



TUGAS AKHIR - TF 181801

**OPTIMISASI KONSUMSI ENERGI PADA PROSES
KOLOM DEPROPANIZER BERBASIS *NEURAL
NETWORK - PARTICLE SWARM OPTIMIZATION*
(NN-PSO)**

IAN HAIKAL AMIR AKBAR
NRP. 0231164000021

Dosen Pembimbing:
Prof. Dr. Ir. Aulia Siti Aisjah, M.T.
Dr. Ir. Syamsul Arifin, M.T.

Departemen Teknik Fisika
Fakultas Teknologi Industri dan Rekayasa Sistem
Institut Teknologi Sepuluh Nopember
2020



TUGAS AKHIR - TF 181801

**OPTIMISASI KONSUMSI ENERGI PADA PROSES
KOLOM DEPROPANIZER BERBASIS *NEURAL
NETWORK - PARTICLE SWARM OPTIMIZATION*
(*NN-PSO*)**

**IAN HAIKAL AMIR AKBAR
NRP. 0231164000021**

Dosen Pembimbing 1 :
Prof. Dr. Ir. Aulia Siti Aisjah, M.T.

Dosen Pembimbing 2 :
Dr. Ir. Syamsul Arifin, M.T.

Departemen Teknik Fisika
Fakultas Teknologi Industri dan Rekayasa Sistem
Institut Teknologi Sepuluh Nopember
2020

Halaman ini sengaja dikosongkan



FINAL PROJECT - TF 181801

***ENERGY CONSUMPTION OPTIMIZATION IN
DEPROPANIZER COLUMN PROCESS BASED ON
NEURAL NETWORK - PARTICLE SWARM
OPTIMIZATION (NN-PSO)***

IAN HAIKAL AMIR AKBAR
NRP. 0231164000021

Supervisors 1:
Prof. Dr. Ir. Aulia Siti Aisjah, M.T.

Supervisors 2:
Dr. Ir. Syamsul Arifin, M.T.

*Department of Engineering Physics
Faculty of Industrial Technology and System Engineering
Institut Teknologi Sepuluh Nopember
2020*

Halaman ini sengaja dikosongkan

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini.

Nama : Ian Haikal Amir Akbar
NRP : 02311640000021
Departemen / Prodi : Teknik Fisika / S1 Teknik Fisika
Fakultas : Fakultas Teknologi - Industri dan Rekayasa Sistem
(FT-IRS)
Perguruan Tinggi : Institut Teknologi Sepuluh Nopember (ITS)

Dengan ini menyatakan bahwa Tugas Akhir dengan judul “OPTIMISASI KONSUMSI ENERGI PADA PROSES KOLOM DEPROPANIZER BERBASIS *NEURAL NETWORK – PARTICLE SWARM OPTIMIZATION* (NN-PSO)” adalah benar karya saya sendiri dan bukan plagiat dari karya orang lain. Apabila di kemudian hari terbukti terdapat plagiat pada Tugas Akhir ini, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenarnya-benarnya.

Surabaya, 26 Juli 2020

Yang membuat pernyataan,



Ian Haikal Amir Akbar
NRP. 02311640000021

Halaman ini sengaja dikosongkan

**LEMBAR PENGESAHAN
TUGAS AKHIR**

**OPTIMISASI KOMSUMSI ENERGI PADA PROSES
KOLOM DEPROPANIZER BERBASIS NEURAL
NETWORK – PARTICLE SWARM OPTIMIZATION
(NN-PSO)**

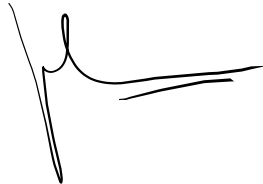
Oleh:



IAN HAIKAL AMIR AKBAR
NRP. 0231164000021

Surabaya,

**Menyetujui,
Pembimbing I**



Prof. Dr. Ir. Aulia Siti Aisjah, M.T.
NIP. 19660116 198903 2 001

**Menyetujui,
Pembimbing II**



Dr. Ir. Syamsul Arifin, M.T.
NIP. 19630907 198903 1 004

**Mengetahui,
Kepala Departemen
Teknik Fisika FT-IRS ITS**



Dr. Suvanto, S.T., M.T.
NIP. 19711113 199512 1 002

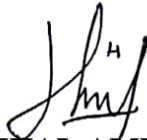
Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN
OPTIMISASI KONSUMSI ENERGI PADA PROSES KOLOM
DEPROPANIZER BERBASIS NEURAL NETWORK – PARTICLE
SWARM OPTIMIZATION (NN-PSO)

TUGAS AKHIR

Diajukan untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Program Studi S-1 Departemen Teknik Fisika
Fakultas Teknologi Industri dan Rekayasa Sistem (FT-IRS)
Institut Teknologi Sepuluh Nopember

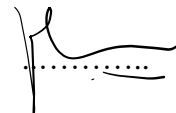


Oleh:



IAN HAIKAL AMIR AKBAR

NRP. 02311640000021

Disetujui oleh Tim Penguji Tugas Akhir:

- | | | |
|---|--|-----------------|
| 1. Prof. Dr. Ir. Aulia Siti Aisjah, M.T. |  | (Pembimbing I) |
| 2. Dr. Ir. Syamsul Arifin, M.T. |  | (Pembimbing II) |
| 3. Hendra Cordova, S.T., M.T. |  | (Ketua Penguji) |
| 4. Ir. Harsono Hadi, M.T., Ph.D. |  | (Penguji I) |
| 5. Dr. rer. nat. Ruri Agung Wahyuono, S.T.,
M.T. |  | (Penguji II) |

SURABAYA

2020

Halaman ini sengaja dikosongkan

OPTIMISASI KONSUMSI ENERGI PADA PROSES KOLOM DEPROPANIZER BERBASIS NEURAL NETWORK – PARTICLE SWARM OPTIMIZATION (NN-PSO)

Nama : Ian Haikal Amir Akbar
NRP : 02311640000021
Departemen : Teknik Fisika FT-IRS ITS
Dosen Pembimbing : Prof. Dr. Ir. Aulia Siti Aisjah, M.T.
Dr. Ir. Syamsul Arifin, M.T.

ABSTRAK

Kolom Distilasi merupakan plant yang digunakan untuk memisahkan komponen senyawa *hydrocarbon* pada *crude oil* dan *natural gas* berdasarkan nilai volatilitasnya. Penggunaan energi yang tinggi pada equipment *Condenser* dan *Reboiler* menuntut adanya optimisasi konsumsi energi kolom distilasi untuk menekan biaya utilitas *cooling water* dan *steam*. Metode optimisasi dilakukan dengan cara mencari nilai variabel operasional pada kolom distilasi tersebut yang menghasilkan konsumsi energi minimum. Proses pencarian nilai variabel operasional kolom distilasi dilakukan dengan menggunakan metode *Stochastic Algorithm* yaitu *Particle Swarm Optimization* (PSO). Kolom distilasi yang bersifat kompleks dan non linier dimodelkan dengan *Neural Network* (NN), dan keluaran dari model NN digunakan untuk penentuan fungsi obyektif pada algoritma PSO. Variabel operasional dari kolom depropanizer yang akan dilakukan optimisasi yaitu *Feed – Molar Flow* (F), *Feed – Temperature* (Tf), *Condenser – Pressure* (Pc), *Reboiler – Pressure* (Pr). Performansi arsitektur *Neural Network Feed Forward - Backpropagation* dengan algoritma training *Lavenberg Marquart* pada jumlah *neuron* 19 menghasilkan nilai RMSE Qc, RSME Qr, dan RMSE total berturut – turut sebesar $5,541 \times 10^{-4}$; $6,310 \times 10^{-4}$; $8,398 \times 10^{-4}$. Metode optimisasi konsumsi energi menggunakan algoritma PSO menghasilkan penghematan konsumsi energi pada *Reboiler* dengan nilai efisiensi Er sebesar 29,27 %, sedangkan pada *equipment Condenser* terjadi peningkatan konsumsi energi dibandingkan dengan kondisi eksisting sehingga nilai efisiensi Ec sebesar -33,75 %, sehingga secara keseluruhan dapat memberikan penghematan biaya utilitas sebesar 16,92 %.

Kata Kunci: Kolom Depropanizer, Kolom Distilasi, *Neural Network*, Optimisasi, *Particle Swarm Optimization*.

Halaman ini sengaja dikosongkan

**ENERGY CONSUMPTION OPTIMIZATION IN
DEPROPANIZER COLUMN PROCESS BASED ON NEURAL
NETWORK - PARTICLE SWARM OPTIMIZATION (NN-PSO)**

Name : Ian Haikal Amir Akbar
NRP : 02311640000021
Department : Engineering Physics FT-IRS ITS
Supervisors : Prof. Dr. Ir. Aulia Siti Aisjah, M.T.
Dr. Ir. Syamsul Arifin, M.T.

ABSTRACT

Distillation column is a plant that is used to separate components of hydrocarbon compounds in crude oil and natural gas based on their volatility values. The high energy usage of Condenser and Reboiler equipment demands the optimization of the distillation column energy consumption to reduce the utility cost of cooling water and steam. Optimization methods performed by searching operational condition variable values distillation column to produce minimum energy consumption, the process of finding the value of distillation column operational variables using Stochastic Algorithm, that is Particle Swarm Optimization (PSO). Distillation columns that are complex and non-linear can be modeled using Neural Network (NN), and the output from the NN model is used to determine the objective function in the PSO algorithm. The operational variables of the Depropanizer Column to be optimized are Feed - Molar Flow (F), Feed - Temperature (Tf), Condenser - Pressure (Pc), Reboiler - Pressure (Pr). Architecture performance of Neural Network Feed Forward - Backpropagation with Lavenberg Marquart's training algorithm on the number of neurons 19 produces RMSE_{Qc}, RSME_{Qr}, and RMSE_{total} values respectively $5,541 \times 10^{-4}$; $6,310 \times 10^{-4}$; $8,398 \times 10^{-4}$. Energy consumption optimization method using the PSO algorithm result in savings of energy consumption in the reboiler with E_r efficiency value of 29.27%, while in Condenser equipment there is an increase in energy consumption compared to existing conditions, so the E_c efficiency value of -33.75%, so that as a whole can provide utility cost savings amounting 16.92%.

Keywords: *Depropanizer Column, Distillation Column, Neural Network, Optimization, Particle Swarm Optimization.*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur penulis kepada Allah SWT, karena rahmat dan hikmat-Nya sehingga penulis diberikan kesehatan, kemudahan dan kelancaran dalam menyusun laporan tugas akhir ini. Sehingga penulis mampu menyusun laporan tugas akhir yang berjudul:

“OPTIMISASI KONSUMSI ENERGI PADA PROSES KOLOM DEPROPANIZER BERBASIS *NEURAL NETWORK – PARTICLE SWARM OPTIMIZATION* (NN-PSO)”

Tugas akhir ini merupakan salah satu persyaratan akademik yang harus dipenuhi dalam menempuh program studi S-1 Departemen Teknik Fisika FT-IRS ITS. Penulis menyampaikan terima kasih yang sebesar- besarnya kepada:

1. Bapak Dr. Suyanto, S.T., M.T. selaku Kepala Departemen Teknik Fisika FT-IRS ITS.
2. Ibu Prof. Dr. Ir. Aulia Siti Aisjah, M.T. dan Bapak Dr. Ir. Syamsul Arifin, M.T. selaku dosen pembimbing tugas akhir yang dengan sabar memotivasi dan membimbing penulis untuk menyelesaikan tugas akhir ini.
3. Ibu Dr. Katherin Indriawati, S.T., M.T. selaku dosen wali dengan sabar memotivasi dan membimbing penulis selama kuliah di Teknik Fisika.
4. Segenap keluarga penulis yang telah memberikan dukungan penuh terhadap penyelesaian tugas akhir ini.
5. Teman-teman Teknik Fisika Angkatan 2016, teman-teman Asisten Laboratorium Workshop Instrumentasi yang telah membantu dan memotivasi saya dalam pelaksanaan dan penyusunan laporan tugas akhir ini.

Penulis menyadari bahwa mungkin masih ada kekurangan dalam laporan ini, sehingga kritik dan saran penulis terima. Semoga laporan ini dapat berguna dan bermanfaat bagi penulis dan pihak yang membacanya.

Surabaya, 26 Juli 2020

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

HALAMAN JUDUL.....	i
COVER PAGE.....	iii
PERNYATAAN BEBAS PLAGIASI.....	v
LEMBAR PENGESAHAN	vii
LEMBAR PENGESAHAN	ix
ABSTRAK	xi
ABSTRACT	xiii
KATA PENGANTAR	xv
DAFTAR ISI.....	xvii
DAFTAR GAMBAR	xxi
DAFTAR TABEL.....	xxiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Tujuan.....	3
1.4 Batasan Masalah.....	3
1.5 Sistematika Laporan	4
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	5
2.1 Kajian Pustaka	5
2.2 Kolom Distilasi.....	7
2.2.1 Kolom Depropanizer	8
2.2.2 Model Matematis Kolom Distilasi	8
2.2.3 <i>Physical properties of Natural Gas</i>	12
2.3 Optimisasi Konsumsi Energi Kolom Distilasi	14

2.3.1.	Proses Overview Kolom Distilasi.....	14
2.3.2.	Perhitungan Konsumsi Energi.....	16
2.3.3.	Biaya Operasional pada Kolom Distilasi.....	17
2.4	<i>Neural Network</i>	18
2.5	PSO (<i>Particle Swarm Optimization</i>).....	21
2.5.1	Proses Inisiasi	23
2.5.2	Proses Iterasi dan Pembaharuan Parameter	23
BAB III METODOLOGI PENELITIAN		25
3.1	Diagram Alir Penelitian	25
3.2	Pengumpulan Data Proses Kolom Depropanizer.....	26
3.3	Perhitungan Nilai <i>Critical Constant</i> (Pcr dan Tcr).....	27
3.4	Pemodelan Matematis Kolom Depropanizer	29
3.5	Perancangan Simulasi Plant Kolom Depropanizer pada <i>Software Hysys</i> untuk Pengambilan Data Training <i>Neural Network</i> (NN).	29
3.6	Validasi Model Simulasi Plant Kolom Depropanizer pada <i>software Hysys</i> . 31	
3.7	Perancangan Arsitektur <i>Neural Network – Particle Swarm Optimization</i> (NN-PSO).	32
3.7.1	Pengambilan Data Training <i>Neural Network</i> (NN) menggunakan <i>software Matlab & Hysys</i>	33
3.7.2	Pemodelan Kolom Depropanizer dengan <i>Neural Network</i> (NN)....	35
3.7.3	Perancangan Algoritma PSO.....	37
3.8	Validasi Hasil Optimisasi Konsumsi Energi.....	41
3.9	Analisa Performansi Hasil Optimisasi Konsumsi Energi.....	41
BAB IV HASIL DAN PEMBAHASAN.....		43
4.1	Pengambilan Data Training <i>Neural Network</i> (NN) pada <i>software Hysys & Matlab</i>	43

4.2	Pelatihan & Validasi Model <i>Neural Network</i> (NN).....	45
4.3	Optimisasi Konsumsi Energi Kolom Depropanizer menggunakan PSO. 48	
4.4	Validasi Hasil Optimisasi Konsumsi Energi	56
4.5	Analisa Performansi Hasil Optimisasi Konsumsi Energi.....	57
BAB V KESIMPULAN DAN SARAN.....		61
5.1	Kesimpulan.....	61
5.2	Saran.....	61
DAFTAR PUSTAKA		63
LAMPIRAN.....		67
BIODATA PENULIS		81

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2. 1 Proses Kolom Distilasi (Hoffman, Lupfer, Kane, Jensen, & Lipták, 2005).	7
Gambar 2. 2 Kolom Depropanizer (Miller & Dennison, 2015).	8
Gambar 2. 3 Kesetimbangan Massa tiap Tray (R. Biyanto, 2007).....	9
Gambar 2. 4 Kesetimbangan massa pada feed tray (R. Biyanto, 2007).	10
Gambar 2. 5 Kesetimbangan massa pada Condenser dan Reflux Drum	10
Gambar 2. 6 Kesetimbangan massa pada Reboiler dan Base Column.....	11
Gambar 2. 7 Struktur dasar Neuron (Buscema et al., 2018).....	18
Gambar 2. 8 Arsitektur <i>Feed Forward – Backpropagation</i> (Fausett, 2005).....	19
Gambar 2. 9 Flowchart Algoritma PSO (Ceylan & Koyuncu, 2019).	22
Gambar 3. 1 Diagram Alir Metode Penelitian Tugas Akhir.....	26
Gambar 3. 2 Model PFD Steady State Kolom Depropanizer pada software Hysys.	30
Gambar 3. 3 Skema optimasi berbasis NN-PSO.	33
Gambar 3. 4 Pemodelan input-output Kolom Depropanizer.....	34
Gambar 3. 5 Arsitektur <i>Neural Network</i>	36
Gambar 3. 6 Flowchart Perancangan Algoritma PSO.....	38
Gambar 3. 7 Fungsi Objektif Algoritma PSO.	40
Gambar 4. 1 Plot data Training NN Feed - Molar flow.	43
Gambar 4. 2 Plot data Training NN Feed - Temperature.	43
Gambar 4. 3 Plot data Training NN Condenser - Pressure.....	44
Gambar 4. 4 Plot data Training NN Reboiler - Pressure.....	44
Gambar 4. 5 Plot data Training NN Condenser Duty - Heat Flow.	44
Gambar 4. 6 Plot data Training NN Reboiler Duty - Heat Flow.....	45
Gambar 4. 7 Grafik regression model Neural Network (NN) dengan jumlah hidden neurons 19.	47
Gambar 4. 8 Grafik konvergensi ($c_1 = c_2 = 0,001$) pada running 1.....	49
Gambar 4. 9 Grafik konvergensi ($c_1 = c_2 = 0,001$) pada running 2.....	49
Gambar 4. 10 Grafik konvergensi ($c_1 = c_2 = 0,001$) pada running 3.....	50

Gambar 4. 11	Grafik konvergensi ($c1 = c2 = 0,0015$) pada running 1	51
Gambar 4. 12	Grafik konvergensi ($c1 = c2 = 0,0015$) pada running 2	52
Gambar 4. 13	Grafik konvergensi ($c1 = c2 = 0,0015$) pada running 3	52
Gambar 4. 14	Grafik konvergensi ($c1 = c2 = 0,002$) pada running 1	54
Gambar 4. 15	Grafik konvergensi ($c1 = c2 = 0,002$) pada running 2	54
Gambar 4. 16	Grafik konvergensi ($c1 = c2 = 0,002$) pada running 3	55
Gambar 4. 17	Grafik hasil optimisasi konsumsi energi dengan algoritma PSO..	58
Gambar 4. 18	Grafik hasil optimisasi nilai operating costs dengan algoritma PSO	58
Gambar B. 1	Model PFD Kolom Depropanizer pada <i>software</i> Hysys.	74
Gambar B. 2	Skema optimisasi berbasis NN-PSO.	75
Gambar B. 3	Skema pengambilan data training <i>Neural Network</i>	75
Gambar B. 4	Arsitektur <i>Neural Network</i>	75
Gambar B. 5	Flowchart perancangan algoritma PSO.	76
Gambar B. 6	Fungsi objektif algoritma PSO.	76

DAFTAR TABEL

Tabel 2. 1 <i>Physical Constants for Pure Components</i> (Saeid M. & Y., 2012).....	13
Tabel 2. 2 Harga <i>Steam</i> dan <i>Cooling Water</i> (Ching, Nandong, & Getu, 2016)...	17
Tabel 3. 1 Kondisi operasional Kolom Depropanizer pada <i>Feed Stream</i> (Ching et al., 2016).....	26
Tabel 3. 2 Spesifikasi Kolom Depropanizer (Ching et al., 2016).	27
Tabel 3. 3 Perhitungan nilai <i>Critical Constant</i> pada <i>feed stream</i>	28
Tabel 3. 4 Hasil simulasi <i>steady state</i> Kolom Depropanizer pada kondisi eksisting.	30
Tabel 3. 5 Validasi model simulasi plant Kolom Depropanizer pada software Hysys.....	32
Tabel 3. 6 Rincian pengambilan pasangan data input - output.....	34
Tabel 3. 7 Arsitektur Neural Network.....	35
Tabel 3. 8 Tabel parameter normalisasi model <i>Neural Network</i> (NN).	37
Tabel 3. 9 Range operasional variabel optimasi.....	38
Tabel 3. 10 Inisialisasi Parameter PSO.	39
Tabel 3. 11 Tabel biaya utilitas.	40
Tabel 4. 1 Hubungan nilai RMSE dengan jumlah <i>hidden neurons</i>	46
Tabel 4. 2 Parameter PSO.	48
Tabel 4. 3 Hasil Optimasi dengan nilai $c1 = c2 = 0,001$	50
Tabel 4. 4 Hasil Optimasi dengan nilai $c1 = c2 = 0,0015$	53
Tabel 4. 5 Hasil Optimasi dengan nilai $c1 = c2 = 0,002$	55
Tabel 4. 6 Validasi hasil optimisasi konsumsi energi.	56
Tabel 4. 7 Tabel hasil optimisasi menggunakan algoritma PSO.....	58
Tabel 4. 8 Perbandingan Sebelum Optimasi dan Sesudah Optimasi.	59
Tabel 4. 9 Efisiensi Penghematan konsumsi energi dan <i>operating costs</i>	60
Tabel A. 1 Data spesifikasi Kolom Depropanizer.....	67
Tabel A. 2 Data <i>stream</i> simulasi <i>steady state</i> Kolom Depropanizer.....	67
Tabel A. 3 Validasi model proses Kolom Depropanizer.....	69
Tabel A. 4 Range operasional variabel optimasi.....	69

Tabel A. 5 Arsitektur <i>Neural Network</i>	69
Tabel A. 6 Parameter normalisasi data <i>Neural Network</i>	70
Tabel A. 7 Parameter PSO.....	71
Tabel A. 8 Tabel RMSE model <i>Neural Network</i>	71
Tabel A. 9 Hasil optimisasi konsumsi energi menggunakan algoritma PSO.	72
Tabel A. 10 Validasi hasil optimisasi konsumsi energi.....	73
Tabel C. 1 <i>Source code</i> pengambilan data training <i>Neural Network</i>	76
Tabel C. 2 Source Code Algoritma PSO	77

BAB I

PENDAHULUAN

1.1 Latar Belakang

Konsumen gas bumi nasional dari sektor rumah tangga adalah LPG (Liquified Petroleum Gas), komposisi rantai hidrocarbon pada LPG didominasi oleh Propana dan Butana, Menurut SK Dirjen Migas No. 26525.K/10/DJM.T/2009, spesifikasi LPG minimal mengandung 95% Propana (M. Hasan Syukur, ST, 2011), dimana proses fraksinasi propana dari kandungan butana dan fraksi berat lainnya terjadi pada Kolom Distilasi Depropanizer. Distilasi sendiri merupakan teknik pemisahan campuran yang terdiri dari dua atau lebih komponen menjadi komponen yang mempunyai tingkat kemurnian sesuai yang diinginkan berdasarkan perbedaan volatilitas komposisinya. Permasalahan dalam industri *Refinery* dalam pengolahan gas alam adalah penggunaan energi yang sangat tinggi pada plant Kolom Distilasi, penggunaan energi tersebut terletak pada equipment *condenser* dan *reboiler* yang akan berakibat pada besarnya *utility cost* yaitu *cooling water* pada *condenser* dan *Steam* pada *reboiler* (R. Biyanto, 2007).

Berbagai industri *Refinery* berupaya untuk melakukan penghematan konsumsi energi pada Kolom Distilasi, salah satu upaya tersebut adalah dengan melakukan pengintegrasian dari Kolom Distilasi dengan keseluruhan proses yang secara teori dapat menghasilkan penghematan energi yang signifikan sampai dengan 28-33% (R. Biyanto, 2007). Teknik lain adalah dengan melakukan optimasi menggunakan *Stochastic Algorithm* untuk mencari kondisi operasi yang membutuhkan energi terkecil.

Stochastic Algorithm banyak digunakan untuk Teknik optimasi, Banyak persoalan keputusan yang dapat dimodelkan dengan menggunakan program matematika atau *objective function* yang bertujuan menentukan nilai maksimum atau minimum. Beberapa jenis *Stochastic Algorithm* yang berkembang saat ini antara lain GA (*Genetic Algorithm*), ICA (*Imperialist Competitive Algorithm*), PSO (*Particle Swarm Optimization*), ACO (*Ant Colony Optimization*) dan TS (*Tabu Search*). Pada sebagian besar metode *Stochastic Algorithm* seperti GA dan TS merupakan tipe yang lamban dalam menentukan solusi optimal sedangkan pada

beberapa metode stokastik baru seperti PSO, ACO dan ICA memiliki beberapa keunggulan diantaranya dapat menyelesaikan permasalahan yang sulit dan kompleks, hal ini sangat cocok untuk diaplikasikan pada permasalahan di Kolom Distilasi yang sangat kompleks. tidak hanya memberikan respon yang lebih baik tetapi juga lebih cepat mencapai konvergen dibandingkan algoritma evolusi biasa (Fitriyani, Ruki, & Ph, 2015). Dari beberapa metode stokastik baru seperti PSO, ACO dan ICA, PSO memiliki keunggulan dalam aspek pencarian ruang.

Secara teori, konsep PSO didasarkan pada perilaku sebuah kawanan serangga, seperti semut, rayap, lebah atau burung. Algoritma PSO meniru perilaku sosial organisme ini, ketika salah satu individu menemukan jalan yang tepat atau pendek menuju ke sumber makanan, sisa kelompok yang lain juga akan dapat segera mengikuti jalan tersebut. Dalam hal ini optimasi energi pada Kolom Distilasi menggunakan metode PSO akan mencari solusi susunan konfigurasi yang menghasilkan energi paling kecil dari beberapa susunan konfigurasinya (Ji, Wang, Chu, & Wu, 2009).

Sehingga pada penelitian ini yang berjudul “*Optimasi Konsumsi Energi pada Proses Kolom Depropanizer berbasis Neural Network - Particle Swarm Optimization*” digunakan Pemodelan Kolom Distilasi Depropanizer dengan memanfaatkan kelebihan yang dimiliki *Neural Network* (NN) dalam membangun hubungan *nonlinier* antara *input-output* MIMO (*Multi Input Multi Output*) variabel sehingga memiliki karakteristik yang sama dengan proses yang ditinjau. *Network type* NN yang digunakan adalah *Feed Forward – Backpropagation*. dengan *Training Function* adalah TRAINLM (*Lavenberg Marquart*) dan *Adaption Learning Function* menggunakan LEARNGDM (*Gradien Descent with Momentum*), untuk mengetahui performansi NN digunakan *Performance function* MSE (*Mean Square Error*).

1.2 Rumusan Masalah

Adapun rumusan masalah pada tugas akhir ini adalah sebagai berikut :

- a) Bagaimana merancang arsitektur Neural Network untuk pemodelan plant Kolom Depropanizer yang menghasilkan nilai RMSE terkecil?

- b) Bagaimana merancang arsitektur integrasi NN-PSO untuk mengoptimasi konsumsi energi pada Kolom Depropanizer?
- c) Berapa nilai parameter PSO yang dapat memberikan penghematan konsumsi energi yang signifikan pada Kolom Depropanizer ?
- d) Apakah Teknik Optimisasi menggunakan sistem integrasi NN-PSO mampu memberikan penghematan konsumsi energi?

1.3 Tujuan

Adapun Tujuan dari penelitian tugas akhir ini adalah sebagai berikut :

- a) Merancang arsitektur Neural Network untuk pemodelan plant Kolom Depropanizer yang menghasilkan nilai RMSE terkecil.
- b) Merancang arsitektur integrasi NN-PSO untuk mengoptimasi konsumsi energi pada Kolom Depropanizer.
- c) Menentukan nilai parameter PSO yang dapat memberikan penghematan konsumsi energi yang signifikan pada Kolom Depropanizer.
- d) Membandingkan konsumsi energi antara kondisi existing dengan kondisi setelah dilakukan optimisasi menggunakan sistem integrasi NN-PSO.

1.4 Batasan Masalah

Adapun Batasan masalah pada tugas akhir ini antara lain sebagai berikut :

- a) Plant yang dijadikan fokus utama pada tugas akhir ini adalah Kolom Depropanizer.
- b) Variabel proses pada Kolom Depropanizer yang akan dilakukan manipulasi menggunakan algoritma PSO agar dapat menghasilkan konsumsi energi terkecil adalah *Feed – Molar Flow (F)*, *Feed – Temperature (Tf)*, *Condenser – Pressure (Pc)* dan *Reboiler – Pressure (Pr)*.
- c) Fungsi objektif pada algoritma PSO adalah menggunakan fungsi minimum dengan memasukan perhitungan biaya utilitas *steam* dan *cooling water* sebagai penentu nilai minimum.
- d) Data pelatihan Neural Network diambil dari Software Hysys dengan mensimulasikan plant secara *steady state*.
- e) Rancangan script program integrasi NN-PSO menggunakan software Matlab.

1.5 Sistematika Laporan

Sistematika penulisan laporan tugas akhir ini adalah sebagai berikut.

BAB I Pendahuluan

Bab I ini terdiri dari latar belakang, rumusan masalah, tujuan, batasan masalah dan sistematika laporan.

BAB II Tinjauan Pustaka & Dasar Teori

Bab II ini dibahas mengenai kajian pustaka dan teori-teori yang berkaitan dengan penelitian yang akan dilakukan, seperti kolom distilasi, kolom depropanizer, pemodelan matematis kolom distilasi, *Neural Network (NN)*, *Feed Forward - Backpropagation*, TRAINLM (*Lavenberg Marquarth*), LEARNGDM (*Gradient with Momentum*), dan Particle Swarm Optimization (PSO).

BAB III Metodologi Penelitian

Bab III ini berisi mengenai rancangan dari penelitian yang dilakukan, metode, dan langkah-langkah dalam penelitian.

BAB IV Hasil dan Pembahasan

Bab IV ini berisi tentang data hasil penelitian dari hasil Optimasi konsumsi energi pada Kolom Depropanizer dengan NN-PSO.

BAB V Kesimpulan dan Saran

Bab V ini diberikan kesimpulan tentang tugas akhir ini dan saran yang membangun sebagai penunjang maupun pengembangan tugas akhir selanjutnya.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Kajian Pustaka

Salah satu penelitian yang relevan tentang optimisasi konsumsi energi pada Kolom Distilasi adalah penelitian yang dilakukan oleh (R. Biyanto, 2007) dengan judul “Algoritma Genetika untuk mengoptimasi konsumsi energi pada proses Kolom Metanol-Air”. Pada penelitian ini, proses optimisasi konsumsi energi menggunakan metode *Genetic Algorithm* (GA) yang diintegrasikan dengan *Neural Network* (NN) sebagai pemodelan plant Kolom Metanol-Air. Dengan menggunakan arsitektur *Neural Network* (NN) NARX (*Nonlinear Auto Regressive with eXternal input*) dan metode pembelajaran algoritma Lavenburg Marquarth, Hasil optimisasi dengan metode *Genetic Algorithm* (GA) mendapatkan variabel proses *pressure top* 106.846 Kpa, *level condenser* 30.289%, *temperatur feed* 83.48 °C, *fraksi feed* 0.5258, dan *flow feed* 493.518 Kgmol/jam. Dari variabel proses tersebut dapat menghasilkan penurunan konsumsi energi *Steam* pada *Reboiler* dan *Cooling water* pada *Condenser* sebesar 46.2% dibandingkan dengan kondisi eksisting.

Penelitian tentang optimisasi pada Kolom Distilasi juga dilakukan oleh (Fitriyani et al., 2015) dengan judul “Optimasi Kondisi Kolom Distilasi Biner untuk Mencapai Kualitas Produk dengan Menggunakan Imperialist Competitive Algorithm (ICA)”. Pada penelitian ini, optimisasi dilakukan untuk mencari kondisi operasional yang mampu meningkatkan kualitas produk atau dengan kata lain meningkatkan performansi Kolom Distilasi dalam memisahkan komponen *Distillate* dan *Bottom Product*. Proses optimisasi pada penelitian ini menggunakan metode *Imperialist Competitive Algorithm* (ICA) yang diintegrasikan dengan *Neural Network* (NN) sebagai pemodelan plant Kolom Distilasi. Dengan menggunakan arsitektur *Neural Network* (NN) NARX (*Nonlinear Auto Regressive with eXternal input*) dan algoritma pembelajaran Lavenberg Maquardt. Hasil optimisasi menggunakan metode *Imperialist Competitive Algorithm* (ICA) didapatkan kondisi operasional yang optimal pada kolom distilasi biner sebesar 0.932 untuk mol fraksi *Distillate* dan 0.9932 untuk mol fraksi *Bottom Product*.

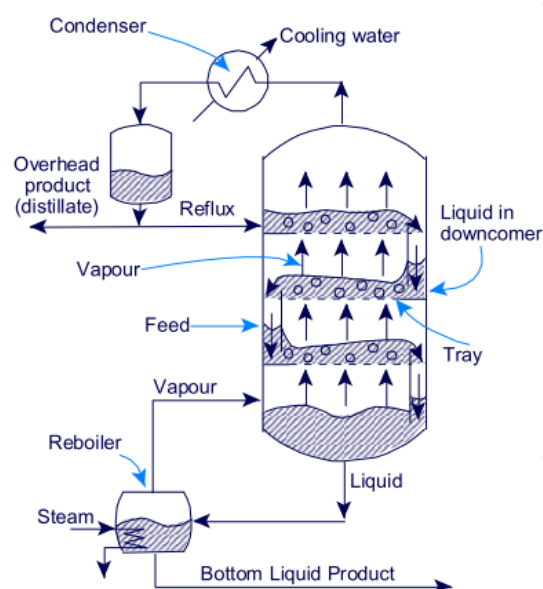
Penelitian lain tentang Kolom Distilasi dimodelkan menggunakan *Neural Network* (NN) juga dilakukan oleh (Biyanto, Widjiantoro, Jabal, & Budiati, 2011) dengan judul “Artificial neural network based modeling and controlling of distillation column system”. Pada penelitian ini menjelaskan tentang sistem pengendalian pada Kolom Distilasi menggunakan *Neural Network - Internal Model Control* (NN-IMC). Referensi yang dapat diambil dari penelitian ini adalah *Neural Network* (NN) yang digunakan sebagai pemodelan plant Kolom Distilasi. Pemodelan *Neural Network* (NN) terdiri dari 3 layer yaitu 1 *input layer*, 1 *hidden layer*, dan 1 *output layer*. Pada *input layer* terdiri dari 4 variabel yaitu X_d (fraksi mol *Distillate*), L (laju reflux), Q_r (*Heat flow reboiler*) dan X_b (fraksi mol *Bottom Product*), pada *hidden layer* terdiri dari 13 *neuron*, dan pada *output layer* terdiri dari 2 variabel yaitu X_d (fraksi mol *Distillate*) dan X_b (fraksi mol *Bottom Product*). Dari rincian tersebut dimodelkan dengan arsitektur NARX (*Nonlinear Auto Regressive with eXternal input*) dan metode pembelajaran algoritma Lavenburg Marquarth menghasilkan nilai RMSE sebesar $5,9908 \times 10^{-5}$ dan $1,2686 \times 10^{-4}$ pada X_d (fraksi mol *Distillate*) dan X_b (fraksi mol *Bottom Product*).

Penelitian Le dan Zhou yang berjudul “A Comparative Study of PSO-ANN, GA-ANN, ICA-ANN, and ABC-ANN in Estimating the Heating Load of Buildings’ Energy Efficiency for Smart City Planning” menjelaskan mengenai perbandingan performansi beberapa metode optimasi efisiensi energi pada sebuah gedung untuk mengoptimalkan perencanaan kota pintar (Le, Nguyen, Dou, & Zhou, 2019). Penelitian ini mengusulkan empat teknik kecerdasan buatan (AI) baru untuk meramalkan beban pemanasan efisiensi energi bangunan dengan menggunakan *Neural Network* (NN) yang dikombinasikan dengan metode optimasi diantaranya *Artificial Bee Colony* (ABC), *Particle Swarm Optimization* (PSO), *Imperialist Competitive Algorithm* (ICA), dan *Genetic Algorithm* (GA). Atau disingkat sebagai model ABC-ANN, PSO-ANN, ICA-ANN, dan GA-ANN. Sebuah gedung dianalisis berdasarkan parameter yang berpengaruh, seperti area distribution (GLAD), glazing area (GLA), orientation (O), overall height (OH), roof area (RA), wall area (WA), surface area (SA), relative compactness (RC), for estimating heating load (HL). Tiga kriteria statistik, seperti *Root Mean Square Error* (RMSE), koefisien *determinasi* (R^2), dan *Mean Absolute Error* (MAE), digunakan untuk

menilai potensi model diatas. Hasil menunjukkan bahwa model GA-ANN memberikan kinerja tertinggi dalam memperkirakan beban pemanasan efisiensi energi bangunan, dengan RMSE 1,625, R2 0,980, dan MAE 0,798. Model yang lain seperti PSO-ANN, ICA-ANN, ABC-ANN menghasilkan kinerja yang lebih rendah dengan RMSE sebesar 1.932, 1.982, 1.878; R2 dari 0,972, 0,970, 0,973; MAE masing-masing 1,027, 0,980, 0,957.

2.2 Kolom Distilasi

Proses distilasi merupakan salah satu proses utama dalam proses pengolahan *crude oil* dan *natural gas*. Kolom Distilasi berfungsi untuk memisahkan komponen dari sebuah campuran. Pemisahan ini dilakukan berdasarkan perbedaan nilai volatilitas komponen pada campuran tersebut. Komponen yang memiliki volatilitas besar akan lebih mudah dipisahkan sedangkan komponen yang memiliki volatilitas kecil akan lebih sulit dipisahkan. Proses yang ada pada proses Kolom Distilasi ditunjukkan pada **Gambar 2.1** sebagai berikut:



Gambar 2. 1 Proses Kolom Distilasi (Hoffman, Lupfer, Kane, Jensen, & Lipták, 2005).

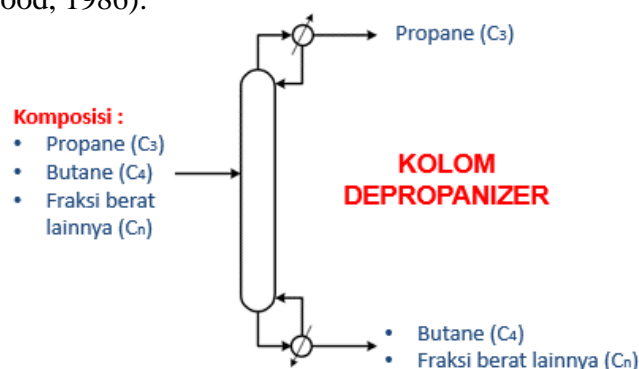
Kolom Distilasi terdiri atas beberapa komponen yaitu sebagai berikut:

- a. *Shell*, atau kolom merupakan tempat terjadinya pemisahan komponen.

- b. *Tray* digunakan untuk memperbesar kontak antara cairan dan gas sehingga komponen dapat dipisahkan sesuai dengan rapat jenisnya, dalam bentuk gas atau cairan.
- c. *Reboiler*, merupakan tempat penguapan cairan yang keluar dari bagian bawah kolom.
- d. Kondenser, merupakan tempat untuk mendinginkan atau mengondensasikan uap yang meninggalkan bagian atas Kolom Distilasi.
- e. *Reflux Drum*, untuk menampung uap yang terkondensasi dari kolom bagian atas sehingga cairan (*reflux*) dapat diumpanbalikkan ke kolom.

2.2.1 Kolom Depropanizer

Kolom depropanizer merupakan kolom distilasi biner yang digunakan untuk memisah propana dari campuran yang mengandung butana dan komponen lainnya. Masukan kolom depropanizer merupakan bottom product dari kolom deethanizer. Depropanizer memfraksinasi feed menjadi produk yang kaya propana dan bottom product yang terdiri dari butana dan minyak gas alam. Keluaran atas dari kolom depropanizer yang kaya akan propana akan dikondensasikan dalam kondensor oleh air pendingin, kemudian kondensat dikumpulkan di reflux drum. Sebagian kondensat dikirim kembali ke kolom dengan menggunakan pompa sedangkan sebagian lainnya yang tersisa ditarik sebagai produk propana cair. Bottom product dari kolom depropanizer kemudian dikirim ke debuthanizer (Carling & Wood, 1986).



Gambar 2. 2 Kolom Depropanizer (Miller & Dennison, 2015).

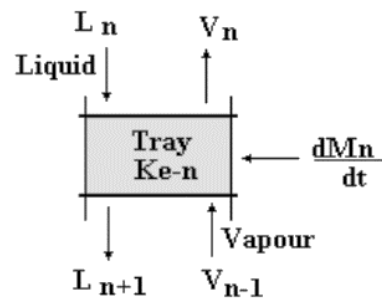
2.2.2 Model Matematis Kolom Distilasi

Model matematis Kolom Distilasi ini merupakan model secara *general* atau umum untuk menggambarkan proses pada Kolom Distilasi yang

direpresentasikan dalam Hukum Kestimbangan Massa, Hukum Kestimbangan Massa Komponen, dan Hukum Kestimbangan Energi pada Tray, *Reboiler & Base Column*, dan *Condenser & Reflux Drum* dalam fase *vapor-liquid*. Pemodelan matematis kolom Depropanizer dibagi menjadi beberapa section diantaranya : **Pemodelan matematis pada tray, Pemodelan matematis pada feed tray, Pemodelan matematis pada Condenser & Reflux Drum, dan Pemodelan Base Column & Reboiler.**

a. Pemodelan Matematis pada Main Tower.

Pada Kolom Distilasi terdapat beberapa tray, Tray digunakan untuk memperbesar kontak antara cairan dan gas sehingga komponen dapat dipisahkan sesuai dengan rapat jenisnya, dalam bentuk gas atau cairan. Cairan pada masing-masing Tray diasumsikan tercampur sempurna dengan komposisi X_N , yang ditunjukkan pada **Gambar 2.3**. Persamaan matematis yang menyatakan proses dalam Kolom Distilasi dapat dijelaskan sebagai berikut (Luyben, 1990) (Biyanto et al., 2011) :



Gambar 2. 3 Kestimbangan Massa tiap Tray (R. Biyanto, 2007).

Hukum Kestimbangan Massa Tray :

$$\frac{dM_N}{dt} = L_{N+1} - L_N + V_{N-1} - V_N \quad (2.1)$$

Hukum Kestimbangan Komponen Tray :

$$\frac{d(M_N X_N)}{dt} = L_{N+1} X_{N+1} - L_N X_N + V_{N-1} Y_{N-1} - V_N Y_N \quad (2.2)$$

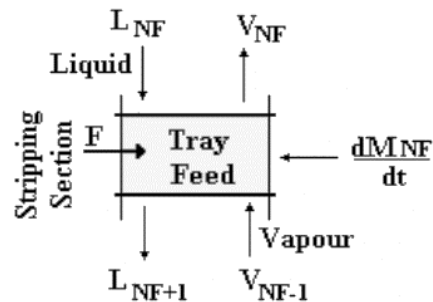
Hukum Kestimbangan Energi Tray :

$$\frac{d(M_N h_N)}{dt} = L_{N+1} h_{N+1} - L_N h_N + V_{N-1} H_{N-1} - V_N H_N \quad (2.3)$$

Feed Tray (N=NF)

Aliran *Feed* tunggal dimasukkan sebagai *saturated liquid* ke tray umpan NF.

Dengan *Feed flow rate* adalah F (mol / jam) dan komposisi X_F adalah komponen dengan fraksi mol volatilitas lebih tinggi (Biyanto et al., 2011).



Gambar 2. 4 Kestimbangan massa pada feed tray (R. Biyanto, 2007).

Hukum Kestimbangan Massa Tray Feed :

$$\frac{dM_{NF}}{dt} = L_{NF+1} - L_{NF} + V_{NF-1} - V_{NF} + F \quad (2.4)$$

Hukum Kestimbangan Komponen Tray Feed :

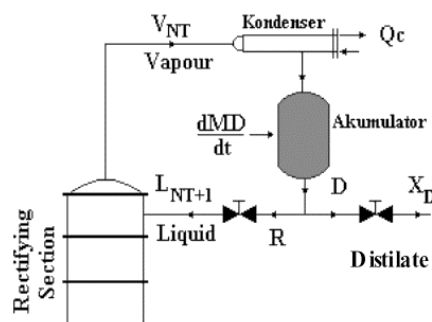
$$\frac{d(M_{NF}X_{NF})}{dt} = L_{NF+1}X_{NF+1} - L_{NF}X_{NF} + V_{NF-1}Y_{NF-1} - V_{NF}Y_{NF} + FX_F \quad (2.5)$$

Hukum Kestimbangan Energi Tray Feed :

$$\frac{d(M_{NF}h_{NF})}{dt} = L_{NF+1}h_{NF+1} - L_{NF}h_{NF} + V_{NF-1}H_{NF-1} - V_{NF}H_{NF} + Fh_F \quad (2.6)$$

b. Pemodelan Matematis pada *Condenser dan Reflux Drum*.

Kondensor merupakan tempat untuk mendinginkan atau mengondensasikan uap yang meninggalkan bagian atas Kolom Distilasi (V_{NT}), uap overhead V_{NT} akan mengalami kondensasi pada kondensor dan mengalir ke Refluks Drum. Refluks dipompa kembali ke tray bagian atas kolom dengan flowrate (L), sedangkan gas keluar sebagai *distillate* (D).



Gambar 2. 5 Kestimbangan massa pada Condenser dan Reflux Drum

(R. Biyanto, 2007).

Sehingga rumus matematis yang berlaku pada Kondensor dan Reflux Drum dapat dijelaskan sebagai berikut (Biyanto et al., 2011) :

Hukum Kesetimbangan Massa Kondensor dan Refluks Drum :

$$\frac{dM_D}{dt} = V_{NT} - L_{NT+1} - D \quad (2.7)$$

Hukum Kesetimbangan Komponen Kondensor dan Refluks Drum :

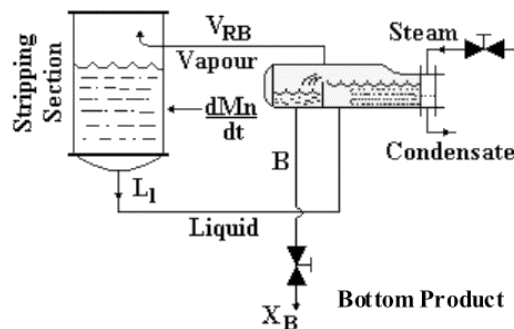
$$\frac{d(M_D X_D)}{dt} = V_{NT} Y_{NT} - (L_{NT+1} + D) X_D \quad (2.8)$$

Hukum Kesetimbangan Energi Kondensor dan Refluks Drum :

$$\frac{d(M_D h_D)}{dt} = V_{NT} H_{NT} - L_{NT+1} h_{NT+1} - D h_D - Q_C \quad (2.9)$$

c. Pemodelan Matematis pada Reboiler dan Base Column

Pada dasar kolom, terdapat produk dasar (*bottom product*) dalam fasa cair. *Bottom product* masuk ke dalam Reboiler dengan *flowrate* L_1 . *bottom product* (L_1) yang memasuki Reboiler di uapkan kembali, hasil penguapan *bottom product* pada Reboiler (V_{RB}) di umpan balik ke tray bagian bawah kolom, sedangkan komponen pada Reboiler yang tidak menguap atau *liquid* akan di alirkan sebagai *bottom product* (B) murni (terbebas dari kandungan *distillate*). Sehingga berlaku rumus matematis sebagai berikut (Biyanto et al., 2011) :



Gambar 2. 6 Kesetimbangan massa pada Reboiler dan Base Column

(R. Biyanto, 2007).

Hukum Kesetimbangan Massa Reboiler dan Base Column :

$$\frac{dM_N}{dt} = L_1 - V_{RB} - B \quad (2.10)$$

Hukum Kesetimbangan Komponen Reboiler dan Base Column :

$$\frac{d(M_B X_B)}{dt} = L_1 X_1 - V_{RB} Y_B - B X_B \quad (2.11)$$

Hukum Kesetimbangan Energi Reboiler dan Base Column :

$$\frac{d(M_B h_B)}{dt} = L_1 h_1 - V_{RB} H_B - B h_B + Q_R \quad (2.12)$$

d. Perhitungan Rasio Kolom Distilasi

Kolom Distilasi memiliki spesifikasi rasio antara lain *Reflux ratio*, *Distillate to feed ratio*, dan *Bottom Product to feed ratio*.

Reflux ratio merupakan perbandingan laju *flowrate Reflux* dengan laju *flowrate Distillate*, sehingga perhitungan *Reflux ratio* dapat dihitung dengan **Persamaan 2.13** sebagai berikut :

$$\text{Reflux ratio} = \frac{\dot{R} \text{ (kgmole/h)}}{\dot{D} \text{ (kgmole/h)}} \quad (2.13)$$

Distillate to feed ratio merupakan perbandingan laju *flowrate Distillate* dengan *flowrate feed*, sehingga perhitungan *Distillate to feed ratio* dapat dihitung dengan **Persamaan 2.14** sebagai berikut :

$$\text{Distillate to feed ratio} = \frac{\dot{D} \text{ (kgmole/h)}}{\dot{F} \text{ (kgmole/h)}} \quad (2.14)$$

Bottom product to feed ratio merupakan perbandingan laju *flowrate Bottom product* dengan *flowrate feed*, sehingga perhitungan *Bottom product to feed ratio* dapat dihitung dengan **Persamaan 2.15** sebagai berikut :

$$\text{Bottom product to feed ratio} = \frac{\dot{B} \text{ (kgmole/h)}}{\dot{F} \text{ (kgmole/h)}} \quad (2.15)$$

2.2.3 Physical properties of Natural Gas.

Natural Gas adalah campuran hidrokarbon alami yang ditemukan di dalam perut bumi pada kondisi tekanan dan suhu yang tinggi (Saeid M. & Y., 2012). Oleh karena itu penting untuk mengetahui bagaimana sifat fisik dari *natural gas*. Termasuk sifat kritis dari *natural gas*. Menurut (Saeid M. & Y., 2012) , komponen senyawa hidrokarbon memiliki nilai P_{cr} (*Critical Pressure*) dan T_{cr} (*Temperature Critical*), dimana jika kondisi suatu senyawa hidrokarbon melewati salah satu batas P_{cr} ataupun T_{cr} akan mengalami *cracking* pada struktur senyawanya.

Sehingga diharuskan dalam proses fraksinasi *natural gas* harus memperhatikan hal tersebut. Tabel *Physical Constant* untuk komponen murni senyawa hidrokarbon dapat ditunjukkan pada **Tabel 2.1** :

Tabel 2. 1 *Physical Constants for Pure Components* (Saeid M. & Y., 2012).

Compound	Formula	Moleculer Weight	Critical Constant	
			P_{cr} (kPa)	T_{cr} (°K)
Methane	CH ₄	16,043	4604	190,6
Ethane	C ₂ H ₆	30,07	4880	305,4
Propane	C ₃ H ₈	44,097	4249	369,8
n-Butane	C ₄ H ₁₀	58,124	3797	425,2
Isobutane	C ₄ H ₁₀	58,124	3648	408,2
n-Pentane	C ₅ H ₁₂	72,151	3369	469,7
Isopentane	C ₅ H ₁₂	72,151	3381	460,4
Neopentane	C ₅ H ₁₂	72,151	3199	1249,4
n-Hexane	C ₆ H ₁₄	86,178	3012	507,4
n-Heptane	C ₇ H ₁₆	100,205	2736	540,3
n-Octane	C ₈ H ₁₈	114,232	2486	568,8
n-Nonane	C ₉ H ₂₀	128,3	2289	594,6
n-Decane	C ₁₀ H ₂₂	142,3	2096	617,7
Ethylene	C ₂ H ₄	28,054	5032	282,6
Propene	C ₃ H ₆	42,081	4613	364,9
Acetylene	C ₂ H ₂	26,038	6139	308,5
Carbon dioxide	CO ₂	44,01	7382	304,2
Hydrogen sulfide	H ₂ S	34,076	9005	373,6
Sulfur dioxide	SO ₂	64,059	7894	430,8
Nitrogen	N ₂	28,013	3399	126,3
Water	H ₂ O	18,015	22105	647,2

Tabel diatas merupakan tabel *Physical Constant* untuk komponen murni senyawa hidrokarbon, untuk menghitung nilai P_{cr} dan T_{cr} pada komponen campuran dapat menggunakan **Persamaan 2.16** :

$$P_{cr} = \sum_i^n P_{cr i} * y_i \quad \text{dan} \quad T_{cr} = \sum_i^n T_{cr i} * y_i \quad (2.16)$$

Dimana $P_{cr i}$ dan $T_{cr i}$ adalah nilai *Critical Pressure* dan *Critical Temperature* pada komponen i , dan y_i adalah *mole fraction* komponen i .

2.3 Optimisasi Konsumsi Energi Kolom Distilasi

Terdapat beberapa cara dalam melakukan optimisasi konsumsi energi pada Kolom Distilasi, yang pertama adalah dengan melakukan *Re-design* pada Kolom Distilasi baik dalam spesifikasi equipment yang digunakan pada Kolom Distilasi maupun pengintegrasian kalor yang dapat menghasilkan penghematan konsumsi energi. Cara ini sangatlah efektif dalam menghasilkan penghematan energi, akan tetapi dalam pelaksanaannya membutuhkan banyak waktu dan juga biaya proyek yang tinggi.

Cara yang kedua dalam melakukan optimisasi konsumsi energi pada Kolom Distilasi adalah dengan mengubah atau memanipulasi kondisi atau nilai pada beberapa variabel operasional Kolom Distilasi yang dapat menghasilkan penghematan konsumsi energi pada equipment *Condenser* dan *Reboiler*. Secara umum optimisasi dengan melakukan manipulasi variabel operasional Kolom Distilasi ini menghasilkan penghematan konsumsi energi yang lebih kecil dibandingkan dengan cara yang pertama yaitu dengan melakukan *Re-design*, akan tetapi cara kedua ini memiliki keuntungan diantaranya biaya proyek yang kecil dan waktu yang sedikit dibandingkan dengan cara pertama. Pada tugas akhir ini dilakukan optimisasi konsumsi energi pada Kolom Distilasi dengan memanipulasi nilai beberapa variabel operasional Kolom Distilasi.

2.3.1. Proses Overview Kolom Distilasi

Kolom Distilasi merupakan suatu plant terdapat pada industri *Refinery* dan *Petrochemical* yang digunakan untuk memisahkan komponen senyawa *hydrocarbon* berdasarkan nilai volatilitas atau titik didihnya, dalam proses distilasinya, Kolom Distilasi membutuhkan energi berupa konsumsi *cooling water* pada *Condenser* dan *Steam* pada *Reboiler*. Aliran *feed* pada Kolom Distilasi dipanaskan atau dikondisikan pada temperature tertentu terlebih dahulu

untuk mencapai kondisi fase antara Liquid dan Gas untuk mencapai proses yang optimal.

Beban energi *Condenser* (Q_c) pada Kolom Distilasi yaitu equipment *Condenser* harus dapat mengkondensasi fase gas *flowrate Distillate* menjadi 100% liquid, sedangkan beban energi *Reboiler* (Q_r) pada Kolom Distilasi yaitu equipment *Reboiler* harus dapat menguapkan kembali *Bottom Product* untuk menghindari komponen senyawa *Distillate* pada *Bottom Product* atau dengan kata lain untuk meningkatkan kualitas pemurnian/pemisahan senyawa *hydrocarbon* pada Kolom Distilasi.

Sehingga dalam menentukan variabel operasional yang akan dilakukan manipulasi dapat dengan melakukan studi teori Termodinamika pada Kolom Distilasi sebagai berikut :

a. Hubungan *flowrate feed* pada beban Q_c dan Q_r .

Hubungan *flowrate feed* (F) pada beban Q_c dan Q_r dapat dilihat pada **Persamaan 2.9** dan **Persamaan 2.12** yaitu persamaan kesetimbangan energi serta **Persamaan 2.14**. yaitu persamaan spesifikasi *Distillate to feed ratio* Kolom Distilasi, sehingga dapat ditentukan bahwa perubahan nilai variabel *flowrate feed* (F) dapat mempengaruhi secara langsung beban Q_c dan Q_r . Semakin besar nilai *flowrate feed* (F) maka energi yang dibutuhkan pada equipment *Condenser* dan *Reboiler* semakin besar juga, begitu juga sebaliknya.

b. Hubungan *temperature feed* pada beban Q_c dan Q_r .

Nilai *temperature feed* akan mempengaruhi *stream feed phase* pada Kolom Distilasi, dengan mengubah nilai *temperature feed* maka fase *stream feed* akan berubah antara *liquid* dan *vapour*, hal tersebut akan mempengaruhi nilai beban energi pada equipment *Condenser* dan *Reboiler*. Karena secara mendasar bahwa Kolom Distilasi memisahkan komponen senyawa *hydrocarbon* berdasarkan nilai volatilitas atau titik didihnya, komponen yang memiliki nilai volatilitas yang tinggi akan mudah menguap dan menuju ke *top stage* Kolom Distilasi dan akan melalui *Condenser* untuk proses kondensasi ke fase *liquid*, sedangkan komponen yang memiliki nilai volatilitas rendah akan susah menguap dan menuju ke bagian *bottom stage* Kolom Distilasi dan akan melalui *Reboiler* untuk proses penguapan kembali. Sehingga secara langsung bahwa

fase *stream feed* akan mempengaruhi nilai beban energi pada equipment *Condenser* dan *Reboiler*, dan cara memanipulasi fase *stream feed* adalah dengan mengubah temperatur aliran streamnya.

c. Hubungan *pressure Condenser* dan *Reboiler* pada beban Q_c dan Q_r .

Menurut (Mcgoodwin, 2016) Suatu cairan menguap ketika tekanan uap jenuhnya menjadi sama dengan tekanan eksternalnya (tekanan atmosfer untuk sistem terbuka) pada cairan. Jika cairan terpapar normal tekanan atmosfer di permukaan laut, maka cairan akan mendidih ketika tekanan uap jenuhnya menjadi sama dengan 1 atm (atau 101325 Pa atau 101.325 kPa atau 760 mmHg). Sebagai contoh di permukaan laut dengan tekanan 1 atm, air murni akan menguap ketika suhunya mencapai sekitar 100 °C. Tetapi pada tekanan yang berbeda seperti di area pegunungan, air akan mendidih pada temperatur yang berbeda. Begitu juga dalam kasus ini senyawa *hydrocarbon* yang terdapat pada *Reboiler* akan menguap pada temperatur yang berbeda dan tekanan yang berbeda. Sehingga nilai tekanan pada *Reboiler* akan berpengaruh pada beban energi *Reboiler* (Q_r).

Sedangkan pada bagian *Condenser*, Beban energi *Condenser* (Q_c) pada Kolom Distilasi adalah mengkondensasi fase gas *flowrate Distillate* menjadi 100% liquid. Suatu cairan memiliki nilai *Melting Point* (titik pencairan). Sama halnya dengan *Boiling Point* (titik didih), titik pencairan suatu cairan memiliki nilai yang berbeda pada tekanan yang berbeda, sehingga senyawa *hydrocarbon* pada *Condenser* akan terkondensasi pada tekanan tertentu.

2.3.2. Perhitungan Konsumsi Energi

Perhitungan konsumsi energi pada Kolom Distilasi dapat dijelaskan dalam beberapa persamaan berikut :

Perhitungan konsumsi energi *Condenser* dan *Reboiler* per satuan laju massa menggunakan **Persamaan 2.17** dan **Persamaan 2.18** :

$$E_c = \frac{\dot{Q}_c \text{ (kW)}}{\dot{D} \text{ (kgmole/h)}} \quad (2.17)$$

$$E_r = \frac{\dot{Q}_r \text{ (kW)}}{\dot{B} \text{ (kgmole/h)}} \quad (2.18)$$

Perhitungan efisiensi konsumsi energi *Condenser* dan *Reboiler* menggunakan **Persamaan 2.19** dan **Persamaan 2.20** :

$$\mu Ec (\%) = \frac{Ec_{eksisting} - Ec_{optimasi}}{Ec_{eksisting}} \times 100\% \quad (2.19)$$

$$\mu Er (\%) = \frac{Er_{eksisting} - Er_{optimasi}}{Er_{eksisting}} \times 100\% \quad (2.20)$$

2.3.3. Biaya Operasional pada Kolom Distilasi

Operating Costs pada kolom distilasi ini berhubungan dengan besarnya energi yang di konsumsi pada equipment *Condenser* dan *Reboiler*, berikut *Operating Costs* pada kolom distilasi :

- Proses *Heating by steam*, proses ini terjadi pada equipment reboiler pada kolom distilasi dimana besarnya laju panas pada reboiler adalah Q_r (Heat Flow Reboiler), Biaya yang terkait adalah biaya bahan bakar untuk menghasilkan steam.
- Proses *Cooling by cooling water*, proses ini terjadi pada equipment condenser pada kolom distilasi, biaya yang terkait adalah biaya yang digunakan untuk mengalirkan air ke *Condenser* menggunakan pompa, *Costs Cooling* memiliki biaya yang lebih rendah dibandingkan dengan *Heating Costs*.

Tabel 2. 2 Harga *Steam* dan *Cooling Water* (Ching, Nandong, & Getu, 2016).

Utility Data	Harga (\$(/kW.year))
Cooling Water	90
Steam	300

Berdasarkan point a. b. dan **Tabel 2.3** dapat ditentukan nilai *operating costs* sebagai berikut :

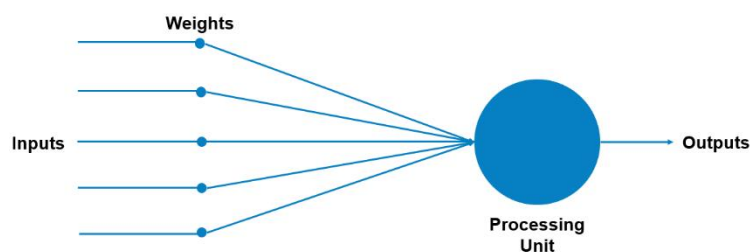
$$Ocosts = \text{Biaya cooling water} + \text{Biaya steam} \quad (2.21)$$

Perhitungan efisiensi *operating costs* :

$$\mu Ocosts (\%) = \frac{Ocosts_{eksisting} - Ocosts_{optimasi}}{Ocosts_{eksisting}} \times 100\% \quad (2.22)$$

2.4 *Neural Network*

Secara umum *Neural Network* (NN) adalah jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan jaringan syaraf manusia. *Neural Network* (NN) ini merupakan sistem adaptif yang dapat merubah strukturnya untuk memecahkan masalah berdasarkan informasi eksternal maupun internal yang mengalir melalui jaringan tersebut. Secara sederhana *Neural Network* (NN) adalah sebuah alat pemodelan data statistik non-linear. *Neural Network* (NN) dapat digunakan untuk memodelkan hubungan yang kompleks antara input dan output untuk menemukan pola-pola pada data. Secara mendasar, sistem pembelajaran merupakan proses penambahan pengetahuan pada *Neural Network* (NN) yang sifatnya kontinuitas sehingga pada saat digunakan pengetahuan tersebut akan dieksploitasikan secara maksimal dalam mengenali suatu objek. *Neuron* adalah bagian dasar dari pemrosesan suatu *Neural Network* (NN) (Buscema et al., 2018). Struktur dasar *neuron* dapat dilihat pada **Gambar 2.7**.



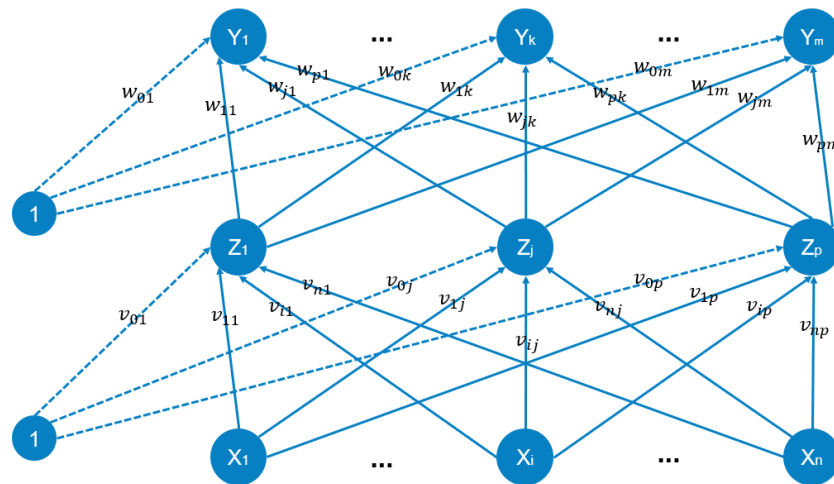
Gambar 2.7 Struktur dasar Neuron (Buscema et al., 2018).

Bagian dari struktur neuron adalah sebagai berikut :

- a. Input, merupakan masukan yang digunakan baik saat pembelajaran maupun dalam mengenali suatu objek.
- b. Wight, beban yang selalu berubah setiap kali diberikan input sebagai proses pembelajaran.
- c. Processing Unit, merupakan tempat berlangsungnya proses pengenalan suatu objek berdasarkan pembebanan yang diberikan.
- d. Output, keluaran dari hasil pengenalan suatu objek

2.4.1. Feed Forward - Backpropagation

Feed Forward - Backpropagation merupakan arsitektur tipe *Neural Network* (NN) yang memiliki algoritma pembelajaran sendiri yaitu *backpropagation*. Arsitektur *Feed Forward – Backpropagation* memiliki struktur *network* seperti pada tipe *network feed forward* pada umumnya. Sehingga dapat dilihat struktur arsitektur *Feed Forward – Backpropagation* pada **Gambar 2.8.** sebagai berikut :



Gambar 2. 8 Arsitektur *Feed Forward – Backpropagation* (Fausett, 2005)

Dengan unit X sebagai input layer, unit Z sebagai hidden layer dan unit Y sebagai output layer dengan posisi bias seperti gambar diatas. *Backpropagation* yang merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh perceptron dengan banyak lapisan untuk mengubah bobot- bobot yang terhubung dengan neuron–neuron yang ada pada lapisan tersembunyinya algoritma *Backpropagation* mengubah *error* output untuk mengubah nilai bobot-bobotnya dalam arah mundur. untuk mendapatkan *error* ini, tahapan perambatan maju harus dikerjakan terlebih dahulu. Pada saat perambatan maju neuron diaktifkan dengan menggunakan fungsi aktivasi yang dapat dideferensiasikan, seperti (Rahmatullah, 2017) :

Fungsi aktivasi Sigmoid :

$$y = f(x) = \frac{1}{1 + e^{-\sigma x}} \quad (2.23)$$

Fungsi aktivasi Tan Sigmoid :

$$y = f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (2.24)$$

Fungsi aktivasi Pure linear :

$$y = f(x) = x \quad (2.25)$$

2.4.2. Merancang Arsitektur *Neural Network* (NN)

Ada beberapa yang harus diperhatikan pada saat merancang arsitektur *Neural Network* (NN) diantaranya :

- Menentukan persentase data pelatihan, dan validasi.
- Menentukan jumlah lapisan (input layer, hidden layer dan output layer).
- Menentukan jumlah neuron pada hidden layer.
- Menentukan jumlah iterasi maksimum.
- Menentukan nilai error goal.
- Menentukan teknik normalisasi data min dan max atau mean dan deviasi.
- Menentukan algoritma pelatihan.

Rumus normalisasi data dengan cara interpolasi data :

$$\frac{X - X_{min}}{X_{max} - X_{min}} = \frac{Y - Y_{min}}{Y_{max} - Y_{min}} \quad (2.26)$$

2.4.3. TRAINLM (*Lavenberg Marquart*)

Algoritma pembelajaran yang digunakan dalam penelitian ini adalah algoritma Levenberg Marquardt. Meskipun algoritma ini lebih kompleks dibandingkan algoritma backpropagation, tetapi algoritma ini dapat memberikan hasil yang lebih baik, terutama pada proses dengan orde banyak (lebih dari satu) (Industri, 2017). Algoritma Lavenberg Marquart dapat diringkas sebagai berikut (Sjoberg, 2001) :

- a. Pilih vector bobot awal $w^{(0)}$ dan harga awal $\lambda^{(0)}$.
- b. Dimana w adalah bobot dan λ diberikan harga awal. Tentukan arah pencarian.

$$[R(w^{(i)} + \lambda^{(i)}I)]f^{(i)} = -G(w^{(i)}) \quad (2.27)$$

diperoleh f dan dimasukkan ke :

$$w = \arg \min V_N(w, Z^N) \quad (2.28)$$

Jika $V_N(w^{(i)} + f^{(i)}, Z^N) < V_N(w^{(i)}, Z^N)$ sehingga memenuhi $w^{(i+1)} = w^{(i)} + f^{(i)}$ sebagai iterasi baru, maka $\lambda^{(i+1)} = \lambda^{(i)}$. Jika tidak maka mencari harga baru dari r

$$r^{(i)} = \frac{V_N(w^{(i)}, Z^N) - V_N(w^{(i)} + f^{(i)}, Z^N)}{V_N(w^{(i)}, Z^N) - L^{(i)}(w^{(i)} + f^{(i)})} \quad (2.29)$$

Jika $r^{(i)} > 0,75$ maka $\lambda^{(i)} = \lambda^{(i)}/2$

Jika $r^{(i)} > 0,25$ maka $\lambda^{(i)} = 2\lambda^{(i)}$

- c. Jika kriteria tercapai, maka perhitungan berhenti. Jika kriteria belum tercapai maka mengulangi langka nomer 2.

2.4.4. LEARNINGDM (*Gradient Descent with Momentum*)

Kriteria yang digunakan untuk menghentikan suatu pelatihan terdapat dua macam cara yaitu :

- Menggunakan tingkat galat yang diinginkan.
- Menentukan jumlah maksimum epoch yang ingin dilakukan.

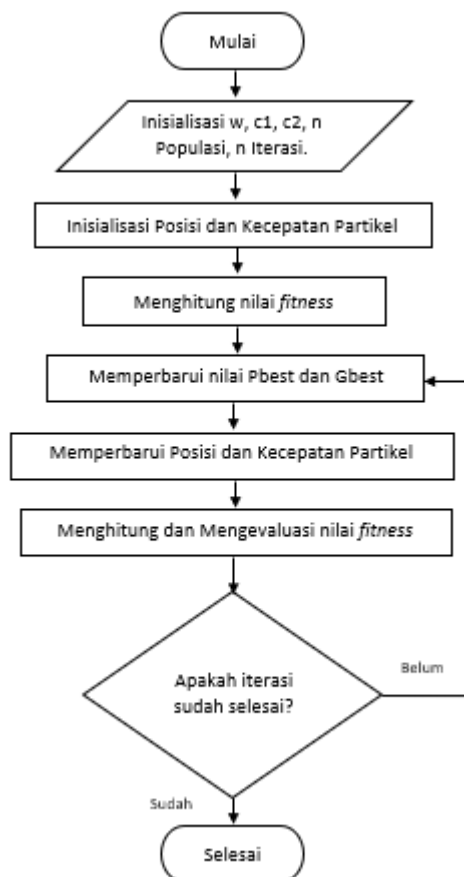
Metode LEARNINGDM ini adalah bagian dari SGD (Stochastic Gradient Descent) yang merupakan solusi untuk menyelesaikan issue GD (Gradient Descent). SGD akan melakukan update weight tanpa menunggu 1 epoch selesai. SGD menggunakan konsep yang mirip dengan batching dengan membagi data training menjadi beberapa batch. Weight akan diupdate dalam setiap batch selesai diproses. Momentum yang dimaksud dari LEARNINGDM ini adalah metoda yang ditambahkan pada GD untuk mencegah optimisasi berhenti pada local minimum (Widyaningrum & Romadhon, 2014).

2.5 PSO (*Particle Swarm Optimization*)

PSO (*Particle Swarm Optimization*) merupakan salah satu dalam *Genetic Algorithm* yang digunakan sebagai metode optimasi berdasarkan data stokastik. Metode PSO secara mendasar termasuk dalam kategori *swarm intelligence* dimana

logika optimasi mengikuti perilaku suatu kelompok partikel (sebagai contoh burung) dalam mencari sumber makanan terdekat.

Dalam konteks optimasi multivariabel, kawanan diasumsikan mempunyai ukuran tertentu atau tetap dengan setiap partikel posisi awalnya yang terletak di suatu lokasi yang acak dalam ruang multidimensi. Setiap partikel diasumsikan memiliki dua parameter, yaitu posisi dan kecepatan. Setiap partikel bergerak dalam ruang tertentu dan mengingat posisi terbaik yang pernah dilalui atau ditemukan terhadap sumber makanan yang akan dituju atau nilai fungsi objektif. Setiap partikel menyampaikan informasi atau posisi terbaiknya kepada partikel yang lain sehingga partikel lain dapat menyesuaikan posisi dan kecepatan masing-masing berdasarkan informasi yang diterima mengenai informasi posisi terbaik tersebut. Berikut merupakan tahap dari PSO (Rahmatullah, 2017) :



Gambar 2. 9 Flowchart Algoritma PSO (Ceylan & Koyuncu, 2019).

2.5.1 Proses Inisiasi

Inisialisasi sekumpulan partikel secara random dimana setiap partikel merepresentasikan solusi yang mungkin untuk masalah optimasi. Pada tahap ini biasanya beberapa parameter particle memiliki batasan-batasan pada masing-masing kasus yang akan dioptimasi. Hal ini dilakukan untuk mencegah nilai random inisialisasi yang terlalu jauh.

Parameter yang sangat penting diinisialisasi pada algoritma PSO adalah posisi dari setiap particle (X_i), kecepatan dari setiap particle (V_i), dan posisi terbaik dari setiap particle (P_i). Ketiga parameter ini dapat berupa lebih dari satu parameter yang akan dioptimasi atau biasa disebut dengan multi objective. Kemudian parameter lainnya yang harus diinisialisasi adalah nilai-nilai konstanta, seperti c_1 , c_2 , r_1 , dan r_2 . Keempat konstanta tersebut dapat diperbaharui setiap iterasi nantinya atau hanya ditentukan sebuah angka (0 sampai 1).

2.5.2 Proses Iterasi dan Pembaharuan Parameter

Pada proses ini parameter-parameter penting yang sudah diinisiasi pada tahap sebelumnya masuk ke dalam tahap pembaharuan nilai-nilai parameter untuk mendapatkan nilai yang terbaik. Adapun formula untuk menghitung nilai-nilai parameter yang akan diperbaharui pada tahap iterasi adalah sebagai berikut (Moravej, Jazaeri, & Gholamzadeh, 2012).

$$v_{k+1} = w_k v_k + c_1 r_1 (P_{best} - x_k) + c_2 r_2 (G_{best} - x_k) \quad (2.28)$$

$$x_{k+1} = x_k + v_k \quad (2.29)$$

Dimana :

w_k = *inertia weight*

c_1 = faktor *learning* untuk *particle*

c_2 = faktor *learning* untuk *swarm*

r_1 = random nilai 1 (0 sampai 1)

r_2 = random nilai 2 (0 sampai 1)

v_k = kecepatan setiap *particle*

x_k = posisi setiap *particle*

v_{k+1} = kecepatan baru setiap *particle*

- x_{k+1} = posisi baru setiap *particle*
 P_{best} = posisi terbaik *particle*
 G_{best} = posisi terbaik dari populasi

Nilai dari w_k disesuaikan dengan permasalahan optimasi yang diinginkan. Hal ini dikarenakan nilai dari formula untuk mencari nilai w_k adalah beragam. Pemilihan nilai w_k yang tepat akan mempercepat proses optimasi pada permasalahan yang diambil karena peran dari w_k adalah pengontrol keseimbangan antara eksplorasi global dan local. Nilai w_k yang tinggi menambah porsi pencarian global (global exploration), sedangkan nilai w_k yang rendah lebih menekankan pencarian lokal (local search). Untuk tidak terlalu menitikberatkan pada salah satu bagian dan tetap mencari area pencaarian yang baru dalam ruang berdimensi tertentu, maka perlu dicari nilai bobot inersia w_k yang secara seimbang menjaga pencarian global dan lokal. Untuk mencapai itu dan mempercepat konvergensi, suatu bobot inersia yang mengecil nilainya dengan bertambahnya iterasi digunakan dengan formula (Ji et al., 2009) :

$$w_k(i) = w_{k \max} - \left(\frac{w_{k \max} - w_{k \min}}{i_{\max}} \right) i \quad (2.29)$$

Dimana :

- $w_k(i)$ = *inertia weight* pada iterasi ke – i.
 $w_{k \max}$ = *inertia weight* maksimum.
 $w_{k \min}$ = *inertia weight* minimum.
 i_{\max} = iterasi maksimum.
 i = iterasi ke – i.

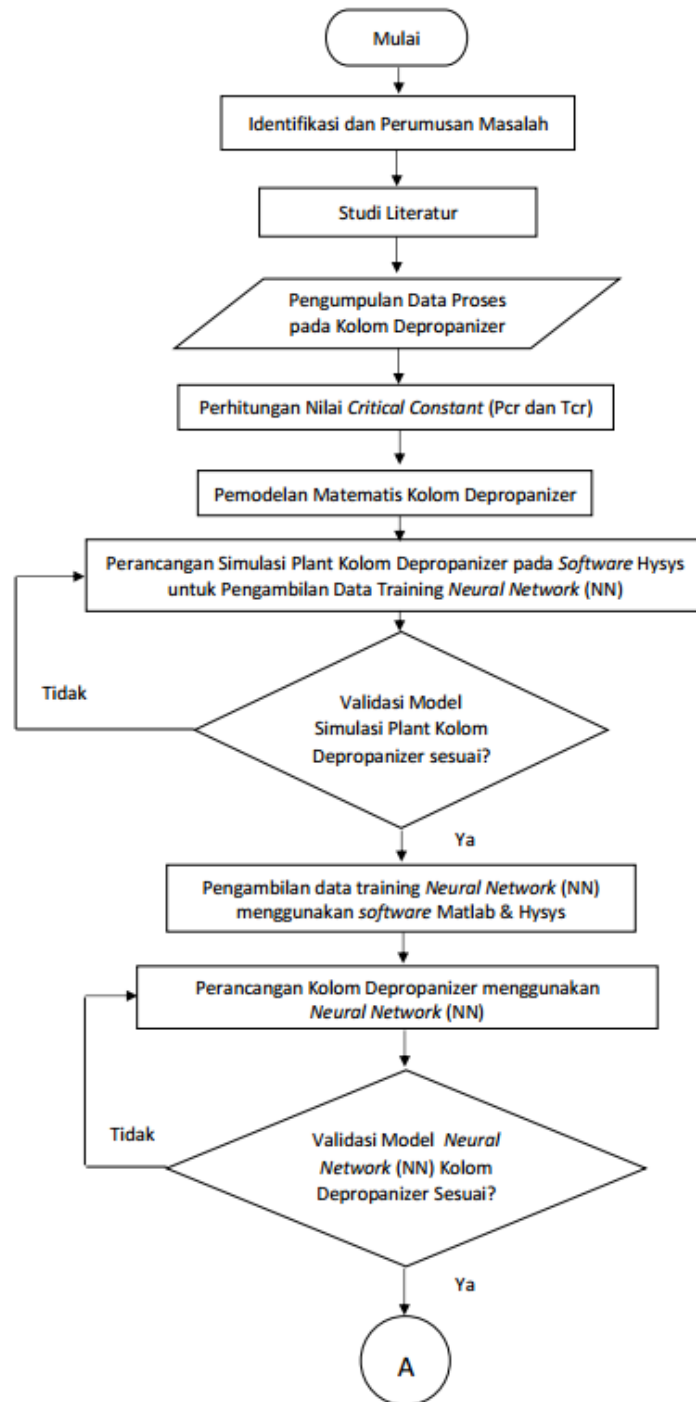
dimana $w_{k \max}$ dan $w_{k \min}$ masing-masing adalah nilai awal dan nilai akhir bobot inersia, i_{\max} adalah jumlah iterasi maksimum yang digunakan dan i adalah iterasi yang sekarang. Biasanya digunakan nilai $\theta_{\max} = 0.9$ dan $\theta_{\min} = 0.4$ (Ji et al., 2009).

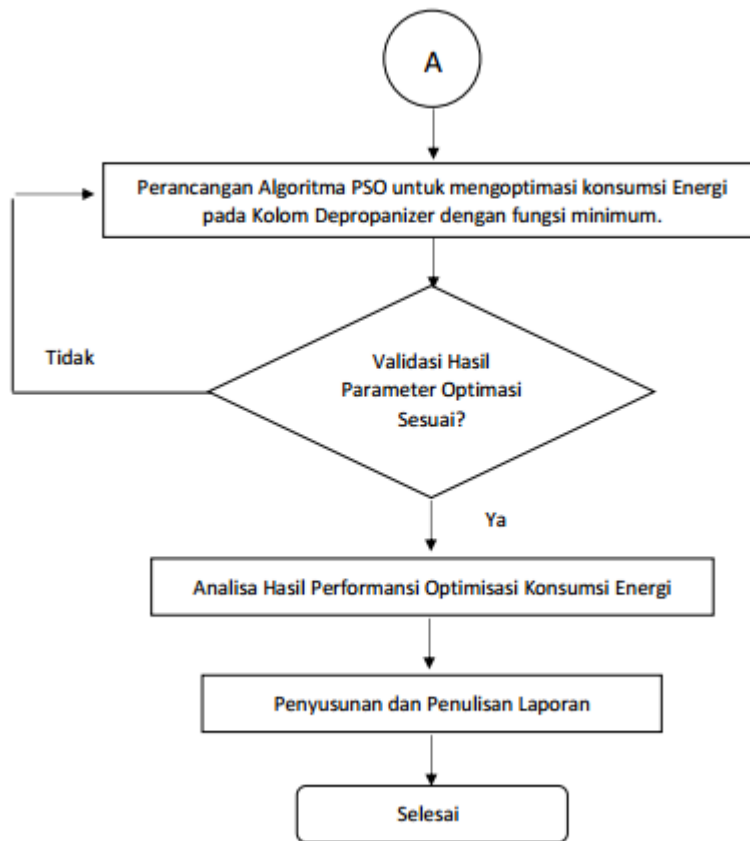
BAB III

METODOLOGI PENELITIAN

3.1 Diagram Alir Penelitian

Adapun diagram alir penelitian yang digunakan pada tugas akhir ini adalah seperti berikut :





Gambar 3. 1 Diagram Alir Metode Penelitian Tugas Akhir

3.2 Pengumpulan Data Proses Kolom Depropanizer

Pada penelitian ini proses yang digunakan dimodelkan menggunakan perangkat lunak Aspen Hysys V8.8. Data yang dikumpulkan meliputi depropanizer *column internal* datasheet yang berisikan kondisi masing-masing *stream*, komposisi masing-masing komponen pada setiap *stream* dan skema PFD eksisting yang merupakan data sekunder dari Penelitian (Ching et al., 2016). Tipe condensor yang digunakan pada penelitian (Ching et al., 2016) menggunakan Total/Full Condensing. Data proses Kolom Depropanizer dari penelitian (Ching et al., 2016) dapat dilihat pada **Tabel 3.1** dan **Tabel 3.2**.

Tabel 3. 1 Kondisi operasional Kolom Depropanizer pada *Feed Stream* (Ching et al., 2016).

Variabel	Nilai	Satuan
T_f (<i>Temperature feed</i>)	71,11	°C

P_f (Pressure feed)	16	Bar
\dot{F} (Flow feed)	890,3	kgmol/h
Composition (mole fraction)		
Ethane	(C ₂)	0,0001
Propane	(C ₃)	0,5959
<i>i</i> -Butane	(iC ₄)	0,1638
<i>n</i> -Butane	(nC ₄)	0,1214
<i>i</i> -Pentane	(iC ₅)	0,0523
<i>n</i> -Pentane	(nC ₅)	0,0271
<i>n</i> -Hexane	(nC ₆)	0,0192
<i>n</i> -Heptane	(nC ₇)	0,0113
<i>n</i> -Octane	(nC ₈)	0,0070
<i>n</i> -Nonane	(nC ₉)	0,0019

Tabel 3. 2 Spesifikasi Kolom Depropanizer (Ching et al., 2016).

Variabel	Nilai	Satuan
N_t (Number of trays)	50	Tray
N (Feed tray location)	25	Feed 1 and 2
$R_T C_3$ (Recovery of C3 at top stream)	0,999	Mole fraction
$R_B C_4$ (Recovery of C4 at bottom stream)	0,990	Mole fraction
\dot{D} (Molar Flow of top product stream)	531,6	kgmol/h
\dot{B} (Molar Flow of bottom product stream)	358,8	kgmol/h
P_c (Pressure of condenser)	14,91	Bar
P_r (Pressure of reboiler)	16,20	Bar

3.3 Perhitungan Nilai Critical Constant (P_{cr} dan T_{cr}).

Pada tugas akhir ini, optimisasi konsumsi energi pada Kolom Depropanizer dilakukan dengan mencari kondisi/nilai dari variabel optimasi (*flow feed*, *temperature feed*, *pressure condenser*, dan *pressure reboiler*) yang akan menghasilkan konsumsi energi minimum pada *equipment condenser* dan *reboiler*. Oleh karena itu perlu dilakukan perhitungan nilai *Critical Constant* (P_{cr} dan T_{cr})

untuk menentukan apakah hasil optimisasi konsumsi energi dengan nilai variabel *pressure* dan *temperature* tertentu memungkinkan untuk diterapkan atau diaplikasikan dan juga untuk dilakukan pengecekan pada kondisi eksisting aliran *feed stream*. Perhitungan nilai *Critical Constant* P_{cr} dan T_{cr} menggunakan **Persamaan 2.16** Berikut merupakan tabel perhitungan nilai *Critical Constant* pada *feed stream*.

Tabel 3. 3 Perhitungan nilai *Critical Constant* pada *feed stream*.

<i>Compounds</i>		<i>Mole Fraction</i>	Pcr (kPa)	Pcr*yi	Tcr (°K)	Tcr*yi
<i>Ethane</i>	(C ₂)	0,0001	4880	0,488	305,4	0,03054
<i>Propane</i>	(C ₃)	0,5959	4249	2531,979	369,8	220,3638
<i>i-Butane</i>	(iC ₄)	0,1638	3648	597,5424	408,2	66,86316
<i>n-Butane</i>	(nC ₄)	0,1214	3797	460,9558	425,2	51,61928
<i>i-Pentane</i>	(iC ₅)	0,0523	3381	176,8263	460,4	24,07892
<i>n-Pentane</i>	(nC ₅)	0,0271	3369	91,2999	469,7	12,72887
<i>n-Hexane</i>	(nC ₆)	0,0192	3012	57,8304	507,4	9,74208
<i>n-Heptane</i>	(nC ₇)	0,0113	2736	30,9168	540,3	6,10539
<i>n-Octane</i>	(nC ₈)	0,007	2486	17,402	568,8	3,9816
<i>n-Nonane</i>	(nC ₉)	0,0019	2289	4,3491	594,6	1,12974
			$\sum_i^n P_{cr i} * y_i$	3969,59	$\sum_i^n T_{cr i} * y_i$	396,6434

Dari hasil perhitungan diatas dapat dilihat nilai P_{cr} pada *feed stream* sebesar 3969,59 kPa dan nilai T_{cr} sebesar 396,64 °K (123,64°C), sehingga dapat ditentukan bahwa aliran *feed stream* Kolom Depropanizer pada kondisi eksisting dengan nilai *pressure* 16 bar dan *temperature* sebesar 71,11°C memungkinkan untuk diaplikasikan karena masih dibawah nilai P_{cr} dan T_{cr} . Dari hasil perhitungan nilai P_{cr} dan T_{cr} juga dapat digunakan sebagai pertimbangan untuk range variabel *pressure* dan *temperature* pada proses optimisasi agar tidak melebihi nilai *Critical constant pressure* dan *temperature*.

3.4 Pemodelan Matematis Kolom Depropanizer

Berdasarkan **Tabel 3.1** dan **Tabel 3.2**, diketahui nilai \dot{F} (*Flow feed*), \dot{D} (*Molar Flow of top product stream*), dan \dot{B} (*Molar Flow of bottom product stream*). Sehingga spesifikasi Kolom Depropanizer *Distillate to feed ratio* dan *Bottom Product to feed ratio* dapat dihitung menggunakan **Persamaan 2.14** dan **Persamaan 2.15**, dan ditampilkan pada perhitungan **Persamaan 3.1** dan **Persamaan 3.2** sebagai berikut :

$$\text{Distillate to feed ratio} = \frac{\dot{D}}{\dot{F}} = \frac{531,6 \text{ (kgmole/h)}}{890,3 \text{ (kgmole/h)}} = 0,597 \quad (3.1)$$

$$\text{Bottom product to feed ratio} = \frac{\dot{B}}{\dot{F}} = \frac{358,8 \text{ (kgmole/h)}}{890,3 \text{ (kgmole/h)}} = 0,403 \quad (3.2)$$

Dari nilai rasio diatas akan digunakan untuk perhitungan perubahan *flowrate Distillate* dan *flowrate Bottom product* :

$$\dot{D}_{new} = \text{Distillate to feed ratio} \times \dot{F}_{new} = 0,597 \times \dot{F}_{new} \quad (3.3)$$

$$\dot{B}_{new} = \text{Bottom product to feed ratio} \times \dot{F}_{new} = 0,403 \times \dot{F}_{new} \quad (3.4)$$

Sehingga perhitungan konsumsi energi *Condenser* dan *Reboiler* per satuan laju massa dapat ditentukan sebagai berikut :

$$E_{C_{new}} = \frac{\dot{Q}_{c_{new}} \text{ (kW)}}{\dot{D}_{new} \text{ (kgmole/h)}} \quad (3.5)$$

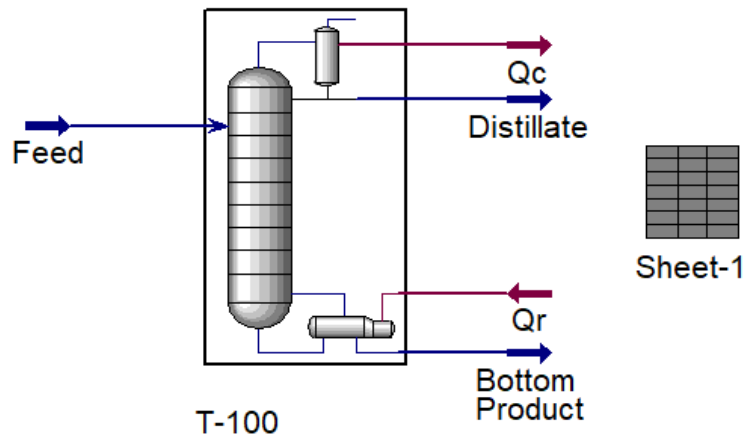
$$E_{R_{new}} = \frac{\dot{Q}_{r_{new}} \text{ (kW)}}{\dot{B}_{new} \text{ (kgmole/h)}} \quad (3.6)$$

Dari seluruh perhitungan dan pemodelan matematis Kolom Depropanizer diatas kemudian akan digunakan pada pemodelan algoritma PSO untuk proses optimisasi konsumsi energi.

3.5 Perancangan Simulasi Plant Kolom Depropanizer pada Software Hysys untuk Pengambilan Data Training Neural Network (NN).

Data proses yang didapatkan dari penelitian (Ching et al., 2016) pada **Tabel 3.1** dan **Tabel 3.2** kemudian digunakan untuk perancangan simulasi plant Kolom Depropanizer menggunakan *software* Aspen Hysys V8.8.. Dari hasil perancangan ini akan digunakan untuk proses pengambilan data training *Neural Network* (NN) untuk pemodelan plant Kolom Depropanizer. Perancangan simulasi plant dilakukan

secara *steady state* dengan menggunakan Peng-Robinson *equation of state*, berikut **Gambar 3.2** merupakan tampilan simulasi kolom depropanizer saat kondisi *steady state*.



Gambar 3. 2 Model PFD Steady State Kolom Depropanizer pada software Hysys.

Berikut merupakan hasil simulasi *steady state* Kolom Depropanizer pada kondisi eksisting dapat disajikan pada **Tabel 3.4** :

Tabel 3. 4 Hasil simulasi *steady state* Kolom Depropanizer pada kondisi eksisting.

Stream	Variabel	Nilai	Satuan
Distillate Stream	T_t (Temperature Distillate)	43,72	°C
	P_t (Pressure Distillate)	14,91	Bar
	\dot{D} (Distillate flow)	532,6	kgmol/h
	Composition (mole fraction)		
	Ethane	(C ₂)	0,0002
	Propane	(C ₃)	0,9951
	<i>i</i> -Butane	(iC ₄)	0,0047
	<i>n</i> -Butane	(nC ₄)	0,0000
	<i>i</i> -Pentane	(iC ₅)	0,0000
	<i>n</i> -Pentane	(nC ₅)	0,0000
<i>n</i> -Hexane	(nC ₆)	0,0000	
<i>n</i> -Heptane	(nC ₇)	0,0000	
<i>n</i> -Octane	(nC ₈)	0,0000	

	<i>n-Nonane</i>	(nC ₉)	0,0000
	<i>T_b (Temperature Bottom Product)</i>	110,1	°C
	<i>P_b (Pressure Bottom Product)</i>	16,2	Bar
	<i>\dot{B} (Bottom Product flow)</i>	357,7	kgmol/h
	<i>Composition (mole fraction)</i>		
Bottom Product Stream	<i>Ethane</i>	(C ₂)	0,0000
	<i>Propane</i>	(C ₃)	0,0015
	<i>i-Butane</i>	(iC ₄)	0,4006
	<i>n-Butane</i>	(nC ₄)	0,3022
	<i>i-Pentane</i>	(iC ₅)	0,1302
	<i>n-Pentane</i>	(nC ₅)	0,0675
	<i>n-Hexane</i>	(nC ₆)	0,0478
	<i>n-Heptane</i>	(nC ₇)	0,0281
	<i>n-Octane</i>	(nC ₈)	0,0174
	<i>n-Nonane</i>	(nC ₉)	0,0047
Qc	Heat Flow	2,091 x 10 ⁷	kJ/h
Strean	Power	5809	kW
Qr	Heat Flow	1,737 x 10 ⁷	kJ/h
Stream	Power	4826	kW

3.6 Validasi Model Simulasi Plant Kolom Depropanizer pada *software* Hysys.

Validasi hasil perancangan model simulasi plant Kolom Depropanizer pada *software* Hysys ini diperlukan untuk menentukan apakah model simulasi Kolom Depropanizer dapat dikatakan benar dan valid. Validasi dilakukan dengan membandingkan beberapa parameter hasil simulasi Hysys dengan data dari penelitian (Ching et al., 2016). Berikut merupakan tabel perbandingan parameter hasil simulasi Hysys dengan data dari penelitian (Ching et al., 2016) ditampilkan pada **Tabel 3.5**:

Tabel 3. 5 Validasi model simulasi plant Kolom Depropanizer pada software Hysys.

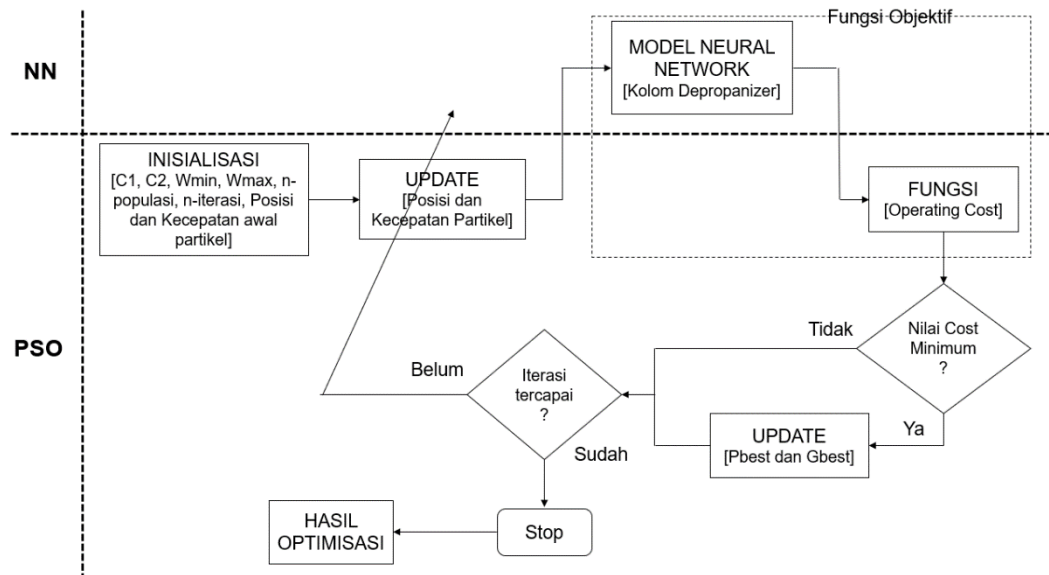
<i>Stream</i>	<i>Variabel</i>	Data penelitian (Ching et al., 2016)	Hysys	Error (%)
<i>Feed Stream</i>	<i>Temperature</i>	71,11 °C	71,11 °C	0
	<i>Pressure</i>	16 Bar	16 Bar	0
	<i>Molar flow</i>	890,3 kgmole/h	890,3 kgmole/h	0
<i>Distillate Stream</i>	<i>Temperature</i>	-	43,72 °C	-
	<i>Pressure</i>	14,91 Bar	14,91 Bar	0
	<i>Molar flow</i>	531,6 kgmole/h	532,6 kgmole/h	0,188
<i>Bottom Product Stream</i>	<i>Temperature</i>	-	43,72 °C	-
	<i>Pressure</i>	16,2 Bar	16,2 Bar	0
	<i>Molar flow</i>	358,8 kgmole/h	357,7kgmole/h	0,307

Dari **Tabel 3.5.** diatas, dapat diketahui hasil validasi model simulasi plant Kolom Depropanizer pada *software* Hysys secara keseluruhan menghasilkan nilai error dibawah 1%, nilai error ini sangat kecil, sehingga ketika model simulasi plant ini digunakan untuk pengambilan data training *Neural Network* (NN), nilai error dari hasil pengambilan data dapat ditoleransi, sehingga hasil perancangan model simulasi plant Kolom Depropanizer dapat dikatakan valid.

3.7 Perancangan Arsitektur *Neural Network* – *Particle Swarm Optimization* (NN-PSO).

Secara keseluruhan proses optimasi konsumsi energi pada Kolom Depropanizer menggunakan arsitektur integrasi *Neural Network* – *Particle Swarm Optimization* (NN-PSO). Pada arsitektur integrasi NN-PSO, *Neural Network* (NN) digunakan sebagai fungsi objektif pada algoritma optimasi PSO. Dengan kemampuan *Neural Network* (NN) yang dapat memodelkan input-output yang non linier, *Neural Network* (NN) digunakan untuk menghitung nilai Q_c (*Heat flow condenser*) dan Q_r (*Heat flow reboiler*) dari variabel input *Neural Network* (NN)

(yaitu variabel yang di optimasi). Berikut merupakan skema optimasi NN-PSO dapat ditunjukkan pada **Gambar 3.3**.



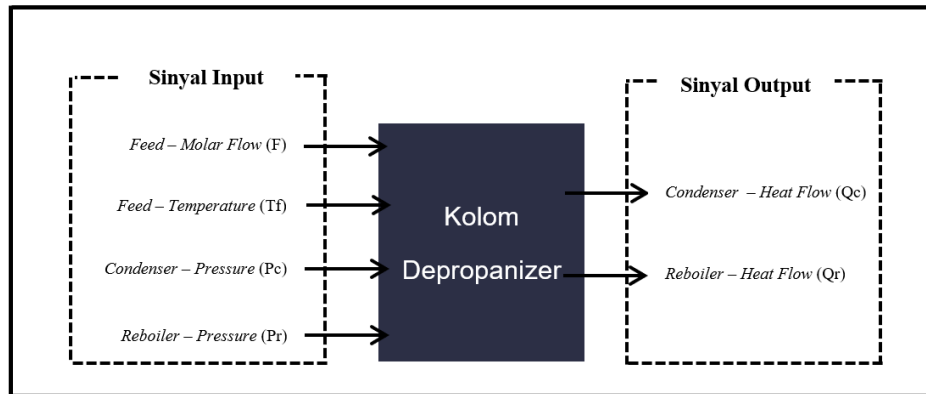
Gambar 3.3 Skema optimasi berbasis NN-PSO.

Untuk merancang arsitektur integrasi NN-PSO terdapat 3 tahapan, dengan urutan sesuai pada point 3.6.1 , 3.6.2 , 3.6.3.

3.7.1 Pengambilan Data Training *Neural Network* (NN) menggunakan software Matlab & Hysys.

Pengambilan data training *Neural Network* (NN) dilakukan dengan mengintegrasikan *software* Matlab dan Hysys. *Software* Matlab digunakan sebagai operator yang memberikan input sinyal random ke model simulasi plant Kolom Depropanizer pada *software* Hysys serta merekam hasil sinyal output dari *software* Hysys. Sedangkan *software* Hysys digunakan untuk memproses atau mengkalkulasi sinyal input dari *software* Matlab dan menghasilkan sinyal output.

Terdapat 4 variabel pada sinyal input yaitu *Feed – Molar Flow* (F), *Feed – Temperature* (Tf), *Condenser – Pressure* (Pc) dan *Reboiler – Pressure* (Pr) serta 2 variabel pada sinyal output yaitu Qr (Heat Flow Reboiler) dan Qc (Heat Flow Condenser). Pengambilan data training *Neural Network* (NN) dapat diilustrasikan pada **Gambar 3.4**.



Gambar 3. 4 Pemodelan input-output Kolom Depropanizer.

Sinyal input yang diberikan berupa sinyal random dari *software* Matlab yang dilakukan sebanyak 10000x *looping* agar dapat menghasilkan pasangan data input – output sebanyak 10000 data. Berikut rincian pengambilan data input – output dapat dilihat pada **Tabel 3.6** dan untuk source code pengambilan data training *Neural Network* (NN) dapat dilihat pada **Lampiran C.1**.

Tabel 3. 6 Rincian pengambilan pasangan data input - output.

Variabel Optimasi	Range
Feed – Molar Flow (F)	700 – 1100 (kgmole/h)
Feed – Temperature (Tf)	55 – 105 (°C)
Condenser – Pressure (Pc)	1440 – 1540 (kPa)
Reboiler – Pressure (Pr)	1570 – 1670 (kPa)
Komposisi Data	
Jumlah Data	10000 data
Data training <i>Neural Network</i>	7500 data
Data validasi <i>Neural Network</i>	2500 data

Data yang sudah diperoleh kemudian dibagi menjadi 2 bagian yaitu data training (pelatihan) dan data validasi, dimana data yang digunakan untuk proses pelatihan lebih banyak dibandingkan proses validasi, dengan rincian 75% dari total keseluruhan data untuk pelatihan sedangkan sisanya 25% untuk validasi. Data-data tersebut nantinya akan berfungsi sebagai masukan neural network dan juga sebagai validasi dari hasil keluaran neural network.

3.7.2 Pemodelan Kolom Depropanizer dengan *Neural Network* (NN).

Pada penelitian ini pemodelan proses Kolom Depropanizer dilakukan dengan menggunakan struktur *Feed Forward - Backpropagation*. Terdiri dari 1 layer input, 1 layer hidden dan 1 layer output dengan fungsi aktivasi Tansig pada hidden neuron dan Purelinear pada output neuron.

Fungsi Aktivasi Tan Sigmoid (pada *hidden layer*) :

$$y = f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (3.7)$$

Fungsi Aktivasi Pure Linier (pada *output layer*) :

$$y = f(x) = x \quad (3.8)$$

Pada Perancangan Arsitektur Neural Network dapat dijelaskan pada **Tabel 3.7**.

Tabel 3. 7 Arsitektur Neural Network.

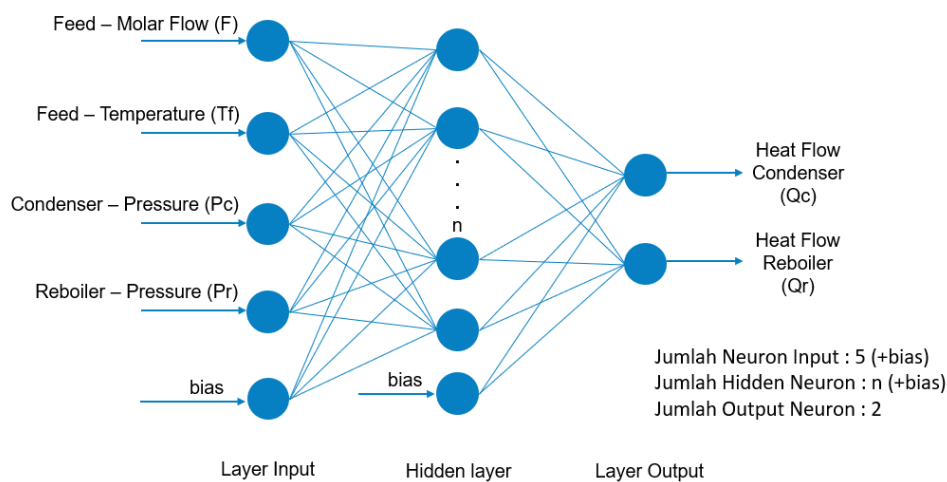
No.	Arsitektur	Type
1	Network type	Feed Forward - Backpropagation
2	Training Function	TRAINLM (Lavenberg Marquart)
3	Adaption Learning Function	LEARNLMD (Gradient Descent with Momentum)
4	Input Neurons	<ul style="list-style-type: none"> • <i>Feed – Molar Flow</i> (F). • <i>Feed – Temperature</i> (Tf). • <i>Condenser – Pressure</i> (Pc). • <i>Reboiler – Pressure</i> (Pr). Jumlah input <i>neuron</i> sebanyak 4
5	Hidden Neurons	Dilakukan jumlah variasi <i>hidden neuron</i> sebanyak 4 sampai 20 <i>neuron</i>
6	Output Neurons	<ul style="list-style-type: none"> • <i>Condenser Duty – Heat Flow</i> (Qc). • <i>Reboiler Duty – Heat Flow</i> (Qr). Jumlah output <i>neuron</i> sebanyak 2
7	Performance Parameter	MSE (Mean Square Error)
8	Fungsi Aktivasi	Tan Sigmoid pada <i>hidden layer</i> Pure linear pada <i>output layer</i>
9	Jumlah Pasangan Data	10000
10	Komposisi Data	<ul style="list-style-type: none"> • 75% Data Training [5x7500] Input Training [2x7500] Target Training

- 25% Data Validasi
[5x2500] Input Validasi
[2x2500] Target Validasi

Parameter Training

1	Epochs	1000
2	Time	infinite
3	Error Goal	1e-25
4	Minimum Gradient	1e-10

Dari tabel rincian arsitektur *Neural Network* (NN) diatas, maka dapat dibuat model *network* dapat dilihat pada **Gambar 3.5**.



Gambar 3. 5 Arsitektur *Neural Network*

Pada model *Neural Network* (NN) diatas, pada *layer input* terdiri dari 4 variabel diantaranya : *Feed – Molar Flow* (F), *Feed – Temperature* (Tf), *Condenser – Pressure* (Pc) dan *Reboiler – Pressure* (Pr). Pada *hidden layer* dilakukan variasi jumlah *hidden neuron* 4 sampai 20 *neuron*. Dan pada *layer output* terdiri dari 2 variabel yaitu : *Condenser - Heat Flow* (Qc), dan *Reboiler – Heat Flow* (Qr).

Sebelum dilakukan proses training pada model *Neural Network* (NN), data training dan data validasi di lakukan normalisasi terlebih dahulu pada setiap variabelnya agar menghasilkan model yang baik dengan nilai RMSE yang kecil menggunakan rumus interpolasi pada **Persamaan 3.9** berikut :

$$\frac{X - X_{min}}{X_{max} - X_{min}} = \frac{Y - Y_{min}}{Y_{max} - Y_{min}} \quad (3.9)$$

Tabel 3.8 berikut merupakan rincian nilai maksimum dan minimum pada setiap variabel operasional Kolom Depropanizer :

Tabel 3. 8 Tabel parameter normalisasi model *Neural Network* (NN).

Variabel Optimasi	Parameter Normalisasi			
Feed – Molar Flow (F) [kgmole/h]	Xmin	700	Ymin	0
	Xmax	1100	Ymax	1
Feed – Temperature (Tf) [°C]	Xmin	55	Ymin	0
	Xmax	105	Ymax	1
Condenser – Pressure (Pc) [kPa]	Xmin	1440	Ymin	0
	Xmax	1540	Ymax	1
Reboiler – Pressure (Pr) [kPa]	Xmin	1570	Ymin	0
	Xmax	1670	Ymax	1
Condenser – Heat Flow (Qc) [kW]	Xmin	0	Ymin	0
	Xmax	10000	Ymax	1
Reboiler – Heat Flow (Qr) [kW]	Xmin	0	Ymin	0
	Xmax	10000	Ymax	1

Validasi model *Neural Network* (NN) yang telah dibuat dilakukan dengan memberikan input yang belum pernah dilatihkan kepada *Neural Network* (NN). Baik tidaknya model *Neural Network* (NN) yang dibuat dapat dilihat dari *Root Mean Squared Error* (RMSE) yang diperoleh sepanjang N sample. Formula *Root Mean Squared Error* (RMSE), dapat ditulis sebagai berikut :

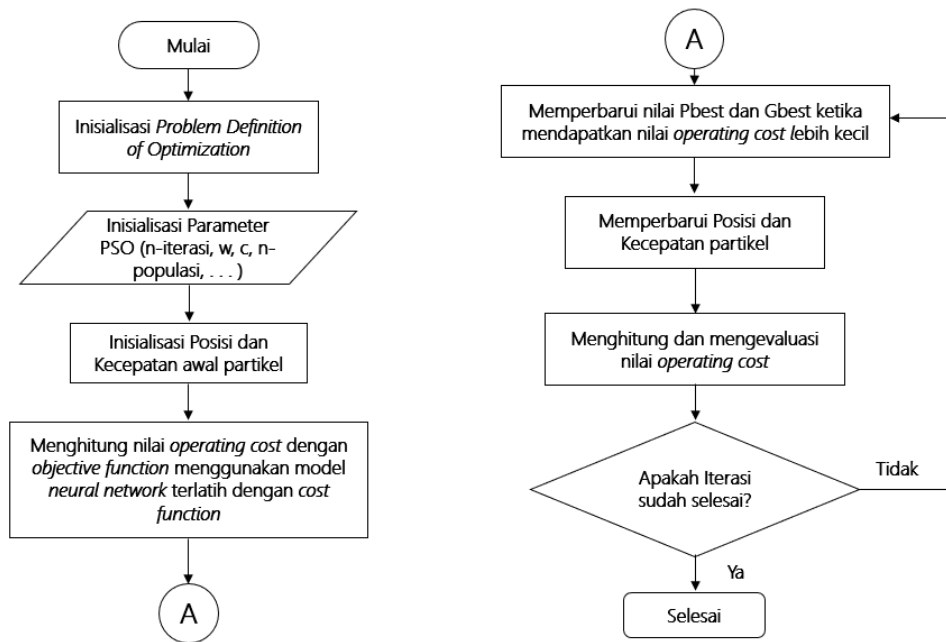
$$RMSE = \sqrt{\frac{\sum_{i=1}^N (y - y_i)^2}{N}} \quad (3.10)$$

Simulasi dilakukan dengan menggunakan nntool Matlab, dengan variasi jumlah *hidden neuron* sebanyak 4 sampai 20 *neuron*. Sehingga dapat ditentukan dari variasi jumlah *hidden neuron* akan dicari dengan hasil RMSE terkecil, dan kemudian akan di gunakan sebagai fungsi objektif pada Algoritma PSO.

3.7.3 Perancangan Algoritma PSO.

Pada saat melakukan optimasi konsumsi energi pada Kolom Depropanizer menggunakan algoritma PSO, dilakukan beberapa langkah guna

untuk mencapai nilai Q_c dan Q_r yang minimum. Flowchart dalam penggunaan algoritma PSO pada optimisasi konsumsi energi dilakukan untuk mempermudah penyusunan algoritma. Flowchart algoritma PSO disusun seperti **Gambar 3.6**.



Gambar 3. 6 Flowchart Perancangan Algoritma PSO.

Penjelasan tentang sistematika penyusunan flowchart algoritma PSO untuk optimisasi konsumsi energi pada Kolom Depropanizer dapat dijelaskan sebagai berikut :

a) *Inisialisasi Problem Definition of Optimization.*

Pada tahap ini, dilakukan penjabaran permasalahan yang akan di optimasi, yaitu dengan menentukan jumlah variabel yang di optimasi sebagai komponen dalam posisi, menentukan range operasional dari variabel yang di optimasi. Pada tugas akhir ini digunakan 4 variabel yang akan di optimasi dengan range operasional sebagai berikut :

Tabel 3. 9 Range operasional variabel optimasi.

Variabel Optimasi	Range
Feed – Molar Flow (F)	700 – 1100 (kgmole/h)
Feed – Temperature (Tf)	55 – 105 (°C)
Condenser – Pressure (Pc)	1440 – 1540 (kPa)
Reboiler – Pressure (Pr)	1570 – 1670 (kPa)

b) Inisialisasi Parameter PSO.

Pada tahap ini, dilakukan proses inisialisasi parameter PSO diantaranya : n-iterasi, n Populasi, w_k (weight inertia), c_1 (faktor *learning* untuk *particle*), c_2 (faktor *learning* untuk *swarm*), pada tahap ini akan dilakukan inisialisasi ulang jika hasil optimasi belum valid atau sesuai dengan mengubah nilai w_k , c_1 , dan c_2 dengan melakukan 3x running dengan nilai paramter yang sama. berikut inisialisasi parameter PSO dapat dijabarkan pada **Tabel 3.10** :

Tabel 3. 10 Inisialisasi Parameter PSO.

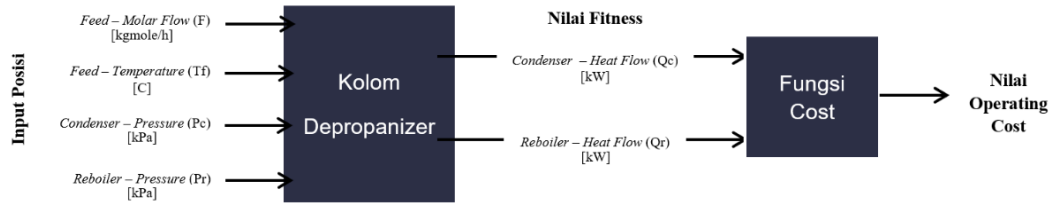
Parameter PSO	Nilai
n – Iterasi	1000
n – Populasi	25
$w_{k \max}$ (weight inertia maksimum)	0,9
$w_{k \min}$ (weight inertia minimum)	0,4
c_1 (faktor <i>learning</i> untuk <i>particle</i>)	0,001 ; 0,0015 ; 0,002
c_2 (faktor <i>learning</i> untuk <i>swarm</i>)	0,001 ; 0,0015 ; 0,002

c) Inisialisasi Posisi dan Kecepatan awal partikel.

Pada tahap ini, dilakukan inisialisasi Posisi dan Kecepatan awal partikel, Posisi partikel di inisialisasikan dengan variabel optimasi yaitu F, Tf, Pc, Pr. inisialisasi awal posisi partikel menggunakan bilangan random sesuai dengan range nya pada **Tabel 3.10**. sedangkan kecepatan awal setiap partikel di inisialisasikan memiliki kecepatan awal nol.

d) Menghitung nilai *operating costs* dengan fungsi objektif.

Algoritma PSO yang dirancang menghitung nilai *operating costs* dengan fungsi objektif menggunakan model *Neural Network* (NN) yang sudah terlatih untuk menghitung nilai Qc dan Qr, selanjutnya akan dihitung biaya utilitas *steam* dan *cooling water* untuk menentukan biaya yang minimum (sebagai fungsi minimum).



Gambar 3. 7 Fungsi Objektif Algoritma PSO.

Tabel 3. 11 Tabel biaya utilitas.

Utility Data	Harga (\$(kW.year))
Cooling Water	90
Steam	300

$$\begin{aligned}
 \text{Nilai Fitness} &= Q_c(kW) \text{ dan } Q_r(kW) \\
 \text{Ocosts} \left(\frac{\$}{kgmole} \right) &= \frac{Q_c(kW) * 90 \left(\frac{\$}{kW \cdot year} \right)}{\dot{D} \left(\frac{kgmole}{h} \right)} + \frac{Q_r(kW) * 300 \left(\frac{\$}{kW \cdot year} \right)}{\dot{B} \left(\frac{kgmole}{h} \right)} \\
 &= \frac{Q_c(kW) * 90 \left(\frac{\$}{kW \cdot year} \right)}{8766 * \dot{D} \left(\frac{kgmole}{year} \right)} + \frac{Q_r(kW) * 300 \left(\frac{\$}{kW \cdot year} \right)}{8766 * \dot{B} \left(\frac{kgmole}{year} \right)} \\
 &= \left(\frac{Q_c * 90}{8766 * \dot{D}} + \frac{Q_r * 300}{8766 * \dot{B}} \right) \left(\frac{\$}{kgmole} \right) \tag{3.11}
 \end{aligned}$$

e) Memperbarui nilai P_{best} dan G_{best}

Nilai P_{best} dari setiap partikel akan diperbarui ketika mendapatkan nilai *operating costs* yang lebih kecil atau minimum, dan setiap partikel akan mengingat posisi terbaiknya, nilai G_{best} akan diperbarui ketika mendapatkan nilai *operating costs* yang lebih kecil dari semua partikel pada setiap iterasi. sehingga pada akhir iterasi (1000) didapatkan *operating costs* yang paling minimum yang linier dengan besarnya energi.

f) Memperbarui Posisi dan Kecepatan partikel

Proses update beberapa parameter PSO terjadi pada tahap ini. Parameter yang diperbaharui antara lain adalah posisi (x), dan kecepatan (v). **Persamaan 3.12** dan **Persamaan 3.13** berikut digunakan untuk memperbaharui nilai dari parameter posisi (x), dan kecepatan (v).

$$v_{k+1} = w_k v_k + c_1 r_1 (P_{best} - x_k) + c_2 r_2 (G_{best} - x_k) \tag{3.12}$$

$$x_{k+1} = x_k + v_k \quad (3.13)$$

g) Menghitung dan mengevaluasi nilai *operating costs*.

Pada tahap ini, dilakukan perhitungan dan evaluasi nilai *operating costs* dengan fungsi objektif menggunakan model *Neural Network* yang sudah terlatih untuk menghitung nilai Qc dan Qr serta menghitung nilai *operating costs* nya. Sama pada tahap d).

h) Mengecek n-iterasi.

Pada tahap ini, algoritma PSO akan melakukan pengecekan jumlah iterasi yang sudah dilakukan, ketika iterasi mencapai batas maksimum yang ditentukan maka algoritma PSO akan memberhentikan iterasi, dan ketika jumlah iterasi belum tercapai maka algoritma PSO akan melakukan looping untuk melakukan iterasi selanjutnya hingga tercapai iterasi maksimum.

Sehingga dari penjelasan flowchart diatas dapat dilakukan proses pembuatan source code algoritma PSO pada software Matlab dapat dilihat pada **Lampiran C.2**.

3.8 Validasi Hasil Optimisasi Konsumsi Energi.

Setelah dilakukan proses optimisasi konsumsi energi menggunakan algoritma PSO, selanjutnya akan dilakukan validasi hasil optimisasi konsumsi energi untuk membuktikan bahwa parameter hasil optimisasi konsumsi energi dapat dikatakan benar. Metode validasi yang digunakan adalah dengan mengujikan hasil parameter optimisasi (*Feed – Molar Flow (F)*, *Feed – Temperature (Tf)*, *Condenser – Pressure (Pc)* dan *Reboiler – Pressure (Pr)*) ke model proses Kolom Depropanizer pada *software* Hysys serta membandingkan nilai Qr (*Heat Flow Reboiler*) dan Qc (*Heat Flow Condenser*).

3.9 Analisa Performansi Hasil Optimisasi Konsumsi Energi.

Dari hasil optimisasi konsumsi energi menggunakan algoritma PSO, akan dicari hasil terbaik yaitu nilai *operating costs* minimum, nilai *operating costs* direpresentasikan sebagai besarnya biaya konsumsi energi (\$) persatuan massa

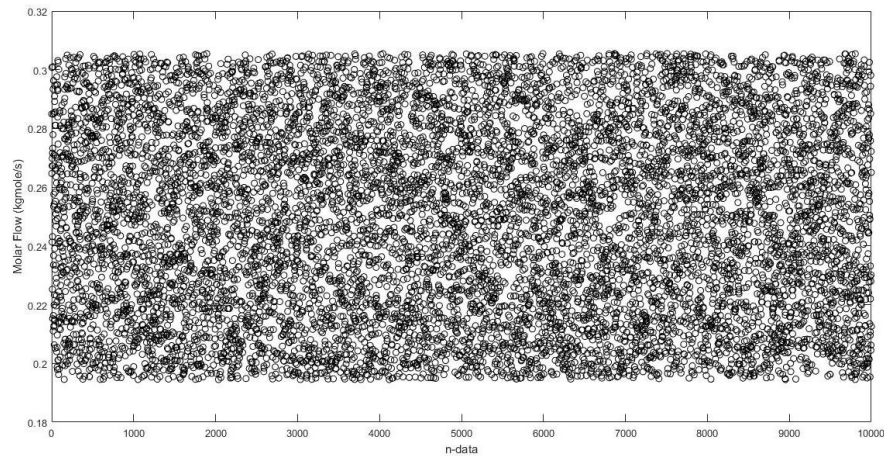
(kgmole), sehingga dapat diartikan bahwa operating costs adalah besarnya biaya konsumsi energi yang dibutuhkan untuk mengolah 1 kgmole *nautral gas*.

BAB IV

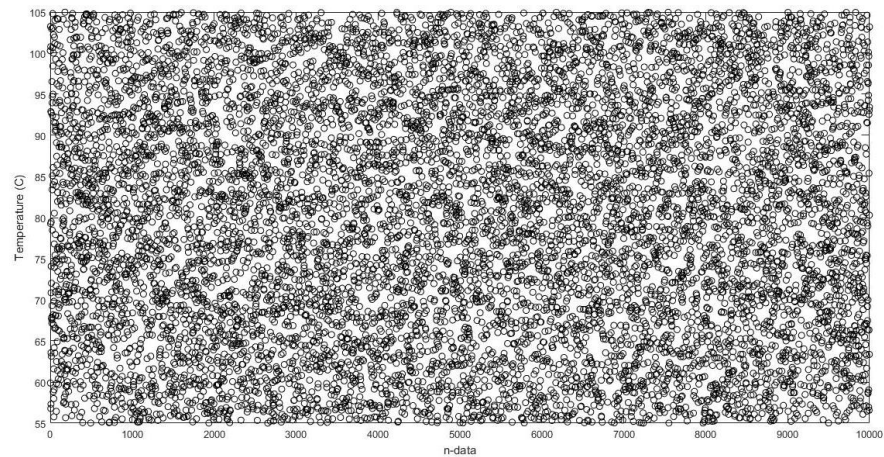
HASIL DAN PEMBAHASAN

4.1 Pengambilan Data Training *Neural Network* (NN) pada software Hysys & Matlab.

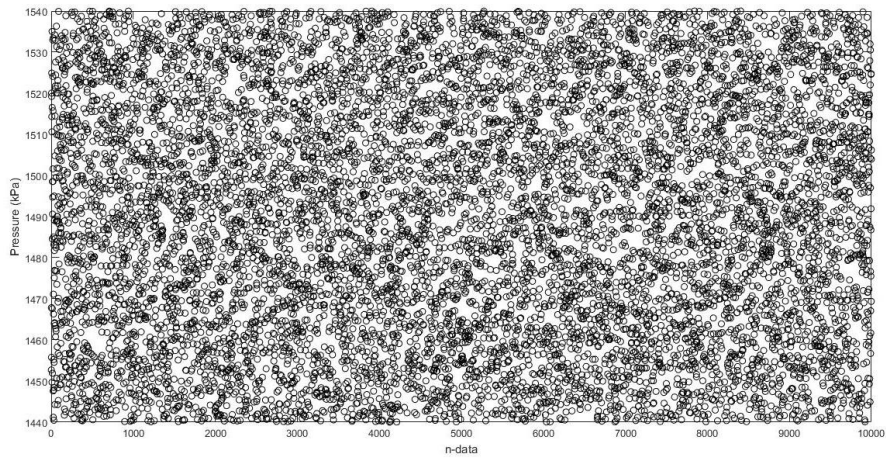
Pengambilan data training *Neural Network* (NN) pada software Hysys ini terdapat 4 variabel input yang diberikan kepada plant yaitu *Feed – Molar Flow* (F), *Feed – Temperature* (Tf), *Condenser – Pressure* (Pc) dan *Reboiler – Pressure* (Pr) serta menghasilkan dua variabel output yaitu Qr (*Heat Flow Reboiler*) dan Qc (*Heat Flow Condenser*). Berikut merupakan hasil pengambilan data training Neural Network dapat ditampilkan dalam bentuk plot grafik sebagai berikut :



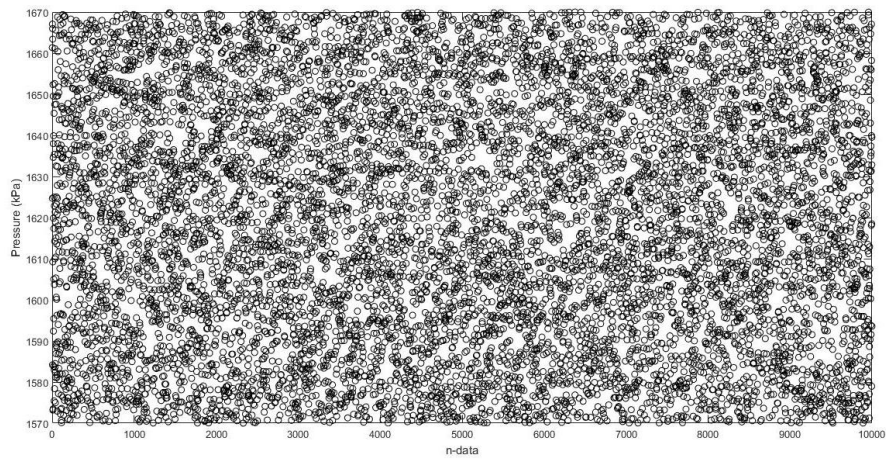
Gambar 4. 1 Plot data Training NN Feed - Molar flow.



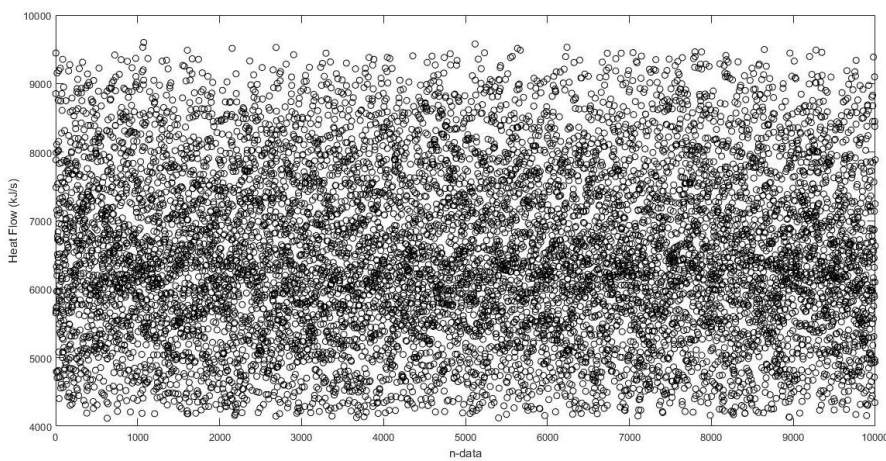
Gambar 4. 2 Plot data Training NN Feed - Temperature.



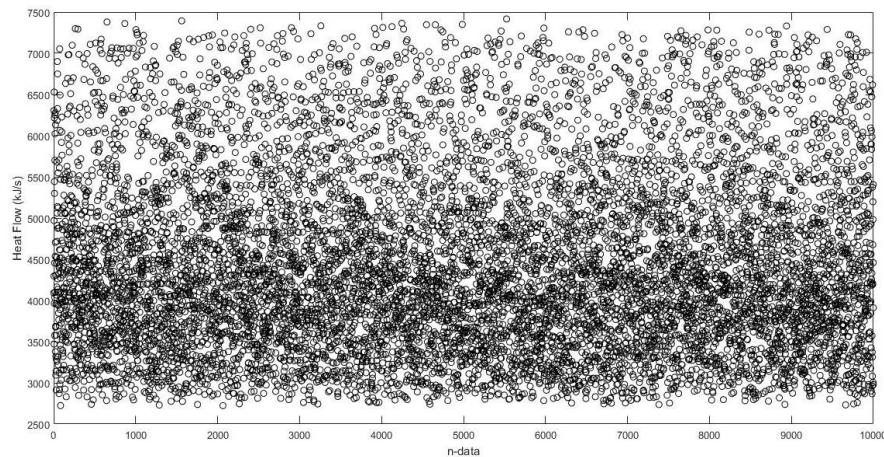
Gambar 4. 3 Plot data Training NN Condenser - Pressure.



Gambar 4. 4 Plot data Training NN Reboiler - Pressure.



Gambar 4. 5 Plot data Training NN Condenser Duty - Heat Flow.



Gambar 4. 6 Plot data Training NN Reboiler Duty - Heat Flow.

4.2 Pelatihan & Validasi Model *Neural Network* (NN).

Proses pelatihan dan validasi model *Neural Network* (NN) dilakukan secara offline. Struktur neural network yang digunakan adalah *Feed Forward - Backpropagation* dengan 1 *input layer*, 1 *hidden layer* dan 1 *output layer*. Fungsi aktivasi yang digunakan adalah *Tan Sigmoid* pada hidden layer dan *Pure Linear* pada output layer. Sedangkan untuk metode pelatihan menggunakan algoritma pelatihan *Levenberg Marquard*. Hasil dari pemrosesan tersebut akan menampilkan nilai Root Mean Squared Error (RMSE) validasi. Nilai RMSE terkecil menunjukkan bahwa struktur tersebut adalah pemodelan terbaik yang dapat digunakan.

Pada pemodelan *Neural Network* (NN) ini data yang diambil seluruhnya berjumlah 10000 sampel, dimana 75% data digunakan untuk pelatihan dan 25% data sisanya digunakan untuk validasi model *Neural Network* (NN) yang telah didapat. Maksimum epoch di set sebanyak 1000 kali iterasi. Kemudian dilakukan variasi perubahan *hidden neurons* mulai dari 4 hingga 20 hidden neurons pada hidden layer untuk mendapatkan hasil pelatihan dan validasi terbaik.

Validasi model *Neural Network* (NN) dilakukan dengan cara memberikan input data validasi yang sudah disediakan dengan matriks 4x2500 kepada masing-masing model *Neural Network* (NN) yang telah dibuat, kemudian nilai keluaran dari masing-masing model tersebut dibandingkan dengan data Target Validasi dengan matriks 2x2500 sehingga dari sini dapat dihitung nilai RMSE masing-

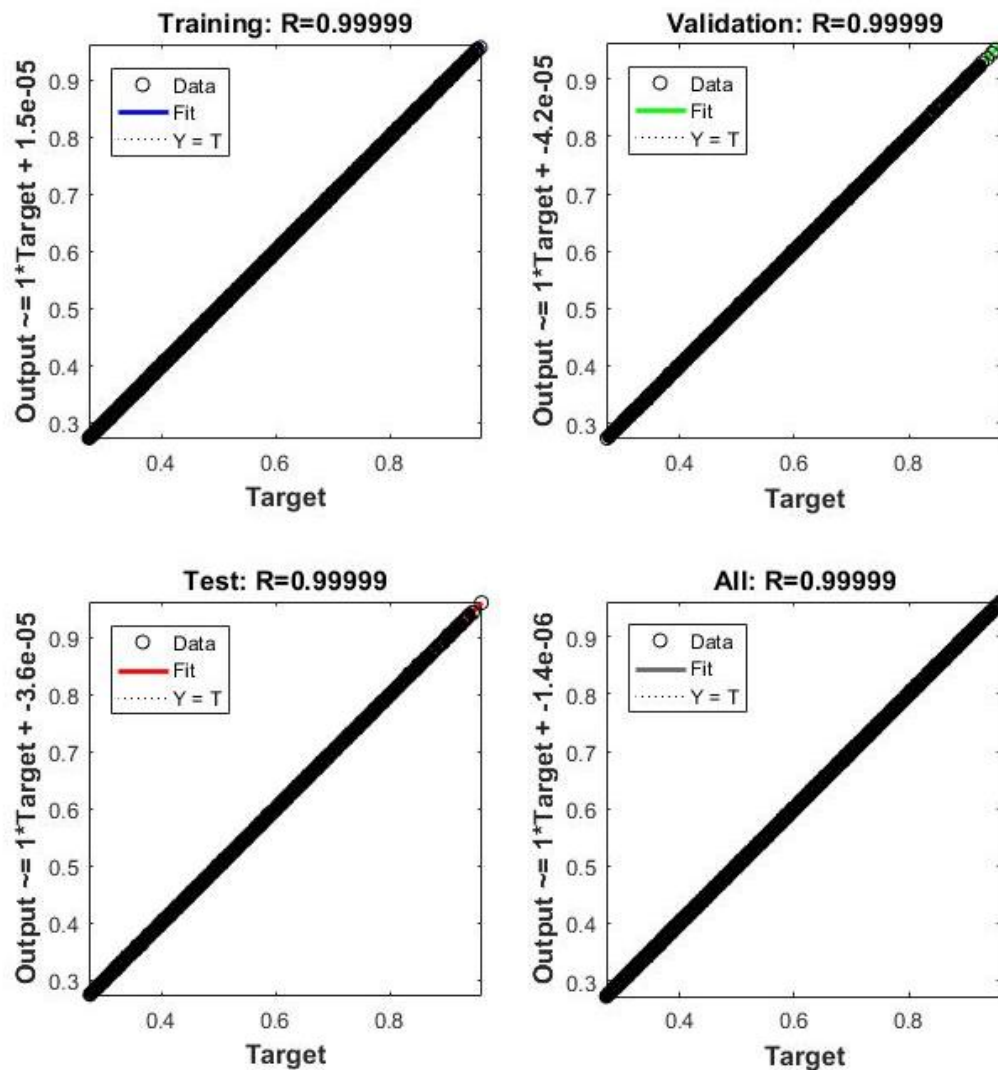
masing model *Neural Network* (NN) dari model neural network 4 *hidden neurons* hingga 20 *hidden neurons*. Berikut adalah hubungan antara nilai RMSE dengan jumlah hidden neurons ditunjukkan pada **Tabel 4.1**.

Tabel 4. 1 Hubungan nilai RMSE dengan jumlah *hidden neurons*.

Jumlah <i>Hidden neurons</i>	RMSE		
	Qc	Qr	Total
4	$5,099 \times 10^{-3}$	$6,169 \times 10^{-3}$	$8,004 \times 10^{-3}$
5	$2,432 \times 10^{-3}$	$2,743 \times 10^{-3}$	$3,666 \times 10^{-3}$
6	$1,351 \times 10^{-3}$	$1,798 \times 10^{-3}$	$2,249 \times 10^{-3}$
7	$8,605 \times 10^{-4}$	$1,311 \times 10^{-3}$	$1,568 \times 10^{-3}$
8	$7,393 \times 10^{-4}$	$1,008 \times 10^{-3}$	$1,250 \times 10^{-3}$
9	$6,603 \times 10^{-4}$	$8,581 \times 10^{-4}$	$1,083 \times 10^{-3}$
10	$6,113 \times 10^{-4}$	$7,889 \times 10^{-4}$	$9,980 \times 10^{-4}$
11	$6,533 \times 10^{-4}$	$7,643 \times 10^{-4}$	$1,005 \times 10^{-3}$
12	$5,697 \times 10^{-4}$	$7,253 \times 10^{-4}$	$9,223 \times 10^{-4}$
13	$5,761 \times 10^{-4}$	$7,125 \times 10^{-4}$	$9,163 \times 10^{-4}$
14	$6,519 \times 10^{-4}$	$8,114 \times 10^{-4}$	$1,041 \times 10^{-3}$
15	$6,876 \times 10^{-4}$	$8,127 \times 10^{-4}$	$1,065 \times 10^{-3}$
16	$5,826 \times 10^{-4}$	$6,953 \times 10^{-4}$	$9,071 \times 10^{-4}$
17	$5,751 \times 10^{-4}$	$6,999 \times 10^{-4}$	$9,059 \times 10^{-4}$
18	$5,794 \times 10^{-4}$	$7,012 \times 10^{-4}$	$9,096 \times 10^{-4}$
19	$5,541 \times 10^{-4}$	$6,310 \times 10^{-4}$	$8,398 \times 10^{-4}$
20	$5,519 \times 10^{-4}$	$6,487 \times 10^{-4}$	$8,517 \times 10^{-4}$

Dari **Tabel 4.1** menunjukkan bahwa secara umum proses pemodelan Kolom Depropanizer menggunakan *Neural Network* (NN) ini berjalan dengan baik, dikarenakan nilai-nilai RMSE validasi didapatkan nilai yang kecil. Kemudian dari **Tabel 4.1** diatas juga dapat dilihat bahwa model *Neural Network* (NN) terbaik didapatkan dari hasil training dengan jumlah hidden neurons 19 dengan rincian RMSE Qc, RMSE Qr, dan RMSE total berturut – turut sebesar $5,541 \times 10^{-4}$; $6,310 \times 10^{-4}$; $8,398 \times 10^{-4}$. Berikut merupakan grafik regression hasil pelatihan model

Neural Network (NN) dengan jumlah hidden neurons 19 dapat dilihat pada **Gambar 4.7.**



Gambar 4. 7 Grafik regression model *Neural Network* (NN) dengan jumlah hidden neurons 19.

Dari Grafik *regression* model *Neural Network* (NN) dengan jumlah *hidden neurons* 19 diatas menunjukkan bahwa pemodelan proses Kolom Depropanizer menggunakan *Neural Network* (NN) menghasilkan model yang baik, hal ini dapat dilihat pada nilai regresi *training*, *validation* dan *test* dengan nilai regresi 0,99999, jika nilai regresi semakin mendekati nilai 1 maka semakin baik model *Neural Network* (NN) yang di buat, hal ini dikarenakan bahwa nilai regresi menunjukkan kesesuaian data output *Neural Network* (NN) dengan data target $Y=T$, dengan regresi target adalah 1.

4.3 Optimisasi Konsumsi Energi Kolom Depropanizer menggunakan PSO.

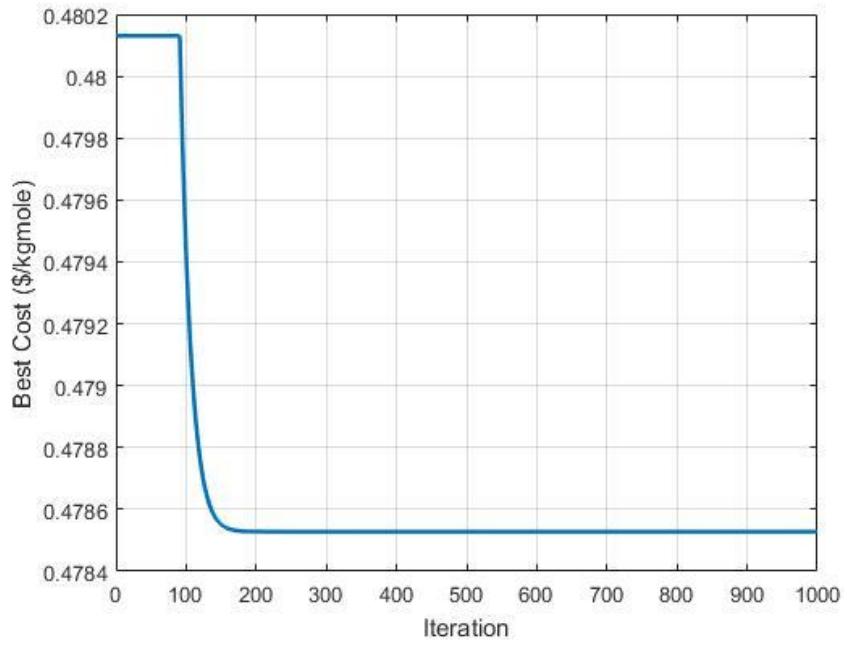
Optimisasi Konsumsi Energi pada Kolom Depropanizer menggunakan algoritma PSO dilakukan 1000 kali iterasi dengan jumlah populasi sebanyak 25 partikel, serta dilakukan variasi parameter PSO yaitu nilai c_1 dan c_2 dengan nilai $c_1 = c_2 = 0,001 ; 0,0015 ; 0,002$ serta dilakukan update *weight inertia* pada setiap iterasi dengan batas $w_{k \min}$ dan $w_{k \max}$ adalah 0,4 dan 0,9. Masing – masing nilai c_1 dan c_2 dilakukan 3x running pada software Matlab.

Tabel 4. 2 Parameter PSO.

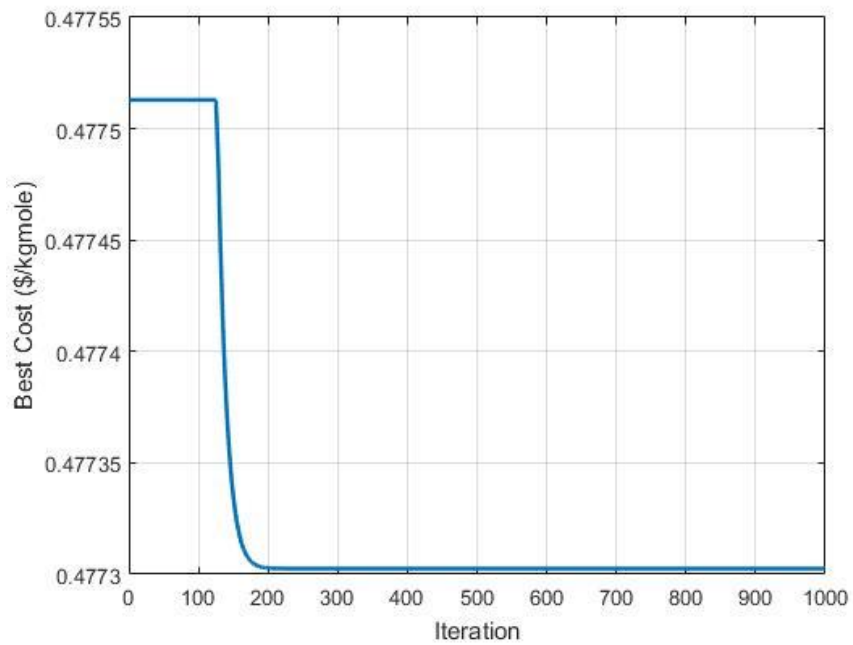
Parameter PSO	Nilai
n – Iterasi	1000
n – Populasi	25
$w_{k \max}$ (weight inertia maksimum)	0,9
$w_{k \min}$ (weight inertia minimum)	0,4
c_1 (faktor <i>learning</i> untuk <i>particle</i>)	0,001 ; 0,0015 ; 0,002
c_2 (faktor <i>learning</i> untuk <i>swarm</i>)	0,001 ; 0,0015 ; 0,002

a) Hasil Optimisasi dengan nilai $c_1 = c_2 = 0,001$.

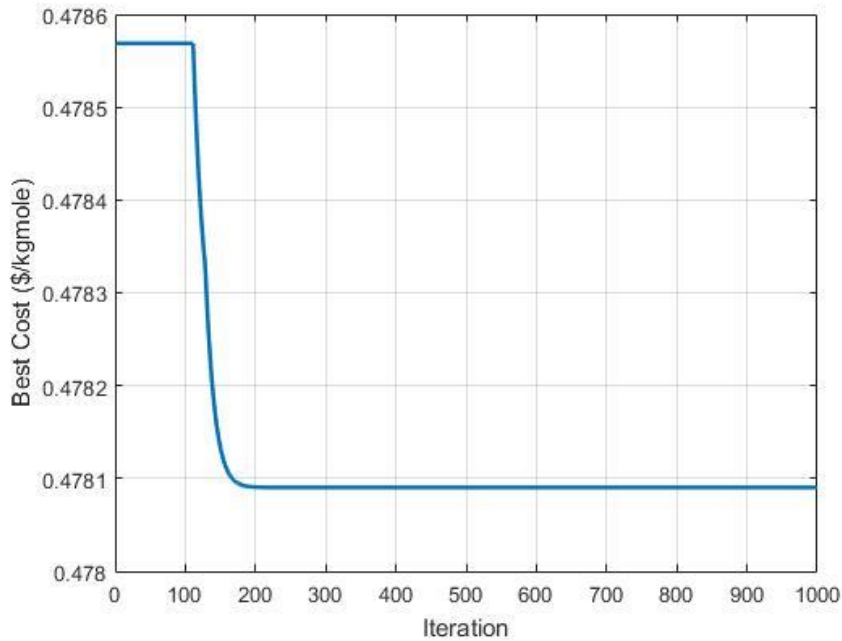
Berikut merupakan hasil optimisasi dengan nilai $c_1 = c_2 = 0,001$ *running* pertama, kedua dan ketiga dapat direpresentasikan dalam bentuk grafik konvergensi fungsi minimum, grafik konvergensi ditampilkan dalam bentuk grafik hubungan *operating cost* dan n-iterasi sebagai berikut :



Gambar 4. 8 Grafik konvergensi ($c1 = c2 = 0,001$) pada running 1



Gambar 4. 9 Grafik konvergensi ($c1 = c2 = 0,001$) pada running 2.



Gambar 4. 10 Grafik konvergensi ($c_1 = c_2 = 0,001$) pada running 3.

Pada Grafik konvergensi dengan nilai parameter $c_1 = c_2 = 0,001$ pada running pertama menunjukkan konvergen pada iterasi ke-164 dengan nilai *fitness* (*operating costs*) sebesar 0,47853 \$/kgmole, nilai ini menunjukkan bahwa hasil optimasi dengan nilai parameter $c_1 = c_2 = 0,001$ pada *running* pertama menghasilkan biaya operasional (*cooling water* dan *steam*) sebesar 0,47853 \$/kgmole, demikian juga pada running kedua menunjukkan konvergen pada iterasi ke – 179 dengan nilai *fitness* (*operating costs*) sebesar 0,47730 \$/kgmole dan pada *running* ketiga konvergen pada iterasi ke – 175 dengan nilai *fitness* (*operating costs*) sebesar 0,47809 \$/kgmole. Berikut merupakan tabel hasil optimisasi dengan nilai $c_1 = c_2 = 0,001$ dapat dilihat pada **Tabel 4.3**.

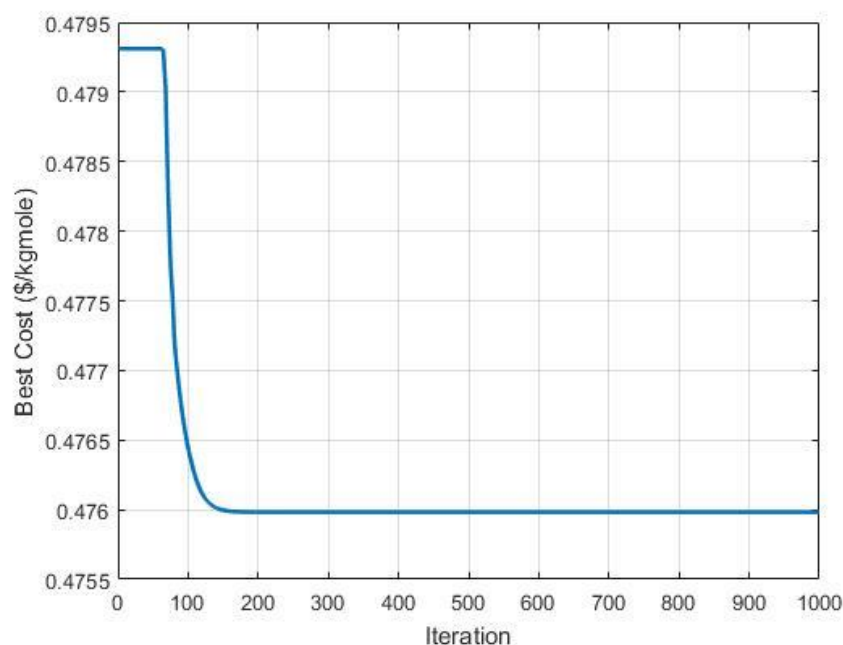
Tabel 4. 3 Hasil Optimisasi dengan nilai $c_1 = c_2 = 0,001$.

	Running		
	Ke-1	Ke-2	Ke-3
Position :			
F (kgmole/h)	928,584	826,707	938,201
Tf (°C)	103,161	104,918	102,005

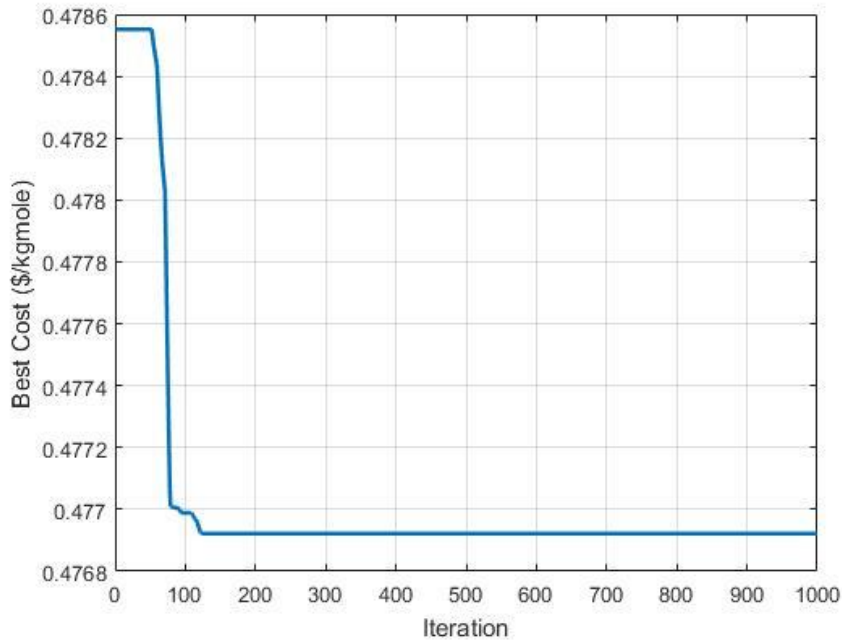
Pc (kPa)	1527,69	1515,541	1531,303
Pr (kPa)	1579,63	1574,9	1570,141
Fitness :			
Qc (kW)	8037,441	7186,329	8085,745
Qr (kW)	3604,877	3191,231	3644,445
Operating Cost (\$/kgmole)	0,47853	0,47730	0,47809
Iterasi	164	179	175

b) Hasil Optimisasi dengan nilai $c_1 = c_2 = 0,0015$.

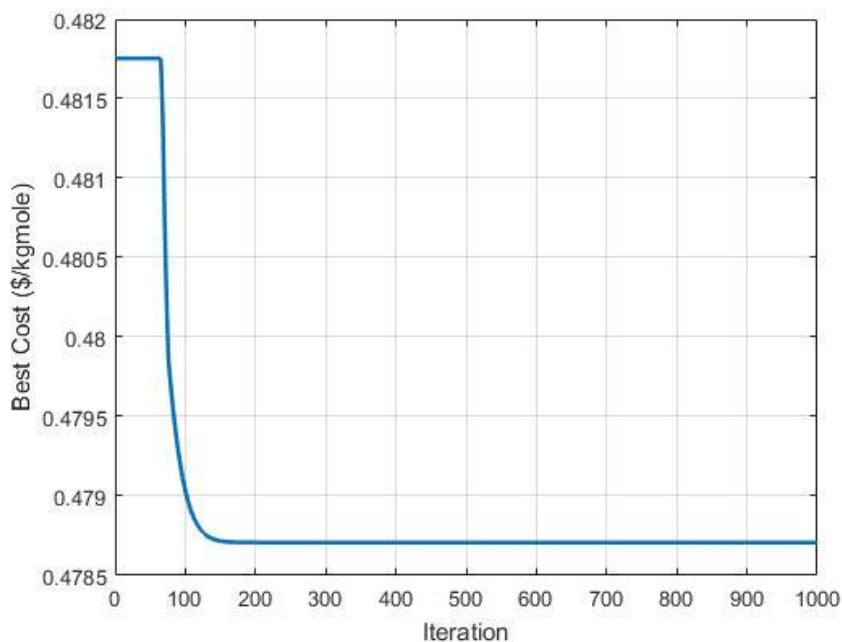
Berikut merupakan hasil optimisasi dengan nilai $c_1 = c_2 = 0,0015$ *running* pertama, kedua dan ketiga dapat direpresentasikan dalam bentuk grafik konvergensi fungsi minimum, grafik konvergensi ditampilkan dalam bentuk grafik hubungan *operating cost* dan n-iterasi sebagai berikut :



Gambar 4. 11 Grafik konvergensi ($c_1 = c_2 = 0,0015$) pada running 1.



Gambar 4. 12 Grafik konvergensi ($c1 = c2 = 0,0015$) pada running 2



Gambar 4. 13 Grafik konvergensi ($c1 = c2 = 0,0015$) pada running 3

Pada Grafik konvergensi dengan nilai parameter $c1 = c2 = 0,0015$ pada running pertama menunjukkan konvergen pada iterasi ke-171 dengan nilai *fitness* (*operating costs*) sebesar 0,47853 \$/kgmole, nilai ini menunjukkan bahwa hasil optimasi dengan nilai parameter $c1 = c2 = 0,0015$ pada *running* pertama

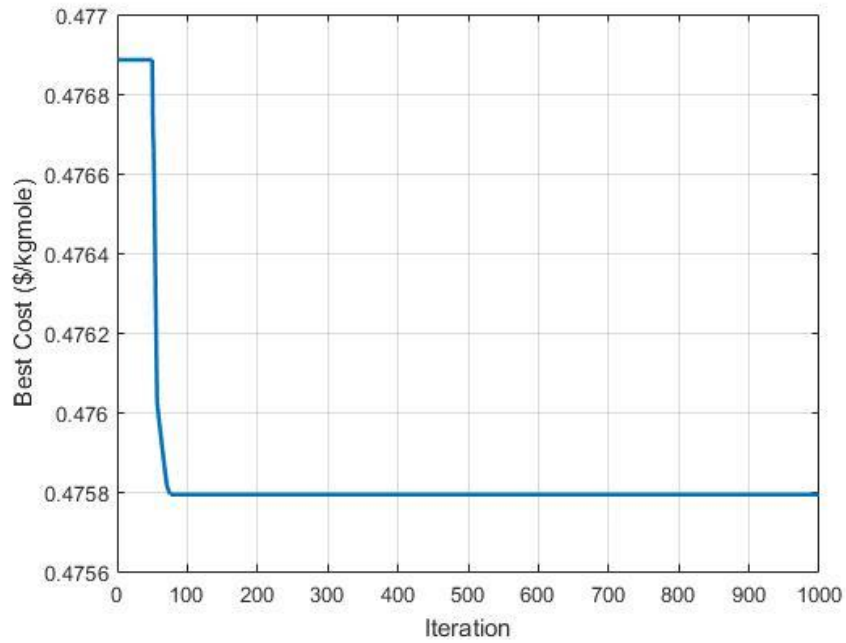
menghasilkan biaya operasional (*cooling water* dan *steam*) sebesar 0,47598 \$/kgmole, demikian juga pada *running* kedua menunjukkan konvergen pada iterasi ke – 123 dengan nilai *fitness (operating costs)* sebesar 0,47692 \$/kgmole dan pada *running* ketiga konvergen pada iterasi ke – 152 dengan nilai *fitness (operating costs)* sebesar 0,47871 \$/kgmole. Berikut merupakan tabel hasil optimisasi dengan nilai $c_1 = c_2 = 0,0015$ dapat dilihat pada **Tabel 4.4**.

Tabel 4. 4 Hasil Optimisasi dengan nilai $c_1 = c_2 = 0,0015$.

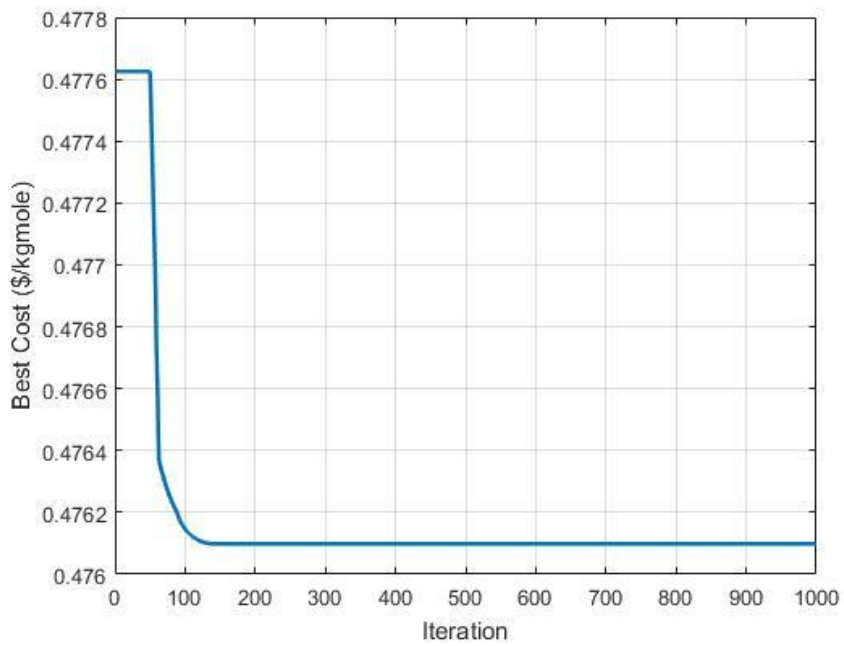
	Running		
	Ke-1	Ke-2	Ke-3
Position :			
F (kgmole/h)	700	901,5794	751,3687
Tf (°C)	105	105	105
Pc (kPa)	1460,61	1518,218	1440,068
Pr (kPa)	1570,1	1570	1597,593
Fitness :			
Qc (kW)	6104,445	7836,556	6563,987
Qr (kW)	2687,281	3476,335	2906,245
Operating Cost (\$/kgmole)	0,47598	0,47692	0,47871
Iterasi	171	123	152

c) Hasil Optimisasi dengan nilai $c_1 = c_2 = 0,002$.

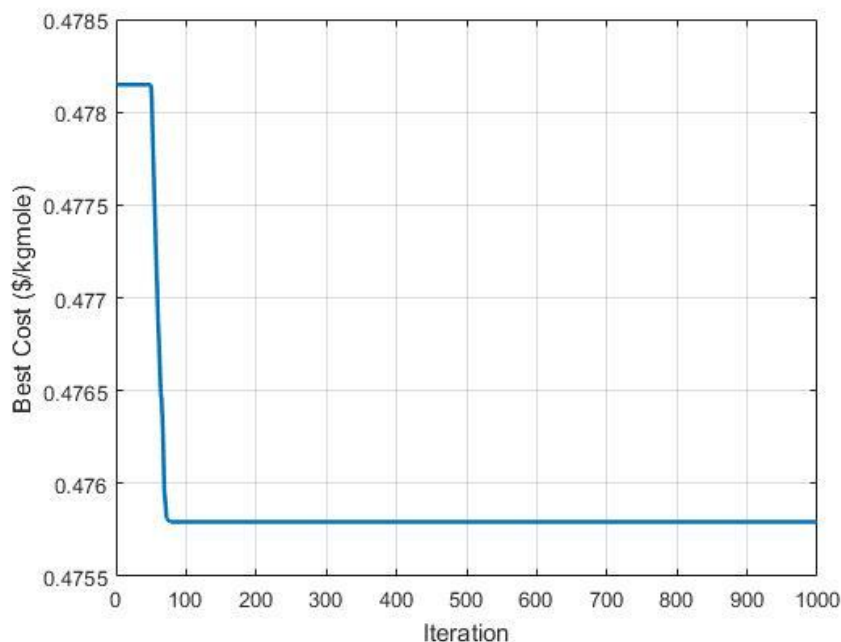
Berikut merupakan hasil optimisasi dengan nilai $c_1 = c_2 = 0,002$ *running* pertama, kedua dan ketiga dapat direpresentasikan dalam bentuk grafik konvergensi fungsi minimum, grafik konvergensi ditampilkan dalam bentuk grafik hubungan *operating cost* dan n-iterasi sebagai berikut :



Gambar 4. 14 Grafik konvergensi ($c_1 = c_2 = 0,002$) pada running 1



Gambar 4. 15 Grafik konvergensi ($c_1 = c_2 = 0,002$) pada running 2.



Gambar 4. 16 Grafik konvergensi ($c_1 = c_2 = 0,002$) pada running 3

Pada Grafik konvergensi dengan nilai parameter $c_1 = c_2 = 0,002$ pada running pertama menunjukkan konvergen pada iterasi ke-75 dengan nilai *fitness (operating costs)* sebesar 0,47579 \$/kgmole, nilai ini menunjukkan bahwa hasil optimasi dengan nilai parameter $c_1 = c_2 = 0,0015$ pada *running* pertama menghasilkan biaya operasional (*cooling water* dan *steam*) sebesar 0,47598 \$/kgmole, demikian juga pada running kedua menunjukkan konvergen pada iterasi ke – 125 dengan nilai *fitness (operating costs)* sebesar 0,47610 \$/kgmole dan pada *running* ketiga konvergen pada iterasi ke – 77 dengan nilai *fitness (operating costs)* sebesar 0,47579 \$/kgmole. Berikut merupakan tabel hasil optimisasi dengan nilai $c_1 = c_2 = 0,002$ dapat dilihat pada **Tabel 4.5**.

Tabel 4. 5 Hasil Optimasi dengan nilai $c_1 = c_2 = 0,002$.

	Running		
	Ke-1	Ke-2	Ke-3
Position :			
F (kgmole/h)	1100	700	1100
Tf (°C)	105	105	105
Pc (kPa)	1440	1476,993	1440

Pr (kPa)	1570	1570	1570
Fitness :			
Qc (kW)	9602,715	6098,802	9602,715
Qr (kW)	4218,381	2689,375	4218,381
Operating Cost (\$/kgmole)	0,475792786	0,476098215	0,4757928
Iterasi	75	125	77

4.4 Validasi Hasil Optimisasi Konsumsi Energi

Setelah dilakukan proses optimisasi konsumsi energi menggunakan algoritma PSO, selanjutnya akan dilakukan validasi hasil optimisasi konsumsi energi untuk membuktikan bahwa parameter hasil optimisasi konsumsi energi dapat dikatakan benar. Metode validasi yang digunakan adalah dengan mengujikan hasil parameter optimisasi (*Feed – Molar Flow (F)*, *Feed – Temperature (Tf)*, *Condenser – Pressure (Pc)* dan *Reboiler – Pressure (Pr)*) ke model proses Kolom Depropanizer pada *software* Hysys serta membandingkan nilai Qr (*Heat Flow Reboiler*) dan Qc (*Heat Flow Condenser*). **Tabel 4.6** berikut merupakan tabel validasi hasil optimisasi konsumsi energi :

Tabel 4. 6 Validasi hasil optimisasi konsumsi energi.

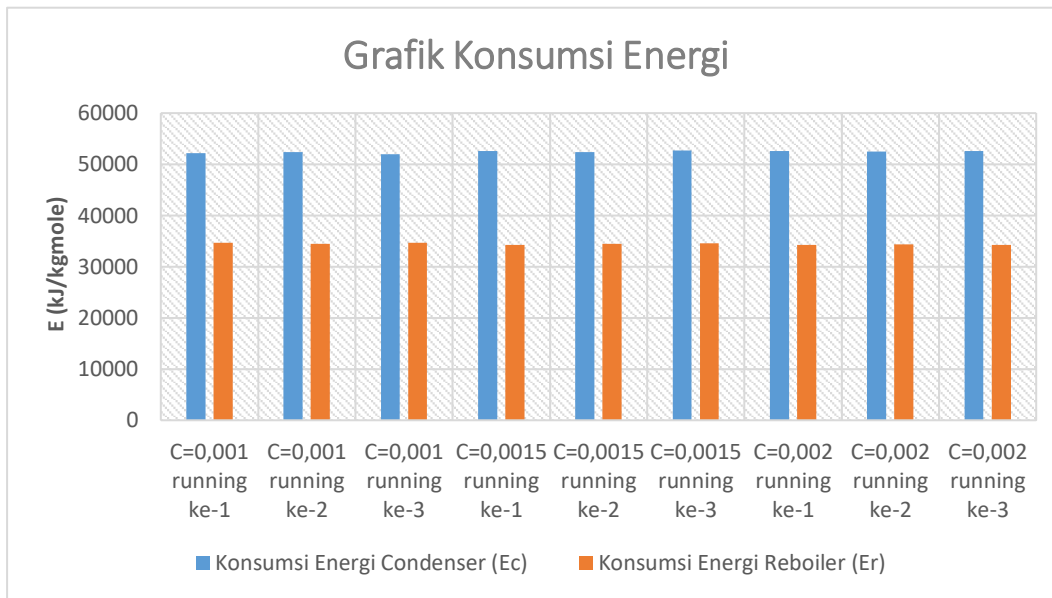
	Variabel	Matlab	Hysys	Error (%)
C=0,001	Qc (kW)	8037,441	8029,000	-0,10513
running ke-1	Qr (kW)	3604,877	3601,000	-0,10765
C=0,001	Qc (kW)	7186,329	7187,000	0,009333
running ke-2	Qr (kW)	3191,231	3186,000	-0,16419
C=0,001	Qc (kW)	8085,745	8080,000	-0,0711
running ke-3	Qr (kW)	3644,445	3637,000	-0,20471
C=0,0015	Qc (kW)	6104,445	6103,000	-0,02368
running ke-1	Qr (kW)	2687,281	2685,000	-0,08496
C=0,0015	Qc (kW)	7836,556	7836,000	-0,0071
running ke-2	Qr (kW)	3476,335	3469,000	-0,21145

C=0,0015	Qc (kW)	6563,987	6565,000	0,015423
running ke-3	Qr (kW)	2906,245	2903,000	-0,11177
C=0,002	Qc (kW)	9602,715	9602,000	-0,00745
running ke-1	Qr (kW)	4218,381	4215,000	-0,08021
C=0,002	Qc (kW)	6098,802	6097,000	-0,02955
running ke-2	Qr (kW)	2689,375	2687,000	-0,08841
C=0,003	Qc (kW)	9602,715	9602,000	-0,00745
running ke-3	Qr (kW)	4218,381	4215,000	-0,08021

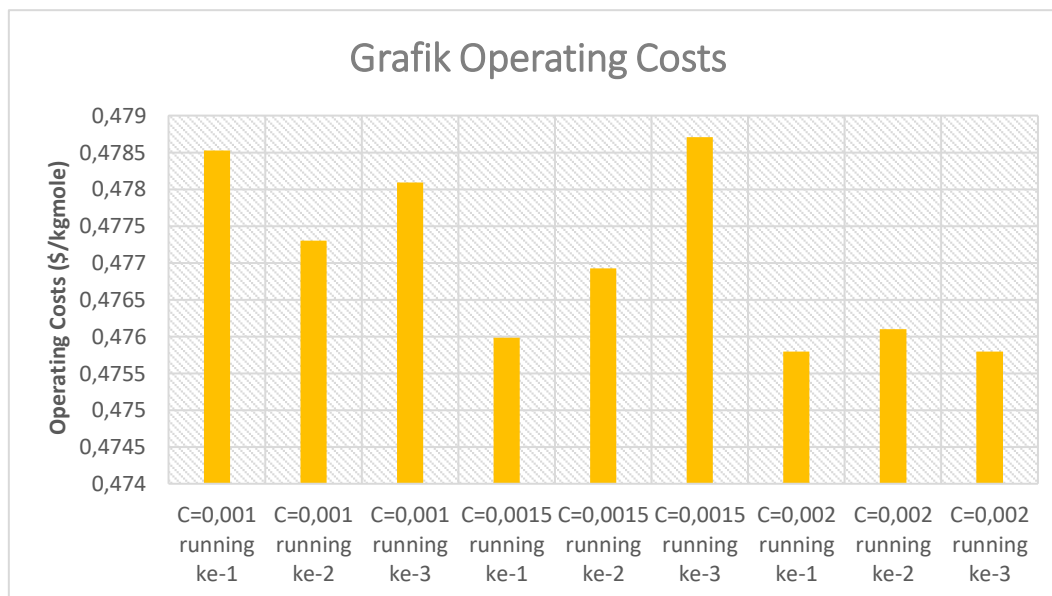
Dari **Tabel 4.6** menunjukkan bahwa proses optimisasi konsumsi energi menggunakan algoritma PSO menghasilkan nilai error yang kecil yaitu dibawah 1 %. Nilai error ini diantaranya dikarenakan kesalahan perhitungan (error) model *neural network* untuk menghitung nilai *fitness* (Qc dan Qr). Sehingga dengan nilai error dibawah 1 % maka secara keseluruhan proses optimisasi konsumsi energi pada Kolom Depropanizer dapat dikatakan valid.

4.5 Analisa Performansi Hasil Optimisasi Konsumsi Energi

Dari hasil optimisasi konsumsi energi menggunakan algoritma PSO, akan dicari hasil terbaik yaitu nilai operating costs minimum, nilai operating costs direpresentasikan sebagai besarnya biaya konsumsi energi (\$) persatuan massa (kgmole), sehingga dapat diartikan bahwa operating costs adalah besarnya biaya konsumsi energi yang dibutuhkan untuk mengolah 1 kgmole *neutral gas*. berikut merupakan grafik hubungan nilai c_1 dan c_2 dengan nilai konsumsi energi dan operating costs :



Gambar 4. 17 Grafik hasil optimisasi konsumsi energi dengan algoritma PSO



Gambar 4. 18 Grafik hasil optimisasi nilai operating costs dengan algoritma PSO

Tabel 4. 7 Tabel hasil optimisasi menggunakan algoritma PSO

Parameter PSO	Ec (kJ/kgmole)	Er (kJ/kgmole)	Operating costs (\$/kgmole)
$c = 0,001$ running ke-1	52194,478	34678,992	0,478528
$c = 0,001$ running ke-2	52418,355	34482,902	0,477303
$c = 0,001$ running ke-3	51969,963	34700,290	0,47809

$c = 0,0015$ running ke-1	52586,748	34293,555	0,475983
$c = 0,0015$ running ke-2	52414,259	34444,122	0,476922
$c = 0,0015$ running ke-3	52679,644	34552,268	0,478707
$c = 0,002$ running ke-1	52641,654	34257,099	0,475793
$c = 0,002$ running ke-2	52538,135	34320,283	0,476098
$c = 0,002$ running ke-3	52641,654	34257,099	0,475793

Dari grafik dan tabel diatas dapat ditentukan bahwa hasil optimisasi konsumsi energi dengan penggunaan energi yang optimal yaitu nilai E_c sebesar 52641,654 kJ/kgmole dan nilai E_r sebesar 34257,099 kJ/kgmole dengan nilai *operating costs* sebesar 0,47579 \$/kgmole. Hasil tersebut didapatkan dengan nilai Parameter PSO $c_1 = c_2 = 0,002$ pada *running* pertama dan ketiga. Kemudian hasil tersebut dibandingkan dengan kondisi eksisting atau sebelum di optimasi apakah hasil optimisasi konsumsi energi menggunakan algoritma PSO mampu memberikan penghematan energi. Berikut **Tabel 4.8** merupakan tabel perbandingan konsumsi energi sesudah optimisasi dan sebelum optimisasi.

Tabel 4. 8 Perbandingan Sebelum Optimasi dan Sesudah Optimasi.

	Sebelum Optimasi	Sesudah Optimasi
F (kgmole/h)	890,3	1100
Tf (°C)	71,11	105
Pc (kPa)	1491	1440
Pr (kPa)	1620	1570
Qc (kW)	5810,744	9602,715
Qr (kW)	4827,393	4218,381
Konsumsi energi :		
E_c (kJ/kgmole)	39357,141	52641,654
E_r (kJ/kgmole)	48436,610	34257,099
Operating Cost (\$/kgmole)	0,572703	0,475793

Dari **Tabel 4.8** dapat dilihat perbandingan besarnya konsumsi energi sebelum dan sesudah dilakukan optimisasi, terjadi perbedaan yang signifikan dikarenakan nilai

variabel – variabel operasional pada kondisi eksisting yang berbeda pula, sehingga dapat dicari persentase penghematan konsumsi energi dan *operating costs* dengan **Persamaan 2.19 ; 2.20 ; 2.22.**

Tabel 4. 9 Efisiensi Penghematan konsumsi energi dan *operating costs*.

	Efisiensi (%)
μ Ec (%)	-33,75
μ Er (%)	29,27
μ Operating Cost (%)	16,92

Dari **Tabel 4.9** diatas dapat disimpulkan bahwa Teknik Optimisasi konsumsi energi pada Kolom Depropanizer menggunakan Algoritma PSO menghasilkan penghematan konsumsi energi pada *Reboiler* dengan nilai efisiensi Er sebesar 29,27 %, sedangkan pada *equipment Condenser* terjadi peningkatan konsumsi energi dibandingkan dengan kondisi eksisting sehingga nilai efisiensi Ec sebesar -33,75 %, sehingga secara keseluruhan Teknik Optimisasi menggunakan algoritma PSO dapat memberikan penghematan biaya utilitas sebesar 16,92 %.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis yang telah dilakukan, kesimpulan dari tugas akhir ini adalah sebagai berikut :

- a) Pemodelan plant Kolom Depropanizer menggunakan *Neural Network* (NN) didapatkan nilai RMSE terkecil pada jumlah *hidden neurons* sebanyak 19. dengan rincian RMSE Qc, RMSE Qr, dan RMSE total berturut – turut sebesar $5,541 \times 10^{-4}$; $6,310 \times 10^{-4}$; $8,398 \times 10^{-4}$.
- b) Nilai Parameter PSO yang dapat memberikan penghematan konsumsi energi paling signifikan didapatkan pada $c_1 = c_2 = 0,002$ pada running pertama dan ketiga dengan jumlah iterasi 1000 kali, jumlah populasi sebanyak 25 partikel, $w_{k \min}$ dan $w_{k \max}$ adalah 0,4 dan 0,9. Sehingga didapatkan kondisi variabel operasional *flow feed* sebesar 1100 kgmole/h, *temperature feed* sebesar 105 °C, *pressure condenser* sebesar 1440 kPa, dan *pressure reboiler* sebesar 1570 kPa.
- c) Hasil optimisasi menghasilkan penghematan konsumsi energi pada *Reboiler* dengan nilai efisiensi Er sebesar 29,27 %, sedangkan pada *equipment Condenser* terjadi peningkatan konsumsi energi dibandingkan dengan kondisi eksisting sehingga nilai efisiensi Ec sebesar -33,75 %, sehingga secara keseluruhan Teknik Optimisasi menggunakan algoritma PSO dapat memberikan penghematan biaya utilitas sebesar 16,92 %.

5.2 Saran

Dari hasil tugas akhir ini dapat diberikan saran untuk pengembangan penelitian selanjutnya yaitu : untuk memodelkan sistem MIMO (Multi Input Multi Output) plant Kolom Depropanizer menggunakan *Neural Network* (NN) dengan dua output (Qc dan Qr) akan lebih baik jika dimodelkan menjadi dua model *Neural Network* (NN) MISO (Multi Input Single Output) 1 dan 2 agar mendapatkan nilai RMSE yang lebih kecil.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- Biyanto, T. R., Widjiantoro, B. L., Jabal, A. A., & Budiati, T. (2011). Artificial neural network based modeling and controlling of distillation column system. *International Journal of Engineering, Science and Technology*, 2(6), 177–188. <https://doi.org/10.4314/ijest.v2i6.63709>
- Buscema, P. M., Massini, G., Breda, M., Lodwick, W. A., Newman, F., & Asadi-Zeydabadi, M. (2018). Artificial neural networks. In *Studies in Systems, Decision and Control*. https://doi.org/10.1007/978-3-319-75049-1_2
- Carling, G. A., & Wood, R. K. (1986). The Dynamics and Control of a Depropanizer. *IFAC Proceedings Volumes*, 19(15), 167–173. [https://doi.org/10.1016/s1474-6670\(17\)59417-3](https://doi.org/10.1016/s1474-6670(17)59417-3)
- Ceylan, R., & Koyuncu, H. (2019). A Novel Rotation Forest Modality Based on Hybrid NNs: RF (ScPSO-NN). *Journal of King Saud University - Computer and Information Sciences*, 31(2), 235–251. <https://doi.org/10.1016/j.jksuci.2017.10.011>
- Ching, T. C., Nandong, J., & Getu, M. (2016). Retrofitting Options for Natural Gas Liquid (NGL) Fractionation Trains Using the Concept of Single Column Development. *Procedia Engineering*, 148, 923–931. <https://doi.org/10.1016/j.proeng.2016.06.503>
- Fausett, L. (2005). Fundamentals of Neural Networks: Architectures, Algorithms and Applications Fundamentals of Neural Networks: Architectures, Algorithms and Applications. *Journal of Chromatography. B, Analytical Technologies in the Biomedical and Life Sciences*, 814(2), 315–323.
- Fitriyani, N., Ruki, T., & Ph, B. (2015). *Optimasi Kondisi Kolom Distilasi Biner untuk Mencapai Kualitas Produk dengan Menggunakan Imperialist Competitive Algorithm (ICA)*. 1–6.
- Hoffman, H. L., Lupfer, D. E., Kane, L. A., Jensen, B. A., & Lipták, B. G. (2005). Distillation: Basic controls. *Instrument Engineers Handbook, Fourth Edition: Process Control and Optimization*, 2(1985), 1820–1854. <https://doi.org/10.1016/b978-0-7506-2255-4.50122-7>

- Industri, F. T. (2017). *PERANCANGAN SISTEM PENGENDALI NEURAL (NNMPC) PADA KOLOM DEPROPANIZER*.
- Ji, Z., Wang, Y., Chu, Y., & Wu, Q. (2009). Bacterial particle swarm optimization. *Chinese Journal of Electronics*, 18(2), 195–199.
- Le, L. T., Nguyen, H., Dou, J., & Zhou, J. (2019). A comparative study of PSO-ANN, GA-ANN, ICA-ANN, and ABC-ANN in estimating the heating load of buildings' energy efficiency for smart city planning. *Applied Sciences (Switzerland)*, 9(13). <https://doi.org/10.3390/app9132630>
- M. Hasan Syukur, ST, M. (2011). Penggunaan Liquified Petroleum Gases (Lpg). *Forum Teknologi*, 01(2), 1–14. Retrieved from http://pusdiklatmigas.esdm.go.id/new/pusdiklatmigas/file/t2-_LPG_---_M_hasan_Syukur.pdf
- Mcgoodwin, M. C. (2016). *Introduction Catalog Course Description*.
- Miller, R., & Dennison, J. (2015). Lecture 1. *An Outline of Psychiatry in Clinical Lectures*, 3–7. https://doi.org/10.1007/978-3-319-18051-9_1
- Moravej, Z., Jazaeri, M., & Gholamzadeh, M. (2012). Optimal coordination of distance and over-current relays in series compensated systems based on MAPSO. *Energy Conversion and Management*. <https://doi.org/10.1016/j.enconman.2011.11.024>
- R. Biyanto, T. (2007). Algoritma Genetika untuk Mengoptimasi Konsumsi Energi pada Proses Kolom Distilasi Metanol-Air. *Jurnal Teknik Elektro*, 7(1), 43–49. <https://doi.org/10.9744/jte.7.1.43-49>
- Rahmatullah, D. (2017). *Setting Docr Adaptif Pada Sistem Distribusi Dengan Pembangkit Tersebar Menggunakan Algoritma Pso - Neural Network Adaptive Docr Setting on Distribution Systems With Distributed Generation Using Pso-Neural Network Algorithm*.
- Saeid M., W. A., & Y., J. (2012). Handbook of Natural Gas Transmission and Processing. In *Foreign Affairs* (Vol. 91). <https://doi.org/10.1017/CBO9781107415324.004>
- Sjoberg, J. (2001). Neural networks for modelling and control of dynamic systems, M. Nørgaard, O. Ravn, N. K. Poulsen and L. K. Hansen, Springer, London, 2000, xiv+246pp. *International Journal of Robust and Nonlinear Control*.

<https://doi.org/10.1002/rnc.585>

Widyaningrum, V. T., & Romadhon, A. S. (2014). *PENGARUH PEMBERIAN MOMENTUM PADA ARTIFICIAL NEURAL NETWORK BACKPROPAGATION*. (November).

Halaman ini sengaja dikosongkan

LAMPIRAN

A. Data Simulasi (Hysys & Matlab).

Tabel A. 1 Data spesifikasi Kolom Depropanizer.

Variabel	Nilai	Satuan
N_t (Number of trays)	50	Tray
N (Feed tray location)	25	Feed 1 and 2
$R_T C_3$ (Recovery of C3 at top stream)	0,999	Mole fraction
$R_B C_4$ (Recovery of C4 at bottom stream)	0,990	Mole fraction
\dot{D} (Molar Flow of top product stream)	531,6	kgmol/h
\dot{B} (Molar Flow of bottom product stream)	358,8	kgmol/h
P_c (Pressure of condenser)	14,91	Bar
P_r (Pressure of reboiler)	16,20	Bar

Tabel A. 2 Data stream simulasi *steady state* Kolom Depropanizer.

Stream	Variabel	Nilai	Satuan
Feed Stream	T_f (Temperature feed)	71,11	°C
	P_f (Pressure feed)	16	Bar
	\dot{F} (Flow feed)	890,3	kgmol/h
	Composition (mole fraction)		
	Ethane	(C ₂)	0.0001
	Propane	(C ₃)	0.5959
	<i>i</i> -Butane	(iC ₄)	0.1638
	<i>n</i> -Butane	(nC ₄)	0.1214
	<i>i</i> -Pentane	(iC ₅)	0.0523
	<i>n</i> -Pentane	(nC ₅)	0.0271
	<i>n</i> -Hexane	(nC ₆)	0.0192
	<i>n</i> -Heptane	(nC ₇)	0.0113
	<i>n</i> -Octane	(nC ₈)	0.0070
<i>n</i> -Nonane	(nC ₉)	0.0019	

Distillate Stream	T_t (Temperature Distillate)	43,72	°C
	P_t (Pressure Distillate)	14,91	Bar
	\dot{D} (Distillate flow)	532,6	kgmol/h
	Composition (mole fraction)		
	Ethane	(C ₂)	0,0002
	Propane	(C ₃)	0,9951
	<i>i</i> -Butane	(iC ₄)	0,0047
	<i>n</i> -Butane	(nC ₄)	0,0000
	<i>i</i> -Pentane	(iC ₅)	0,0000
	<i>n</i> -Pentane	(nC ₅)	0,0000
	<i>n</i> -Hexane	(nC ₆)	0,0000
	<i>n</i> -Heptane	(nC ₇)	0,0000
<i>n</i> -Octane	(nC ₈)	0,0000	
<i>n</i> -Nonane	(nC ₉)	0,0000	
Bottom Product Stream	T_b (Temperature Bottom Product)	110,1	°C
	P_b (Pressure Bottom Product)	16,2	Bar
	\dot{B} (Bottom Product flow)	357,7	kgmol/h
	Composition (mole fraction)		
	Ethane	(C ₂)	0,0000
	Propane	(C ₃)	0,0015
	<i>i</i> -Butane	(iC ₄)	0,4006
	<i>n</i> -Butane	(nC ₄)	0,3022
	<i>i</i> -Pentane	(iC ₅)	0,1302
	<i>n</i> -Pentane	(nC ₅)	0,0675
	<i>n</i> -Hexane	(nC ₆)	0,0478
	<i>n</i> -Heptane	(nC ₇)	0,0281
<i>n</i> -Octane	(nC ₈)	0,0174	

	<i>n-Nonane</i>	(nC ₉)	0,0047
Qc	<i>Heat Flow</i>	2,091 x 10 ⁷	kJ/h
Strean	<i>Power</i>	5809	kW
Qr	<i>Heat Flow</i>	1,737 x 10 ⁷	kJ/h
Stream	<i>Power</i>	4826	kW

Tabel A. 3 Validasi model proses Kolom Depropanizer.

<i>Stream</i>	Variabel	Data penelitian (Ching et al., 2016)	Hysys	Error (%)
Feed Stream	<i>Temperature</i>	71,11 °C	71,11 °C	0
	<i>Pressure</i>	16 Bar	16 Bar	0
	<i>Molar flow</i>	890,3 kgmole/h	890,3 kgmole/h	0
Distillate Stream	<i>Temperature</i>	-	43,72 °C	-
	<i>Pressure</i>	14,91 Bar	14,91 Bar	0
	<i>Molar flow</i>	531,6 kgmole/h	532,6 kgmole/h	0,188
Bottom Product Stream	<i>Temperature</i>	-	43,72 °C	-
	<i>Pressure</i>	16,2 Bar	16,2 Bar	0
	<i>Molar flow</i>	358,8 kgmole/h	357,7kgmole/h	0,307

Tabel A. 4 Range operasional variabel optimasi.

Variabel Optimasi	Range
Feed – Molar Flow (F)	700 – 1100 (kgmole/h)
Feed – Temperature (Tf)	55 – 105 (°C)
Condenser – Pressure (Pc)	1440 – 1540 (kPa)
Reboiler – Pressure (Pr)	1570 – 1670 (kPa)

Tabel A. 5 Arsitektur *Neural Network*.

No.	Arsitektur	Type
1	Network type	Feed Forward - Backpropagation
2	Training Function	TRAINLM (Lavenberg Marquart)

3	Adaption Learning Function	LEARNGDM (Gradient Descent with Momentum)
4	Input Neurons	<ul style="list-style-type: none"> • <i>Feed – Molar Flow (F)</i>. • <i>Feed – Temperature (Tf)</i>. • <i>Condenser – Pressure (Pc)</i>. • <i>Reboiler – Pressure (Pr)</i>. Jumlah input <i>neuron</i> sebanyak 4
5	Hidden Neurons	Dilakukan jumlah variasi <i>hidden neuron</i> sebanyak 4 sampai 20 <i>neuron</i>
5	Output Neurons	<ul style="list-style-type: none"> • <i>Condenser Duty – Heat Flow (Qc)</i>. • <i>Reboiler Duty – Heat Flow (Qr)</i>. Jumlah output <i>neuron</i> sebanyak 2
6	Performance Parameter	MSE (Mean Square Error)
7	Fungsi Aktivasi	Tan Sigmoid pada <i>hidden layer</i> Pure linear pada <i>output layer</i>
7	Jumlah Pasangan Data	10000
8	Komposisi Data	<ul style="list-style-type: none"> • 75% Data Training [5x7500] Input Training [2x7500] Target Training • 25% Data Validasi [5x2500] Input Validasi [2x2500] Target Validasi
Parameter Training		
1	Epochs	1000
2	Time	infinite
3	Error Goal	1e-25
4	Minimum Gradient	1e-10

Tabel A. 6 Parameter normalisasi data *Neural Network*.

Variabel Optimasi	Parameter Normalisasi			
Feed – Molar Flow (F) [kgmole/h]	Xmin	700	Ymin	0
	Xmax	1100	Ymax	1
Feed – Temperature (Tf) [°C]	Xmin	55	Ymin	0
	Xmax	105	Ymax	1
Condenser – Pressure (Pc)	Xmin	1440	Ymin	0

[kPa]	Xmax	1540	Ymax	1
Reboiler – Pressure (Pr)	Xmin	1570	Ymin	0
[kPa]	Xmax	1670	Ymax	1
Condenser – Heat Flow (Qc)	Xmin	0	Ymin	0
[kW]	Xmax	10000	Ymax	1
Reboiler – Heat Flow (Qr)	Xmin	0	Ymin	0
[kW]	Xmax	10000	Ymax	1

Tabel A. 7 Parameter PSO.

Parameter PSO	Nilai
n – Iterasi	1000
n – Populasi	25
$w_{k \max}$ (weight inertia maksimum)	0,9
$w_{k \min}$ (weight inertia minimum)	0,4
c_1 (faktor <i>learning</i> untuk <i>particle</i>)	0,001 ; 0,0015 ; 0,002
c_2 (faktor <i>learning</i> untuk <i>swarm</i>)	0,001 ; 0,0015 ; 0,002

Tabel A. 8 Tabel RMSE model *Neural Network*.

Jumlah <i>Hidden neurons</i>	RMSE		
	Qc	Qr	Total
4	$5,099 \times 10^{-3}$	$6,169 \times 10^{-3}$	$8,004 \times 10^{-3}$
5	$2,432 \times 10^{-3}$	$2,743 \times 10^{-3}$	$3,666 \times 10^{-3}$
6	$1,351 \times 10^{-3}$	$1,798 \times 10^{-3}$	$2,249 \times 10^{-3}$
7	$8,605 \times 10^{-4}$	$1,311 \times 10^{-3}$	$1,568 \times 10^{-3}$
8	$7,393 \times 10^{-4}$	$1,008 \times 10^{-3}$	$1,250 \times 10^{-3}$
9	$6,603 \times 10^{-4}$	$8,581 \times 10^{-4}$	$1,083 \times 10^{-3}$
10	$6,113 \times 10^{-4}$	$7,889 \times 10^{-4}$	$9,980 \times 10^{-4}$
11	$6,533 \times 10^{-4}$	$7,643 \times 10^{-4}$	$1,005 \times 10^{-3}$
12	$5,697 \times 10^{-4}$	$7,253 \times 10^{-4}$	$9,223 \times 10^{-4}$
13	$5,761 \times 10^{-4}$	$7,125 \times 10^{-4}$	$9,163 \times 10^{-4}$

14	$6,519 \times 10^{-4}$	$8,114 \times 10^{-4}$	$1,041 \times 10^{-3}$
15	$6,876 \times 10^{-4}$	$8,127 \times 10^{-4}$	$1,065 \times 10^{-3}$
16	$5,826 \times 10^{-4}$	$6,953 \times 10^{-4}$	$9,071 \times 10^{-4}$
17	$5,751 \times 10^{-4}$	$6,999 \times 10^{-4}$	$9,059 \times 10^{-4}$
18	$5,794 \times 10^{-4}$	$7,012 \times 10^{-4}$	$9,096 \times 10^{-4}$
19	$5,541 \times 10^{-4}$	$6,310 \times 10^{-4}$	$8,398 \times 10^{-4}$
20	$5,519 \times 10^{-4}$	$6,487 \times 10^{-4}$	$8,517 \times 10^{-4}$

Tabel A. 9 Hasil optimisasi konsumsi energi menggunakan algoritma PSO.

	Running		
	Ke-1	Ke-2	Ke-3
C = 0,001			
F (kgmole/h)	928,584	826,707	938,201
Tf (°C)	103,161	104,918	102,005
Pc (kPa)	1527,69	1515,541	1531,303
Pr (kPa)	1579,63	1574,9	1570,141
Qc (kW)	8037,441	7186,329	8085,745
Qr (kW)	3604,877	3191,231	3644,445
Iterasi	164	179	175
Ec (kJ/kgmole)	52194,47827	52418,35513	51969,96266
Er (kJ/kgmole)	34678,99174	34482,90169	34700,28978
Operating Cost (\$/kgmole)	0,47852817	0,477302535	0,4780903
C = 0,0015			
F (kgmole/h)	700	901,5794	751,3687
Tf (°C)	105	105	105
Pc (kPa)	1460,61	1518,218	1440,068
Pr (kPa)	1570,1	1570	1597,593
Qc (kW)	6104,445	7836,556	6563,987
Qr (kW)	2687,281	3476,335	2906,245

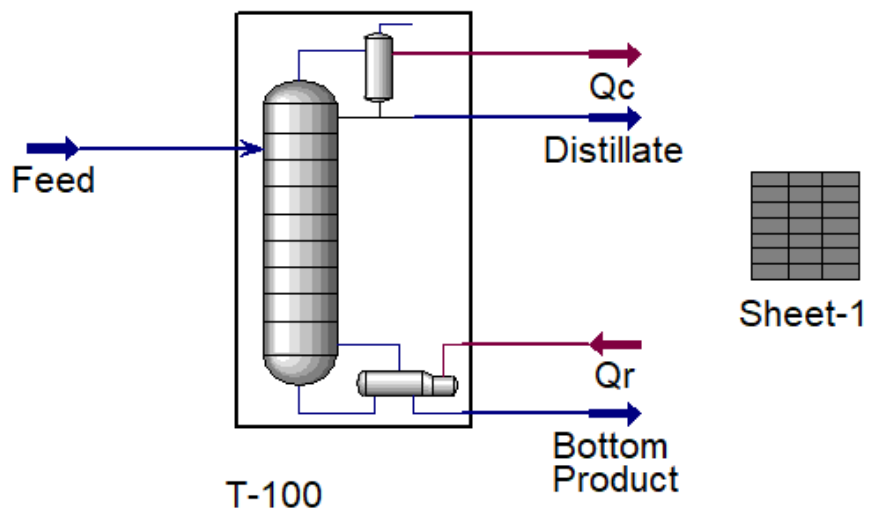
Iterasi	171	123	152
Ec (kJ/kgmole)	52586,74825	52414,25862	52679,64398
Er (kJ/kgmole)	34293,55486	34444,12182	34552,2682
Operating Cost (\$/kgmole)	0,475983	0,476922	0,478707
C = 0,002			
F (kgmole/h)	1100	700	1100
Tf (°C)	105	105	105
Pc (kPa)	1440	1476,993	1440
Pr (kPa)	1570	1570	1570
Qc (kW)	9602,715	6098,802	9602,715
Qr (kW)	4218,381	2689,375	4218,381
Iterasi	75	125	77
Ec (kJ/kgmole)	52641,65374	52538,13529	52641,65374
Er (kJ/kgmole)	34257,09862	34320,28287	34257,09862
Operating Cost (\$/kgmole)	0,475793	0,476098	0,475793

Tabel A. 10 Validasi hasil optimisasi konsumsi energi.

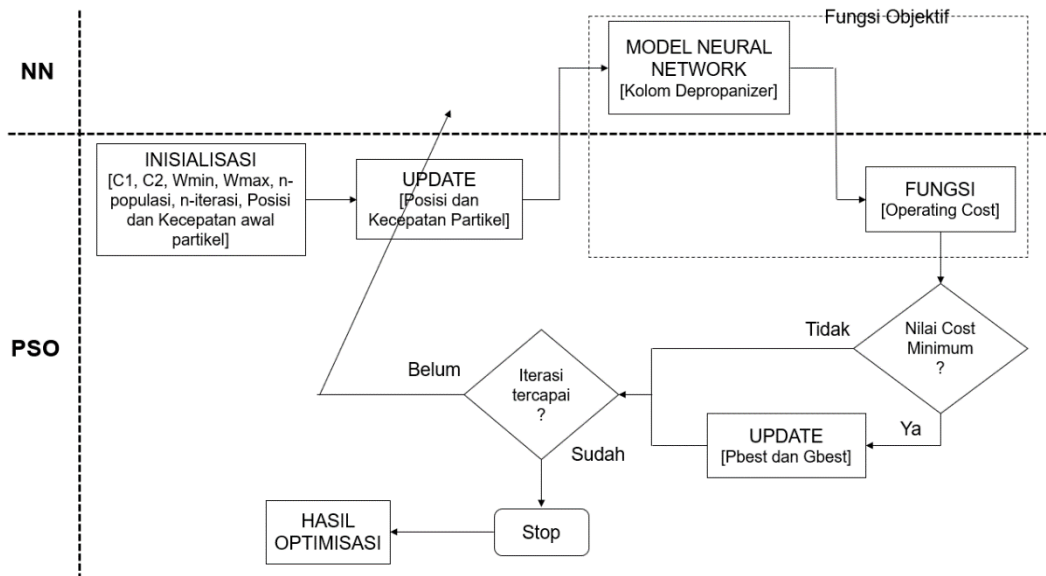
	Variabel	Matlab	Hysys	Error (%)
C=0,001	Qc (kW)	8037,441	8029,000	-0,10513
running ke-1	Qr (kW)	3604,877	3601,000	-0,10765
C=0,001	Qc (kW)	7186,329	7187,000	0,009333
running ke-2	Qr (kW)	3191,231	3186,000	-0,16419
C=0,001	Qc (kW)	8085,745	8080,000	-0,0711
running ke-3	Qr (kW)	3644,445	3637,000	-0,20471
C=0,0015	Qc (kW)	6104,445	6103,000	-0,02368
running ke-1	Qr (kW)	2687,281	2685,000	-0,08496
C=0,0015	Qc (kW)	7836,556	7836,000	-0,0071
running ke-2	Qr (kW)	3476,335	3469,000	-0,21145

C=0,0015	Qc (kW)	6563,987	6565,000	0,015423
running ke-3	Qr (kW)	2906,245	2903,000	-0,11177
C=0,002	Qc (kW)	9602,715	9602,000	-0,00745
running ke-1	Qr (kW)	4218,381	4215,000	-0,08021
C=0,002	Qc (kW)	6098,802	6097,000	-0,02955
running ke-2	Qr (kW)	2689,375	2687,000	-0,08841
C=0,003	Qc (kW)	9602,715	9602,000	-0,00745
running ke-3	Qr (kW)	4218,381	4215,000	-0,08021

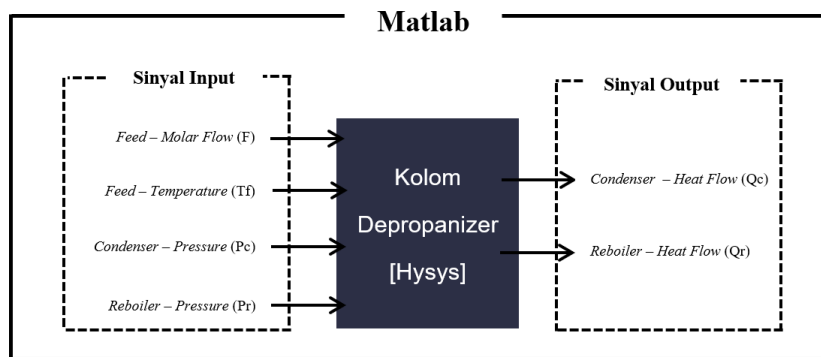
B. Model dan Skema



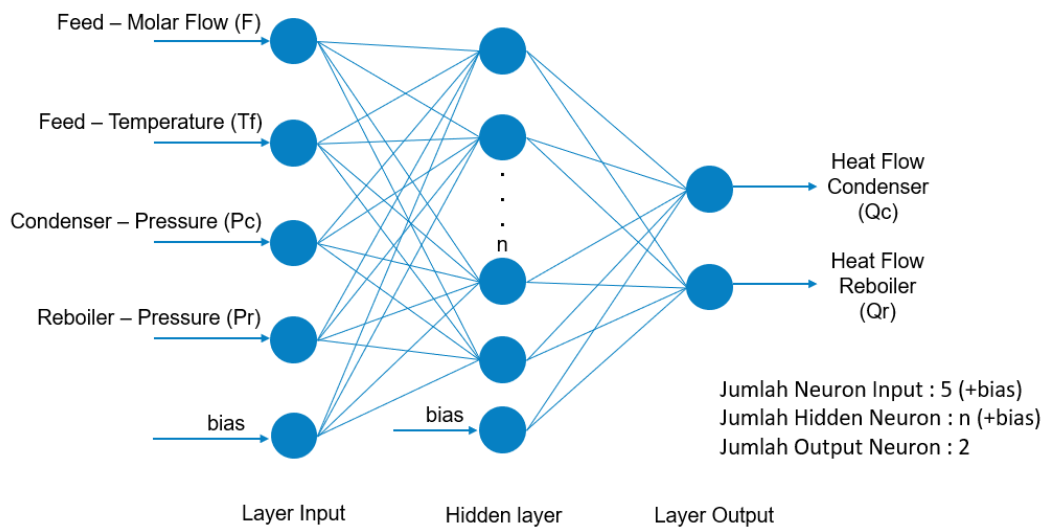
Gambar B. 1 Model PFD Kolom Depropanizer pada *software* Hysys.



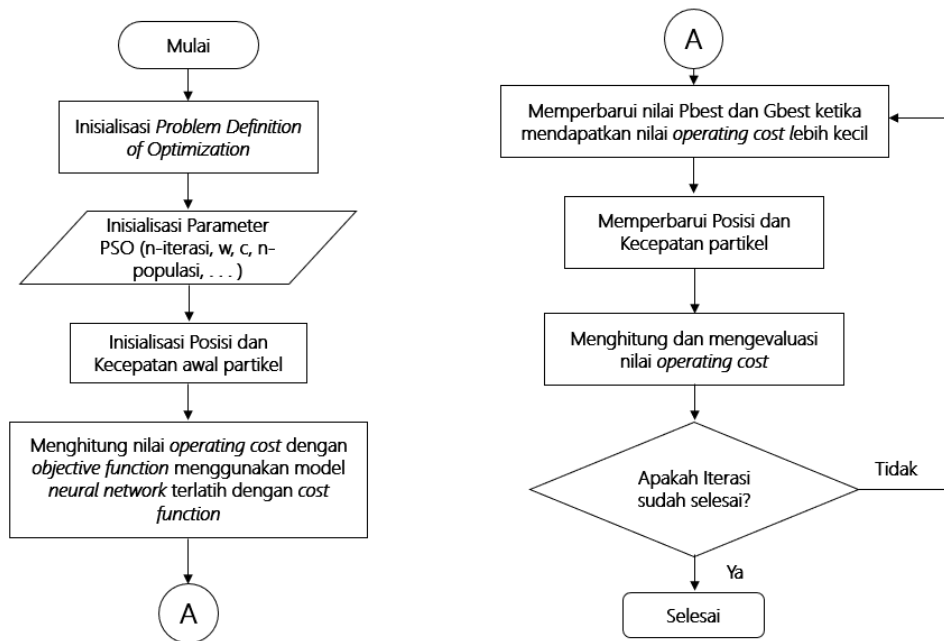
Gambar B. 2 Skema optimisasi berbasis NN-PSO.



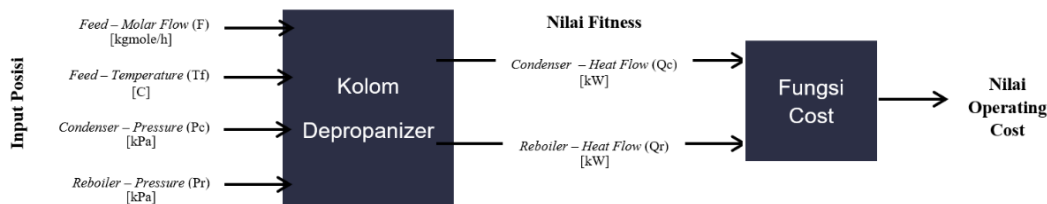
Gambar B. 3 Skema pengambilan data training *Neural Network*.



Gambar B. 4 Arsitektur *Neural Network*.



Gambar B. 5 Flowchart perancangan algoritma PSO.



Gambar B. 6 Fungsi objektif algoritma PSO.

C. Kode Matlab

Tabel C. 1 Source code pengambilan data training Neural Network.

Source code :

```

clc;
clear;
close all;
Hsys=actxserver('Hysys.Application.V8.8'); % Mengkoneksi
software Matlab ke Hysys
[stat,mess]=fileattrib;
simcase = Hsys.SimulationCases.Open([mess.Name '\ss_Dp.hsc']);
% Membuka simulasi Hysys(nama file "ss_Dp.hsc")
simcase.invoke('Activate');
fs=simcase.get('flowsheet'); % Mengakses Flowsheet
op=fs.get('Operations'); % Mengakses parameter operasional pada
Flowsheet
ms=fs.get('MaterialStreams'); % Mengakses Material Stream pada
Flowsheet
  
```

```

ns=fs.get('EnergyStreams'); % Mengakses Energy Stream pada
Flowsheet
sheet = op.Item('Sheet-1'); % Mengakses sheet
HySolver = simcase.Solver; % De/Activating solver pada Hysys

for i=1:1:10000 % Looping 10000x untuk pasangan data input -
output
tic;
HySolver.CanSolve = 0;
HySolver.CanSolve = 1;
input.flowfeed(i)=unifrnd(0.1945, 0.3055, 1); % Membangkitkan
sinyal random flowfeed
input.tempfeed(i)=unifrnd(55, 105, 1); % Membangkitkan sinyal
random tempfeed
input.pcond(i)=unifrnd(1440, 1540, 1); % Membangkitkan sinyal
random pcond
input.preb(i)=unifrnd(1570, 1670, 1); % Membangkitkan sinyal
random preb
sheet.Cell('A1').CellValue=input.flowfeed(i); % Mengirim sinyal
input flowfeed ke Hysys
sheet.Cell('A2').CellValue=input.tempfeed(i); % Mengirim sinyal
input tempfeed ke Hysys
sheet.Cell('A3').CellValue=input.pcond(i); % Mengirim sinyal
input pcond ke Hysys
sheet.Cell('A4').CellValue=input.preb(i); % Mengirim sinyal
input preb ke Hysys
HySolver.CanSolve = 0;
HySolver.CanSolve = 1;
output.condenser_duty(i)=sheet.Cell('B1').CellValue; % Merekam
data sinyal output Qc dari Hysys
output.reboiler_duty(i)=sheet.Cell('B2').CellValue; % Merekam
data sinyal output Qr dari Hysys
toc;
end

```

Tabel C. 2 Source Code Algoritma PSO

Source code :

```

clc;
clearvars -except ff19; % Clear semua variabel pada saat running
kecuali model neural network (ff19)
close all;

%% Mendefinisikan Problem Optimisasi

nVar = 4; % Input Neural Network (F, Tf, Pc, Pr)
VarSize = [nVar 1]; % Matriks untuk input Neural Network (4x1)
VarMin = 0; % Variabel Operasional Minimum nilai
normalisasi(F, Tf, Pc, Pr)
VarMax = 1; % Variabel Operasional Minimum nilai
normalisasi(F, Tf, Pc, Pr)
D = 0.597; % Spesifikasi Distillate to feed ratio
B = 0.403; % Spesifikasi Bottom Product to feed ratio

```

```

%% Mendefinisikan Parameter PSO

MaxIt = 1000;          % Iterasi Maksimum
nPop = 25;            % Ukuran Populasi (jumlah partikel)
w = 1;               % Koefisien Bobot Inersia
wdamp = 0.0005;      % Rasio damping dari Bobot Inersia
c1 = 0.002;          % Personal Acceleration Coefficient
c2 = 0.002;          % Social Acceleration Coefficient

%% Inisialisasi

% Struktur Partikel
empty_particle.Position = [];
empty_particle.Velocity = [];
empty_particle.Fitness = [];
empty_particle.Energy = [];
empty_particle.Cost = [];
empty_particle.Best.Position = [];
empty_particle.Best.Fitness = [];
empty_particle.Best.Energy = [];
empty_particle.Best.Cost = [];

% Membuat Array Populasi
particle = repmat(empty_particle, nPop, 1);

% Inisialisasi Global Best
GlobalBest.Cost = inf;
% Inisialisasi awal partikel dalam Populasi
for i=1:nPop
    % Membangkitkan nilai Posisi awal dengan nilai random
    F = rand(1);      % Komponen variabel operasional Feed
    flow
    Tf = rand(1);    % Komponen variabel operasional
    Temperature feed
    Pc = rand(1);    % Komponen variabel operasional Pressure
    Condenser
    Pr = rand(1);    % Komponen variabel operasional Pressure
    Reboiler

    particle(i).Position = [F;Tf;Pc;Pr];
    % Inisialisasi Kecepatan awal adalah nol
    particle(i).Velocity = zeros(VarSize);
    % Fungsi Evaluasi (fungsi objektif)
    particle(i).Fitness = abs(ff19(particle(i).Position)); %
    Kalkulasi nilai fitness dengan model neural network
    particle(i).Energy = particle(i).Fitness*10000; %
    particle(i).Energy = nilai denormalisasi
    particle(i).Cost =
    ((particle(i).Energy(1,1)/(((particle(i).Position(1,1)*400)+700)
    *D))*90 +
    (particle(i).Energy(2,1)/(((particle(i).Position(1,1)*400)+700)*
    B))*300)/8766; % Kalkulasi biaya Utilitas ($/kgmole)
    % Update nilai Personal Best
    particle(i).Best.Position = particle(i).Position;
    particle(i).Best.Fitness = particle(i).Fitness;

```

```

particle(i).Best.Energy = particle(i).Energy;
particle(i).Best.Cost = particle(i).Cost;
% Update nilai Global Best
if particle(i).Best.Cost < GlobalBest.Cost
    GlobalBest = particle(i).Best;
end
end
% Array untuk menyimpan Nilai Cost Terbaik pada Setiap Iterasi
BestCosts = zeros (MaxIt, 1);

%% Fungsi Main loop Algoritma PSO
for it=1:MaxIt

    for i=1:nPop
        % Update Velocity setiap Partikel
        particle(i).Velocity = w*particle(i).Velocity +
c1*rand(VarSize).*(particle(i).Best.Position -
particle(i).Position) + c2*rand(VarSize).*(GlobalBest.Position -
particle(i).Position);
        % Update Posisi setiap Partikel
        particle(i).Position = particle(i).Position +
particle(i).Velocity;
        % Batas Update Posisi (variabel operasional) setiap
partikel
        particle(i).Position = max(particle(i).Position, VarMin);
particle(i).Position = min(particle(i).Position, VarMax);
        % Fungsi Evaluasi (fungsi objektif)
        particle(i).Fitness = abs(ff19(particle(i).Position));
% Kalkulasi nilai fitness dengan model neural network
        particle(i).Energy = particle(i).Fitness*10000;
% particle(i).Energy = nilai denormalisasi
        particle(i).Cost =
((particle(i).Energy(1,1)/(((particle(i).Position(1,1)*400)+700)
*D))*90 +
(particle(i).Energy(2,1)/(((particle(i).Position(1,1)*400)+700)*
B))*300)/8766; % Kalkulasi biaya Utilitas
        % Update nilai Personal Best
        if particle(i).Cost < particle(i).Best.Cost
            particle(i).Best.Position = particle(i).Position;
            particle(i).Best.Fitness = particle(i).Fitness;
            particle(i).Best.Energy = particle(i).Energy;
            particle(i).Best.Cost = particle(i).Cost;
            % Update nilai Global Best
            if particle(i).Best.Cost < GlobalBest.Cost
                GlobalBest = particle(i).Best;
            end
        end
    end
    % Menyimpan nilai Best Cost
    BestCosts(it) = GlobalBest.Cost;
    % Menampilkan Informasi Iterasi
    disp(['Iteration ' num2str(it) ': Best Cost = $/kgmole '
num2str(BestCosts(it))]);
    % Rasio Damping Bobot Inersia
    w = w - wdamp;
end
%% Hasil

```

```
% De-normalisasi Posisi partikel (Global best)
RealPosition =
[ ((GlobalBest.Position(1,1)*400)+700); ((GlobalBest.Position(2,1)
*50)+55); ((GlobalBest.Position(3,1)*100)+1440); ((GlobalBest.Posi
tion(4,1)*100)+1570)];
% Plot Grafik
figure;
plot(BestCosts, 'LineWidth', 2);
xlabel('Iteration');
ylabel('Best Cost ($/kgmole)');
grid on;
```

BIODATA PENULIS



Ian Haikal Amir Akbar, lahir di Kota Gresik pada tanggal 06 Februari 1999 dari pasangan Bapak Mahrus dan ibu Ena Kowati. Penulis merupakan anak ketiga dari empat bersaudara. Pada tahun 2010 penulis menyelesaikan pendidikan Sekolah Dasar di SDN 1 Pangkah Wetan, pada tahun 2013 menyelesaikan pendidikan Sekolah Menengah Pertama di SMPN 1 Ujung Pangkah, pada tahun 2016 menyelesaikan pendidikan Sekolah Menengah Atas di SMAN 1 Sidayu. Pada tahun yang sama penulis terdaftar sebagai mahasiswa di Departemen Teknik Fisika Institut Teknologi Sepuluh Nopember (ITS). Kegiatan aktif penulis selama menempuh studi S1 Teknik Fisika di ITS diantaranya Panitia EPW 2017-2018, ITS Open 2018, Wakadep Ristek HMTF ITS 2018/2019, serta memenangkan kompetisi keilmiahan dengan meraih Silver Medal pada event IT-FEST, dan PKM terdani oleh dikti bidang Karsa Cipta. Penulis bergabung menjadi asisten laboratorium “Workshop Instrumentasi” pada tahun ketiga dan telah mengikuti proyek “Audit Energi” di TBBM Pertamina Bangka Balam. Pada tahun keempat penulis mengikuti PMMB (Program Magang Mahasiswa Bersertifikat) di PT Pertamina (Persero) RU – V Balikpapan dengan penempatan sebagai Jr. Instrument Inspection Engineer di departemen Project Engineering. Konsentrasi tugas akhir yang didalami adalah bidang rekayasa instrumentasi dan kontrol. Pada bulan Juli 2019 penulis telah menyelesaikan Tugas Akhir dengan judul OPTIMISASI KONSUMSI ENERGI PADA PROSES KOLOM DEPROPANIZER BERBASIS NEURAL NETWORK – PARTICLE SWARM OPTIMIZATION (NN-PSO).

Setiap kali Anda ingin berdiskusi lebih lanjut tentang tugas akhir, serta memberikan kritik dan saran maka dapat menghubungi penulis melalui email: ianhaikalamirakbar@gmail.com.