250	2250	2250
SO	0	650	1650	1650	1650
SC	0.60606	6.66667	6.66667	6.66667	6.66667

Tabel 4. 9 Informasi untuk *Port 2*

<i>Port 2</i>	Periode				
	1	2	3	4	5
OH	1500	0	0	0	8250
IT	0	0	10000	10000	0
IP	1500	0	10000	10000	8250
SR	0	0	0	0	10000
CR	1750	1750	1750	1750	1750
RSP	3000	3000	3000	3000	3000
SO	0	250	1750	1750	0
SC	0.71429	0	4.7619	4.7619	3.92857

Tabel 4. 10 Informasi untuk *Port 3*

<i>Port 3</i>	Periode				
	1	2	3	4	5
OH	2000	600	0	0	0
IT	0	0	0	9000	9000
IP	2000	600	0	9000	9000
SR	0	0	0	0	0
CR	1400	1400	1400	1400	1400
RSP	2670	2670	2670	2670	2670
SO	0	0	800	1400	1400
SC	1.0989	0.32967	0	4.94505	4.94505

Tabel 4. 11 Informasi untuk *Port 4*

<i>Port 4</i>	Periode				
	1	2	3	4	5
OH	800	0	0	0	0
IT	0	0	13000	13000	13000
IP	800	0	13000	13000	13000
SR	0	0	0	0	0
CR	800	800	800	800	800
RSP	2100	2100	2100	2100	2100
SO	0	0	800	800	800
SC	0.66667	0	10.8333	10.8333	10.8333

Tabel 4. 12 Informasi untuk *Port 5*

<i>Port 5</i>	Periode				
	1	2	3	4	5
OH	1000	50	0	0	0
IT	12000	12000	12000	12000	12000
IP	13000	12050	12000	12000	12000
SR	0	0	0	0	0
CR	950	950	950	950	950
RSP	2850	2850	2850	2850	2850
SO	0	0	900	950	950
SC	0.52632	6.34211	6.31579	6.31579	6.31579

Tabel 4.8 - 4.12 menunjukkan informasi mengenai *On-Hand, Intransit Inventory, Inventory Position, Scheduled Receipt, Consumption Rate, Re-shipment Point, Stock Out, dan Stock Criticality* untuk masing-masing *port* selama horison perencanaan yang ditetapkan. Berdasarkan hasil yang didapatkan dari enumerasi, didapatkan solusi penjadwalan dan penugasan kapal dengan total biaya yang dikeluarkan sebesar Rp. 12.158.865,56.

4.4.2 Analisis Uji Validasi

Analisis uji validasi pada algoritma SIM-A yang dikembangkan dilakukan dengan cara membandingkan hasil dari proses enumerasi dengan hasil *running* kode pemrograman MATLAB R2010a. Berdasarkan hasil yang didapatkan dari enumerasi dan *running* MATLAB dapat diketahui bahwa solusi permasalahan yang dihasilkan dari keduanya bernilai sama yang meliputi penjadwalan dan penugasan masing-masing kapal dan informasi masing-masing *port* selama horison perencanaan yang ditetapkan. Di samping itu juga, dilihat dari total biaya yang dikeluarkan keduanya menunjukkan nilai yang sama. Hasil *running* MATLAB untuk kasus sederhana dapat dilihat pada gambar 4.7 - 4.9 sedangkan hasil enumerasi ditunjukkan pada tabel 4.6 – 4.11. Karena dari enumerasi dan *running* permasalahan sederhana didapatkan solusi yang sama maka dapat dikatakan bahwa algoritma SIM – A yang dikembangkan valid.

BAB 5

APLIKASI ALGORITMA UNTUK SISTEM RIIL

Pada bab ini akan dilakukan aplikasi algoritma untuk sistem riil mengenai penjadwalan dan penugasan kapal yang ada di PT. X. Tahap ini dibagi menjadi dua tahapan yaitu pengumpulan data dan pengolahan data. Langkah pertama adalah melakukan pengumpulan data yang diperlukan dalam pengerjaan penelitian. Setelah data terkumpul, selanjutnya dilakukan pengolahan data sesuai dengan *Simple Iterative Mutation Algorithm* yang dirancang.

5.1 Pengumpulan Data

Dalam tahap ini dilakukan pengumpulan data yang diperlukan dalam penyelesaian masalah penjadwalan dan penugasan kapal dengan menggunakan algoritma SIM – A yang dikembangkan. Adapun data yang digunakan dalam penelitian ini meliputi parameter pelabuhan, parameter kapal, parameter pelayaran, parameter biaya. Data terkait parameter pelabuhan yang meliputi jumlah *packing plant* yang harus dikunjungi, *consumption rate*, kapasitas silo, dan kedalaman dermaga didapatkan dari PT. X. Data terkait parameter kapal yang meliputi kapasitas kapal, kecepatan *loading*, kecepatan *unloading* kapal, dan *minimum requirement draft* kapal juga didapatkan dari PT. X. Data untuk parameter biaya seperti biaya *loading*, biaya *unloading*, *rate* waktu tunggu sebelum *loading* dan *unloading* serta *rate* waktu tunggu silo didapatkan dengan cara melakukan penyesuaian dengan menggunakan data biaya pengiriman yang didapatkan dari anak perusahaan yang sama. Salah satu caranya adalah dengan melakukan interpolasi dari data tersebut. Misalnya untuk *rate* sewa kapal per hari, interpolasi dilakukan dengan cara jika untuk kapal dengan kapasitas 7517 ton *rate* sewa kapalnya sebesar Rp. 154,908,039.68 juta per hari dan kapal dengan kapasitas 10.052 ton *rate* sewa kapal per hari sebesar Rp.194,648,897.35 per hari maka dengan menggunakan konsep interpolasi akan didapatkan *rate* sewa kapal per hari untuk kapal dengan kapasitas 6150 ton, begitu juga untuk kapal dengan kapasitas yang lain. Data terkait dengan *sailing time* diestimasi dengan

bantuan *google map* untuk melihat jarak lokasi depot ke *port (packing plant)* yang kemudian penentuan lama durasi *sailing time* dari depot ke *port (packing plant)* tujuan didasarkan dengan melihat data historis waktu pengiriman dan jarak tempuh (lokasi) *port* tujuan dari depot salah satu anak perusahaan dari PT. X. Data terkait *lead time* pengiriman dari depot ke *port* tujuan didapatkan dengan mempertimbangkan waktu *loading*, *sailing time*, dan waktu *unloading*. Oleh karena itu, dalam penelitian ini sebagian data yang digunakan merupakan data asli yang didapatkan dari PT. X dan sebagian lagi merupakan hasil estimasi dengan menggunakan cara – cara yang telah disebut.

Tabel 5.1 merupakan data terkait parameter pelabuhan *loading* dan *unloading* pada PT. X.

Tabel 5. 1 Parameter Pelabuhan *Loading* dan *Unloading* PT. X

Port	Demand (ton)	Kedalaman (m)	Kapasitas (ton)
Depot	-	-	-
1	240	12	5500
2	300	7	6000
3	1157	10	22000
4	742	7	10000
5	845	10	10000
6	2586	10	20000
7	926	12	10000

Tabel 5.2 di bawah ini merupakan data terkait parameter kapal yang digunakan pada pengiriman semen curah di PT. X.

Tabel 5. 2 Parameter Kapal yang Digunakan Pada PT. X

Nama kapal	Kapasitas (ton)	Kecepatan Loading (ton/jam)	Kecepatan Unloading (ton / jam)	Minimum Draft yang dibutuhkan (meter)
A	6000	500	600	6,81
B	6150	700	600	7,15
C	6700	425	350	7,08
D	7800	650	700	7,65
E	6900	550	500	7,70
F	21000	900	600	9,36
G	4500	500	150	6,23
H	5000	300	180	6,75
I	8800	500	600	7,31
J	5900	500	100	6,77
K	6500	500	150	7,15
L	8000	400	100	7,3
M	9500	500	280	7,31

Tabel 5. 3 Parameter Biaya yang Digunakan Pada PT. X

Nama kapal	Harga Sewa Kapal (Rp/Hari)	LCK (Rp.)	ULCK (Rp.)	RDC (Rp/jam)	RWTSL (Rp/jam)	RWTSU (Rp/jam)	RWTS (Rp/jam)
A	136,589,638.21	9,066,000.00	24,438,000.00	58,800.00	58,800.00	58,800.00	58,800.00
B	138,403,009.78	9,292,650.00	25,048,950.00	60,270.00	60,270.00	60,270.00	60,270.00
C	145,052,038.84	10,123,700.00	27,289,100.00	65,660.00	65,660.00	65,660.00	65,660.00
D	158,350,096.95	11,785,800.00	31,769,400.00	76,440.00	76,440.00	76,440.00	76,440.00
E	147,469,867.58	10,425,900.00	28,103,700.00	67,620.00	67,620.00	67,620.00	67,620.00
F	317,926,794.38	31,731,000.00	85,533,000.00	182,708.23	182,708.23	182,708.23	182,708.23

Nama kapal	Harga Sewa Kapal (Rp/Hari)	LCK (Rp.)	ULCK (Rp.)	RDC (Rp/jam)	RWTSL (Rp/jam)	RWTSU (Rp/jam)	RWTS (Rp/jam)
G	118,455,922.60	6,799,500.00	18,328,500.00	44,100.00	44,100.00	44,100.00	44,100.00
H	124,500,494.47	7,555,000.00	20,365,000.00	49,000.00	49,000.00	49,000.00	49,000.00
I	170,439,240.70	13,296,800.00	35,842,400.00	86,240.00	86,240.00	86,240.00	86,240.00
J	135,380,723.84	8,914,900.00	24,030,700.00	57,820.00	57,820.00	57,820.00	57,820.00
K	142,634,210.09	9,821,500.00	26,474,500.00	63,700.00	63,700.00	63,700.00	63,700.00
L	160,767,925.70	12,088,000.00	32,584,000.00	78,400.00	78,400.00	78,400.00	78,400.00
M	178,901,641.32	14,354,500.00	38,693,500.00	93,100.00	93,100.00	93,100.00	93,100.00

Tabel 5.3 merupakan tabel parameter biaya yang bukan merupakan data riil dari PT. X melainkan merupakan data yang didapatkan dengan cara melakukan estimasi seperti yang telah dijelaskan sebelumnya.

Parameter pelayaran dari depot menuju masing – masing *port* tujuan (*packing plant*) ditunjukkan pada tabel 5.4 di bawah ini.

Tabel 5. 4 Parameter Pelayaran

<i>Port</i>	Depot	
	<i>Sailing Time</i> (Jam)	<i>Lead Time</i> (Hari)
1	33.6	3.32
2	40.8	3.62
3	52.8	4.09
4	100.8	6.12
5	81.6	5.35
6	36	3.38
7	40.8	3.58

Lama waktu *sailing time* dan *lead time* yang ditunjukkan pada tabel 5.4 di didapatkan dengan cara bantuan *google map* untuk melihat jarak (lokasi)

port tujuan dari depot kemudian dengan melihat data historis pengiriman kapal dari salah satu perusahaan PT. X yang akan dijadikan sebagai acuan dalam menentukan lama durasi *sailing time* dan *lead time* dalam penelitian ini.

5.2 Pengolahan Data

Pada bab ini dilakukan pengolahan data terhadap data waktu tunggu sebelum *loading* (WTSL) dan waktu tunggu sebelum *unloading* (WTSU) beserta probabilitas kejadian untuk masing - masing kategori waktu tunggu.

5.2.1 Distribusi Waktu Tunggu Sebelum *Loading* (WTSL)

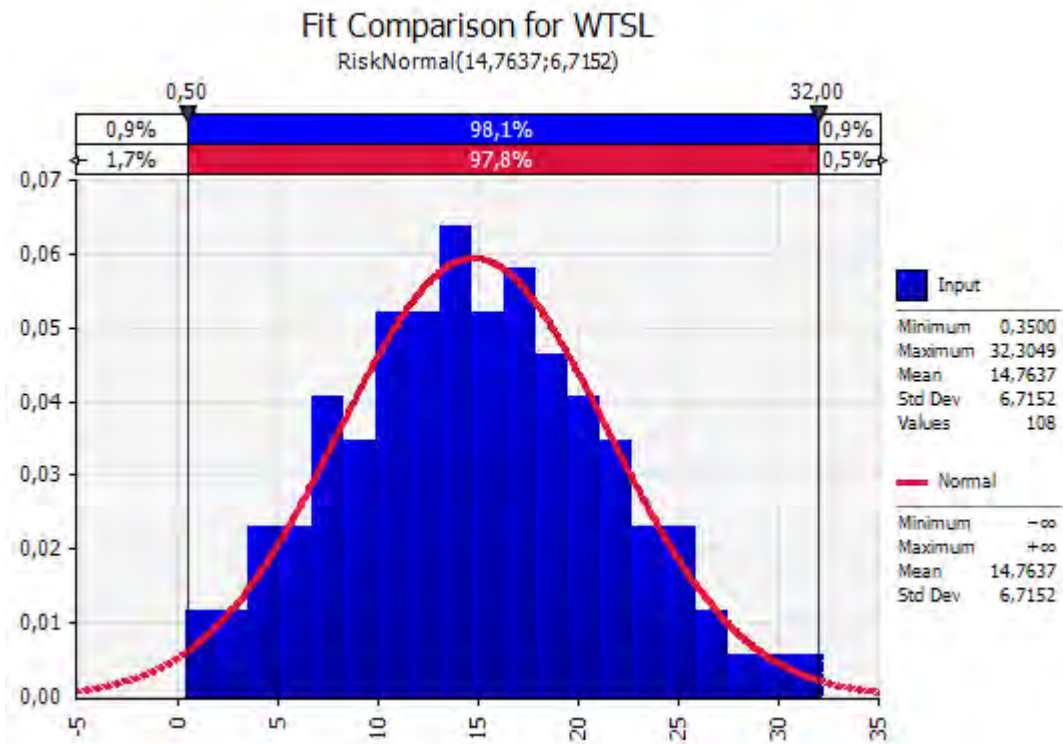
Dari data historis pengiriman kapal, frekuensi kejadian kapal menunggu sebelum *loading* dicari dengan menggunakan persamaan 5.1 di bawah ini.

$$\text{Frekuensi WTSL} = \frac{\text{Jumlah terjadinya WTSL}}{\text{Jumlah data keseluruhan}} \quad (5.1)$$

Frekuensi terjadinya WTSL akan dijadikan sebagai acuan untuk menentukan probabilitas kejadian WTSL di dalam algoritma yang dikembangkan. Berikut ini ditunjukkan perhitungan frekuensi WTSL.

$$\text{Frekuensi WTSL} = \frac{79}{108} = 0.728$$

Untuk parameter distribusi waktu tunggu sebelum *loading* dicari dengan melakukan *fitting* distribusi terhadap data historis lama waktu tunggu data historis. *Fitting* distribusi untuk waktu tunggu dilakukan dengan menggunakan bantuan *software @RISK*. Gambar 5.1 menunjukkan hasil *fitting* distribusi untuk WTSL.



Gambar 5. 1 *Fitting* Distribusi WTSU

Berdasarkan hasil *fitting* distribusi dapat diketahui bahwa WTSU berdistribusi normal dengan rata – rata sebesar 14.7637 jam dan standar deviasi sebesar 6.715 jam.

Berdasarkan hasil perhitungan frekuensi WTSU dan *fitting* distribusi di atas, maka di dalam kode pemrograman kapal akan mengalami WTSU jika bilangan random yang dibangkitkan (r) ≤ 0.728 dengan durasi berdistribusi normal (14.7637;6.715) jam.

5.2.2 Distribusi Waktu Tunggu Sebelum Unloading (WTSU)

Frekuensi waktu tunggu sebelum *unloading* (WTSU) dicari dengan mengacu pada data historis, dimana dari data historis dihitung frekuensi terjadinya waktu tunggu sebelum *unloading* di *port* tujuan dengan menggunakan persamaan 5.2 di bawah ini.

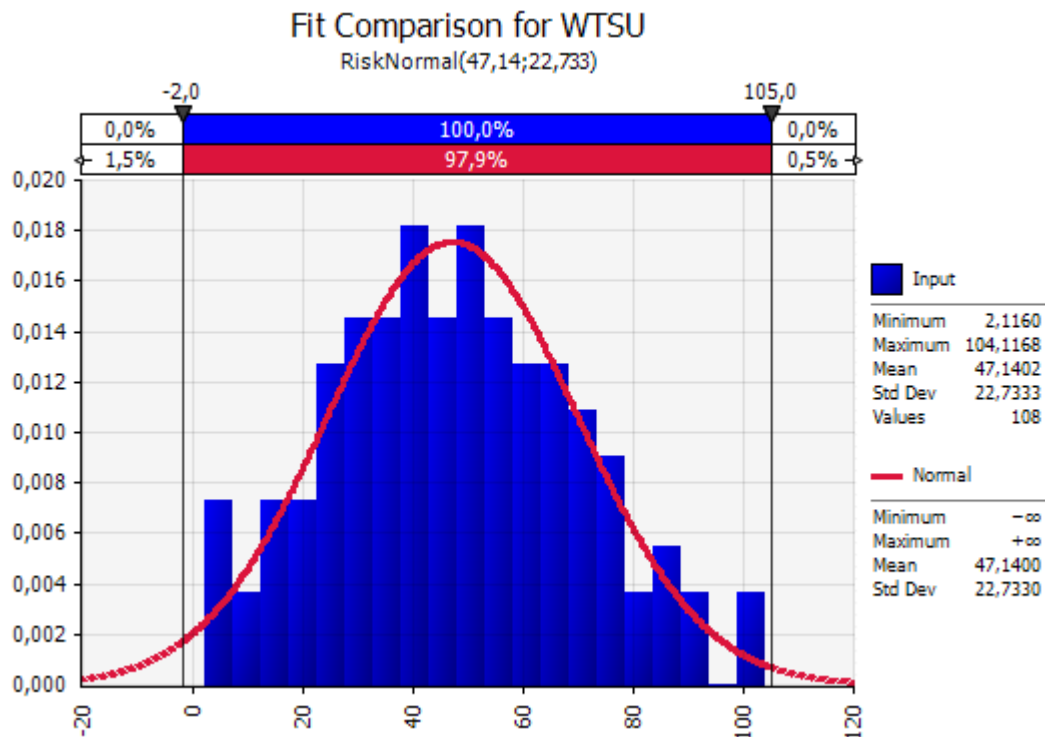
$$\text{Frekuensi WTSU} = \frac{\text{Jumlah terjadinya WTSU}}{\text{Jumlah data keseluruhan}} \quad (5.2)$$

Frekuensi waktu tunggu sebelum *unloading* akan digunakan untuk menentukan probabilitas WTSU di dalam algoritma yang dikembangkan. Berdasarkan hasil perhitungan didapatkan frekuensi WTSU sebesar 0.9, maka probabilitas waktu tunggu yang akan digunakan dalam algoritma juga bernilai sama yaitu 0.9. Persamaan 5.2 menunjukkan hasil perhitungan frekuensi WTSU.

$$\text{Frekuensi WTSU} = \frac{97}{108} = 0.9$$

Distribusi WTSU dicari dengan melakukan *fitting* distribusi pada data historis WTSU menggunakan bantuan *software @RISK*. Parameter WTSU akan digunakan sebagai *input* di dalam kode program algoritma *Simple Iterative Mutation Algorithm* yang dikembangkan baik untuk scenario 1 maupun skenario 2. Begitu juga untuk parameter WTSL akan digunakan sebagai *input* dalam *running* algoritma.

Berdasarkan hasil *fitting* distribusi untuk waktu tunggu sebelum *unloading* (WTSU) yang ditunjukkan oleh Gambar 5.2 di bawah ini, dapat diketahui bahwa jika kapal mengalami proses menunggu sebelum melakukan *unloading* maka kapal akan menunggu (WTSL) dengan durasi berdistribusi normal (47,14 ; 22,73).



Gambar 5. 2 Hasil *Fitting* Distribusi WTSU

5.2.3 Running Algoritma SIM – A

Pada bagian ini dilakukan *running* kode Algoritma SIM – A yang telah dikembangkan dengan menggunakan data yang telah dikumpulkan pada bagian sebelumnya. Pada *running* algoritma SIM – A yang dikembangkan digunakan parameter *input* dengan jumlah *port* yang harus dilayani sebanyak 7, jumlah kapal yang tersedia yang bisa digunakan sebesar 13, jumlah populasi atau solusi yang dibangkitkan sebesar (N) 1000 solusi, rasio sampel elit yang dipilih (ρ) sebesar 0.1, fraksi *safety stock* sebesar 0.05, dan jumlah iterasi maksimum yang digunakan sebesar 100 iterasi. Besarnya *penalty cost* akibat keterlambatan (*stock out*) dikenakan sebesar Rp. 10.000.000, sedangkan akibat kelebihan *On-Hand* dikenakan *penalty cost* sebesar Rp. 5.000.000. Parameter *input* tersebut digunakan untuk kedua skenario yang dikembangkan. Inisial *On-Hand* yang digunakan dalam *running* algoritma untuk *packing plant* 1, 2, 3, 4, 5, 6, dan 7 berturut - turut yaitu 1000, 3500, 2000, 2500, 1000, 4000, dan 2300 ton.

5.2.3.1 Hasil Running Skenario 1

Pada bagian ini akan ditunjukkan hasil *running* kode algoritma SIM – A yang dikembangkan yaitu proses *loading* di depot dan proses *unloading* di port hanya boleh dimulai dari jam 08:00 – 18:00. Di bawah ini ditunjukkan rekap data solusi optimum penjadwalan dan penugasan kapal hasil *running* kode algoritma SIM – A untuk skenario 1.

Tabel 5. 5 Rekap Data Penjadwalan dan Penugasan Kapal Skenario 1

Periode	Kapal	Port	Periode	Kapal	Port
1	E	3	16	-	-
	C	5		-	-
2	H	4	17	-	-
	I	6		-	-
3	G	1	18	E	3
	B	7		-	-
4	A	6	19	D	6
5	-	-	20	H	4
6	J	4	21	C	7
7	-	-	22	-	-
8	-	-	23	B	5
	-	-		I	6
9	D	3	24	G	1
	M	6		-	-
10	G	2	25	M	3
	L	5		-	-
11	-	-	26	-	-
12	B	7	27	A	4
13	I	6	28	D	6
14	-	-	29	-	-
15	-	-	30	J	2

Tabel 5.5 menunjukkan bahwa pada periode berapa saja kapal ditugaskan dan jika ditugaskan maka kapal ditugaskan ke *port* tujuan yang mana selama 30 hari horison perencanaan. Total biaya yang dikeluarkan dari penjadwalan dan penugasan kapal tersebut yaitu sebesar Rp. 635,380,394,107.29. Tabel 5.6 merupakan tabel status kapal dari waktu ke waktu selama 30 hari horison perencanaan berdasarkan solusi optimum yang didapatkan dari skenario 1. Pada periode 5, 7, dan 8 tidak ada kapal yang dijadwalkan padahal masih ada kapal yang tersedia pada periode tersebut. Hal tersebut terjadi karena beberapa hal diantaranya kapal yang tersedia tidak *feasible* ditugaskan ke *port* tujuan kritis dan juga terdapat *port* yang belum kritis pada periode tersebut. Begitu juga untuk periode selanjutnya jika tidak ada kapal yang ditugaskan pada *port* tujuan manapun maka kemungkinan yang terjadi adalah kapal yang *available* pada periode tersebut tidak *feasible* untuk ditugaskan karena tidak memenuhi konstrain MRD dan kapasitas atau juga memang tidak ada kapal yang *available* pada periode tersebut. Akan tetapi pada kasus ini, berdasarkan solusi optimum yang didapatkan selalu ada kapal yang *available* di setiap periode dalam 30 hari horison perencanaan seperti yang ditunjukkan pada tabel 5.6 tentang status kapal.

Tabel 5. 6 Status Kapal Berdasarkan Solusi Optimum Skenario 1

Nama Kapal	Periode																														Avalable	Non Available	Utilisasi	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30				
A	A	A	A	D	B	B	B	B	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	D	B	B	B	21	9	30%
B	A	A	D	B	B	B	B	B	B	B	A	D	B	B	B	B	B	A	A	A	A	D	B	B	B	B	B	B	B	B	7	23	77%	
C	D	B	B	B	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A	D	B	B	B	B	B	B	B	A	A	12	18	60%		
D	A	A	A	A	A	A	A	A	D	B	B	B	B	B	B	B	B	D	B	B	B	B	B	B	A	A	D	B	B	10	20	67%		
E	D	B	B	B	B	B	B	B	B	A	A	A	A	A	A	A	D	B	B	B	B	B	B	B	B	A	A	A	A	12	18	60%		
F	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	30	0	0%		
G	A	A	D	B	B	B	B	B	B	D	B	B	B	B	B	B	B	B	A	A	A	A	D	B	B	B	B	B	B	6	24	80%		
H	A	D	B	B	B	B	B	B	B	B	B	B	B	B	B	A	A	A	A	D	B	B	B	B	B	B	B	B	B	5	25	83%		
I	A	D	B	B	B	B	B	B	B	A	A	A	D	B	B	B	B	B	B	B	A	D	B	B	B	B	B	A	A	7	23	77%		
J	A	A	A	A	A	D	B	B	B	B	B	B	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A	A	D	15	15	50%		
K	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	30	0	0%		
L	A	A	A	A	A	A	A	A	A	D	B	B	B	B	B	B	B	B	B	B	B	A	A	A	A	A	A	A	A	17	13	43%		
M	A	A	A	A	A	A	A	A	D	B	B	B	B	B	B	A	A	A	A	A	A	A	A	D	B	B	B	B	B	17	13	43%		

Keterangan :

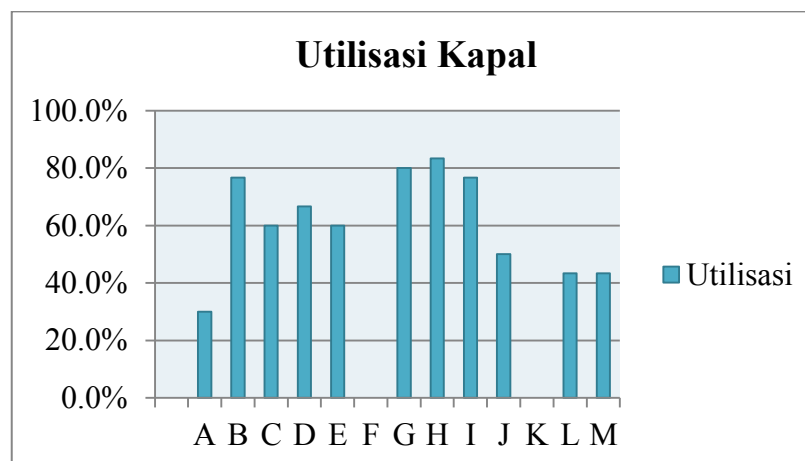
A = Available

D = Dijadwalkan (Awal periode dijadwalkan)

B = Busy

Berdasarkan tabel 5.6 dapat diketahui status masing – masing kapal dari waktu ke waktu selama 30 hari horison perencanaan. Di samping itu juga didapatkan informasi berupa utilisasi masing – masing kapal. Utilisasi kapal merupakan rasio antara jumlah sibuk dengan lama horison perencanaan. Berdasarkan hasil perhitungan dapat diketahui ternyata kapal F dan K utilitasinya sebesar 0% karena selama berdasarkan hasil *running* skenario 1, kedua kapal tersebut tidak pernah dipilih untuk ditugaskan untuk pengiriman semen curah ke *packing plant* manapun. Hal itu terjadi karena dalam proses pemilihan kapal yang terpilih untuk ditugaskan dalam algoritma dilakukan dengan cara random menggunakan matrik transisi dan berdasarkan hasil *running* kebetulan kedua kapal tersebut tidak terpilih meskipun jumlah solusi yang dibangkitkan sebanyak 1000 solusi pada tahap inisialisasi. Berdasarkan hasil perhitungan utilisasi ternyata terdapat beberapa kapal yang utilitasinya di bawah 50%, oleh karena itu perlu dilakukan upaya untuk meningkatkan utilisasi masing – masing kapal. Di samping itu juga, dengan adanya kapal yang tidak digunakan sama sekali selama horison perencanaan maka kapal tersebut tidak perlu di sewa oleh PT. X untuk mengurangi biaya sewa kapal.

Gambar 5.3 di bawah ini merupakan grafik utilisasi masing – masing kapal berdasarkan solusi optimum yang didapatkan dari hasil *running* skenario 1.



Gambar 5. 3 Utilisasi Kapal Berdasarkan Solusi Optimum Skenario 1

Berdasarkan Gambar 5.3 di atas, dapat diketahui bahwa kapal H merupakan kapal yang memiliki utilisasi tertinggi dibandingkan dengan kapal

lainnya yaitu sebesar 83.3%, sedangkan kapal dengan utilisasi terendah adalah kapal F dan kapal K dengan utilisasi masing – masing sebesar 0%. Dengan kata lain, kapal F dan kapal K berdasarkan solusi optimum yang didapatkan dari *running* skenario 1 tidak pernah digunakan sekalipun untuk mengirimkan semen curah selama horison perencanaan yang ditetapkan.

Tabel 5.7 – 5.13 memberikan informasi mengenai masing – masing *port* berdasarkan solusi penjadwalan yang didapatkan yang meliputi *On-Hand*, *Intransit Inventory*, *Inventory Position*, *Scheduled Receipt*, *Consumption Rate*, *Re-shipment point*, *Stock Out* dan *Stock Criticality* selama 30 hari horizon perencanaan yang ditetapkan berdasarkan solusi optimum yang didapatkan dari skenario 1.

Tabel 5. 7 Informasi Silo Port Tujuan 1

<i>Port 1</i>	Periode														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OH	1000	760	520	280	40	0	0	0	4260	4020	3780	3540	3300	3060	2820
IT	0	0	4500	4500	4500	4500	4500	4500	0	0	0	0	0	0	0
IP	1000	760	5020	4780	4540	4500	4500	4500	4260	4020	3780	3540	3300	3060	2820
SR	0	0	0	0	0	0	0	0	4500	0	0	0	0	0	0
CR	240	240	240	240	240	240	240	240	240	240	240	240	240	240	240
RSP	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072
SO	0	0	0	0	0	200	240	240	0	0	0	0	0	0	0
SC	1.26	0.95	0.65	6.00	5.70	5.65	5.65	5.65	5.35	5.05	4.74	4.44	4.14	3.84	3.54
<i>Port 1</i>	Periode														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
OH	2580	2340	2100	1860	1620	1380	1140	900	660	420	180	0	0	0	4260
IT	0	0	0	0	0	0	0	0	4500	4500	4500	4500	4500	4500	0
IP	2580	2340	2100	1860	1620	1380	1140	900	5160	4920	4680	4500	4500	4500	4260
SR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4500
CR	240	240	240	240	240	240	240	240	240	240	240	240	240	240	240
RSP	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072
SO	0	0	0	0	0	0	0	0	0	0	0	60	240	240	0
SC	3.24	2.94	2.64	2.33	2.03	1.73	1.43	1.13	0.83	6.17	5.87	5.65	5.65	5.65	5.35

Tabel 5. 8 Informasi Silo Port Tujuan 2

<i>Port 2</i>	Periode														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OH	3500	3200	2900	2600	2300	2000	1700	1400	1100	800	500	200	0	0	0
IT	0	0	0	0	0	0	0	0	0	4500	4500	4500	4500	4500	4500
IP	3500	3200	2900	2600	2300	2000	1700	1400	1100	5300	5000	4700	4500	4500	4500
SR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CR	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300
RSP	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386
SO	0	0	0	0	0	0	0	0	0	0	0	0	100	300	300
SC	3.22	2.95	2.67	2.39	2.12	1.84	1.57	1.29	1.01	0.74	4.60	4.33	4.14	4.14	4.14
<i>Port 2</i>	Periode														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
OH	0	0	0	4200	3900	3600	3300	3000	2700	2400	2100	1800	1500	1200	900
IT	4500	4500	4500	0	0	0	0	0	0	0	0	0	0	0	5900
IP	4500	4500	4500	4200	3900	3600	3300	3000	2700	2400	2100	1800	1500	1200	6800
SR	0	0	0	4500	0	0	0	0	0	0	0	0	0	0	0
CR	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300
RSP	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386
SO	300	300	300	0	0	0	0	0	0	0	0	0	0	0	0
SC	4.14	4.14	4.14	3.87	3.59	3.31	3.04	2.76	2.49	2.21	1.93	1.66	1.38	1.10	0.83

Tabel 5. 9 Informasi Silo *Packing Plant 3*

<i>Port 3</i>	Periode														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OH	2000	843	0	0	0	0	0	5743	4586	3429	2272	1115	0	0	0
IT	6900	6900	6900	6900	6900	6900	6900	0	7800	7800	7800	7800	7800	7800	7800
IP	8900	7743	6900	6900	6900	6900	6900	5743	12386	11229	10072	8915	7800	7800	7800
SR	0	0	0	0	0	0	0	6900	0	0	0	0	0	0	0
CR	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157
RSP	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832
SO	0	0	314	1157	1157	1157	1157	0	0	0	0	0	42	1157	1157
SC	0.423	1.636	1.458	1.458	1.458	1.458	1.458	1.214	0.969	2.373	2.128	1.884	1.648	1.648	1.648
<i>Port 3</i>	Periode														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
OH	6643	5486	4329	3172	2015	858	0	0	5743	4586	3429	2272	1115	0	0
IT	0	0	6900	6900	6900	6900	6900	6900	0	9500	9500	9500	9500	9500	9500
IP	6643	5486	11229	10072	8915	7758	6900	6900	5743	14086	12929	11772	10615	9500	9500
SR	7800	0	0	0	0	0	0	0	6900	0	0	0	0	0	0
CR	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157
RSP	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832
SO	0	0	0	0	0	0	299	1157	0	0	0	0	0	42	1157
SC	1.40	1.16	0.91	2.13	1.88	1.64	1.46	1.46	1.21	0.97	2.73	2.49	2.24	2.01	2.01

Tabel 5. 10 Informasi Silo *Packing Plant 4*

<i>Port 4</i>	Periode														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OH	2500	1758	1016	274	0	0	0	0	0	0	0	4258	3516	2774	2032
IT	0	5000	5000	5000	5000	10900	10900	10900	10900	10900	10900	5900	5900	5900	5900
IP	2500	6758	6016	5274	5000	10900	10900	10900	10900	10900	10900	10158	9416	8674	7932
SR	0	0	0	0	0	0	0	0	0	0	0	5000	0	0	0
CR	742	742	742	742	742	742	742	742	742	742	742	742	742	742	742
RSP	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041
SO	0	0	0	0	468	742	742	742	742	742	742	0	0	0	0
SC	0.55	0.39	1.32	1.16	1.10	1.10	2.40	2.40	2.40	2.40	2.40	2.24	2.07	1.91	1.75
<i>Port 4</i>	Periode														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
OH	7190	6448	5706	4964	4222	3480	2738	1996	1254	512	0	0	4258	3516	2774
IT	0	0	0	0	5000	5000	5000	5000	5000	5000	5000	11000	6000	6000	6000
IP	7190	6448	5706	4964	9222	8480	7738	6996	6254	5512	5000	11000	10258	9516	8774
SR	5900	0	0	0	0	0	0	0	0	0	0	0	5000	0	0
CR	742	742	742	742	742	742	742	742	742	742	742	742	742	742	742
RSP	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041
SO	0	0	0	0	0	0	0	0	0	0	230	742	0	0	0
SC	1.58	1.42	1.26	1.09	0.93	1.87	1.70	1.54	1.38	1.21	1.10	1.10	2.26	2.10	1.93

Tabel 5. 11 Informasi Silo *Packing Plant 5*

<i>Port 5</i>	Periode														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OH	1000	155	0	0	0	0	0	5855	5010	4165	3320	2475	1630	785	0
IT	6700	6700	6700	6700	6700	6700	6700	0	0	8000	8000	8000	8000	8000	8000
IP	7700	6855	6700	6700	6700	6700	6700	5855	5010	12165	11320	10475	9630	8785	8000
SR	0	0	0	0	0	0	0	6700	0	0	0	0	0	0	0
CR	845	845	845	845	845	845	845	845	845	845	845	845	845	845	845
RSP	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021
SO	0	0	690	845	845	845	845	0	0	0	0	0	0	0	60
SC	0.22	1.52	1.48	1.48	1.48	1.48	1.48	1.30	1.11	0.92	2.50	2.32	2.13	1.94	1.77
<i>Port 5</i>	Periode														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
OH	0	0	0	7155	6310	5465	4620	3775	2930	2085	1240	395	0	5305	4460
IT	8000	8000	8000	0	0	0	0	6150	6150	6150	6150	6150	6150	0	0
IP	8000	8000	8000	7155	6310	5465	4620	9925	9080	8235	7390	6545	6150	5305	4460
SR	0	0	0	8000	0	0	0	0	0	0	0	0	0	6150	0
CR	845	845	845	845	845	845	845	845	845	845	845	845	845	845	845
RSP	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021
SO	845	845	845	0	0	0	0	0	0	0	0	0	450	0	0
SC	1.77	1.77	1.77	1.58	1.40	1.21	1.02	0.84	2.01	1.82	1.63	1.45	1.36	1.17	0.99

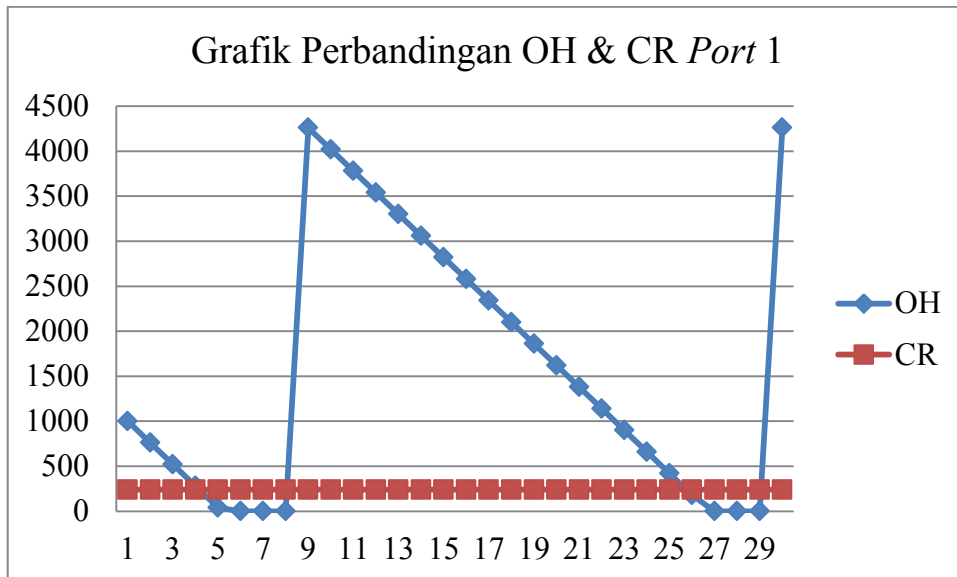
Tabel 5. 12 Informasi Silo *Packing Plant 6*

<i>Port 6</i>	Periode														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OH	4000	1414	0	0	0	0	3414	828	7042	4456	1870	0	0	0	6914
IT	0	8800	8800	14800	14800	14800	8800	8800	9500	9500	9500	9500	18300	18300	8800
IP	4000	10214	8800	14800	14800	14800	12214	9628	16542	13956	11370	9500	18300	18300	15714
SR	0	0	0	0	0	0	6000	0	8800	0	0	0	0	0	9500
CR	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586
RSP	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741
SO	0	0	1172	2586	2586	2586	0	0	0	0	0	716	2586	2586	0
SC	0.46	0.16	1.01	1.01	1.69	1.69	1.40	1.10	0.81	1.60	1.30	1.09	1.09	2.09	1.80
<i>Port 6</i>	Periode														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
OH	4328	1742	0	0	6214	3628	1042	0	0	5214	2628	42	6256	3670	1084
IT	8800	8800	8800	16600	7800	7800	7800	16600	16600	8800	8800	8800	7800	7800	7800
IP	13128	10542	8800	16600	14014	11428	8842	16600	16600	14014	11428	8842	14056	11470	8884
SR	0	0	0	0	8800	0	0	0	0	7800	0	0	8800	0	0
CR	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586
RSP	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741
SO	0	0	844	2586	0	0	0	1544	2586	0	0	0	0	0	0
SC	1.50	1.21	1.01	1.01	1.60	1.31	1.01	0.89	1.90	1.60	1.31	1.01	0.72	1.31	1.02

Tabel 5. 13 Informasi Silo *Packing Plant 7*

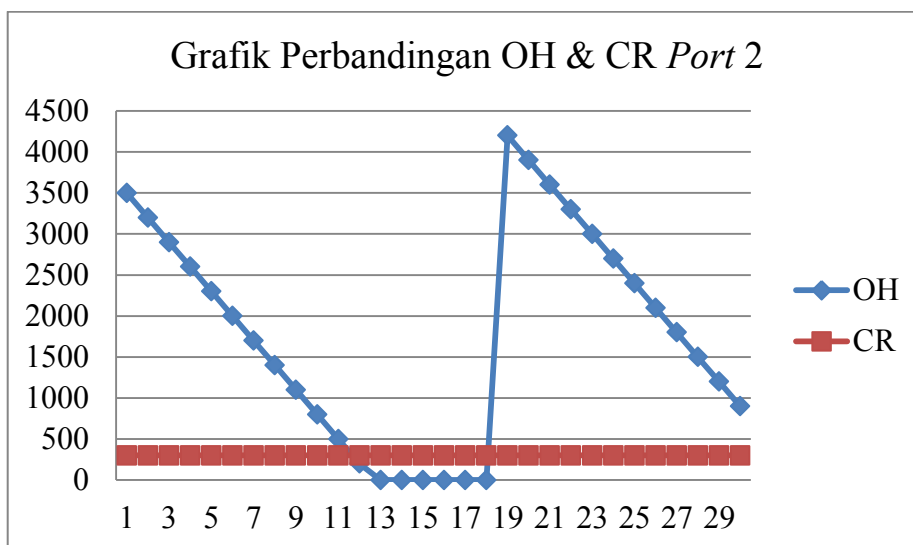
<i>Port 7</i>	Periode														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OH	2300	1374	448	0	0	0	0	0	5224	4298	3372	2446	1520	594	0
IT	0	0	6150	6150	6150	6150	6150	6150	0	0	0	6150	6150	6150	6150
IP	2300	1374	6598	6150	6150	6150	6150	6150	5224	4298	3372	8596	7670	6744	6150
SR	0	0	0	0	0	0	0	0	6150	0	0	0	0	0	0
CR	926	926	926	926	926	926	926	926	926	926	926	926	926	926	926
RSP	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815
SO	0	0	0	478	926	926	926	926	0	0	0	0	0	0	332
SC	0.69	0.41	0.14	1.86	1.86	1.86	1.86	1.86	1.58	1.30	1.02	0.74	2.31	2.03	1.86
<i>Port 7</i>	Periode														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
OH	0	0	5224	4298	3372	2446	1520	594	0	0	0	5774	4848	3922	2996
IT	6150	6150	0	0	0	6700	6700	6700	6700	6700	6700	0	0	0	0
IP	6150	6150	5224	4298	3372	9146	8220	7294	6700	6700	6700	5774	4848	3922	2996
SR	0	0	6150	0	0	0	0	0	0	0	0	6700	0	0	0
CR	926	926	926	926	926	926	926	926	926	926	926	926	926	926	926
RSP	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815
SO	926	926	0	0	0	0	0	0	332	926	926	0	0	0	0
SC	1.86	1.86	1.58	1.30	1.02	0.74	2.48	2.20	2.02	2.02	2.02	1.74	1.46	1.18	0.90

Berdasarkan informasi masing-masing silo dari solusi optimum yang didapatkan dari hasil *running* algoritma SIM – A untuk penjadwalan dan penugasan kapal selama 30 hari horison perencanaan, berikut ditunjukkan grafik perbandingan antara *On-Hand* dan *consumption rate* untuk masing-masing *port*.



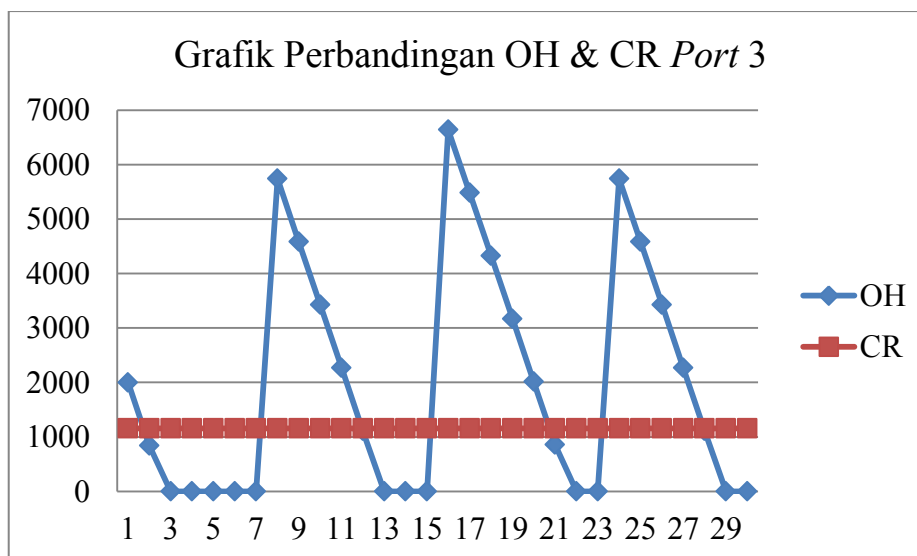
Gambar 5. 4 Grafik Perbandingan OH dan CR di Port 1

Berdasarkan Gambar 5.4 dapat diketahui bahwa *packing plant* 1 mengalami *stock out* sebanyak 7 kali dalam 30 hari periode perencanaan. Dari gambar di atas juga dapat diketahui bahwa ketika *On-Hand* di silo tinggi maka akan dibutuhkan waktu yang cukup lama untuk menghabiskan *On-Hand* tersebut karena *consumption rate* harian di Lhoksumawe relatif sedikit. Oleh karena itu bisa dikatakan rasio antara *On-Hand* dengan *consumption rate* cukup tinggi. Hal ini bisa dilihat pada periode 9 dimana *On-Hand* saat itu sebanyak 4260 ton, *On-Hand* tersebut baru habis pada periode 26.



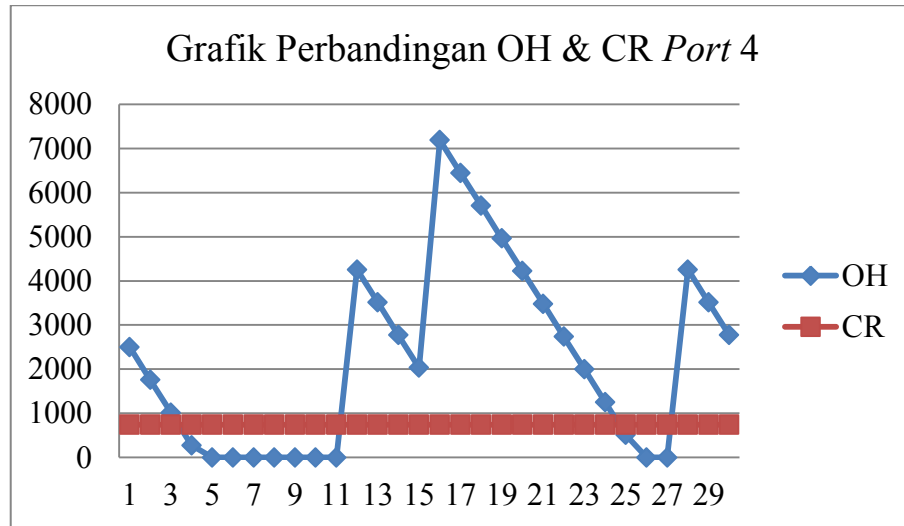
Gambar 5. 5 Grafik Perbandingan OH dan CR di *Port 2*

Berdasarkan Gambar 5.5 dapat diketahui bahwa *packing plant* di *port 2* mengalami *stock out* sebanyak 7 kali dari 30 hari periode perencanaan. Di samping itu juga dapat diketahui bahwa rasio antara *On-Hand* dengan *consumption rate* cukup tinggi. Artinya jika pada saat tertentu jumlah *On-Hand* di silo besar maka akan dibutuhkan waktu yang cukup lama untuk menghabiskan *On-Hand* tersebut dikarenakan *consumption rate* hariannya tidak terlalu besar.



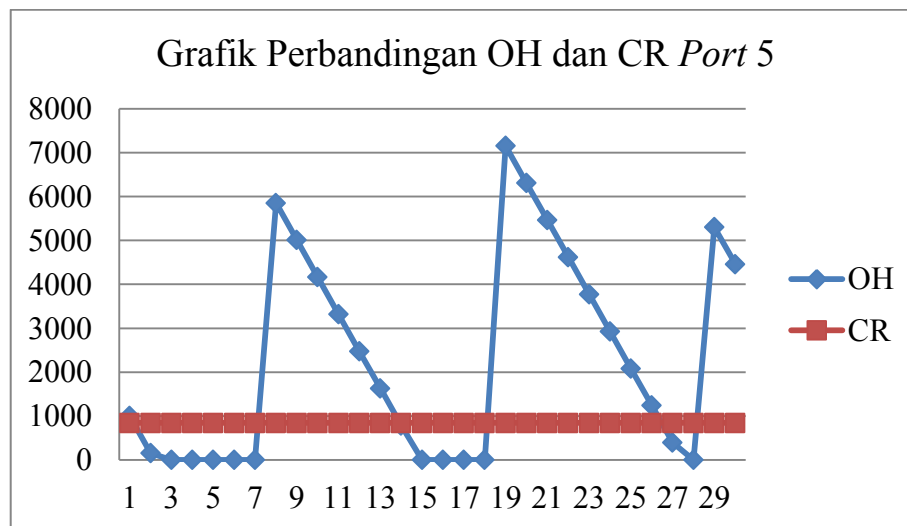
Gambar 5. 6 Grafik Perbandingan OH dan CR *Port 3*

Berdasarkan Gambar 5.6 di atas dapat diketahui bahwa *packing plant* di *port 3* beberapa kali mengalami *stock out*. Rasio antara *On-Hand* dan *consumption rate* di *port 3* relatif kecil karena *consumption rate* hariannya relatif besar.



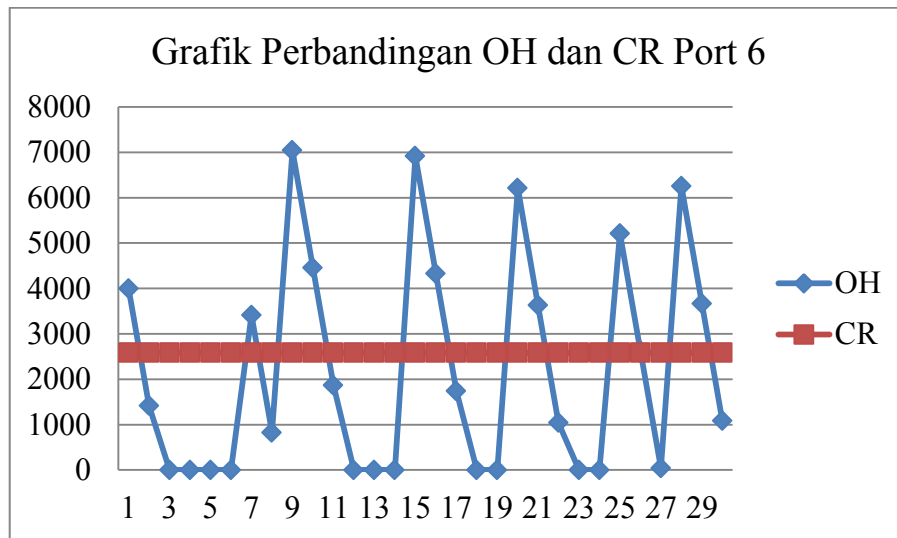
Gambar 5. 7 Grafik Perbandingan OH dan CR *Port 4*

Gambar 5.7 merupakan grafik perbandingan antara *On-Hand* dan *consumption rate* di *packing plant 4*.



Gambar 5. 8 Grafik Perbandingan OH dan CR *Port 5*

Gambar 5.8 merupakan grafik perbandingan antara *On-Hand* dan *consumption rate* di *packing plant (port) 5*.

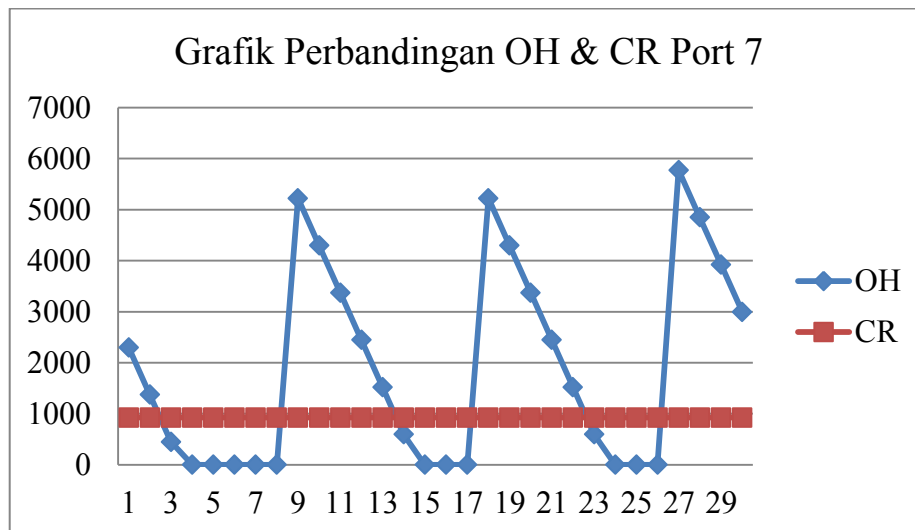


Gambar 5. 9 Grafik Perbandingan OH dan CR *Port 6*

Berdasarkan Gambar 5.9 dapat dilihat bahwa fluktuasi *On-Hand* di *packing plant 6* selama horison perencanaan sanga jelas terlihat. Hal itu dikarenakan berdasarkan solusi optimum penjadwalan dan penugasan kapal yang didapatkan seringkali ada kapal yang ditugaskan ke *port 6* dengan kapasitas yang berbeda dan pada periode yang berbeda yang cukup berdekatan sehingga kapal tersebut akan sampai pada *packing plant* pada periode yang berdekatan pula sehingga ketika ada kapal yang sampai maka *On-Hand* akan langsung di *update* di periode kedatangan kapal. Hal itulah yang membuat fluktuasi *On-Hand* di *port 6* sangat jelas terjadi tidak seperti pada *packing plant* lainnya. Di samping itu juga rasio antara OH dan CR pada *port 6* relatif kecil karena CR harian di *port 6* jumlahnya besar sehingga ketika ada *scheduled receipt* atau barang yang datang dengan jumlah tertentu maka barang tersebut akan habis dalam periode waktu yang relatif singkat sehingga *port 6* dalam waktu yang singkat bisa terpilih lagi menjadi *port* yang kritis. Hal inilah yang mengakibatkan tingkat fluktuasi OH di *port 6* relatif tinggi. Berdasarkan Gambar 5.9 *On – Hand* di *port 6* tidak pernah lebih dari 8000 ton setiap hari (periode) padahal kapasitas silo *port 6* sebesar 20.000 ton. Artinya, setiap periode dalam horison perencanaan terdapat lebih dari 12.000 ton dari kapasitas maksimum silo tidak terisi. Di samping itu juga, selama 30 hari horison perencanaan *port 6* mengalami *stock out* sebanyak 18 periode. Oleh karena itu, maka diperlukan upaya agar *stock out* di *port 6* bisa dikurangi

yaitu dengan cara membuat kapal dengan kapasitas yang besar *dedicated* untuk *port* tujuan dengan kapasitas yang besar. Ketika *port* dengan kapasitas yang besar merupakan *port* kritis terpilih maka kapal yang ditugaskan ke *port* tersebut harusnya kapal dengan kapasitas yang besar pula. Dengan adanya kebijakan tersebut selain frekuensi *stock out* bisa dikurangi di samping itu juga tingkat fluktuasi OH yang sangat tinggi pada *port* 6 bisa berkurang.

Gambar 5.10 di bawah ini merupakan grafik perbandingan *On-Hand* dan *consumption rate* di *packing plant* 7. Pola perubahan *On-Hand* pada *packing plant* 7 kurang lebih sama dengan *packing plant* lainnya kecuali pada *packing plant* 6.



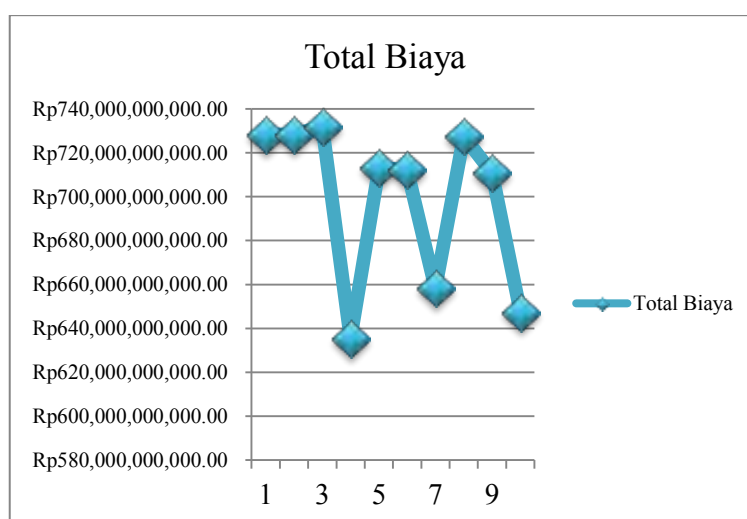
Gambar 5. 10 Grafik Perbandingan OH dan CR *Port* 7

Running kode pemrograman pada skenario 1 dilakukan sebanyak 10 kali percobaan dengan menggunakan parameter *input* yang sama. Rekap data total biaya selama 10 kali percobaan pada skenario 1 ditunjukkan pada tabel 5.14.

Tabel 5. 14 Total Biaya 10 Kali Percobaan Skenario 1

Percobaan Ke-	Total Biaya
1	Rp 728,232,041,910.58
2	Rp 728,437,419,583.91
3	Rp 731,924,041,954.91
4	Rp 635,380,394,107.29
5	Rp 713,142,542,545.18
6	Rp 712,284,328,172.27
7	Rp 658,275,392,983.83
8	Rp 727,605,681,115.53
9	Rp 710,983,194,823.77
10	Rp 647,283,332,293.56

Data hasil *running* percobaan skenario 1 kemudian di *plot* dalam bentuk grafik sesuai yang ditunjukkan oleh gambar 5.11 di bawah ini. Grafik digunakan untuk menunjukkan pola total nilai biaya yang dikeluarkan selama horison perencanaan dari 10 kali percobaan skenario 1.



Gambar 5. 11 Rekap Total Biaya Selama 10 Kali Percobaan

5.2.3.2 Hasil Running Skenario 2

Pada tahap ini dilakukan *running* skenario 2 yaitu penjadwalan dan penugasan kapal dimana waktu mulai proses *loading* bisa dimulai kapanpun dalam rentang 24 jam penuh yaitu jam 0:00 – 24:00, sedangkan waktu mulai proses *unloading* bisa dimulai dari jam 08:00 – 18:00 yang dilihat berdasarkan waktu sampai kapal di *port* tujuan. Tabel 5.15 merupakan rekap data solusi optimum penjadwalan dan penugasan kapal untuk skenario 2.

Tabel 5. 15 Solusi Optimum Penjadwalan dan Penugasan Kapal Pada Skenario 2

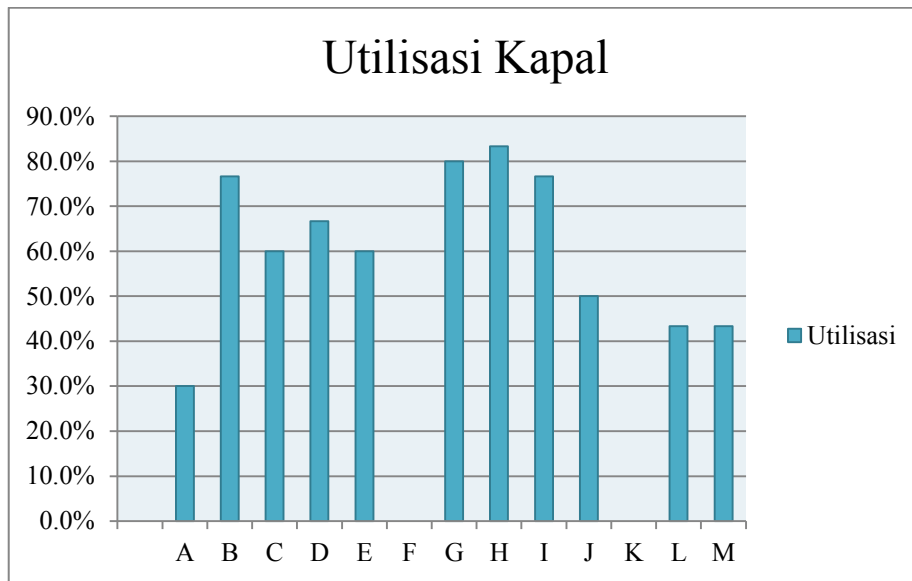
Periode	Kapal	Port	Periode	Kapal	Port
1	G	3	16	-	-
	A	5		-	-
2	J	4	17	-	-
	I	6		-	-
3	H	1	18	D	5
	E	7		A	6
4	B	3	19	J	4
	L	6		-	-
5	-	-	20	E	7
6	-	-	21	G	6
7	-	-	22	-	-
8	-	-	23	A	6
9	-	-	24	-	-
10	G	2	25	-	-
	K	5		-	-
11	E	6	26	H	6
12	F	3	27	B	5
	A	7		-	-
13	H	4	28	G	1
	-	-		D	6
14	I	6	29	A	4
15	-	-	30	J	2

Berdasarkan hasil *running* skenario 2, didapatkan solusi optimal penjadwalan dan penugasan kapal seperti yang ditunjukkan pada tabel 5.13 dengan total biaya sebesar Rp. 503,012,052,848.16. Pada periode 5 sampai 9 tidak satupun kapal yang ditugaskan ke *port* manapun padahal pada periode tersebut ada kapal yang *available*. Hal itu terjadi karena beberapa alasan, yaitu kapal yang tersedia tidak *feasible* ditugaskan karena tidak memenuhi konstrain MRD dan kapasitas dan atau *port* dengan tingkat *stock criticality* terendah masih memilih *inventory position* yang lebih besar dari titik RSP. Periode setelahnya jika tidak ada satupun kapal yang ditugaskan maka kemungkinan yang ada adalah selain kapal tidak memenuhi konstrain MRD dan kapasitas atau tidak ada *port* yang kritis, di samping itu mungkin dikarenakan tidak ada satupun kapal yang *available* pada periode tersebut. Akan tetapi dalam kasus ini, selama horison perencanaan yang ditetapkan, berdasarkan hasil solusi optimum yang didapatkan dari skenario 2, setiap periode selalu ada kapal yang *available* pada periode tersebut seperti yang ditunjukkan pada tabel 5.16 tentang status kapal. Status masing – masing kapal selama 30 hari horizon perencanaan yang didapatkan dari solusi optimum pada skenario 2 di tunjukkan pada table 5.16 di bawah ini.

Tabel 5. 16 Status Kapal Berdasarkan Solusi Optimum Solusi 2

Nama Kapal	Periode																														Available	Non Available	Utilisasi
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30			
A	D	B	B	B	B	B	B	B	B	B	A	D	B	B	B	B	D	B	B	B	B	D	B	B	B	A	A	D	B	3	27	90.00%	
B	A	A	A	D	B	B	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A	A	A	A	A	A	D	B	B	B	17	13	43.33%
C	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	30	0	0.00%
D	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	D	B	B	B	B	B	B	B	B	B	A	D	B	B	18	12	40.00%
E	A	A	D	B	B	B	B	B	B	A	D	B	B	B	B	A	A	A	A	D	B	B	B	B	B	B	B	A	A	A	10	20	66.67%
F	A	A	A	A	A	A	A	A	A	A	D	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A	A	A	A	A	23	7	23.33%
G	D	B	B	B	B	B	B	B	A	D	B	B	B	B	B	B	B	B	B	D	B	B	B	B	B	B	B	D	B	B	1	29	96.67%
H	A	A	D	B	B	B	B	B	B	B	A	A	D	B	B	B	B	B	B	B	B	B	B	B	B	D	B	B	B	B	4	26	86.67%
I	A	D	B	B	B	B	B	B	A	A	A	A	A	D	B	B	B	B	B	A	A	A	A	A	A	A	A	A	A	A	16	14	46.67%
J	A	D	B	B	B	B	B	B	B	B	B	B	B	B	A	A	A	A	D	B	B	B	B	B	B	B	B	B	B	D	5	25	83.33%
K	A	A	A	A	A	A	A	A	A	D	B	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A	A	A	A	A	21	9	30.00%
L	A	A	A	D	B	B	B	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	20	10	33.33%
M	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	30	0	0.00%

Berdasarkan Tabel 5.16 di atas dapat diketahui bahwa terdapat beberapa kapal yang tidak pernah digunakan sama sekali untuk mengirimkan semen curah yaitu kapal C dan kapal M dengan tingkat utilisasi 0%. Di samping itu juga terdapat beberapa kapal dengan tingkat utilisasi di bawah 50% yaitu kapal B, D, F, I, K dan L. Berikut ini ditunjukkan gambar grafik utilisasi masing – masing kapal berdasarkan solusi optimum pada skenario 2.



Gambar 5. 12 Utilisasi Kapal Berdasarkan Solusi Optimum Skenario 2

Kapal dengan tingkat utilisasi terbesar adalah kapal G dengan utilisasi sebesar 96.7%. Artinya, kapal G merupakan kapal yang paling sering beroperasi yang digunakan untuk mengirimkan semen curah ke *packing plant*.

Informasi untuk masing – masing *port* tujuan berdasarkan hasil *running* yang dilakukan ditunjukkan pada tabel di bawah ini.

Tabel 5. 17 Informasi Silo *Packing Plant 1* (Skenario 2)

<i>Port 1</i>	Periode														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OH	1000	760	520	280	40	0	0	0	0	4760	4520	4280	4040	3800	3560
IT	0	0	5000	5000	5000	5000	5000	5000	5000	0	0	0	0	0	0
IP	1000	760	5520	5280	5040	5000	5000	5000	5000	4760	4520	4280	4040	3800	3560
SR	0	0	0	0	0	0	0	0	0	5000	0	0	0	0	0
CR	240	240	240	240	240	240	240	240	240	240	240	240	240	240	240
RSP	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072
SO	0	0	0	0	0	200	240	240	240	0	0	0	0	0	0
SC	1.255	0.954	0.653	6.627	6.325	6.275	6.275	6.275	6.275	5.974	5.673	5.371	5.070	4.769	4.468
<i>Port 1</i>	Periode														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
OH	3320	3080	2840	2600	2360	2120	1880	1640	1400	1160	920	680	440	200	0
IT	0	0	0	0	0	0	0	0	0	0	0	0	4500	4500	4500
IP	3320	3080	2840	2600	2360	2120	1880	1640	1400	1160	920	680	4940	4700	4500
SR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CR	240	240	240	240	240	240	240	240	240	240	240	240	240	240	240
RSP	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072	1072
SO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40
SC	4.17	3.87	3.56	3.26	2.96	2.66	2.36	2.06	1.76	1.46	1.15	0.85	0.55	5.90	5.65

Tabel 5. 18 Informasi Silo *Packing Plant 2* (Skenario 2)

<i>Port 2</i>	Periode														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OH	3500	3200	2900	2600	2300	2000	1700	1400	1100	800	500	200	0	0	0
IT	0	0	0	0	0	0	0	0	0	4500	4500	4500	4500	4500	4500
IP	3500	3200	2900	2600	2300	2000	1700	1400	1100	5300	5000	4700	4500	4500	4500
SR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CR	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300
RSP	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386
SO	0	0	0	0	0	0	0	0	0	0	0	0	100	300	300
SC	3.22	2.95	2.67	2.39	2.12	1.84	1.57	1.29	1.01	0.74	4.60	4.33	4.14	4.14	4.14
<i>Port 2</i>	Periode														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
OH	0	0	0	4200	3900	3600	3300	3000	2700	2400	2100	1800	1500	1200	900
IT	4500	4500	4500	0	0	0	0	0	0	0	0	0	0	0	5900
IP	4500	4500	4500	4200	3900	3600	3300	3000	2700	2400	2100	1800	1500	1200	6800
SR	0	0	0	4500	0	0	0	0	0	0	0	0	0	0	0
CR	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300
RSP	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386	1386
SO	300	300	300	0	0	0	0	0	0	0	0	0	0	0	0
SC	4.14	4.14	4.14	3.87	3.59	3.31	3.04	2.76	2.49	2.21	1.93	1.66	1.38	1.10	0.83

Tabel 5. 19 Informasi Silo *Packing Plant 3* (Skenario 2)

<i>Port 3</i>	Periode														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OH	2000	843	0	0	0	0	3343	2186	1029	0	4993	3836	2679	1522	365
IT	4500	4500	4500	10650	10650	10650	6150	6150	6150	6150	0	21000	21000	21000	21000
IP	6500	5343	4500	10650	10650	10650	9493	8336	7179	6150	4993	24836	23679	22522	21365
SR	0	0	0	0	0	0	4500	0	0	0	6150	0	0	0	0
CR	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157
RSP	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832
SO	0	0	314	1157	1157	1157	0	0	0	128	0	0	0	0	0
SC	0.42	1.13	0.95	0.95	2.25	2.25	2.01	1.76	1.52	1.30	1.06	0.81	5.00	4.76	4.51
<i>Port 3</i>	Periode														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
OH	0	19843	18686	17529	16372	15215	14058	12901	11744	10587	9430	8273	7116	5959	4802
IT	21000	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IP	21000	19843	18686	17529	16372	15215	14058	12901	11744	10587	9430	8273	7116	5959	4802
SR	0	21000	0	0	0	0	0	0	0	0	0	0	0	0	0
CR	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157	1157
RSP	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832	5832
SO	792	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SC	4.44	4.19	3.95	3.70	3.46	3.22	2.97	2.73	2.48	2.24	1.99	1.75	1.50	1.26	1.01

Tabel 5. 20 Informasi Silo *Packing Plant* 4 (Skenario 2)

<i>Port</i> 4	Periode														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OH	2500	1758	1016	274	0	0	0	0	0	0	5158	4416	3674	2932	2190
IT	0	5900	5900	5900	5900	5900	5900	5900	5900	5900	0	0	5000	5000	5000
IP	2500	7658	6916	6174	5900	5900	5900	5900	5900	5900	5158	4416	8674	7932	7190
SR	0	0	0	0	0	0	0	0	0	0	5900	0	0	0	0
CR	742	742	742	742	742	742	742	742	742	742	742	742	742	742	742
RSP	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041
SO	0	0	0	0	468	742	742	742	742	742	0	0	0	0	0
SC	0.55	0.39	1.52	1.36	1.30	1.30	1.30	1.30	1.30	1.30	1.14	0.97	0.81	1.75	1.58
<i>Port</i> 4	Periode														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
OH	1448	706	0	0	0	4258	3516	2774	2032	1290	6448	5706	4964	4222	3480
IT	5000	5000	5000	10900	10900	5900	5900	5900	5900	5900	0	0	0	6000	6000
IP	6448	5706	5000	10900	10900	10158	9416	8674	7932	7190	6448	5706	4964	10222	9480
SR	0	0	0	0	0	5000	0	0	0	0	5900	0	0	0	0
CR	742	742	742	742	742	742	742	742	742	742	742	742	742	742	742
RSP	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041	5041
SO	0	0	36	742	742	0	0	0	0	0	0	0	0	0	0
SC	1.42	1.26	1.10	1.10	2.40	2.24	2.07	1.91	1.75	1.58	1.42	1.26	1.09	0.93	2.09

Tabel 5. 21 Informasi Silo *Packing Plant 5* (Skenario 2)

<i>Port 5</i>	Periode														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OH	1000	155	0	0	0	0	0	5155	4310	3465	2620	1775	930	85	0
IT	6000	6000	6000	6000	6000	6000	6000	0	0	6500	6500	6500	6500	6500	6500
IP	7000	6155	6000	6000	6000	6000	6000	5155	4310	9965	9120	8275	7430	6585	6500
SR	0	0	0	0	0	0	0	6000	0	0	0	0	0	0	0
CR	845	845	845	845	845	845	845	845	845	845	845	845	845	845	845
RSP	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021
SO	0	0	690	845	845	845	845	0	0	0	0	0	0	0	760
SC	0.22	1.36	1.33	1.33	1.33	1.33	1.33	1.14	0.95	0.77	2.02	1.83	1.64	1.46	1.44
<i>Port 5</i>	Periode														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
OH	5655	4810	3965	3120	2275	1430	585	7540	6695	5850	5005	4160	3315	2470	1625
IT	0	0	7800	7800	7800	7800	7800	0	0	0	0	6150	6150	6150	6150
IP	5655	4810	11765	10920	10075	9230	8385	7540	6695	5850	5005	10310	9465	8620	7775
SR	6500	0	0	0	0	0	0	7800	0	0	0	0	0	0	0
CR	845	845	845	845	845	845	845	845	845	845	845	845	845	845	845
RSP	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021	5021
SO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SC	1.25	1.06	0.88	2.42	2.23	2.04	1.85	1.67	1.48	1.29	1.11	0.92	2.09	1.91	1.72

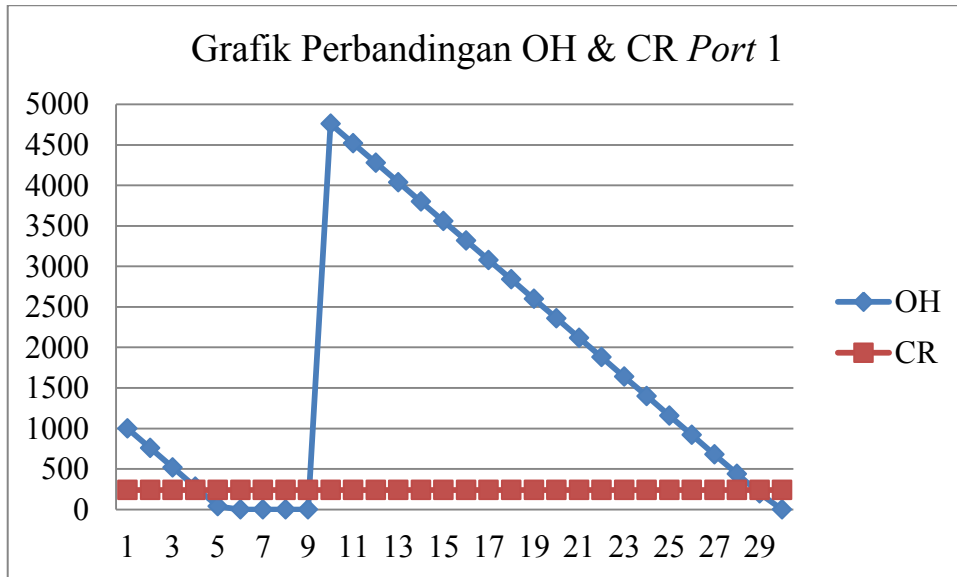
Tabel 5. 22 Informasi Silo *Packing Plant* 6 (Skenario 2)

<i>Port</i> 6	Periode														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OH	4000	1414	0	0	0	0	0	6214	3628	1042	0	5414	2828	242	4556
IT	0	8800	8800	16800	16800	16800	16800	8000	8000	8000	14900	6900	6900	15700	8800
IP	4000	10214	8800	16800	16800	16800	16800	14214	11628	9042	14900	12314	9728	15942	13356
SR	0	0	0	0	0	0	0	8800	0	0	0	8000	0	0	6900
CR	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586
RSP	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741
SO	0	0	1172	2586	2586	2586	2586	0	0	0	1544	0	0	0	0
SC	0.46	0.16	1.01	1.01	1.92	1.92	1.92	1.63	1.33	1.03	0.92	1.41	1.11	0.82	1.53
<i>Port</i> 6	Periode														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
OH	1970	0	0	6214	3628	7042	4456	1870	0	3414	5328	2742	156	2570	0
IT	8800	8800	14800	6000	6000	4500	4500	10500	10500	4500	5000	5000	12800	7800	7800
IP	10770	8800	14800	12214	9628	11542	8956	12370	10500	7914	10328	7742	12956	10370	7800
SR	0	0	0	8800	0	6000	0	0	0	6000	4500	0	0	5000	0
CR	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586	2586
RSP	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741	9741
SO	0	616	2586	0	0	0	0	0	716	0	0	0	0	0	16
SC	1.23	1.01	1.01	1.40	1.10	0.81	1.02	0.73	1.20	0.91	0.61	0.89	0.59	1.19	0.89

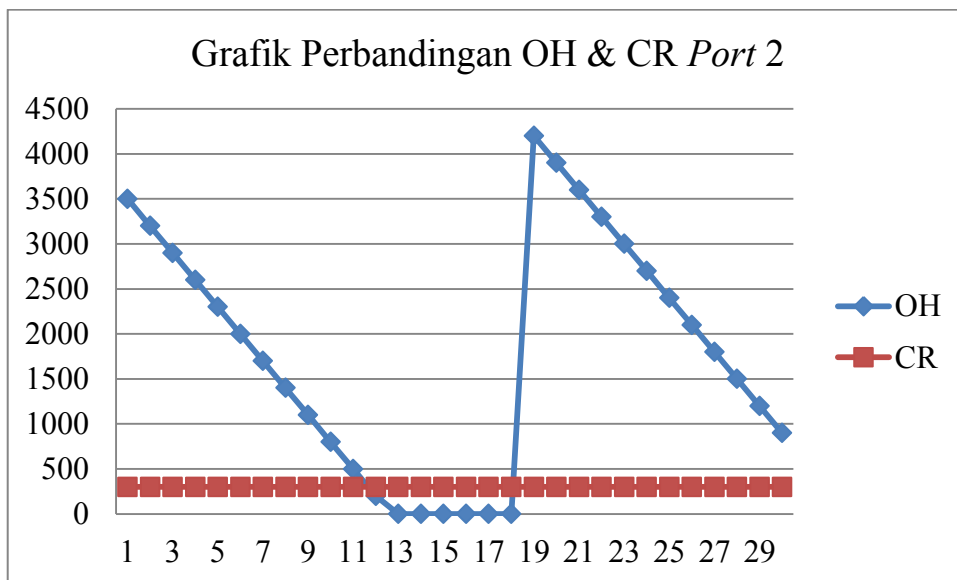
Tabel 5. 23 Informasi Silo *Packing Plant 7* (Skenario 2)

<i>Port 7</i>	Periode														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OH	2300	1374	448	0	0	0	0	5974	5048	4122	3196	2270	1344	418	0
IT	0	0	6900	6900	6900	6900	6900	0	0	0	0	6000	6000	6000	6000
IP	2300	1374	7348	6900	6900	6900	6900	5974	5048	4122	3196	8270	7344	6418	6000
SR	0	0	0	0	0	0	0	6900	0	0	0	0	0	0	0
CR	926	926	926	926	926	926	926	926	926	926	926	926	926	926	926
RSP	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815
SO	0	0	0	478	926	926	926	0	0	0	0	0	0	0	508
SC	0.69	0.41	0.14	2.08	2.08	2.08	2.08	1.80	1.52	1.24	0.96	0.68	2.22	1.94	1.81
<i>Port 7</i>	Periode														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
OH	0	5074	4148	3222	2296	1370	444	0	0	0	0	5974	5048	4122	3196
IT	6000	0	0	0	6900	6900	6900	6900	6900	6900	6900	0	0	0	0
IP	6000	5074	4148	3222	9196	8270	7344	6900	6900	6900	6900	5974	5048	4122	3196
SR	0	6000	0	0	0	0	0	0	0	0	0	6900	0	0	0
CR	926	926	926	926	926	926	926	926	926	926	926	926	926	926	926
RSP	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815	3815
SO	926	0	0	0	0	0	0	482	926	926	926	0	0	0	0
SC	1.81	1.53	1.25	0.97	0.69	2.49	2.22	2.08	2.08	2.08	2.08	1.80	1.52	1.24	0.96

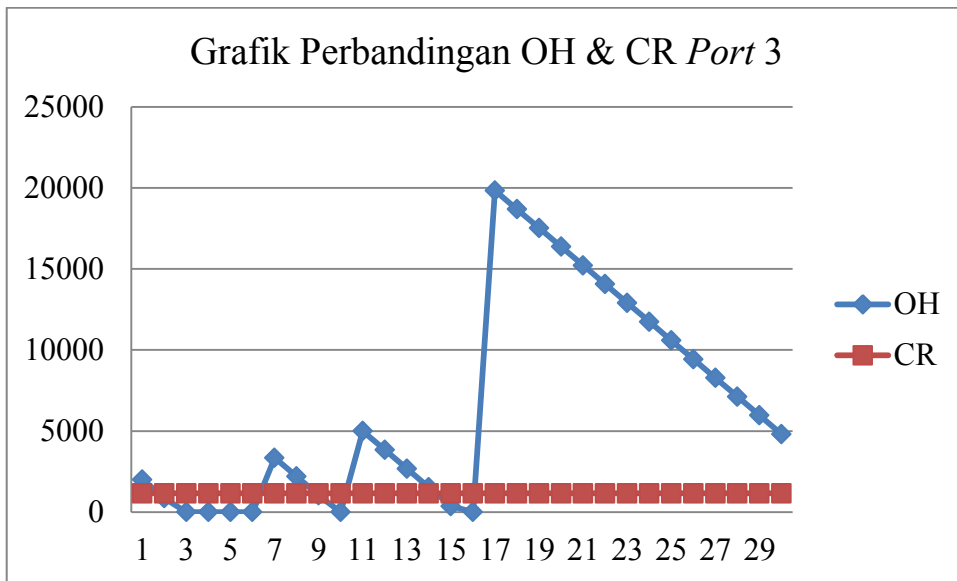
Berdasarkan solusi optimum hasil penjadwalan dan penugasan kapal yang didapatkan pada skenario 2, untuk masing-masing *packing plant* dibuat grafik perbandingan *On-Hand* dan *consumption rate* untuk masing – masing *port (packing plant)* yang ditunjukkan pada gambar di bawah ini.



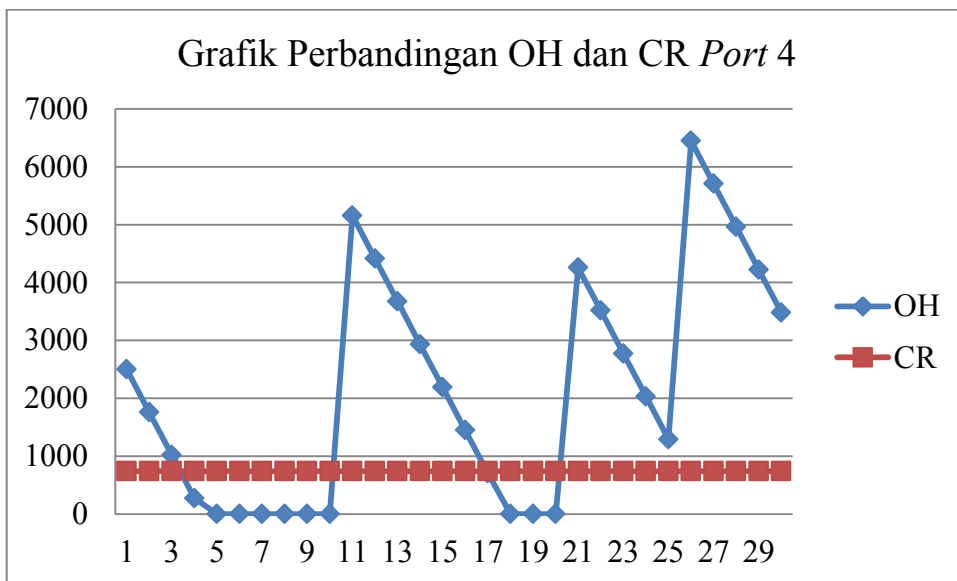
Gambar 5. 13 Grafik Perbandingan OH dan CR *Port 1* (Skenario 2)



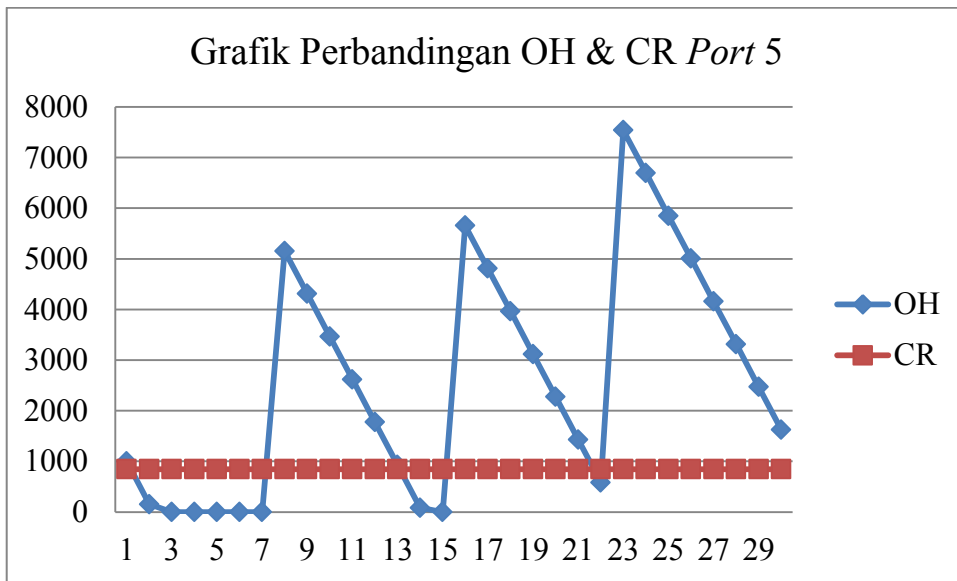
Gambar 5. 14 Grafik Perbandingan OH dan CR *Port 2* (Skenario 2)



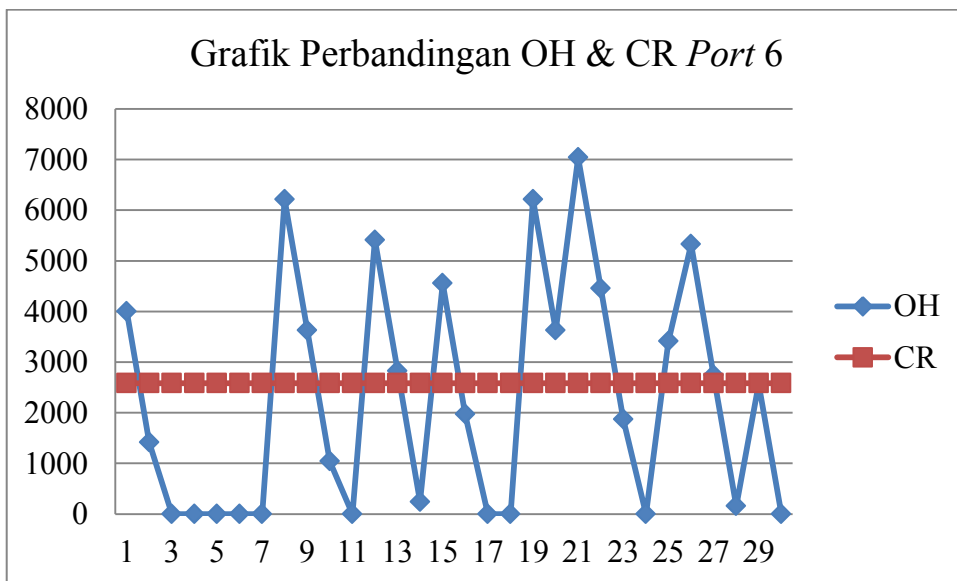
Gambar 5. 15 Grafik Perbandingan OH dan CR di Port 3 (Skenario 2)



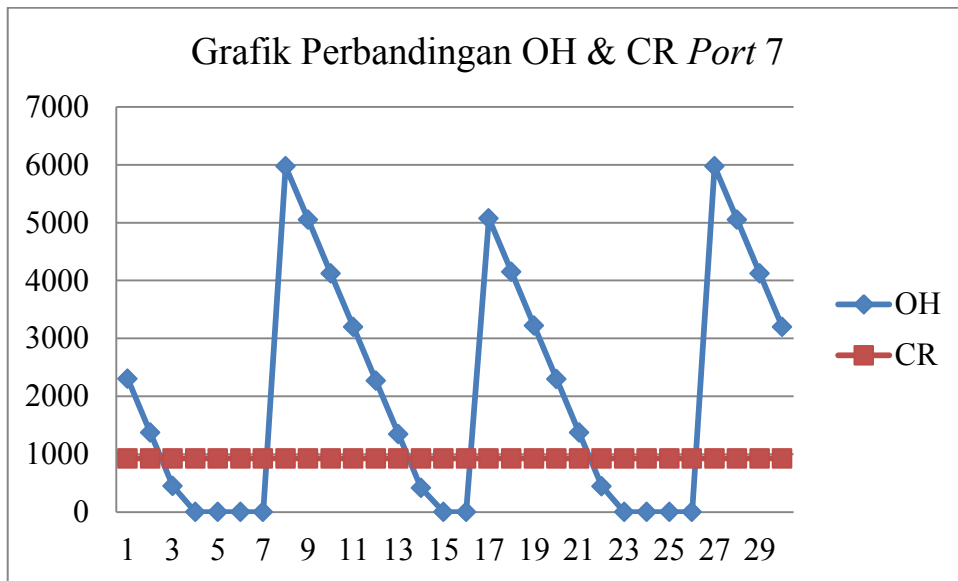
Gambar 5. 16 Grafik Perbandingan OH dan CR Port 4 (Skenario 2)



Gambar 5. 17 Grafik Perbandingan OH dan CR *Port 5* (Skenario 2)



Gambar 5. 18 Grafik Perbandingan OH dan CR di *Port 6* (Skenario 2)



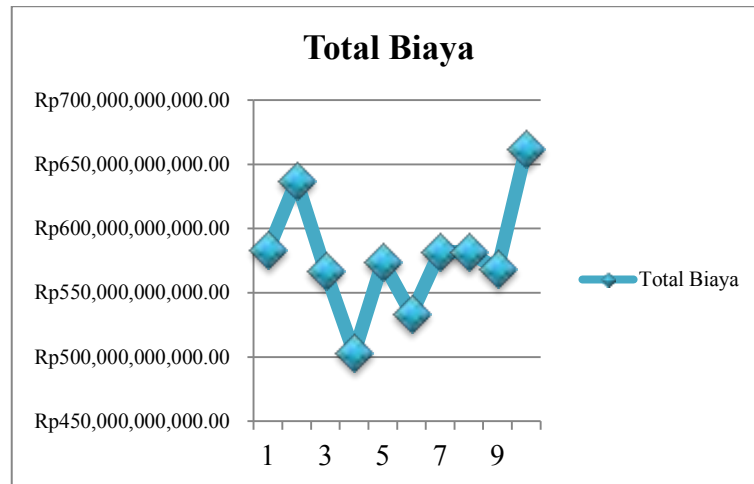
Gambar 5. 19 Grafik Perbandingan OH dan CR di Port 7 (Skenario 2)

Sama halnya dengan skenario 1, pada skenario 2 juga dilakukan *running* percobaan sebanyak 10 kali percobaan. Tabel 5.24 menunjukkan total biaya yang dikeluarkan dari 10 kali percobaan skenario 2.

Tabel 5. 24 Total Biaya 10 Kali Percobaan Skenario 2

Percobaan Ke-	Total Biaya
1	Rp 583,301,392,537.03
2	Rp 637,273,669,000.98
3	Rp 567,118,829,245.65
4	Rp 503,012,052,848.16
5	Rp 574,266,570,871.44
6	Rp 533,532,579,885.83
7	Rp 582,056,594,247.59
8	Rp 581,573,006,395.01
9	Rp 568,632,622,776.89
10	Rp 662,069,873,671.52

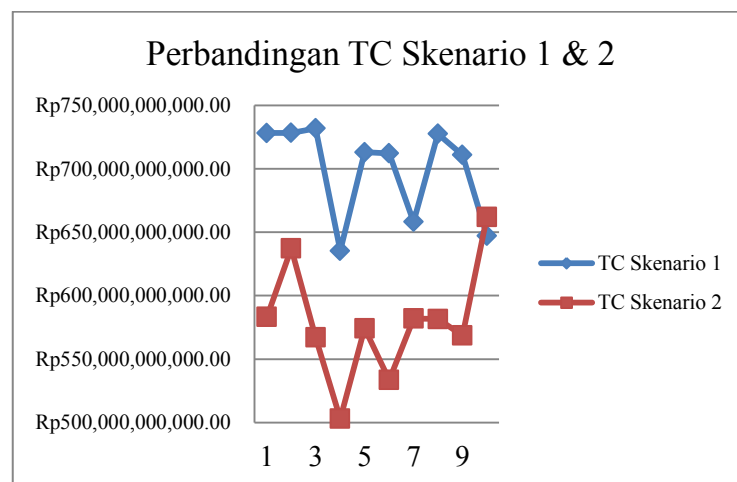
Selanjutnya total biaya selama 10 kali percobaan untuk skenario 2 di-plot untuk melihat perubahan nilai total biaya dari setiap percobaan. Gambar 5.20 menunjukkan hasil *plot* total biaya pada skenario 2 selama 10 percobaan.



Gambar 5. 20 Grafik Total Biaya 10 Kali Percobaan Skenario 2

5.2.3.3 Perbandingan Total Biaya Skenario 1 dan 2

Pada tahap ini dilakukan perbandingan hasil total biaya selama 10 kali percobaan antara skenario 1 dan 2. Gambar 5.21 merupakan grafik perbandingan total biaya dari dua skenario yang dikembangkan.



Gambar 5. 21 Grafik Perbandingan Total Biaya Skenario 1 dan 2

Gambar 5.21 merupakan grafik perbandingan total biaya skenario 1 dan 2 selama 10 kali percobaan. Berdasarkan grafik tersebut dapat diketahui bahwa total biaya yang dikeluarkan oleh solusi dari skenario 2 pada percobaan 1 sampai 9 lebih minimum dibandingkan total biaya yang dihabiskan oleh skenario 1, sedangkan pada percobaan ke – 10 total biaya yang dihasilkan oleh skenario 2 lebih besar dibanding skenario 1. Berdasarkan grafik tersebut dapat dikatakan bahwa, solusi optimum yang dihasilkan oleh skenario 2 cenderung lebih minimum dibandingkan skenario 1 karena pada skenario 2 kapal memiliki fleksibilitas yang lebih tinggi untuk memulai proses *loading* di depot kapanpun kapal sampai di depot dengan syarat kapal tidak mengalami proses waktu tunggu sebelum *loading* akibat ada kondisi tertentu di depot, misalnya depot sedang melayani kapal lain untuk melakukan *loading*.

(halaman ini sengaja dikosongkan)

BAB 6

ANALISIS HASIL DAN REKOMENDASI

Pada bab ini dilakukan analisis terhadap hasil dari aplikasi algoritma untuk sistem riil pada bab sebelumnya. Kemudian setelah itu dilanjutkan dengan pemberian rekomendasi berdasarkan hasil analisis yang dilakukan.

6.1 Analisis Hasil Algoritma SIM – A

SIM – A merupakan salah satu metode metaheuristik yang dapat menyelesaikan permasalahan yang tergolong ke dalam kategori *non polynomial – hard problem*. Algoritma ini mampu menyelesaikan permasalahan kompleks dengan waktu komputasi yang relatif lebih singkat dibandingkan metoda eksak. Berdasarkan hasil dari aplikasi algoritma untuk sistem riil pada bab 5 serta diperkuat dengan uji verifikasi dan validasi, maka algoritma *Simple Iterative Mutation Algorithm* telah berhasil dikembangkan sesuai dengan batasan – batasan yang dipertimbangkan di dalamnya. Verifikasi memastikan bahwa algoritma SIM – A yang dikembangkan telah sesuai dengan logika operasional sehingga bisa dikatakan bahwa algoritma SIM – A *verified*. Uji validasi semakin memperkuat alasan bahwa model tidak hanya *verified* dari sisi ketiadaan *error* dalam kode pemrograman dan telah sesuai dengan logika operasional tetapi juga memastikan bahwa hasil dari *running* algoritma telah valid dengan cara membandingkan antara hasil *running* pada kasus sederhana dengan hasil enumerasi. Analisis hasil algoritma dilakukan pada skenario 1 dan skenario 2.

6.1.1 Analisis Hasil Algoritma SIM – A Skenario 1

Skenario 1 merupakan skenario yang membahas mengenai penjadwalan dan penugasan kapal dimana waktu mulai *loading* di depot dan *unloading* di *port* hanya bisa dilakukan diantara jam 08:00 – 18:00. Berdasarkan hasil aplikasi algoritma (pengumpulan dan pengolahan data) dapat diketahui bahwa dalam melakukan pengiriman semen curah, PT. X menggunakan 13 kapal sewa dengan spesifikasi yang berbeda – beda dengan jumlah *packing plant* yang harus dilayani sebanyak 7 unit yang tersebar di berbagai daerah di Indonesia.

Dengan menggunakan data *input* yang telah dijelaskan pada bab 5 dan dengan parameter jumlah populasi yang dibangkitkan sebesar (N) 1000 solusi, rasio sampel elit yang dipilih (ρ) sebesar 0.1, fraksi *safety stock* sebesar 0.05, dan jumlah iterasi maksimum yang digunakan sebanyak 100 iterasi dengan 10 kali percobaan didapatkan total biaya optimum yang dikeluarkan sebesar Rp. 635,380,394,107.286. Adapun penjadwalan dan penugasan kapal yang optimum pada yang didapatkan dari hasil *running* skenario 1 ditunjukkan pada tabel 5.5. Informasi mengenai masing – masing silo *packing plant* yang dilayani telah ditunjukkan pada tabel 5.7 sampai tabel 5.13. Adapun informasi yang ditunjukkan untuk masing – masing silo meliputi *On-Hand*, *Intransit Inventory*, *Inventory Position*, *Scheduled Receipt*, *Consumption Rate*, *Re-shipment Point*, *Stock Out*, dan *Stock Criticality* masing – masing silo *port* tujuan.

Berdasarkan Gambar 5.11 Rekap Total Biaya Selama 10 Kali Percobaan untuk skenario 1 dapat diketahui bahwa perubahan total biaya yang dikeluarkan untuk pengiriman semen curah (penjadwalan dan penugasan kapal) selama horison perencanaan yang ditetapkan mengalami perubahan yang cukup signifikan antara solusi yang dihasilkan pada masing – masing percobaan. Solusi dengan total biaya didapatkan pada *running* ke – 4. Perbedaan total biaya yang relatif signifikan antara masing – masing percobaan terjadi akibat pemilihan kapal yang ditugaskan untuk mengirimkan semen dipilih secara random berdasarkan matrik probabilitas transisi, sehingga solusi penjadwalan yang didapatkan berbeda – beda, dimana hal tersebut berdampak pada total biaya yang dihabiskan yang berbeda pula untuk masing – masing percobaan. Solusi yang berbeda – beda tersebut juga mengakibatkan jumlah *stock out* yang terjadi pada masing – masing *port* juga berbeda – beda jumlahnya. Dengan adanya *stock out* maka akan berdampak pada dikenakan *penalty cost* dengan jumlah yang sangat besar per satuan ton yaitu sebesar 10.000.000 per ton kekurangan. *Penalty cost* tersebut berperan besar dalam penentuan total biaya yang dihabiskan, sehingga sangat wajar jika masing – masing solusi di setiap percobaan menghasilkan perbedaan biaya yang relatif besar. *Packing plant* 6 merupakan *packing plant* yang paling sering mengalami *stock out* dibandingkan dengan *packing plant* lainnya karena *consumption rate* hariannya lebih tinggi dibandingkan yang lainnya. *Packing*

plant yang lainnya juga mengalami *stock out* di beberapa periode hanya saja jumlahnya tidak sebanyak *packing plant* yang ada di *port* 6. Hal inilah (*stock out* di *packing plant*) yang mempengaruhi jumlah total biaya pengiriman (*objective function*) menjadi bernilai sangat besar. Hasil *running algoritma* SIM – A untuk skenario 1 telah memenuhi konstrain – konstrain yang telah dipertimbangkan. Oleh karena itu dapat dikatakan bahwa SIM – A mampu menyelesaikan permasalahan *Ship Scheduling and Assignment* untuk distribusi semen curah di PT. X.

Gambar 5.3 merupakan grafik utilisasi masing – masing kapal yang didapatkan dari solusi optimum pada skenario 1. Berdasarkan grafik tersebut dapat diketahui bahwa terdapat kapal yang tidak pernah digunakan sama sekali untuk mengirimkan kapal yaitu kapal F dan K dengan tingkat utilisasi 0%. Kapal H merupakan kapal dengan tingkat utilisasi sebesar 83.3%. Berdasarkan grafik tersebut dapat diketahui bahwa utilisasi masing – masing kapal besarnya berbeda – beda, oleh karena itu maka diperlukan upaya untuk menyeragamkan utilisasi masing – masing kapal dengan menambahkan konstrain batas minimal dan maksimal utilisasi kapal. Dengan adanya dua kapal dengan utilisasi 0%, hal tersebut menandakan bahwa dari 13 kapal yang disewa hanya 11 kapal yang digunakan untuk mengirimkan semen ke *packing plant*. Dari 11 kapal yang digunakan, terdapat beberapa kapal yang tingkat utilitasinya di bawah 50%.

6.1.2 Analisis Hasil Algoritma SIM – A Skenario 2

Algoritma SIM – A skenario 2 digunakan untuk menyelesaikan permasalahan penjadwalan dan penugasan kapal dimana waktu mulai proses *loading* di depot dapat dilakukan dalam rentang 24 jam penuh yaitu jam 0:00 – 24:00 sedangkan proses *unloading* di *port* bisa dimulai dari jam 08:00 – 18:00. Artinya jika kapal datang di *port* sebelum pukul 8 maka kapal harus menunggu sampai waktu telah menunjukkan jam 8. Jika kapal di tiba di *port* tujua pada periode t lewat dari jam 18:00 maka kapal baru bisa melakukan pada periode $t + 1$ paling awal jam 8 pagi. Dengan menggunakan parameter *input* yang sama seperti halnya pada skenario 1, dari 10 kali percobaan *running* algoritma SIM – A skenario 2 mampu menyelesaikan permasalahan *Ship Scheduling and Assignment*

dengan total biaya minimum sebesar Rp 635,380,394,107.29 yang didapatkan pada percobaan ke - 4. Adapun solusi penjadwalan dan penugasan untuk masing – masing kapal ditunjukkan pada tabel 5.15 sedangkan informasi untuk masing – masing silo *port* tujuan ditunjukkan pada tabel 5.17 sampai 5.23. Berdasarkan tabel 5.15 dapat diketahui bahwa selama 30 hari horison perencanaan, di masing – masing periode jumlah kapal yang dijadwalkan maksimal sebanyak 2 unit untuk 2 *port* yang kritis. Artinya, algoritma SIM – A yang dikembangkan untuk skenario 2 telah memenuhi konstrain yang dipertimbangkan.

Berdasarkan total biaya dari 10 kali percobaan *running* skenario 2 yang ditunjukkan oleh Gambar 5.20 dapat diketahui bahwa perubahan total biaya yang dikeluarkan dari percobaan 1 sampai ke 10 memiliki tingkat fluktuasi yang cukup tinggi dikarenakan solusi yang didapatkan dari percobaan 1 sampai ke 10. Dalam *running* algoritma jumlah populasi yang dibangkitkan sebanyak 1000 populasi, artinya pada tahap inialisasi dibangkitkan 1000 solusi yang kemudian diurutkan dari solusi yang memberikan total biaya terendah sampai terbesar. Dimana setiap hari penjadwalan dari 1000 solusi yang dibangkitkan terdapat 2 kapal yang ditugaskan yang dipilih secara *random* berdasarkan matriks probabilitas transisi dimana setiap kapal yang fleksibel memiliki peluang yang sama untuk terpilih. Sebanyak 100 solusi dari 1000 solusi yang dibangkitkan akan mengalami proses elitisme pada iterasi berikutnya sedangkan 900 solusi yang lainnya dibangkitkan secara *random* sesuai dengan urutan proses yang ada. Dimana iterasi akan berhenti ketika jumlah iterasi maksimum tercapai yaitu sebanyak 100 iterasi. Pemilihan kapal yang dilakukan secara *random* berpengaruh terhadap terjadinya jumlah *stock out* di masing – masing *port* tujuan selama horizon perencanaan yang ditetapkan berbeda – beda di setiap *running* percobaan. Dengan besaran *penalty cost* sebesar Rp. 10.000.000,00 per ton *stock out* akan mengakibatkan total biaya yang dihabiskan juga berbeda – beda di setiap solusi percobaan dan perbedaanya relatif besar antar *running* percobaan.

Berdasarkan Gambar 5.21 tentang utilisasi masing – masing kapal berdasarkan solusi optimum skenario 2 dapat diketahui bahwa terdapat 2 kapal yang sama sekali tidak digunakan untuk mengirimkan semen ke *packing plant* selama horizon perencanaan yang ditetapkan yaitu kapal C dan M dengan tingkat

utilisasi sebesar 0%. Kapal G merupakan kapal dengan tingkat utilisasi terbesar dibandingkan kapal lainnya yaitu dengan utilisasi sebesar 96.7%. Berdasarkan gambar tersebut juga dapat diketahui terdapat beberapa kapal dengan tingkat utilisasi di bawah 50% yaitu kapal B, D, F, I, K, dan L. Berdasarkan grafik tersebut juga diketahui bahwa *gap* utilisasi antara kapal dengan utilisasi terbesar dan terendah sangat besar. Oleh karena itu diperlukan upaya untuk menyeragamkan utilisasi masing – masing kapal dengan menambah batasan utilisasi minimum dan maksimum kapal. Dari 13 kapal yang disewa untuk mengirimkan semen curah ke *packing plant* hanya digunakan 11 kapal. Artinya, ada dua kapal yang sebenarnya tidak perlu disewa untuk mengirimkan semen curah karena kapal lain masih bisa digunakan untuk mengirimkan semen tersebut ke *packing plant*. Oleh karena itu, penentuan jumlah kapal yang perlu di sewa perlu dijadikan pertimbangan dalam penjadwalan dan penugasan kapal.

6.1.3 Perbandingan Hasil Skenario 1 dan Skenario 2

Berdasarkan Gambar 5. 5 Grafik Perbandingan Total Biaya Skenario 1 dan 2 dapat diketahui bahwa berdasarkan 10 kali percobaan untuk skenario 1 dan skenario 2, keduanya menghasilkan total biaya yang cukup berbeda signifikan dimana total biaya yang dikeluarkan oleh skenario 2 relatif lebih sedikit dibandingkan skenario 1 hal itu terjadi karena fleksibilitas waktu mulai untuk melakukan *loading* di depot pada skenario 2 [0:00 – 24:00] lebih lebar dibandingkan skenario 1 [08:00 – 18:00] sehingga hal itu berdampak pada waktu mulai *loading* yang bisa kapanpun dan berakibat pada kecepatan pelayanan kapal sampai ke *port* tujuan yang mengakibatkan kemungkinan adanya *stock out* lebih sedikit dibandingkan skenario 1 (kondisi eksisting).

6.2 Rekomendasi

Berdasarkan analisis hasil yang telah dilakukan, maka ada beberapa rekomendasi yang bisa diajukan ke PT. X mengenai penjadwalan dan penugasan kapal untuk mengirimkan semen curah ke masing – masing *packing plant*

1. Penyelesaian masalah penjadwalan dan penugasan kapal yang mempertimbangkan *minimum requirement draft* dapat diselesaikan dengan menggunakan algoritma SIM – A yang telah dikembangkan.
2. Berdasarkan kondisi operasional penjadwalan dan penugasan kapal yang ada di PT. X, skenario 1 mampu meniru untuk digunakan menyelesaikan permasalahan tersebut.
3. Berdasarkan hasil solusi optimum skenario 1 dan 2, dari 13 kapal yang disewa untuk mengirimkan semen curah ke *packing plant*, jumlah kapal yang digunakan hanya 11 kapal, sehingga penentuan jumlah kapal yang harus disewa perlu dikaji ulang supaya biaya yang dihabiskan bisa minimum.
4. Jika ingin memperkecil total biaya pengiriman semen curah dari PT. X menuju masing – masing *packing plant* maka skenario 2 bisa dijadikan alternatif mekanisme penjadwalan dan penugasan kapal di PT. X.

BAB 7

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dari hasil penelitian yang telah dilakukan, dapat ditarik beberapa kesimpulan yaitu:

1. Telah didapatkan algoritma *Simple Iterative Mutation Algorithm* (SIM – A) untuk menyelesaikan permasalahan *Ship Scheduling and Assignment* yang mempertimbangkan *minimum requirement draft* kapal untuk bisa singgah di *port*.
2. Algoritma SIM – A yang dikembangkan mampu menyelesaikan permasalahan *Ship Scheduling and Assignment* pada distribusi semen curah di PT. X.
3. Dengan menggunakan dengan parameter jumlah populasi yang dibangkitkan sebesar (N) 1000 solusi, rasio sampel elit yang dipilih (ρ) sebesar 0.1, fraksi *safety stock* sebesar 0.05, dan jumlah iterasi maksimum yang digunakan sebanyak 100 iterasi dengan 10 kali percobaan didapatkan total biaya optimum pada skenario 1 (kondisi eksisting) yang dikeluarkan sebesar Rp. 635,380,394,107.286.
4. Dari dua skenario algoritma SIM – A yang dikembangkan dengan parameter *input* yang sama, skenario 2 cenderung menghasilkan nilai yang lebih baik dibanding skenario 1 karena skenario 2 memiliki fleksibilitas yang lebih lebar untuk memulai waktu *loading* di depot.

6.2 Saran

Adapun saran yang dapat diberikan untuk penelitian selanjutnya yaitu sebagai berikut.

1. Algoritma SIM – A dapat dikembangkan dengan memberikan batasan *stock out* tidak boleh terjadi pada silo *packing plant*.
2. Algoritma SIM – A bisa dikembangkan dengan membuat besarnya *consumption rate* per hari bersifat stokastik dengan distribusi permintaan tertentu.
3. Penentuan jumlah optimum kapal yang harus digunakan (disewa) dapat dijadikan skenario dalam algoritma SIM – A.
4. Algoritma SIM – A dapat dikembangkan dengan memberikan batasan minimum dan maksimum utilisasi kapal untuk menyeragamkan penggunaan kapal selama horizon perencanaan yang ditetapkan.
5. Penjadwalan di setiap periode tidak dibatasi berdasarkan jumlah maksimum kapal yang diijinkan untuk melakukan *loading* di *port*.

DAFTAR PUSTAKA

- Al-Khayyal, F., & Hwang, S. (2005). *Inventory Constraints Maritime Routing and Scheduling for Multi-Commodity Liquid Bulk*. Ph.D. Thesis. School of Industrial and Systems Engineering, Georgia Institut of Technology.
- Almeida, R., (2013). *Chinese Shipyard Win Order for Dozen Deltamarin-Designed Bulk Carriers*. <http://gcaptain.com/chinese-shipyards-orders-dozen/>.
- Almeida, R., (2013). *Pioneer Marine Launches With A Newbuild Order for 4 Bulk Carriers*. <http://gcaptain.com/pioneer-marine-launches-pankaj-khanna/>.
- Andersson, H., Duesund, J. M., Fagerholt, K. (2011). *Ship Routing and Scheduling With Cargo Coupling and Synchronization Constraints*. *Computers & Industrial Engineering* (2011), 1107-1116.
- Bronmo, G., Christiansen, M., Nygreen, B. (2006). *Ship Routing and Scheduling With Flexible Cargo Sizes*. *Journal of the operational research society* (2007) 58, 1167-1177.
- Budiman, M. A., (2012). *Simple Iterative Mutation Algorithm for The Truck Scheduling Problem in A Cross Docking Environment*. Master Thesis. Surabaya : Institut Teknologi Sepuluh Nopember.
- Chopra, S. & Meindl, P., (2007). *Supply Chain Management : Strategy, Planning, and Operation (3th ed.)*. Springer.
- Christiansen, M., Fagerholt, K., Ronen, D., (2003). *Ship Routing and Scheduling : Status and Perspectives*. *Transportation Science* (2004), 38 pp.1-18.
- Christiansen, M., Fagerholt, K., Nygreen, B., Ronen, D., (2007). *Chapter 4 : Maritime Transportation*. In: C. Barhat & G. Laporte (Eds.), *Handbook in OR & MS* (Vol. 14, pp.189-284). Elsevier B. V.
- Damayanti, R., (2013). *Pengembangan Algoritma Hybrid Cross Entropy – Genetic Algorithm dalam Penyelesaian Multi-Product Inventory Ship Routing Problem dengan Heterogeneous Fleet*. Tugas Akhir. Surabaya : Institut Teknologi Sepuluh Nopember.

- Madasari, W. R., 2012. *Analisis Biaya Distribusi dan Transportasi Untuk Jaringan Distribusi Semen Dengan Adanya Packing Plant (Studi Kasus: PT. Semen Gresik (Persero), Tbk.)*. Tugas Akhir. Surabaya : Institut Teknologi Sepuluh Nopember.
- Renspandy, R. P., (2013). *Penjadwalan Dispatching Truck Dari Packer Di Pabrik Semen Indonesia Tuban*. Tugas Akhir. Surabaya : Institut Teknologi Sepuluh Nopember.
- Santosa, B., & Willy, P. (2011). *Metoda Metaheuristik : Konsep dan Implementasi*. Surabaya : Guna Widya.

LAMPIRAN

```
%SIM-A Source Code
%Copyright : KETUT HENDRA HARIANTO 2510100090

function [Best_Cost, Best_Scheduled, Best_Solution,Best_STAT] =
SIM_A(nport, nkapal, nperiod, N, rho, fss,it_ttp_max)

disp ('OPTIMASI SHIP SCHEDULING AND ASSIGNMENT MENGGUNAKAN SIMPLE
ITERATIVE MUTATION ALGORITHM')

WTS = 0;
SUM_KAPAL=[];
ALL_SUM = [];
CWTSL = 0;
CWTS = 0;

OH=zeros(nport, nperiod+1, N);
IT=zeros(nport, nperiod+1, N);
CR=zeros(nport, nperiod+1, N);
SR=zeros(nport, nperiod+1, N);
IP=zeros(nport, nperiod+1, N);
SC=zeros(nport, nperiod+1, N);
SO=zeros(nport, nperiod+1, N);
SS=z
eros(nport,nperiod+1,N);
RSP=zeros(nport,nperiod+1,N); %Re-Shipment Point
STAT = zeros(nkapal, nperiod, N);
ASSIGNED = zeros(1, nperiod, N);
Scheduled = zeros(nkapal,nport*nperiod, N);

for i=1:nport
    OH(i, nperiod+1,:)= i;
    IT(i, nperiod+1,:)= i;
    CR(i, nperiod+1,:)= i;
    SR(i, nperiod+1,:)= i;
    IP(i, nperiod+1,:)= i;
    SC(i, nperiod+1,:)= i;
    SO(i, nperiod+1,:)= i; %SO=stock out
    SS(i,nperiod+1,:)= i;
    RSP(i,nperiod+1,:)= i;
End

WTS = zeros(N,1);
Total_CWTS = zeros(N,1);
CWTS = zeros(N,1);
Total_CWTSL = zeros(N,1);
Total_Loading_Cost = zeros(N,1);
Total_Distribution_Cost = zeros(N,1);
Total_Unloading_Cost = zeros(N,1);
Total_CWTSU = zeros(N,1);
Total_PC = zeros(N,1);
Total_Traveling_Cost = zeros(N,1);
Total_Biaya_Sistem = zeros(N,1);
Total_SO = zeros(N,1);
```

```

Total_Charter_Cost = zeros(N,1);
Total_IC = zeros(N,1);

%Karakteristik Port
TWD %time windows depot (production port)
D %demand (consumption rate masing-masing port tujuan
LT %lead time dari depot menuju masing-masing port tujuan (dalam
satuan hari), include loading unloading
ST %sailing time dalam satuan jam
TW %time windows untuk port tujuan
KAPP %kapasitas pelabuhan port tujuan
BC %kedalaman (h) dermaga masing-masing port tujuan

%Parameter Biaya
LCK %loading cost kapal
ULCK %unloading cost kapal
RDC %Rate Distribution Cost
PC %penalty cost akibat adanya stock out per ton
PCI %Rate Waktu Tunggu Sebelum Loading per jam
RWTSU %Rate Waktu Tunggu Sebelum Unloading per jam
RWTS %Rate waktu tunggu silo per jam
RSK %Rate sewa kapal per hari

%Karakteristik Kapal
KAPK %kapasitas kapal
MRD %minimum requirement draft kapal
LR %loading rate (kecepatan loading) kapal
UR %unloading rate

start = cputime;

%Generate Feasibilitas Kunjungan
FK=zeros(nport, nkapal);
for i=1:nport
    for j = 1:nkapal
        if BC(i) >= MRD(j) && KAPK(j)<= KAPP(i)
            FK(i,j)=1;
        end
    end
end

%Set OnHand Awal di msg2 port
OH_0

%Inisiasi, set period=1
period = 1;

WSA = zeros(nkapal, 1,N);
for a = 1:N
    WSA(:,1:a) = WSA(:,1:a) + TWD(1,1);
end
for port = 1:nport
    for a = 1:N
        OH(port,1,a)= OH_0(port,1);
    end
end

```



```

STAT = STAT+1;
rn = zeros(N,1);

%=====Pembangkitan Solusi=====
    for period = 1:nperiod
        for a = 1:N
            period;
            PMS(a) = 1;
            CR(:,period,a)= D(:);
            Solusi = a;

for port = 1:nport
%Hitung Safety Stock
SS(port, period,a) =(fss*KAPP(port)); % SS = fss% fraksi safety
stock dari kapasitas packing plant
RSP(port,period,a)=(D(port)*LT(port)+ SS(port,period,a));
End

if period > 1
for port = 1:nport
    if OH(port,period-1,a) + SR(port,period,a)-D(port) > 0
        OH(port,period,a)= OH(port,period-1,a)+ SR(port, period,a)-
        D(port);
        SO(port,period,a)= 0;
        if OH(port,period,a) > KAPP(port)
            IC(a)=(OH(port,period,a)- KAPP(port))* PCI;
            Total_IC (a) = Total_IC(a)+ IC(a);
        end
    else
        OH(port, period,a)=0;
        SO(port, period,a)= - (OH(port, period-1,a) + SR(port,
        period,a)- D(port));
        Total_SO (a) = Total_SO (a) + SO(port, period,a);
        Total_PC (a) = Total_SO (a) * PC; %Total Penalty Cost
    end
end
end

%Hitung Stock Criticality
SC(:,period,a)=(OH(:,period,a)+IT(:,period,a))./(D.*LT);
SC_NOW=sortrows([SC(:,nperiod+1,a) SC(:,period,a)],2);

for i = 1:nport
    for nomorport = 1:nport
        if SC_NOW(i,1)== nomorport
            break
        end
    end
end
DET_KAPAL=zeros(1,17);
Kapal_Tersedia = [];

if period == 1
period_check = period;
else

```

```

period_check = period-1;
end

if IP(nomorport,period_check,a) <= RSP(nomorport,period,a)
    porttujuan=SC_NOW(i,1);
    for j = 1: nkapal
        if FK(porttujuan, j) == 1 && STAT(j, period,a) == 1
            Kapal_Tersedia = [Kapal_Tersedia j];
        end
    end
    if size(Kapal_Tersedia) ~= [0 0]
        [jmlrow, jumlah_Kapal_Tersedia]=size(Kapal_Tersedia);
        Random_Kapal=[];
        Prob_Random_Kapal=[];
        for k = 1:jumlah_Kapal_Tersedia
            Random_Kapal=[Random_Kapal k];
            Prob_Random_Kapal=[Prob_Random_Kapal k/jumlah_Kapal_Tersedia];
        end
        Random_Kapal=[Random_Kapal; Kapal_Tersedia; Prob_Random_Kapal];

rn(a,1)=rand;

        for d= 1:jumlah_Kapal_Tersedia
            if rn(a,1) <= Random_Kapal(3,d)
                break
            end
        end
        Kapal_Dipilih = Random_Kapal(2,d);

        %Mulai mencatat aktivitas kapal
        PMS(a) = period;

        r_loading(a) = rand();
        if r_loading(a) <= 0.72857
            WTSL(a) = max([14.7637 + 6.7152.*randn() 0]); %WTSL = Waktu Tunggu
            Sebelum Loading
            CWTSL(a) = WTSL(a) * RWTSL(1,Kapal_Dipilih);
        else
            WTSL(a) = 0;
            CWTSL(a) = 0;
        end
        Total_CWTSL (a) = Total_CWTSL (a) + CWTSL(a);

        if WSA (Kapal_Dipilih, 1,a) > TWD(1,2)
            WML(a) = 24 + TWD (1,1)+ WTSL(a);
        elseif WSA (Kapal_Dipilih, 1,a) < TWD (1,1)
            WML(a) = TWD (1,1)+ WTSL(a);
        else
            WML(a) = WSA (Kapal_Dipilih, 1,a)+ WTSL(a);
        end

        DL(a) = KAPK(Kapal_Dipilih)/LR(Kapal_Dipilih);
        Loading_Cost(a) = LCK(Kapal_Dipilih);
        Total_Loading_Cost(a) = Total_Loading_Cost (a) + Loading_Cost(a);
        WSL(a) = WML(a) + DL(a);
        DS(a) = ST(porttujuan);
        Distribution_Cost(a) = DS(a) * RDC(Kapal_Dipilih,porttujuan);

```

```

Total_Distribution_Cost(a)=Total_Distribution_Cost(a)+
Distribution_Cost(a);

WST(a) = WSL(a) + DS(a);
r_unloading(a) = rand();
if r_unloading(a) <= 0.90
WTSU(a) = max ([47.1402 + 22.73.*randn() 0]);
CWTSU (a) = WTSU(a) * RWTSU(1,Kapal_Dipilih);
else
WTSU(a) = 0;
CWTSU(a) = 0;
end
Total_CWTSU (a)= Total_CWTSU (a)+ CWTSU(a);
periode = floor((WST(a)+WTSU(a))/24);

if (PMS(a) + periode - 1) > nperiod
WTS(a) = 0;
CWTS(a) = 0;
else
%Hitung Empty Space Standard (ESS)
ESS=(UR(Kapal_Dipilih)-D(porttujuan)/24)*
KAPK(Kapal_Dipilih)/UR(Kapal_Dipilih);
if periode == 1
ESR(a)= KAPP(porttujuan)-OH(porttujuan,periode,a);%Empty Space
Real
else
ESR(a) = KAPP(porttujuan) - OH(porttujuan,periode-1,a);
end
if ESR(a) < 0
ESR (a)= 0;
end
if ESR(a) < ESS
WTS(a) = (ESS - ESR(a))/(D(porttujuan)/24);
CWTS(a) = RWTS(Kapal_Dipilih) * WTS(a);
end
Total_CWTS (a)= Total_CWTS (a)+ CWTS(a);
end

if mod(WST(a), 24) > TW(porttujuan, 2)
WMU(a) = WST(a) + 24 + WTSU(a) + WTS(a) + TW(porttujuan, 1) -
mod(WST(a), 24);
elseif mod(WST, 24) < TW(porttujuan, 1)
WMU(a) = WST(a) + WTSU(a) + WTS(a)+TW(porttujuan,1)-mod(WST(a),
24);
else
WMU(a) = WST(a) + WTSU(a) + WTS(a);
end

DU(a) = KAPK(Kapal_Dipilih)/UR(Kapal_Dipilih);
Unloading_Cost(a) = ULCK(Kapal_Dipilih);
Total_Unloading_Cost(a)=Total_Unloading_Cost(a)+
Unloading_Cost(a);
WSU(a) = WMU(a) + DU (a);
DSB(a) = ST (porttujuan);
Traveling_Cost(a)=DSB(a)*
RDC(Kapal_Dipilih,porttujuan);%Traveling_Cost=biaya yang
dikeluarkan dari port tujuan menuju depot

```

```

Total_Traveling_Cost(a)=Total_Traveling_Cost(a)+
Traveling_Cost(a);
WSD(a) = WSU(a) + DSB(a);
PAS(a) = PMS (a) + floor(WSD(a)/24);
JAS(a) = mod(WSD(a), 24);

%-----
DET_KAPAL = [PMS(a) WSA(Kapal_Dipilih, 1,a) WTSL(a) WML(a) DL(a)
WSL(a) DS(a) WST(a) WTSU(a) WTS(a) WMU(a) DU(a) WSU(a) DSB(a)
WSD(a) PAS(a) JAS(a)];
SUM_KAPAL = [SUM_KAPAL; Kapal_Dipilih DET_KAPAL a];

%Set WSA Kapal
WSA(Kapal_Dipilih, 1,a) = JAS(a);
%Update status kapal
STAT(Kapal_Dipilih, period, a)=2;
STAT(Kapal_Dipilih, PMS(a)+1:PAS(a)-1, a)=0;
%Tentukan periode mulai loading (PML)
PML(a) = PMS(a) + floor(WML(a)/24);
%Tentukan periode unloading selesai (PSU)
PSU(a) = PMS(a) + floor(WSU(a)/24);
%Update intransit inventory dan onhand inventory
if PSU(a) <= nperiod
IT(porttujuan, period:PSU(a)-1,a) = IT(porttujuan, period:PSU(a)-
1,a) + KAPK(Kapal_Dipilih);
SR(porttujuan, PSU(a), a)= SR(porttujuan, PSU(a), a)+
KAPK(Kapal_Dipilih);
else
IT(porttujuan, period:nperiod,a) = IT(porttujuan,
period:nperiod,a) + KAPK(Kapal_Dipilih);
end
if period ==1
IT(porttujuan, period,a) = KAPK(Kapal_Dipilih);
end
Scheduled(Kapal_Dipilih,porttujuan+nport*(period-1),a) = 1;
ASSIGNED (1,period,a) = ASSIGNED (1,period,a)+1;
if ASSIGNED (1, period,a) == 2
break
end
end
end
end
IP(:,period,a) = OH(:, period,a) + IT (:, period,a);
Total_Charter_Cost (a) = nperiod*sum(RSK);
end
OH;
IT;
SR;
IP;
RSP;
SO;
Total_CWTSL (a) = Total_CWTSL (a);
Total>Loading_Cost (a) = Total>Loading_Cost (a);
Total>Distribution_Cost (a) = Total>Distribution_Cost (a);
Total>CWTSU (a) = Total>CWTSU (a);
Total_PC (a) = Total_PC (a);
Total_Unloading_Cost(a) = Total_Unloading_Cost(a);

```

```

Total_Traveling_Cost (a) = Total_Traveling_Cost (a);
Total_CWTS (a)= Total_CWTS (a);
Total_IC(a) = Total_IC(a);
end

%=====PERHITUNGAN FUNGSI TUJUAN=====
for i = 1:N
Total_Biaya_Sistem(i,1)=Total_CWTSL (i,1)+ Total_Loading_Cost
(i,1)+ Total_Distribution_Cost (i,1)+ Total_CWTSU (i,1)+ Total_PC
(i,1)+ Total_Traveling_Cost (i,1) + Total_CWTS (i,1) +
Total_Unloading_Cost(i,1) + Total_Charter_Cost(i,1)+ Total_IC(a);
end

%=====Iterasi Simple Iterative Mutation Algorithm=====
it_ttp = 0;
while it_ttp < it_ttp_max
it_ttp = it_ttp+1;

%=====PEMILIHAN SAMPLE ELIT=====
[Total_Biaya_Sistem_Sort I] = sort(Total_Biaya_Sistem);
JSE = ceil(rho*N);
for e = 1:JSE
OH_elite(:, :, e)= OH(:, :, I(e));
IT_elite(:, :, e)= IT(:, :, I(e));
CR_elite(:, :, e)= CR(:, :, I(e));
SR_elite(:, :, e)= SR(:, :, I(e));
RSP_elite(:, :, e)= RSP(:, :, I(e));
IP_elite(:, :, e)= IP(:, :, I(e));
SC_elite(:, :, e)= SC(:, :, I(e));
SO_elite(:, :, e)= SO(:, :, I(e));
STAT_elite(:, :, e) = STAT(:, :, I(e));
ASSIGNED_elite(1, :, e) = ASSIGNED(1, :, I(e));
Scheduled_elite(:, :, e) = Scheduled(:, :, I(e));
Total_CWTSL_elite (e,1)= Total_CWTSL(I(e),1);
Total_Loading_Cost_elite(e,1) = Total_Loading_Cost(I(e),1);
Total_Distribution_Cost_elite(e,1)=
Total_Distribution_Cost(I(e),1);
Total_Unloading_Cost_elite(e,1) = Total_Unloading_Cost(I(e),1);
Total_CWTSU_elite(e,1) = Total_CWTSU (I(e),1);
Total_PC_elite(e,1) = Total_PC(I(e),1);
Total_Traveling_Cost_elite (e,1)= Total_Traveling_Cost(I(e),1);
Total_SO_elite (e,1) = Total_SO(I(e),1);
Total_Charter_Cost_elite(e,1) = Total_Charter_Cost(I(e),1);
Total_Biaya_Sistem_elite(e,1) = Total_Biaya_Sistem(I(e),1);
end

%=====MUTASI=====
SUM_KAPAL=[];
ALL_SUM = [];
CWTSL = 0;
OH_Mutasi=zeros(nport, nperiod+1, N-JSE);
IT_Mutasi=zeros(nport, nperiod+1, N-JSE);
CR_Mutasi=zeros(nport, nperiod+1, N-JSE);
SR_Mutasi=zeros(nport, nperiod+1, N-JSE);
IP_Mutasi=zeros(nport, nperiod+1, N-JSE);
SC_Mutasi=zeros(nport, nperiod+1, N-JSE);
SO_Mutasi=zeros(nport, nperiod+1, N-JSE);

```

```

SS_Mutasi=zeros(nport, nperiod+1, N-JSE);
RSP_Mutasi=zeros(nport, nperiod+1, N-JSE);
STAT_Mutasi = zeros(nkapal, nperiod, N-JSE);
ASSIGNED_Mutasi = zeros(1, nperiod, N-JSE);
Scheduled_Mutasi = zeros(nkapal,nport*nperiod, N-JSE);

for i=1:nport
    OH_Mutasi(i, nperiod+1,:)= i;
    IT_Mutasi(i, nperiod+1,:)= i;
    CR_Mutasi(i, nperiod+1,:)= i;
    SR_Mutasi(i, nperiod+1,:)= i;
    IP_Mutasi(i, nperiod+1,:)= i;
    SC_Mutasi(i, nperiod+1,:)= i;
    SO_Mutasi(i, nperiod+1,:)= i; %SO=stock out
end

Total_CWTSL_Mutasi = zeros(N-JSE,1);
Total>Loading_Cost_Mutasi = zeros(N-JSE,1);
Total>Distribution_Cost_Mutasi = zeros(N-JSE,1);
Total>Unloading_Cost_Mutasi = zeros(N-JSE,1);
Total>CWTSU_Mutasi = zeros(N-JSE,1);
Total>PC_Mutasi = zeros(N-JSE,1);
Total>Traveling_Cost_Mutasi = zeros(N-JSE,1);
Total>Biaya_Sistem_Mutasi = zeros(N-JSE,1);
Total>SO_Mutasi = zeros(N-JSE,1);
Total>Charter_Cost_Mutasi = zeros(N-JSE,1);
Total>CWTS_Mutasi = zeros(N-JSE,1);
Total>IC_Mutasi = zeros(N-JSE,1);

%Inisiasi, set period=1
period = 1;

WSA = zeros(nkapal, 1,N);
for a = 1:N
    WSA(:,1,a) = WSA(:,1,a) + TWD(1,1);
end

for port = 1:nport
    for a = 1:N-JSE
        OH_Mutasi(port,1,a)= OH_0(port,1);
    end
end

STAT_Mutasi = STAT_Mutasi+1;
rn = zeros(N-JSE,1);
%Untuk onhand, period = period + 1

for period = 1:nperiod
    for a = 1:N-JSE
        period;
        PMS(a) = 1;
        Solusi = a;
        for port = 1:nport
            %Hitung Safety Stock
            SS_Mutasi(port, period,a) = (fss*KAPP(port));
        end
    end
end

```

```

RSP_Mutasi(port,period,a)=(D(port)*LT(port)+
SS_Mutasi(port,period,a));
end

if period > 1
for port = 1:nport
if OH_Mutasi(port,period-1,a) + SR_Mutasi(port, period,a)-
D(port)> 0
OH_Mutasi(port,period,a)=OH_Mutasi(port,period-
1,a)+SR_Mutasi(port, period,a)- D(port);
SO_Mutasi(port, period,a)= 0;
if OH_Mutasi(port, period,a) > KAPP(port)
IC(a) = (OH_Mutasi(port, period,a) - KAPP(port))* PCI;
Total_IC_Mutasi (a) = Total_IC_Mutasi(a)+ IC(a);
end
else
OH_Mutasi(port, period,a)=0;
SO_Mutasi(port,period,a)=- (OH_Mutasi(port,period-1,a)+
SR_Mutasi(port, period,a)- D(port));
Total_SO_Mutasi (a) = Total_SO_Mutasi (a) + SO_Mutasi(port,
period,a);
Total_PC_Mutasi (a) = Total_SO_Mutasi (a) * PC; %Total Penalty
Cost
end
end
end

%Hitung SC
SC_Mutasi(:,period,a)=(OH_Mutasi(:,period,a)+IT_Mutasi(:,period,a)
)./(D.*LT);
SC_NOW_Mutasi=sortrows([SC_Mutasi(:,nperiod+1,a)
SC_Mutasi(:,period,a)],2);
for i = 1:nport
DET_KAPAL=zeros(1,16);
Kapal_Tersedia = [];
porttujuan=SC_NOW_Mutasi(i,1);
if period == 1
period_check = period;
else
period_check = period-1;
end
if IP_Mutasi(nomorport,period_check,a)<=
RSP_Mutasi(nomorport,period,a)
porttujuan=SC_NOW_Mutasi(i,1);
for j = 1: nkapal
if FK(porttujuan, j) == 1 && STAT_Mutasi(j, period,a)== 1
Kapal_Tersedia = [Kapal_Tersedia j];
end
end
if size(Kapal_Tersedia)~= [0 0]
[jmlrow,jumlah_Kapal_Tersedia]=size(Kapal_Tersedia);
Random_Kapal=[];
Prob_Random_Kapal=[];
for k = 1:jumlah_Kapal_Tersedia
Random_Kapal=[Random_Kapal k];
end
Random_Kapal=[Random_Kapal; Kapal_Tersedia; Prob_Random_Kapal];

```

```

rn(a,1)=rand;
for d= 1:jumlah_Kapal_Tersedia
if rn(a,1) <= Random_Kapal(3,d)
break
end
end
Kapal_Dipilih = Random_Kapal(2,d);
%Mulai mencatat aktivitas kapal
PMS(a) = period;
r_loading(a) = rand();
if r_loading(a) <= 0.72857
WTSL(a) = max([14.7637 + 6.7152.*randn() 0]); %WTSL = Waktu Tunggu
Sebelum Loading
CWTSL(a) = WTSL(a) * RWTSL(1,Kapal_Dipilih);
else
WTSL(a) = 0;
CWTSL(a) = 0;
end
Total_CWTSL_Mutasi (a) = Total_CWTSL_Mutasi (a) + CWTSL(a);
if WSA (Kapal_Dipilih, 1,a)> TWD(1,2)
WML(a) = 24 + TWD (1,1)+ WTSL(a);
elseif WSA (Kapal_Dipilih, 1,a)< TWD (1,1)
WML(a)= TWD (1,1)+ WTSL(a);
else
WML(a) = WSA (Kapal_Dipilih, 1,a)+ WTSL(a);
end
DL(a) = KAPK(Kapal_Dipilih)/LR(Kapal_Dipilih);
Loading_Cost(a) = LCK(Kapal_Dipilih);
Total_Loading_Cost_Mutasi(a) = Total_Loading_Cost_Mutasi (a) +
Loading_Cost(a);
WSL(a) = WML(a) + DL(a);
DS(a) = ST(porttujuan);
Distribution_Cost(a) = DS(a) * RDC(Kapal_Dipilih,porttujuan);
Total_Distribution_Cost_Mutasi(a) = Total_Distribution_Cost_Mutasi
(a)+ Distribution_Cost(a);
WST(a) = WSL(a) + DS(a);
r_unloading(a) = rand();
if r_unloading(a) <= 0.90
WTSU(a) = max ([47.1402 + 22.73.*randn() 0]);
CWTSU(a) = WTSU(a) * RWTSU(1,Kapal_Dipilih);
else
WTSU(a) = 0;
CWTSU(a) = 0;
end
Total_CWTSU_Mutasi (a)= Total_CWTSU_Mutasi (a)+ CWTSU(a);
periode = floor((WST(a)+WTSU(a))/24);
if (PMS(a) + periode - 1) > nperiod
WTS(a) = 0;
CWTS(a) = 0;
else
%Hitung Empty Space Standard (ESS)
ESS=(UR(Kapal_Dipilih)-D(porttujuan)/24)*
KAPK(Kapal_Dipilih)/UR(Kapal_Dipilih);
if periode == 1
ESR(a) = KAPP(porttujuan) - OH(porttujuan,periode,a);%Empty Space
Real
else
ESR(a) = KAPP(porttujuan) - OH(porttujuan,periode-1,a);

```



```

end
if ESR(a) < 0
ESR(a) = 0;
end
if ESR(a) < ESS
WTS(a) = (ESS - ESR(a))/(D(porttujuan)/24);
CWTS(a) = RWTS(Kapal_Dipilih)* WTS(a);
end
Total_CWTS_Mutasi (a) = Total_CWTS_Mutasi(a) + CWTS(a);
end
if mod(WST(a), 24) > TW(porttujuan, 2)
WMU(a) = WST(a) + 24 + WTSU(a) + TW(porttujuan, 1) - mod(WST(a),
24)+ WTS(a);
elseif mod(WST(a), 24) < TW(porttujuan, 1)
WMU(a) = WST(a) + WTSU(a) + TW(porttujuan, 1) - mod(WST(a), 24)+
WTS(a);
else
WMU(a) = WST(a) + WTSU(a) + WTS(a);
end
DU(a) = KAPK(Kapal_Dipilih)/UR(Kapal_Dipilih);
Unloading_Cost(a) = ULCK(Kapal_Dipilih);
Total_Unloading_Cost_Mutasi (a) = Total_Unloading_Cost_Mutasi (a)+
Unloading_Cost(a);
WSU(a) = WMU(a) + DU(a);
DSB(a) = ST (porttujuan);
R_DC = RDC(Kapal_Dipilih,porttujuan);
Traveling_Cost(a)=DSB(a)*
RDC(Kapal_Dipilih,porttujuan);%Traveling_Cost=biaya yang
dikeluarkan dari port tujuan menuju depot
Total_Traveling_Cost_Mutasi (a) = Total_Traveling_Cost_Mutasi (a)
+ Traveling_Cost(a);
WSD(a) = WSU(a) + DSB(a);
PAS(a) = PMS(a) + floor(WSD(a)/24);
JAS(a) = mod(WSD(a), 24);
%-----
DET_KAPAL = [PMS(a) WSA(Kapal_Dipilih, 1,a) WTSL(a) WML(a) DL(a)
WSL(a) DS(a) WST(a) WTSU(a) WTS(a) WMU(a) DU(a) WSU(a) DSB(a)
WSD(a) PAS(a) JAS(a)];
SUM_KAPAL = [SUM_KAPAL; Kapal_Dipilih DET_KAPAL a];
%Set WSA Kapal
WSA(Kapal_Dipilih, 1,a) = JAS(a);
%Update status kapal
STAT_Mutasi(Kapal_Dipilih, period, a)=2;
STAT_Mutasi(Kapal_Dipilih, PMS(a)+1:PAS(a)-1, a)=0;
%Tentukan periode mulai loading (PML)
PML(a) = PMS(a) + floor(WML(a)/24);
%Tentukan periode unloading selesai (PSU)
PSU(a) = PMS(a) + floor(WSU(a)/24);
%Update intransit inventory dan onhand
if PSU <= nperiod
IT_Mutasi(porttujuan,period:PSU(a)-1,a)=IT_Mutasi(porttujuan,
period:PSU(a)-1,a) + KAPK(Kapal_Dipilih);
SR_Mutasi(porttujuan,PSU(a),a)= SR_Mutasi(porttujuan,PSU(a),a)+
KAPK(Kapal_Dipilih);
else
IT_Mutasi(porttujuan,period:nperiod,a)=IT_Mutasi(porttujuan,
period:nperiod,a) + KAPK(Kapal_Dipilih);

```

```

end
Scheduled_Mutasi(Kapal_Dipilih,porttujuan+nport*(period-1),a) = 1;
ASSIGNED_Mutasi (1,period,a) = ASSIGNED_Mutasi (1,period,a)+1;
if ASSIGNED_Mutasi (1, period,a) == 2
break
end
end
end
end
end
IP_Mutasi(:,period,a)=OH_Mutasi(:,period,a)+IT_Mutasi
(:,period,a);
Total_Charter_Cost_Mutasi (a) = nperiod*sum(RSK);
end
OH_Mutasi;
IT_Mutasi;
SR_Mutasi;
IP_Mutasi;
RSP_Mutasi;
SO_Mutasi;
Total_CWTSL_Mutasi (a) = Total_CWTSL_Mutasi (a);
Total_Loading_Cost_Mutasi (a) = Total_Loading_Cost_Mutasi (a);
Total_Distribution_Cost_Mutasi(a)=Total_Distribution_Cost_Mutasi
(a);
Total_CWTSU_Mutasi (a) = Total_CWTSU_Mutasi (a);
Total_PC_Mutasi (a) = Total_PC_Mutasi (a);
Total_Traveling_Cost_Mutasi (a) = Total_Traveling_Cost_Mutasi
(a);
Total_Unloading_Cost_Mutasi(a) = Total_Unloading_Cost_Mutasi(a);
Total_CWTS_Mutasi (a) = Total_CWTS_Mutasi (a);
Total_IC_Mutasi (a) = Total_IC_Mutasi(a);
end
for i = 1:N-JSE
Total_Biaya_Sistem_Mutasi(i,1)=Total_CWTSL_Mutasi(i,1)+
Total_Loading_Cost_Mutasi(i,1)+Total_Distribution_Cost_Mutasi
(i,1)+
Total_CWTSU_Mutasi(i,1)+Total_PC_Mutasi(i,1)+
Total_Traveling_Cost_Mutasi(i,1)+Total_CWTS_Mutasi(i,1)+
Total_Unloading_Cost_Mutasi(i,1)+Total_Charter_Cost_Mutasi(i,1)+
Total_IC_Mutasi(i,1);
end

%=====PENGGABUNGAN HASIL ELITISME DAN MUTASI=====
Total_Biaya_Sistem=[Total_Biaya_Sistem_elite;
Total_Biaya_Sistem_Mutasi];

for i = 1:N
if i <= JSE;
OH(:, :, i) = OH_elite(:, :, i);
IT(:, :, i) = IT_elite(:, :, i);
CR(:, :, i) = CR_elite(:, :, i);
SR(:, :, i) = SR_elite(:, :, i);
RSP(:, :, i) = RSP_elite(:, :, i);
IP(:, :, i) = IP_elite(:, :, i);
SC(:, :, i) = SC_elite(:, :, i);
SO(:, :, i) = SO_elite(:, :, i);
STAT(:, :, i) = STAT_elite(:, :, i);
ASSIGNED(:, :, i) = ASSIGNED_elite(:, :, i);
Scheduled(:, :, i) = Scheduled_elite(:, :, i);

```

```

else
OH(:, :, i) = OH_Mutasi(:, :, i-JSE);
IT(:, :, i) = IT_Mutasi(:, :, i-JSE);
CR(:, :, i) = CR_Mutasi(:, :, i-JSE);
SR(:, :, i) = SR_Mutasi(:, :, i-JSE);
RSP(:, :, i) = RSP_Mutasi(:, :, i-JSE);
IP(:, :, i) = IP_Mutasi(:, :, i-JSE);
SC(:, :, i) = SC_Mutasi(:, :, i-JSE);
SO(:, :, i) = SO_Mutasi(:, :, i-JSE);
%STAT(:, :, i) = STAT_Mutasi(:, :, i-JSE);
ASSIGNED(:, :, i) = ASSIGNED_Mutasi(:, :, i-JSE);
Scheduled(:, :, i) = Scheduled_Mutasi(:, :, i-JSE);
end
end
end

%=====PENAMPILAIN HASIL=====

Best_Cost = Total_Biaya_Sistem_elite(1,1);
Best_OH = OH_elite(:, :, 1);
Best_IT = IT_elite(:, :, 1);
Best_IP = IP_elite(:, :, 1);
Best_SR = SR_elite(:, :, 1);
Best_CR = CR_elite(:, :, 1);
Best_RSP= RSP_elite(:, :, 1);
Best_SO = SO_elite(:, :, 1);
Best_SC = SC_elite(:, :, 1);
Best_Scheduled = Scheduled_elite(:, :, 1);
Best_STAT = STAT_elite(:, :, 1);

Best_Solution = [];
for x = 1:nport
    for y = 1:nperiod
        Best_Solution_0 = [Best_OH(x, :); Best_IT(x, :);
Best_IP(x, :); Best_SR(x, :); Best_CR(x, :); Best_RSP(x, :);
Best_SO(x, :); Best_SC(x, :)];
    end
    Best_Solution = [Best_Solution; Best_Solution_0];
end

finish = cputime;
Waktu_Komputasi = finish - start;

```

(halaman ini sengaja dikosongkan)

BIOGRAFI PENULIS



Ketut Hendra Harianto lahir di Banyuwatis, Singaraja – Bali pada tanggal 06 Februari 1992. Penulis menempuh pendidikan di TK Citra Bangsa, SDN 1 Banyuwatis, SMP N 2 Banjar, SMA N 1 Banjar. Setelah lulus SMA, penulis berhasil diterima di Jurusan Teknik Industri – Institut Teknologi Sepuluh Nopember (ITS – Surabaya) melalui jalur PMDK Beasiswa pada tahun 2010.

Selama menjadi mahasiswa, penulis aktif sebagai asisten Laboratorium Logistics and Supply Chain Management (Lab. LSCM) di departemen Knowledge Development. Penulis pernah beberapa kali ditugaskan sebagai asisten mata kuliah seperti Statistika Industri 1, Perencanaan dan Pengendalian Produksi (PPC), Manajemen Logistik, dan Perancangan Sistem Industri (PSI). Di luar jurusan, penulis juga pernah menjadi panitia di ‘Hari Budaya ITS’ dan ‘Interval ITS’. Pada April tahun 2014, penulis berhasil menjadi Semifinalis dalam Lomba Data Analysis Competition (DAC) Tingkat Nasional yang diadakan oleh Jurusan Statistika – ITS. Pada bulan Mei 2014, penulis berhasil meraih Juara 1 Tingkat Nasional dalam lomba ‘Industrial Engineering on Action (IDEA)’ yang diadakan oleh Universitas Andalas – Padang, Sumatera Barat.

Dalam rangka pengaplikasian ilmu yang diperoleh, penulis berkesempatan untuk melaksanakan kerja praktik di PT. Kamadjaja Logistics bagian Regional Sourcing Export pada bulan Juni hingga Juli 2013 dengan bidang penelitian *work sampling*, *forklift productivity*, dan *inventory management*. Penulis dapat dihubungi via e-mail di hendrarianro@ymail.com.

(halaman ini sengaja dikosongkan)