

**TUGAS AKHIR - TM 184835**

**OPTIMASI GAYA POTONG MENGGUNAKAN METODE  
BACKPROPAGATION NEURAL NETWORK ARTIFICIAL  
BEE COLONY ALGORITHM DAN DIFFERENTIAL  
EVOLUTION UNTUK PROSES END MILLING PADA  
INCONEL 718**

**NAJIBUL KAFFIN**

**NRP 02111740000130**

Dosen Pembimbing

**Mohammad Khoirul Effendi, S.T., M.Sc.Eng., Ph.D.**

**NIP. 198204142010121001**

**TEKNIK MESIN**

Departemen Teknik Mesin

Fakultas Teknologi Industri dan Rekayasa Sistem

Institut Teknologi Sepuluh Nopember

Surabaya

2022



**TUGAS AKHIR - TM 184835**

**OPTIMASI GAYA POTONG MENGGUNAKAN METODE  
BACKPROPAGATION NEURAL NETWORK ARTIFICIAL  
BEE COLONY ALGORITHM DAN DIFFERENTIAL  
EVOLUTION UNTUK PROSES END MILLING PADA  
INCONEL 718**

**NAJIBUL KAFFIN**  
NRP 02111840000130

Dosen Pembimbing  
**Mohammad Khoirul Effendi, S.T., M.Sc.Eng., Ph.D.**  
NIP 198204142010121001

**TEKNIK MESIN**  
Departemen Teknik Mesin  
Fakultas Teknologi Industri dan Rekayasa Sistem  
Institut Teknologi Sepuluh Nopember  
Surabaya  
2022



FINAL PROJECT - TM 184835

***CUTTING FORCE OPTIMIZATION USING  
BACKPROPAGATION NEURAL NETWORK ARTIFICIAL  
BEE COLONY ALGORITHM AND DIFFERENTIAL  
EVOLUTION FOR END MILLING OF INCONEL 718***

**NAJIBUL KAFFIN**  
NRP 02111840000130

Advisor  
**Mohammad Khoirul Effendi, S.T., M.Sc.Eng., Ph.D.**  
NIP 198204142010121001

**MECHANICAL ENGINEERING**  
Department of Mechanical Engineering  
Faculty of Industrial Technology and System Engineering  
Institut Teknologi Sepuluh Nopember  
Surabaya  
2022

## LEMBAR PENGESAHAN

### OPTIMASI GAYA POTONG MENGGUNAKAN METODE BACKPROPAGATION NEURAL NETWORK ARTIFICIAL BEE COLONY ALGORITHM DAN DIFFERENTIAL EVOLUTION UNTUK PROSES END MILLING PADA INCONEL 718

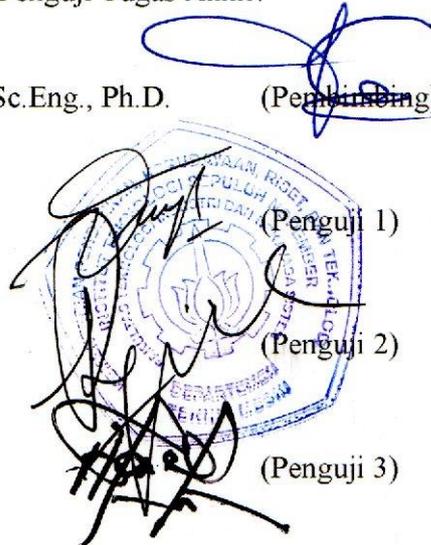
#### TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat  
memperoleh gelar Sarjana Teknik  
Program Studi S-1 Departemen Teknik Mesin  
Fakultas Teknologi Industri dan Rekayasa Sistem  
Institut Teknologi Sepuluh Nopember

Oleh : **NAJIBUL KAFFIN**  
NRP. 0211184000018

Disetujui oleh Tim Penguji Tugas Akhir:

1. Mohammad Khoirul Effendi, S.T., M.Sc.Eng., Ph.D. (Penyiseng)  
NIP 198204142010121001
2. Dinny Harnany, S.T., M.Sc. (Penguji 1)  
NIP 198905132019032013
3. Arif Wahjudi, S.T., M.T., Ph.D. (Penguji 2)  
NIP 197303222001121001
4. Ir. Sampurno, M.T. (Penguji 3)  
NIP 196504041989031002



**SURABAYA**

**Juli, 2022**

## APPROVAL SHEET

### CUTTING FORCE OPTIMIZATION USING BACKPROPAGATION NEURAL NETWORK ARTIFICIAL BEE COLONY ALGORITHM AND DIFFERENTIAL EVOLUTION FOR END MILLING OF INCONEL 718

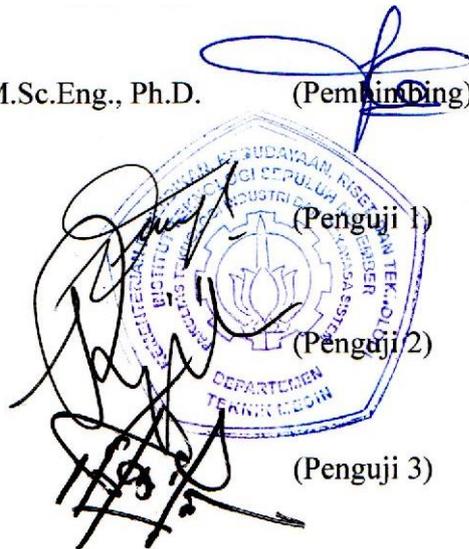
#### FINAL PROJECT

Submitted to fulfill one of the requirements  
for obtaining a degree Bachelor of Engineering at  
Undergraduate Study Program of Mechanical Engineering  
Department of Mechanical Engineering  
Faculty of Industrial Technology and Systems Engineering  
Institut Teknologi Sepuluh Nopember

By : **NAJIBUL KAFFIN**  
NRP. 0211184000018

Approved by Final Project Advisor:

1. Mohammad Khoirul Effendi, S.T., M.Sc.Eng., Ph.D. (Pembimbing)  
NIP 198204142010121001
2. Dinny Harnany, S.T., M.Sc. (Penguji 1)  
NIP 198905132019032013
3. Arif Wahjudi, S.T., M.T., Ph.D. (Penguji 2)  
NIP 197303222001121001
4. Ir. Sampurno, M.T. (Penguji 3)  
NIP 196504041989031002



**SURABAYA**

**July, 2022**

## PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama Mahasiswa / NRP : Najibul Kaffin / 02111840000130  
Departemen : Teknik Mesin  
Dosen Pembimbing / NIP : Mohammad Khoirul Effendi, S.T., M.Sc.Eng., Ph.D. /  
198204142010121001

dengan ini menyatakan bahwa Tugas Akhir dengan judul “Optimasi Gaya Potong Menggunakan Metode *Backpropagation Neural Network Artificial Bee Colony Algorithm* dan *Differential Evolution* untuk Proses *End Milling* pada Inconel 718” adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, 27 Juli 2022

Mengetahui,  
Dosen Pembimbing

Mahasiswa,



Mohammad Khoirul Effendi, S.T., M.Sc.Eng., Ph.D.  
NIP. 198204142010121001



Najibul Kaffin  
NRP. 02111840000130

## STATEMENT OF ORIGINALITY

The undersigned below:

Name of Student / NRP : Najibul Kaffin / 02111840000130  
Department : Mechanical Engineering  
Supervisor / NIP : Mohammad Khoirul Effendi, S.T., M.Sc.Eng., Ph.D. /  
198204142010121001

hereby declare that the Final Project with the title of “Cutting Force Optimization using Backpropagation Neural Network Artificial Bee Colony Algorithm and Differential Evolution for End Milling of Inconel 718” is the result of my own work, is original, and is written by following the rules of scientific writing

If in the future there is a discrepancy with this statement, then I am willing to accept sanctions in accordance with the provisions that apply at Institut Teknologi Sepuluh Nopember.

Surabaya, July 27<sup>th</sup> 2022

Acknowledged,  
Advisor



Mohammad Khoirul Effendi, S.T., M.Sc.Eng., Ph.D.  
NIP. 198204142010121001

Student,



Najibul Kaffin  
NRP. 02111840000130

**OPTIMASI GAYA POTONG MENGGUNAKAN METODE BACKPROPAGATION  
NEURAL NETWORK ARTIFICIAL BEE COLONY ALGORITHM DAN  
DIFFERENTIAL EVOLUTION UNTUK PROSES END MILLING PADA INCONEL  
718**

**Nama Mahasiswa / NRP : Najibul Kaffin**  
**Departemen : Teknik Mesin FTIRS-ITS**  
**Dosen Pembimbing : Mohammad Khoirul Effendi, S.T., M.Sc.Eng., Ph.D.**

**Abstrak**

Proses pemesinan merupakan faktor penting dalam pembuatan suatu produk. Salah satu proses pemesinan konvensional yang sering dilakukan yaitu *end milling*. Proses *end milling* merupakan proses pemotongan menggunakan *end mill* dimana permukaan benda kerja tegak lurus terhadap ujung pahat yang berputar. Pada proses *end milling*, harus memperhatikan gaya potong untuk menghindari defleksi yang mengakibatkan keausan pada pahat. Oleh karena itu perlu dilakukan upaya untuk meningkatkan umur pahat terhadap keausan pada proses *end milling* dengan cara optimasi terhadap parameter *end milling* seperti memvariasikan parameter input (kecepatan potong, kecepatan makan, kedalaman potong, dan lebar pemotongan) agar menghasilkan resultan gaya potong yang minimum, sehingga desain proses *end milling* yang optimal dapat ditentukan.

Penelitian ini berdasarkan pada jurnal dengan judul “*Prediction of Cutting Force in End Milling of Inconel 718*” oleh (Kasim, et al., 2015). Pada penelitian ini digunakan rancangan eksperimen menggunakan 3 variasi kecepatan potong yaitu 100, 120, 140 m/menit, 3 variasi jenis kecepatan makan yaitu 0.1, 0.15, dan 0.2 mm/gigi, 3 variasi kedalaman potong yaitu 0.5, 0.75, dan 1 mm, dan 3 variasi lebar pemotongan yaitu 0.2, 1, dan 1.8 mm. Variasi data diperlukan agar optimasi yang dilakukan menggunakan metode kecerdasan buatan berupa *Backpropagation Neural Network* (BPNN) sebagai alat untuk memodelkan hubungan antara parameter input (kecepatan potong, kecepatan makan, kedalaman potong, dan lebar pemotongan) dengan parameter output (resultan gaya potong) dapat menghasilkan prediksi yang baik. Prediksi tersebut digunakan untuk mendapatkan parameter output berupa resultan gaya potong paling minimum menggunakan metode optimasi yaitu *Artificial Bee Colony Algorithm* (ABC) dan *Differential Evolution* (DE).

Berdasarkan hasil prediksi menggunakan metode *Backpropagation Neural Network* (BPNN) konfigurasi yang menghasilkan kesalahan prediksi paling kecil yaitu BPNN dengan jumlah *hidden layer* 2, jumlah neuron 10 pada *hidden layer* pertama dengan fungsi aktivasi *tansig*, jumlah neuron 10 pada *hidden layer* kedua dengan fungsi aktivasi *tansig* dengan nilai MSE sebesar 0.00096269. Sedangkan hasil optimasi metode *Artificial Bee Colony* (ABC) dan *Differential Evolution* (DE), parameter yang menghasilkan nilai optimal bernilai sama yaitu kecepatan potong 122 m/min, kecepatan makan 0.12 mm/gigi, kedalaman potong 0.5 mm, dan lebar pemotongan 0.2 mm dengan output resultan gaya potong sebesar 139.9599 N. Hasil optimasi resultan gaya potong menggunakan metode optimasi BPNN-ABC dan BPNN-DE sebesar 139.9599 N, lebih kecil 6.8% dibandingkan dengan penelitian sebelumnya yang menggunakan metode RSM sebesar 150.23 N (Kasim et al., 2015).

**Kata Kunci : *End milling, backpropagation neural network, artificial bee colony, differential evolution, resultan gaya potong***

# CUTTING FORCE OPTIMIZATION USING BACKPROPAGATION NEURAL NETWORK ARTIFICIAL BEE COLONY ALGORITHM AND DIFFERENTIAL EVOLUTION FOR END MILLING OF INCONEL 718

**Student Name / NRP** : Najibul Kaffin / 02111840000130  
**Department** : Mechanical Engineering FTIRS-ITS  
**Advisor** : Mohammad Khoirul Effendi, S.T., M.Sc.Eng., Ph.D.

## Abstract

The machining process is an important factor in the manufacture of a product. One of the conventional machining processes that is often carried out is end milling. The end milling process is a cutting process using an end mill where the workpiece surface is perpendicular to the rotating tool tip. In the end milling process, attention must be paid to the cutting force to avoid deflection which results in tool wear. Therefore, it is necessary to make efforts to increase tool life against wear in the end milling process by optimizing the end milling parameters such as varying the input parameters (cutting speed, feeding speed, depth of cut, and cutting width) in order to produce a minimum cutting force resultant, so that the design the optimal end milling process can be determined.

This research is based on a journal entitled "Prediction of Cutting Force in End Milling of Inconel 718" by (Kasim, et al., 2015). In this study, an experimental design was used using 3 variations in cutting speed, namely 100, 120, 140 m/min, 3 variations in feeding speed, namely 0.1, 0.15, and 0.2 mm/tooth, 3 variations in depth of cut, namely 0.5, 0.75, and 1 mm, and 3 variations in cutting width, namely 0.2, 1, and 1.8 mm. Variation of data is needed so that optimization carried out using artificial intelligence methods in the form of Backpropagation Neural Network (BPNN) as a tool to model the relationship between input parameters (cutting speed, feeding speed, depth of cut, and cutting width) with output parameters (maximum cutting force) can produce good prediction. The prediction is used to obtain the output parameter in the form of the minimum maximum cutting force using optimization methods, namely Artificial Bee Colony Algorithm (ABC) and Differential Evolution (DE).

Based on the prediction results using the Backpropagation Neural Network (BPNN) configuration that produces the smallest prediction error, namely BPNN with the number of hidden layer 2, the number of neurons 10 in the first hidden layer with the TANSIG activation function, the number of 10 neurons in the second hidden layer with the TANSIG activation function with MSE value is 0.00096269. While the results of the optimization of the Artificial Bee Colony (ABC) and Differential Evolution (DE) methods, the parameters that produce the optimal value are the same, namely cutting speed 122 m/min, feeding speed 0.12 mm/tooth, depth of cut 0.5 mm, and cutting width 0.2 mm with the resultant cutting force output is 139.9599 N. The resultant cutting force optimization result using the BPNN-ABC and BPNN-DE optimization methods is 139.9599 N, 6.8% smaller than the previous study using the RSM method of 150.23 N (Kasim et al., 2015).

**Keywords** : *End milling, backpropagation neural network, artificial bee colony, cutting force resultant*

## KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa, Maha Pengasih, lagi Maha Penyayang, karena atas berkah dan izin-Nya Tugas Akhir ini dapat terselesaikan. Penulis sangat menyadari bahwa keberhasilan dalam penyelesaian Tugas Akhir ini tidak terlepas dari dukungan dan bantuan dari berbagai pihak. Melalui kesempatan yang baik ini, penulis ingin menyampaikan ucapan terima kasih dan penghargaan yang setinggi-tingginya kepada pihak-pihak yang telah banyak membantu dan mendukung baik secara moril maupun materil dalam proses penyelesaian Tugas Akhir ini, antara lain:

1. Orang tua yang senantiasa memberikan dukungan dan doa hingga penulis dapat menyelesaikan Tugas Akhir ini.
2. Bapak M. Khoirul Effendi, S.T., M.Sc.Eng., Ph.D. selaku dosen pembimbing Tugas Akhir penulis yang selalu memberikan saran, arahan, motivasi, dan ilmunya yang sangat bermanfaat kepada penulis. Terima kasih atas dedikasi dan waktu yang telah diberikan.
3. Ibu Dinny Harnany, S.T., M.Sc., Bapak Arif Wahjudi, S.T., M.T., Ph.D., dan Bapak Ir. Sampurno, M.T., selaku dosen penguji Tugas Akhir penulis. Terima kasih atas waktu dan saran-saran yang telah diberikan.
4. Seluruh Bapak dan Ibu Dosen Teknik Mesin ITS yang telah mendidik penulis dalam mempelajari ilmu teknik mesin dan juga ilmu kehidupan yang sangat luas ini.
5. Keluarga M61 yang telah memberi pengalaman dan pembelajaran berharga kepada penulis selama berkuliah di Teknik Mesin ITS yang sama-sama kita cintai.
6. Rekan-rekan LKKI Ash-Shaff Teknik Mesin ITS yang telah menemani penulis dalam menjalani organisasi di kampus tercinta.
7. Rekan-rekan Laboratorium P3 Teknik Mesin yang telah membantu penulis menyelesaikan Tugas Akhir ini.
8. Seluruh pihak yang tidak dapat penulis sebutkan satu-persatu.

Penulis menyadari masih banyak kekurangan dalam penyusunan Tugas Akhir ini, oleh karena itu saran dan masukan dari semua pihak sangat penulis harapkan. Penulis berharap semoga tugas akhir ini dapat memberikan manfaat dan sumbangsih bagi perkembangan ilmu pengetahuan.

Surabaya, 27 Juli 2022



Penulis

## DAFTAR ISI

<b>LEMBAR PENGESAHAN</b> .....	iii
<b>ABSTRAK</b> .....	vii
<b>KATA PENGANTAR</b> .....	ix
<b>DAFTAR ISI</b> .....	x
<b>DAFTAR GAMBAR</b> .....	xii
<b>DAFTAR TABEL</b> .....	xiii
<b>BAB I</b> .....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	2
1.5 Manfaat Penelitian.....	2
2.1 Penelitian Terdahulu.....	3
2.2 Proses Pemesinan Frais ( <i>Milling</i> ).....	4
2.2.1 Klasifikasi Proses Frais.....	5
2.2.2 Metode Proses Frais.....	6
2.2.3 Jenis Mesin Frais.....	6
2.2.4 Geometri Pahat Frais.....	8
2.2.5 Elemen Dasar Proses Frais.....	9
2.3 Gaya Potong pada Proses <i>End Milling</i> .....	10
2.4 Umur Pahat.....	11
2.4.1 Bidang Aktif Pahat yang Mengalami Kerusakan.....	11
2.4.2 Kriteria Umur Pahat.....	12
2.5 Material Inconel 718.....	12
2.6 <i>Artificial Neural Network</i> .....	13
2.6.1 Fungsi Aktivasi.....	13
2.6.2 Bias.....	15
2.6.3 Pelatihan Jaringan Syaraf Tiruan.....	15
2.6.4 <i>Backpropagation Neural Network</i> .....	15
2.7 <i>Artificial Bee Colony Algorithm</i> .....	16
2.8 <i>Differential Evolution</i> .....	19
<b>BAB III</b> .....	22
3.1 Diagram Alir Penelitian.....	22

3.2	Observasi .....	23
3.3	Perumusan Permasalahan .....	23
3.4	Studi Literatur .....	23
3.5	Pengumpulan Data .....	23
3.5.1	Dimensi dan Material .....	23
3.5.2	Desain Eksperimen Penelitian .....	24
3.6	Pemodelan BPNN .....	25
3.7	Komparasi Hasil .....	29
3.8	Penarikan Kesimpulan dan Saran .....	29
<b>BAB IV</b>	.....	<b>30</b>
4.1	Parameter Input dan Respon .....	30
4.2	Pemodelan BPNN .....	30
4.3	Optimasi dengan Algoritma <i>Artificial Bee Colony</i> .....	34
4.4	Optimasi dengan Algoritma <i>Differential Evolution</i> .....	38
4.5	Penentuan Parameter Berdasarkan Kondisi Sebenarnya .....	41
4.6	Perbandingan Hasil Optimasi Metode ABC dan DE.....	42
4.7	Perbandingan dengan Penelitian Terdahulu .....	43
<b>BAB V</b>	.....	<b>45</b>
5.1	Kesimpulan .....	45
5.2	Saran .....	45
<b>DAFTAR PUSTAKA</b>	.....	<b>46</b>
<b>LAMPIRAN</b>	.....	<b>48</b>
<b>BIODATA PENULIS</b>	.....	<b>59</b>

## DAFTAR GAMBAR

<b>Gambar 2. 1</b>	Pemodelan menggunakan metode RSM (Kasim et al., 2015) .....	3
<b>Gambar 2. 2</b>	Klasifikasi tiga jenis frais (Rochim, 1993) .....	5
<b>Gambar 2. 3</b>	Frais naik dan frais turun (Rochim, 1993) .....	6
<b>Gambar 2. 4</b>	Column and knee type milling machines (Kalpakjian et al., 2014) .....	7
<b>Gambar 2. 5</b>	Bed type milling machines (Kalpakjian et al., 2014) .....	7
<b>Gambar 2. 6</b>	Geometri pahat frais (Rochim, 1993) .....	8
<b>Gambar 2. 7</b>	Tipe jenis end mill (Chong, 2018) .....	9
<b>Gambar 2. 8</b>	Elemen dasar proses frais vertikal dan horizontal (Rochim, 1993) .....	9
<b>Gambar 2. 9</b>	Gaya potong pada proses end milling (Luo et al., 2005) .....	10
<b>Gambar 2. 10</b>	Crater wear dan flank wear (Kalpakjian et al., 2014) .....	11
<b>Gambar 2. 11</b>	Contoh jaringan BPNN .....	16
<b>Gambar 2. 12</b>	Prosedur algoritma artificial bee colony (ABC) .....	17
<b>Gambar 3. 1</b>	Diagram alir penelitian .....	22
<b>Gambar 3. 2</b>	Pahat ball nose end mill (Kasim et al., 2015) .....	24
<b>Gambar 3. 3</b>	Diagram alir BPNN .....	26
<b>Gambar 3. 4</b>	Diagram alir algoritma Artificial Bee Colony .....	27
<b>Gambar 3. 5</b>	Diagram alir algoritma Differential Evolution .....	28
<b>Gambar 4. 1</b>	Struktur jaringan 4-10-10-1 .....	31
<b>Gambar 4. 2</b>	Hasil prediksi BPNN .....	32
<b>Gambar 4. 3</b>	Nilai konstanta R untuk data training, testing, dan validasi .....	34
<b>Gambar 4. 4</b>	Hasil optimasi ABC .....	35
<b>Gambar 4. 5</b>	Hasil optimasi DE .....	38
<b>Gambar 4. 6</b>	Perbandingan hasil optimasi ABC vs DE terhadap jumlah iterasi .....	42
<b>Gambar 4. 7</b>	Perbandingan hasil optimasi ABC vs DE terhadap waktu .....	43

## DAFTAR TABEL

<b>Tabel 2. 1</b> Perbandingan hasil optimasi DE dan RSM (Kamaruddin et al., 2019) .....	4
<b>Tabel 2. 2</b> Parameter optimal untuk kekasaran permukaan (Yusup et al., 2014) .....	4
<b>Tabel 2. 3</b> Spesifikasi CNC.....	8
<b>Tabel 2. 4</b> Batas keausan kritis (Rochim, 1993) .....	12
<b>Tabel 2. 5</b> Komposisi inconel 718 (Kasim et al., 2015) .....	12
<b>Tabel 2. 6</b> Fungsi aktivasi pada MATLAB.....	14
<b>Tabel 3. 1</b> Spesifikasi pahat (Kasim et al., 2015) .....	23
<b>Tabel 3. 2</b> Data input proses end milling .....	24
<b>Tabel 3. 3</b> Hasil eksperimen end milling (Kasim et al., 2015) .....	24
<b>Tabel 4. 1</b> Perbandingan nilai target dan nilai hasil prediksi BPNN .....	32
<b>Tabel 4. 2</b> Nilai optimal hasil optimasi ABC.....	37
<b>Tabel 4. 3</b> Nilai optimal hasil optimasi DE .....	40
<b>Tabel 4. 4</b> Perbandingan hasil optimasi dengan jurnal (Kasim et al., 2015) .....	44

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Industri manufaktur merupakan sektor sekunder dalam menunjang perekonomian di Indonesia. Dalam rentang waktu 10 tahun terakhir, pertumbuhan industri manufaktur di Indonesia mencapai angka 2.2-6.1% dimana sejak tahun 2008 nilai produk domestik bruto (PDB) industri manufaktur mengalami penurunan (Lestari, 2017). Hal tersebut mengharuskan Indonesia untuk lebih memperhatikan perkembangan pada industri manufaktur. Industri pada sektor manufaktur akan mengolah bahan mentah menjadi bahan setengah jadi ataupun bahan jadi melalui proses pemesinan. Dalam dunia manufaktur, proses pemesinan mempunyai peranan penting dalam pembuatan komponen suatu produk. Proses pemesinan adalah proses pemotongan menggunakan *cutting tool* yang dipasangkan dengan mesin perkakas (*machine tools*) dan menghasilkan geram akibat gerak relatif benda kerja terhadap mesin perkakas. Contoh proses pemesinan konvensional yang sering digunakan adalah frais atau *end milling*. *End milling* merupakan proses *milling* dimana permukaan benda kerja tegak lurus terhadap ujung pahat yang berputar. Hasil dari proses *end milling* yaitu terbentuknya slot akibat diameter pahat *end mill* yang lebih kecil dibandingkan lebar benda kerja.

Dengan perkembangan teknologi, proses *end milling* perlu dilakukan optimasi agar menghasilkan produk yang maksimal dengan kerja yang minimum. Optimasi yang dilakukan dapat menggunakan uji eksperimen. Uji eksperimen merupakan salah satu cara untuk menemukan ketepatan geometri dari pahat *end mill*. Dalam melakukan uji eksperimen pada proses *end milling* diperlukan pemilihan variabel yang tepat untuk menghasilkan produk yang baik. Langkah dalam pemilihan variabel diantaranya menganalisa gaya potong yang terjadi saat proses *end milling* dilakukan. Pemilihan variabel pahat *end mill* seperti *cutting speed*, *feed rate*, *depth of cut*, dan *width of cut* harus diperhatikan agar gaya potong yang terjadi saat proses *end milling* dapat diperkecil. Analisa ini perlu dilakukan karena gaya potong dapat menimbulkan defleksi pada pahat *end mill* maupun benda kerja. Adanya defleksi akan mengakibatkan kesalahan geometri pada produk sehingga kualitas produk yang dihasilkan menurun (Jalili, 2009).

Pada penelitian ini digunakan optimasi parameter *end milling* yaitu kecepatan potong, kecepatan makan, kedalaman potong, dan lebar pemotongan berdasarkan jurnal "*Prediction of Cutting Force in End Milling of Inconel 718*" (Kasim et al, 2015). Optimasi dilakukan untuk mendapat resultan gaya potong paling minimum menggunakan metode *Back Propagation Neural Network* (BPNN) untuk memprediksi persamaan polinomial orde tinggi dengan mengatur jumlah *hidden layer* dan neuron setiap *layer*nya. Dengan metode ini, nilai eror yang didapat akan digunakan kembali untuk memperbaiki jaringan di setiap *layer* sehingga didapatkan hasil akhir dari jaringan BPNN memiliki nilai eror yang kecil. Dalam metode BPNN, diterapkan algoritma optimasi *Artificial Bee Colony* (ABC) dan *Differential Evolution* (DE) karena mudah diimplementasikan dan memiliki parameter yang lebih sedikit (Zou et al., 2010). Algoritma lebah akan mensimulasikan koloni lebah berdasarkan perilaku dalam mencari makanan untuk mendapatkan solusi terbaik permasalahan optimal. Pada prosesnya, lebah mencari sumber makanan secara acak lalu membandingkan dengan sumber yang lainnya dengan nilai probabilitas dan dengan pengaruh koloni, lebah dapat berkomunikasi tentang sumber makanan terbaik (solusi) dengan lebah yang lainnya, sehingga ditemukan solusi terbaik di dalam ruang pencarian. Sedangkan algoritma DE mensimulasikan individu dalam suatu populasi yang dipilih secara acak dan akan mengalami diferensiasi dan *crossover* yang terjadi secara berurutan. Hasil proses tersebut akan menghasilkan individu yang akan menggantikan

induk apabila memiliki kecocokan keturunan yang lebih baik dari induknya dan dianggap sebagai solusi terbaik dari proses optimasi yang dilakukan.

## 1.2 Perumusan Masalah

Adapun rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana memilih parameter jumlah *hidden layer*, jumlah neuron setiap *hidden layer*, dan fungsi aktivasi dari *Back Propagation Neural Network* (BPNN) yang paling baik, sehingga menghasilkan nilai kesalahan prediksi target paling kecil?
2. Bagaimana menentukan kecepatan potong, kecepatan makan, kedalaman potong, dan lebar pemotongan untuk mendapatkan respon resultan gaya potong paling minimum menggunakan BPNN-*Artificial Bee Colony* (ABC) dan BPNN-*Differential Evolution* (DE)?
3. Bagaimana perbandingan optimasi metode BPNN-ABC dan BPNN-DE dibandingkan metode *Response Surface Methodology* (RSM)?

## 1.3 Batasan Masalah

Batasan masalah dari penelitian ini antara lain:

1. Data yang digunakan diambil dari penelitian dengan judul "*Prediction of Cutting Force in End Milling of Inconel 718*" oleh Kasim et al. (2015).
2. Perangkat lunak yang digunakan untuk mengolah data adalah MATLAB R2020b.
3. Kombinasi pada pemodelan BPNN yaitu jumlah *hidden layer* adalah 1 sampai 5 *layer*, jumlah neuron setiap *layer* adalah 2 sampai 10 neuron, menggunakan fungsi aktivasi *hardlim*, *hardlims*, *purelin*, *satlin*, *logsig*, dan *tansig* dengan 10 kali pengulangan setiap konfigurasi.
4. Prosentase jumlah data untuk *training*, *testing*, dan *validating* adalah 70%, 15% dan 15%.
5. Pada pemodelan algoritma optimasi ABC dan DE, digunakan iterasi maksimum yaitu 1000 dengan jumlah populasi yaitu 100.

## 1.4 Tujuan Penelitian

Berdasarkan rumusan masalah diatas, tujuan dari penelitian ini adalah sebagai berikut:

1. Mendapatkan parameter dari *Back Propagation Neural Network* (BPNN) yang paling baik, sehingga menghasilkan nilai kesalahan prediksi paling kecil.
2. Menentukan kecepatan potong, kecepatan makan, kedalaman potong, dan lebar pemotongan untuk mendapatkan respon resultan gaya potong paling minimum menggunakan BPNN-*Artificial Bee Colony* (ABC) dan BPNN-*Differential Evolution* (DE).
3. Menentukan perbandingan optimasi metode BPNN-ABC dan BPNN-DE dibandingkan metode *Response Surface Methodology* (RSM).

## 1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini sebagai berikut :

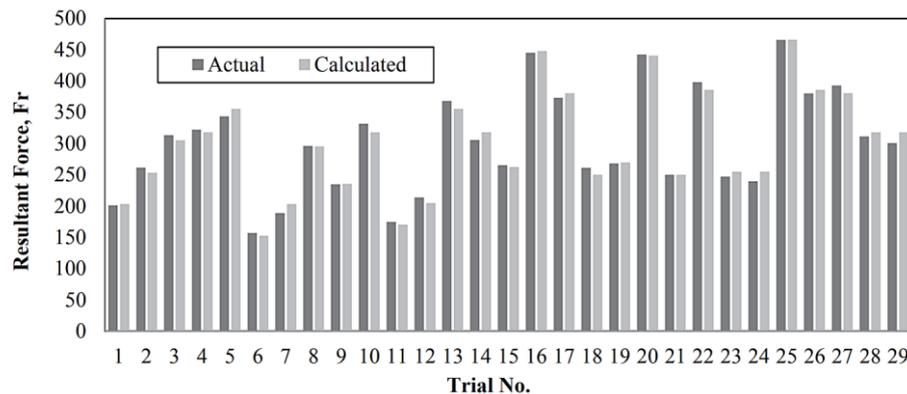
1. Mendapatkan nilai variabel kecepatan potong, kecepatan makan, kedalaman potong, dan lebar pemotongan untuk mendapat resultan gaya potong paling minimum.
2. Sebagai referensi untuk penelitian sejenis dalam rangka pengembangan pengetahuan optimasi dibidang proses *end milling*.

## BAB II TINJAUAN PUSTAKA

### 2.1 Penelitian Terdahulu

Terdapat tiga penelitian terdahulu yang masih berkaitan dengan penelitian ini dan bisa dijadikan sebagai referensi. Referensi pertama diambil dari penelitian Kasim et al. pada tahun 2015 dengan judul “*Prediction of Cutting Force in End Milling of Inconel 718*”. Penelitian dilakukan untuk menganalisa hubungan antara kecepatan potong, kecepatan makan, kedalaman potong, dan lebar pemotongan terhadap gaya potong pada material inconel 718. Penelitian dilakukan menggunakan mesin CNC vertikal DMC 635 V Eco dengan variasi kecepatan potong, kecepatan makan, kedalaman potong, dan lebar pemotongan. Tes *end milling* dilakukan pada benda kerja menggunakan *dynamometer* dan *charge amplifier* Kistler 5070A dengan pahat *ball nose end mill* berdiameter 10 mm. Parameter input yang digunakan yaitu kecepatan potong, kecepatan makan, kedalaman potong, dan lebar pemotongan. Sedangkan parameter output yang dianalisa yaitu gaya potong pada arah normal ( $F_x$ ), arah tangensial ( $F_y$ ), dan arah aksial ( $F_z$ ).

Dari hasil penelitian didapatkan bahwa resultan gaya potong dipengaruhi oleh gaya potong pada arah normal ( $F_x$ ) sebesar 64%, gaya potong arah tangensial ( $F_y$ ) sebesar 19%, dan gaya potong arah aksial ( $F_z$ ) sebesar 17%. Hasil optimasi menggunakan metode *Respon Surface Methodology* (RSM) pada gambar 2.1 didapatkan nilai rerata eror antara hasil eksperimen dan prediksi sebesar 2.6% dimana konfigurasi *end milling* yang menghasilkan resultan gaya potong paling kecil dengan kecepatan potong 125.7 m/menit, kecepatan makan 0.12 mm/gigi, kedalaman potong 0.52 mm, dan lebar pemotongan 0.22 mm dengan hasil resultan gaya potong bernilai 150.23 N (Kasim et al, 2015).



**Gambar 2. 1** Pemodelan menggunakan metode RSM (Kasim et al., 2015)

Referensi kedua diambil dari penelitian Kamaruddin et al. pada tahun 2019 dengan judul “*Optimization of Surface Roughness on CNC Milling Machining using Differential Evolution (DE) Method*”. Penelitian dilakukan menggunakan CNC vertikal AKIRA SEIKI Perfoma SR3 XP dan alat ukur kekasaran permukaan CS-3000. Tes *milling* dilakukan dengan variasi parameter input kecepatan putar, kecepatan makan, dan kedalaman potong. Kecepatan putar divariasikan 2500, 3000, dan 3500 RPM. Kecepatan makan divariasikan menjadi 300, 400, dan 500 mm/putaran. Sedangkan kedalaman potong divariasikan menjadi 0.15, 0.2, dan 0.25 mm. Pada penelitian ini digunakan optimasi menggunakan RSM dan DE untuk menentukan parameter input yang menghasilkan nilai kekasaran permukaan paling minimum. Dari hasil penelitian pada tabel 2.1 didapatkan untuk algoritma DE parameter input yang optimal yaitu kecepatan putar 3500 RPM, kecepatan makan 350.9 mm/putaran, dan kedalaman potong 0.2

mm dengan nilai prediksi kekasaran permukaan bernilai 0.82846. Hasil validasi algoritma DE yang dilakukan didapat nilai kekasaran permukaan 0.863 dimana terdapat eror 4% dari hasil prediksi. Sedangkan pada metode RSM parameter optimal yaitu kecepatan putar 3498.82 RPM, kecepatan makan 351.59 mm/putaran, dan kedalaman potong 0.25 mm dengan prediksi kekasaran permukaan 0.8333. Hasil validasi metode RSM yang dilakukan didapat nilai kekasaran permukaan 0.985 dimana terdapat eror 15.4% dari hasil prediksi.

**Tabel 2. 1** Perbandingan hasil optimasi DE dan RSM (Kamaruddin et al., 2019)

Method	Spindle Speed, a (rpm)	Feed rate, b (mm/rev)	Depth of cut, c (mm)	Surface Roughness, ( $\mu\text{m}$ )		Error (%)
	Optimal	Optimal	Optimal	Pred.	Exp.	
DE	3500	350.9	0.2	0.82846	0.863	4.00
RSM	3498.82	351.59	0.25	0.8333	0.985	15.4

Referensi ketiga diambil dari penelitian Norfadzlan Yusup et al. dengan judul “*Artificial Bee Colony in Optimizing Process Parameters of Surface Roughness in End Milling and Abrasive Waterjet Machining*”. Penelitian yang dilakukan pada tahun 2014 ini melakukan optimasi untuk menentukan kekasaran permukaan minimum pada material Al 7075-T6 menggunakan proses *end milling* dan *abrasive waterjet machining*. Optimasi yang dilakukan menggunakan algoritma *artificial bee colony* dengan variasi kecepatan potong, kecepatan makan, dan sudut pahat pada proses *end milling*. Sedangkan pada proses *abrasive waterjet machining* dilakukan variasi terhadap kecepatan potong, tekanan *waterjet*, ukuran partikel abrasif, dan laju aliran partikel abrasif. Dari hasil penelitian didapatkan bahwa pada proses *end milling*, optimasi menggunakan *artificial bee colony* menurunkan nilai kekasaran permukaan minimum sebesar 10% dan 8% dibandingkan dengan eksperimental dan regresi (Yusup et al., 2014). Sedangkan pada tabel 2.1 menunjukkan algoritma ABC menurunkan nilai kekasaran permukaan minimum untuk proses *abrasive waterjet machining* sebesar 28%, 42%, 2%, dan 0.9% dibandingkan dengan eksperimental, regresi, *genetic algorithm*, dan *simulated annealing* secara berturut-turut.

**Tabel 2. 2** Parameter optimal untuk kekasaran permukaan (Yusup et al., 2014)

Technique	Process parameters					Minimum $R_a$
	(V)	(P)	(h)	(d)	(m)	
Experimental (Caydas and Hascalik 2008)	50	125	1	60	0.5	2.124
Regression (Caydas and Hascalik 2008)	50	125	1	60	2	2.629
ANN (Caydas and Hascalik 2008)	50	125	1	60	2	2.744
GA (Zain et al. 2011)	50.024	125.018	1.636	94.73	0.525	1.554
SA (Zain et al. 2011)	50.003	125.029	1.486	107.737	0.500	1.533
ABC	50	125	1.550	102.521	0.500	1.522

## 2.2 Proses Pemesinan Frais (*Milling*)

Proses *milling* atau frais merupakan proses pemotongan pada sebagian benda kerja menggunakan alat potong yang berotasi. Pemotongan dilakukan menggunakan gigi potong yang mengitari pahat agar menghasilkan proses pemesinan yang lebih cepat. Permukaan yang disayat memiliki geometri datar, menyudut, dan melengkung (Rochim, 1993). Pada proses pemesinan frais terdapat lima elemen dasar yaitu:

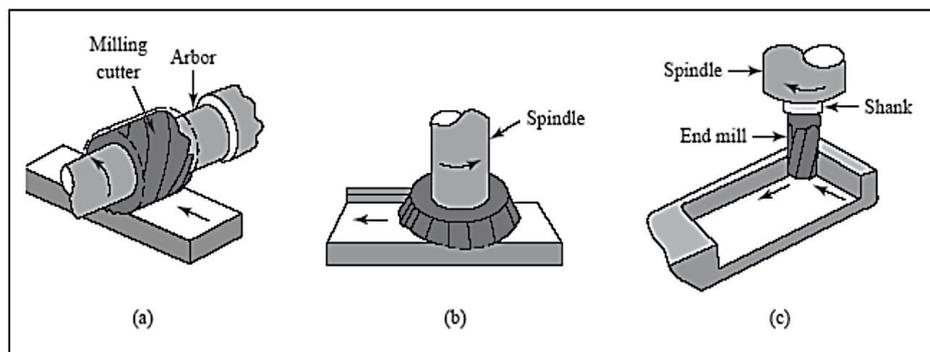
1. Kecepatan potong (*cutting speed*) ;  $v$  (m/min)
2. Kecepatan makan (*feed rate*) ;  $v_f$  (mm/min)
3. Kedalaman potong (*depth of cut*) ;  $\alpha$  (mm)

4. Waktu pemotongan (*cutting time*) ;  $t_c$  (min), dan
5. Kecepatan penghasilan geram (*material removal rate*) ;  $Z$  ( $cm^3$ /menit)

Mesin frais yang digunakan untuk melakukan proses pemesinan dapat dilakukan secara konvensional maupun dengan teknologi CNC. Pada mesin konvensional umumnya menggunakan dua jenis spindel yaitu horizontal dan vertikal. Sedangkan pada mesin frais dengan teknologi CNC menggunakan jenis spindel vertikal.

### 2.2.1 Klasifikasi Proses Frais

Pada gambar 2.2 terdapat jenis-jenis klasifikasi mesin frais. Pada gambar (a) merupakan proses frais periperal, gambar (b) merupakan proses frais muka, dan gambar (c) merupakan proses frais jari (Rochim, 1993).



**Gambar 2. 2** Klasifikasi tiga jenis frais (Rochim, 1993)

Proses pemesinan frais dapat menggunakan berbagai macam jenis konfigurasi operasi pemesinan dengan menggunakan pahat yang menghasilkan geram dalam waktu tertentu. Proses frais diklasifikasikan dalam tiga jenis yaitu:

1. Frais Periperal (*Peripheral Milling*)

Proses frais ini sering disebut dengan *plain milling* atau *slab milling*, dimana permukaan benda kerja yang akan difrais dilakukan oleh gigi potong yang terletak pada permukaan luar badan alat potongnya. Alat potong yang digunakan dapat berupa HSS atau karbida dengan sumbu putaran pahat sejajar dengan permukaan benda kerja. Bentuk gigi yang digunakan pada frais periperal dapat berbentuk lurus maupun helikal dimana bentuk helikal lebih diutamakan karena langsung terikat pada permukaan benda kerja ketika pahat berputar sehingga menghasilkan gaya potong dan torsi yang lebih kecil dan proses pemesinan menjadi lebih halus.

2. Frais Muka (*Face Milling*)

Pada proses frais muka, pahat yang digunakan terpasang pada spindel yang memiliki sumbu putar tegak lurus terhadap permukaan benda kerja. Permukaan hasil proses frais ini didapat dari hasil pemotongan dari ujung dan selubung pahat. Pada proses frais ini dapat menghasilkan *feed marks* dikarenakan gerak relatif antara pahat dan permukaan benda kerja.

3. Frais Jari (*End Milling*)

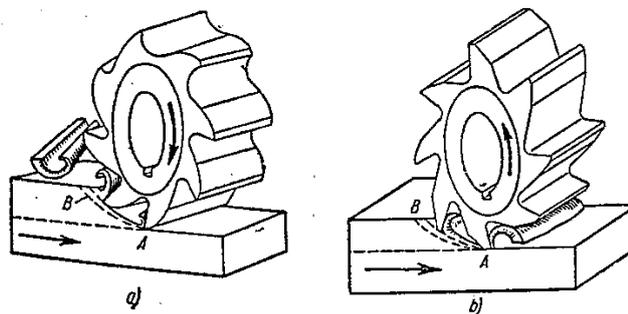
Proses frais ini merupakan proses frais yang paling sering digunakan karena kemampuannya dalam menghasilkan berbagai jenis potongan dan permukaan. Proses frais jari menggunakan ujung pahat yang berputar pada sumbu yang tegak lurus dengan permukaan benda kerja. Pahat bergerak secara menyudut pada benda kerja untuk

menghasilkan bentuk permukaan yang menyudut. Gigi potong yang digunakan umumnya menggunakan material HSS maupun karbida seperti proses frais yang lain dan terletak pada sisi selubung pahat dan ujung badan pahat.

Pada penelitian ini digunakan proses frais jari (*end milling*) untuk mengetahui pengaruh kecepatan potong, kecepatan makan, kedalaman potong, dan lebar pemotongan terhadap gaya potong yang terjadi pada permukaan benda kerja. Hal tersebut digunakan untuk menganalisa pengaruh dari parameter tersebut untuk menentukan resultan gaya potong yang paling optimal menggunakan optimasi artifisial.

### 2.2.2 Metode Proses Frais

Penentuan metode proses frais didasarkan pada arah relatif gerak makan terhadap putaran pahat. Metode proses frais yang umumnya digunakan yaitu frais naik dan frais turun.



**Gambar 2. 3** Frais naik dan frais turun (Rochim, 1993)

Pada gambar 2.3 dijelaskan klasifikasi umum pada proses frais. Gambar (a) merupakan proses frais naik dimana arah pemotongan dari titik A ke titik B atau searah jarum jam sedangkan pada gambar (b) merupakan proses frais turun dimana arah pemotongan dari titik B ke titik A atau berlawanan arah jarum jam.

Pada metode frais naik atau frais konvensional, gerakan putaran dari pahat berlawanan dengan gerak makan meja mesin frais. Jika gerakan pahat berputar searah jarum jam, maka benda kerja yang dipotong akan bergerak dalam arah sebaliknya atau kearah kiri. Proses frais naik membentuk geram seperti tulisan koma dengan ketebalan minimum kemudian menebal. Proses frais naik ini sesuai untuk mesin frais konvensional dikarenakan *backlash* ulir transportirnya relatif lebih besar dan tidak ada *backlash compensation* (Rochim, 1993).

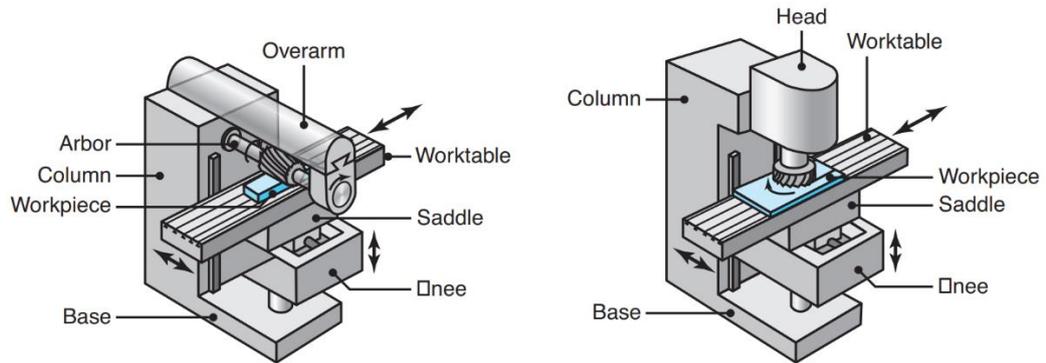
Sedangkan pada metode frais turun atau *climb milling*, gerakan putaran dari pahat searah dengan gerak makan meja mesin frais. Jika gerakan pahat berputar berlawanan arah jarum jam, maka benda kerja yang disayat akan bergerak searah dengan arah putaran atau kearah kiri. Proses frais turun menghasilkan bentuk geram yang berbeda dengan proses frais naik yaitu berbentuk koma yang memiliki ketebalan maksimal kemudian menipis. Proses frais turun umumnya digunakan untuk mesin frais CNC, dikarenakan pada mesin CNC dilengkapi dengan *backlash compensation*. Proses ini tidak cocok jika digunakan pada alat mesin frais konvensional dikarenakan meja frais akan tertekan dan tertarik oleh pahat.

### 2.2.3 Jenis Mesin Frais

Karena kemampuannya dalam melakukan berbagai macam proses pemotongan, proses frais memiliki bermacam jenis yang digunakan sesuai kebutuhan. Terdapat tiga jenis proses mesin frais yang sering digunakan dalam proses pemesinan, yaitu:

1. *Column and knee type machines*
2. *Bed type milling machines*

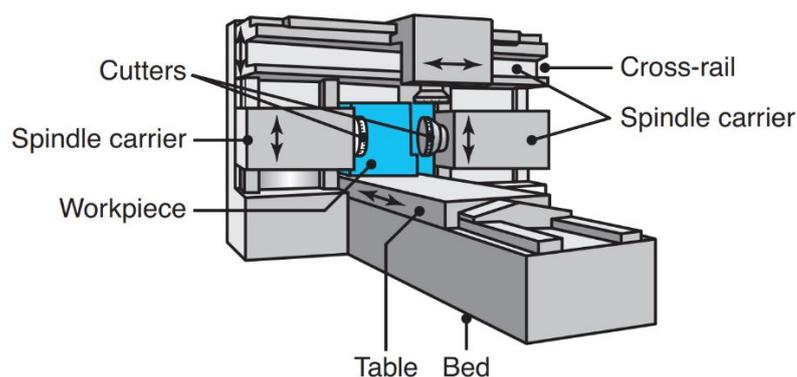
### 3. Computer numerical control machines



**Gambar 2. 4** Column and knee type milling machines (Kalpakjian et al., 2014)

Pada gambar 2.4 mesin jenis *column and knee type machines* pada umumnya memiliki bentuk mesin frais vertikal dan horizontal yang umumnya digunakan dalam semua bentuk operasi frais. Pada mesin frais horizontal umumnya dilakukan proses frais periperal sedangkan pada mesin frais vertikal dilakukan proses frais muka, frais jari, *boring*, dan *drilling*. Komponen dasar pada jenis ini yaitu papan kerja sebagai tempat benda kerja dan bergerak secara sejajar dengan sadel, *knee* untuk pergerakan pada arah vertikal sehingga kedalaman potong dapat diatur, *overarm* pada pergerakan arah horizontal dan *head* sebagai tempat spindel dan gigi potong. Pada jenis *column and knee universal*, papan dapat digerakkan secara horizontal maupun vertikal sehingga dapat menghasilkan bentuk yang kompleks.

Selanjutnya pada tipe *bed type milling machines* seperti pada gambar 2.5, papan kerja dipasang pada *bed* sehingga menggantikan lutut dan hanya bisa bergerak secara longitudinal. Jenis frais ini tidak bisa digunakan untuk semua jenis pemotongan akan tetapi memiliki kekakuan yang tinggi dan umumnya digunakan untuk produksi dalam skala yang besar. Letak spindel dapat berupa horizontal, vertikal, maupun *duplex* yang bisa melakukan berbagai macam pemotongan dalam satu waktu.



**Gambar 2. 5** Bed type milling machines (Kalpakjian et al., 2014)

Pada produksi pemesinan saat ini membutuhkan mesin frais yang bisa menghasilkan berbagai produk dalam waktu yang singkat untuk menghemat biaya produksi sehingga pada saat ini mesin konvensional umumnya sudah digantikan dengan mesin berteknologi *computer numerical control (CNC)*. Mesin dengan CNC dapat melakukan berbagai proses pemesinan mulai dari *milling*, *drilling*, *boring*, maupun *tapping* dengan akurasi yang tinggi dan dalam

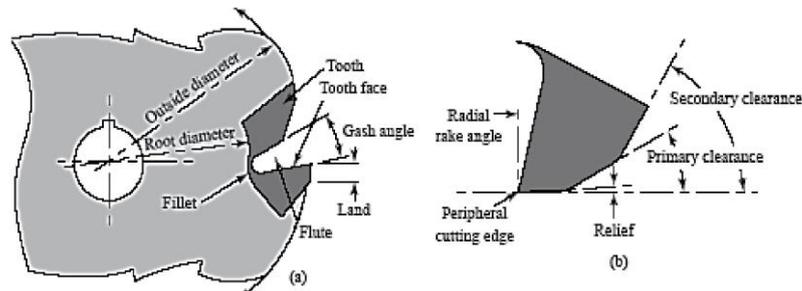
waktu yang singkat. Sehingga mesin frais menjadi sangat fleksibel serta efisien terhadap waktu dan biaya. Pada penelitian ini spesifikasi mesin yang digunakan yaitu berjenis CNC vertikal DMC 635 V Eco dengan spesifikasi seperti pada tabel 2.3 berikut.

**Tabel 2. 3** Spesifikasi CNC

Jenis	DMC 635 V Eco
Tipe	Vertikal
Jarak Sumbu X	635 mm
Jarak Sumbu Y	510 mm
Jarak Sumbu Z	460 mm
Kecepatan Putar	8000 RPM
Maksimum Kecepatan Makan	25 m/min

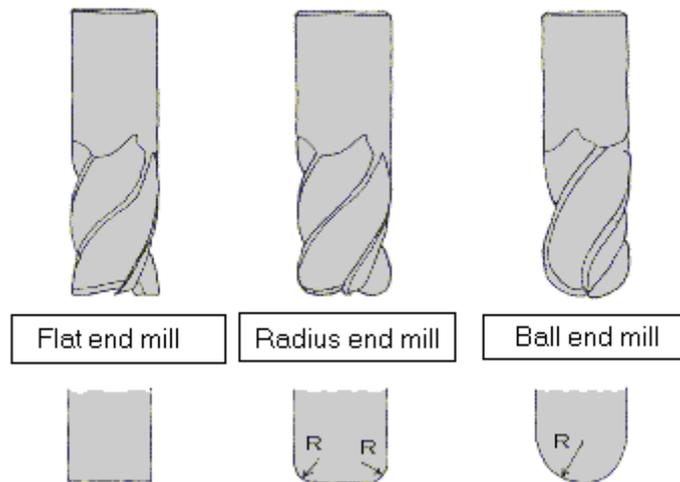
#### 2.2.4 Geometri Pahat Frais

Bentuk umum dari pahat frais identik dengan pahat bubut karena gaya potong terjadi diujung pahat. Pahat frais memiliki bentuk yang lebih rumit dibandingkan dengan pahat bubut dikarenakan memiliki banyak gigi potong, sehingga tergolong proses pemotongan dengan mata potong majemuk. Geometri dari pemotongan pada mesin frais dapat dilihat pada gambar 2.6 berikut.



**Gambar 2. 6** Geometri pahat frais (Rochim, 1993)

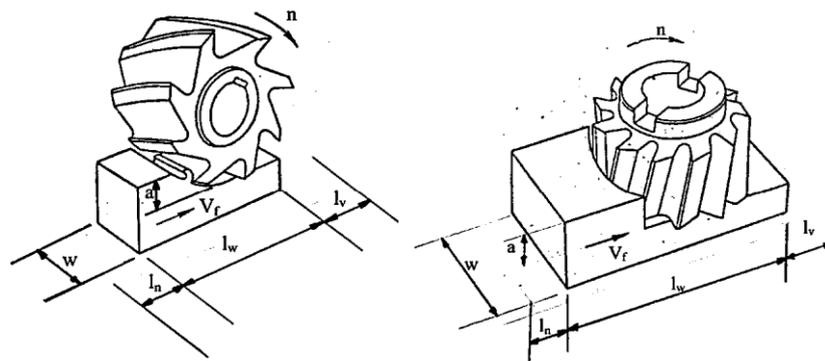
Pada klasifikasi pahat *end mill*, gigi minimum yang digunakan dalam proses penyayatan minimal berjumlah dua buah dan dibagi menjadi beberapa tipe pahat *end mill* yaitu, *flat end mill*, *radius end mill*, dan *ball end mill*. Bentuk geometri dari ketiga jenis pahat ditunjukkan pada gambar 2.7 berikut. Nilai radius dari pahat *end mill* tipe *flat* bernilai 0 mm, sedangkan radius pahat *end mill* bertipe radius bernilai 2 mm, dan nilai radius pahat *end mill* tipe *ball* bernilai 5 mm.



**Gambar 2. 7** Tipe jenis *end mill* (Chong, 2018)

### 2.2.5 Elemen Dasar Proses Frais

Elemen dasar proses frais dapat dinyatakan berdasarkan rumus sebagai berikut:



**Gambar 2. 8** Elemen dasar proses frais vertikal dan horizontal (Rochim, 1993)

Pada gambar 2.8 menunjukkan parameter yang digunakan dalam melakukan proses *end milling*. Gambar (a) merupakan proses frais horizontal dengan arah gerakan pahat searah jarum jam sedangkan benda kerja (laju gerak makan) bergerak dari ke kiri. Gambar (b) menunjukkan proses frais vertikal dimana benda kerja dipotong secara tegak lurus dengan arah putaran pahat searah jarum jam dan arah laju gerak makan bergerak ke kiri.

Untuk benda kerja:

- $w$  = Lebar pemotongan; mm
- $l_w$  = Panjang pemotongan; mm
- $l_t = l_v + l_w + l_n$ ; mm
- $a$  = Kedalaman potong; mm

Untuk pahat frais:

- $d$  = Diameter luar; mm
- $z$  = Jumlah gigi

Untuk mesin frais:

- $n$  = Putaran poros utama; rpm
- $V_f$  = Laju gerak makan; mm/menit

Sehingga didapat perumusan pada proses mesin frais sebagai berikut:

1. Kecepatan potong  

$$V = \frac{\pi \cdot d \cdot n}{1000}; \text{ m/menit} \dots\dots\dots(2.1)$$
2. Gerak makan per gigi  

$$f_z = \frac{V_f}{z \cdot n}; \text{ mm/gigi} \dots\dots\dots(2.2)$$
3. Waktu pemotongan  

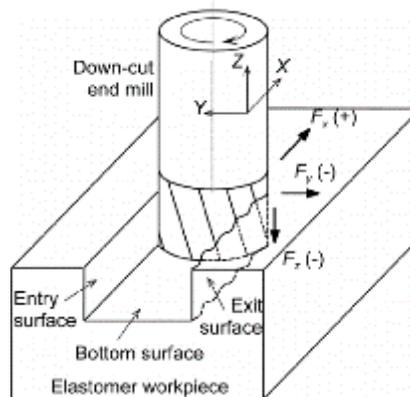
$$t_c = \frac{l_t}{V_f}; \text{ menit} \dots\dots\dots(2.3)$$
4. Kecepatan penghasiian geram  

$$Z = \frac{V_f \cdot a \cdot w}{1000}; \text{ cm}^3/\text{menit} \dots\dots\dots(2.4)$$

**2.3 Gaya Potong pada Proses End Milling**

Gaya potong merupakan gaya yang terjadi akibat reaksi antara pahat dan permukaan benda kerja yang menimbulkan tegangan geser. Tegangan geser maksimum terjadi akibat tegangan pada benda kerja yang mempunyai orientasi yang kompleks dan berada dalam salah satu arah. Apabila tegangan geser melebihi kekuatan dari bahan benda kerja, akan terjadi deformasi plastis pada daerah tegangan terjadi. Deformasi plastis dapat memotong maupun memutuskan material benda kerja pada ujung pahat sehingga membentuk geram. Pada proses *end milling* terdapat pengaruh kecepatan potong dan laju gerak makan terhadap gaya potong dan temperatur pemotongan yang terjadi pada benda kerja. Kecepatan potong berpengaruh terhadap temperatur pemotongan dimana seiring dengan meningkatnya kecepatan potong akan meningkatkan temperatur pemotongan. Sedangkan laju gerak makan berpengaruh terhadap gaya potong dimana bertambahnya laju gerak makan akan diikuti dengan peningkatan gaya potong yang terjadi.

Pada gambar 2.9, gaya-gaya yang terjadi pada proses end milling dapat didefinisikan dalam tiga arah yaitu  $F_x$  sebagai arah gaya pada gerak makan benda kerja,  $F_y$  merupakan arah gaya pada bidang normal, sedangkan  $F_z$  merupakan arah gaya aksial yang tegak lurus bidang benda kerja. Ketiga komponen gaya memiliki nilai yang berbeda tergantung arah gerak makan benda kerja. Gambar 2.9 menunjukkan nilai  $F_x$  bernilai positif karena searah dengan arah gerak makan sedangkan nilai  $F_y$  dan  $F_z$  bernilai negatif karena arah putaran searah dengan jarum jam dan pergerakan putaran menekan kearah sumbu z negatif (J. Luo, 2005).



**Gambar 2. 9** Gaya potong pada proses *end milling* (Luo et al., 2005)

Pada proses *end milling* terdapat gaya potong yang terjadi pada setiap mata potongnya. Gaya potong pada proses *end milling* yang terjadi dirumuskan sebagai berikut:

$$F_k = C_k \cdot a_p^{b_1} \cdot v^{b_2} \cdot v_f^{b_3} \cdot a_e^{b_4}, \text{ dalam satuan N} \dots\dots\dots(2.5)$$

Dimana:

$F_k$  = Gaya potong (N)

$a_p$  = Kedalaman potong (mm)

$v$  = Kecepatan potong (m/min)

$v_f$  = Gerak makan per gigi (mm/gigi)

$a_e$  = Lebar pemotongan (mm)

Sedangkan parameter  $C$ ,  $b_1$ ,  $b_2$ ,  $b_3$ ,  $b_4$  merupakan konstanta model persamaan matematika yang didapatkan dari uji eksperimen (Zenghui et al. 2014).

Beban yang bekerja pada pahat potong dapat menyebabkan terjadinya defleksi pada pahat. Defleksi pada pahat mengakibatkan pergeseran atau perpindahan bentuk dari geometri pahat akibat beban kerja. Contohnya pada proses mesin frais vertikal, gaya potong pada benda kerja yang melebihi gaya dorong roda gigi akan membuat meja terdorong oleh pahat dan benda kerja lebih tertekan ke meja. Jika mesin frais tidak dilengkapi dengan sistem kompensasi keterlambatan gerak balik (*backlash compensator*) akan menyebabkan peningkatan getaran bahkan kerusakan pada benda kerja.

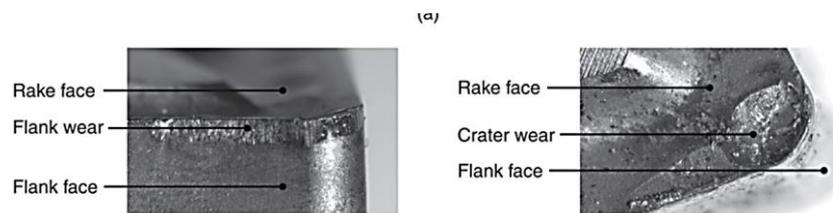
Jika gaya potong yang ditahan setiap gigi untuk melakukan pemotongan semakin besar, akan menyebabkan meningkatnya defleksi yang terjadi pada ujung mata potong pahat. Kekakuan pada pahat pada proses frais dipengaruhi oleh panjang efektif mata potong pahat yang digunakan, semakin besar nilai panjang efektif yang digunakan maka defleksi pada ujung mata potong yang terjadi akan meningkat (Rochim, 1993).

## 2.4 Umur Pahat

Salah satu faktor yang menentukan umur pahat adalah keausan yang terbentuk akibat proses pemesinan. Keausan yang terjadi dapat membesar seiring dengan bertambahnya waktu pemotongan sampai dalam kondisi pahat sudah tidak bisa digunakan karena terdapat tanda-tanda tertentu yang menunjukkan pahat sudah rusak dan tidak bisa digunakan kembali.

### 2.4.1 Bidang Aktif Pahat yang Mengalami Kerusakan

Pada saat proses pemesinan frais, pahat dapat mengalami kegagalan pada saat proses pembentukan geram diakibatkan keausan yang terjadi secara bertahap pada pahat. Contoh jenis keausan yang terjadi pada pahat dapat yaitu keausan kawah (*crater wear*) dan keausan tepi (*flank wear*).



**Gambar 2. 10** *Crater wear* dan *flank wear* (Kalpakjian et al., 2014)

*Flank wear* seperti pada gambar 2.10 umumnya muncul pada bagian sisi samping (tepi) pada permukaan pahat yang bersentuhan dengan benda kerja. Adanya *flank wear* dapat menyebabkan *adhesive* dan *abrasive wear* serta peningkatan temperatur yang dapat mempengaruhi material pahat. Sedangkan *crater wear* muncul pada bagian atas permukaan pahat yang dapat mempengaruhi temperatur pada saat pembentukan geram dan reaksi kimia antara pahat dan benda kerja (*chemical affinity*).

Penyebab kegagalan yang lain yaitu deformasi plastis yang mengubah bentuk geometri pahat maupun retak yang dapat menyebabkan patahan pada pahat akibat gaya potong pada benda kerja. Kegagalan yang terjadi dapat disebabkan karena tingginya tekanan dan temperatur

pada bidang aktif pahat dimana semakin besar nilai tekanan dan temperatur akan membuat kekuatan dan kekerasan material pahat semakin berkurang.

#### 2.4.2 Kriteria Umur Pahat

Berdasarkan jenis keausan yang terjadi pada pahat pada subbab sebelumnya, ditentukan nilai batas keausan pada pahat seperti tabel 2.4 berikut:

**Tabel 2. 4** Batas keausan kritis (Rochim, 1993)

Pahat	Benda Kerja	<i>Flank wear</i> (mm)	<i>Crater wear</i> (mm)
HSS	Baja dan Besi Tuang	0,1 – 0,8	0,3
Karbida	Baja	0,2 – 0,6	0,3
Karbida	Besi Tuang dan <i>Non-Ferrous</i>	0,4 – 0,6	0,3
Keramik	Baja dan Besi Tuang	0,3	0,3

Semakin besar nilai kerusakan yang terjadi pada pahat menyebabkan kondisi pahat tergolong kritis. Apabila digunakan (pahat kritis) akan menyebabkan penambahan keausan yang semakin cepat dan membuat pahat mengalami kegagalan akibat gaya potong yang terlalu tinggi (Rochim, 1993). Hal tersebut jika tidak diatasi dapat membuat pahat tidak bisa digunakan kembali dan dapat membahayakan keselamatan operator mesin.

#### 2.5 Material Inconel 718

Inconel 718 merupakan *superalloy* yang berbasis material nikel yang memiliki ketahanan terhadap suhu tinggi. Sifat dasar dari inconel 718 yang memiliki kemampuan mempertahankan sifat mekanik dan kimiawi pada suhu tinggi dan memiliki kemampuan tahan terhadap korosi menjadikan material ini sebagai bahan yang ideal untuk digunakan pada pembangkit generator, pesawat luar angkasa, dan perlengkapan industri lepas pantai. Komposisi paling besar dari inconel 718 terdiri dari 53% nikel (Ni), 18,7% besi (Fe), 18,3% kromium (Cr), dan disusun oleh material yang lain seperti pada tabel 2.5 berikut:

**Tabel 2. 5** Komposisi inconel 718 (Kasim et al., 2015)

Al	B	C	Cb.Ta	Co	Cr	Cu	Fe	Mn	Mo
0.49	0.004	0.051	5.05	0.30	18.30	0.04	18.70	0.23	3.05
Ni	P	S	Si	Ti					
53.0	< 0.005	< 0.002	0.08	1.05					

Pada proses pemesinan sendiri, pahat yang umumnya digunakan dalam pemesinan inconel 718 yaitu pahat karbida. Pahat karbida yang dilapisi menggunakan TiAlN dapat membuat pahat memiliki permukaan lapisan yang sangat keras dan memiliki ukuran grit yang kecil. Proses pelapisan yang digunakan menggunakan metode *physical vapor deposition* (PVD) dimana material *coating* dibentuk menggunakan ruang hampa bertekanan tinggi. Pemesinan inconel 718 yang menggunakan pahat karbida yang telah dilakukan *coating* dapat meningkatkan produktifitas dikarenakan material inconel 718 merupakan material yang memiliki *machinability* yang buruk. Proses pemesinan inconel 718 yang tidak tepat akan mengakibatkan pahat lebih cepat mengalami aus dikarenakan material ini memiliki konduktifitas termal yang rendah. Sehingga untuk pemesinan inconel 718 sebaiknya dilakukan pada kecepatan potong yang rendah atau menggunakan pelumas (Akhyar et al., 2016).

Berdasarkan penelitian yang dilakukan oleh beberapa jurnal sebelumnya (Kasim et al., 2015) parameter pemesinan yang direkomendasikan pada proses pemesinan inconel 718 yaitu dengan kecepatan potong antara 10 hingga 170 m/min dan kecepatan makan antara 0.02 hingga 0.2 mm/gigi.

## 2.6 Artificial Neural Network

*Artificial neural network* atau jaringan syaraf tiruan adalah bagian dari kecerdasan artifisial yang dimodelkan sebagai kumpulan neuron pada otak manusia. Kumpulan neuron ini (*neural networks*) akan membentuk program yang dapat bekerja selayaknya otak manusia untuk menyelesaikan permasalahan dan membuat keputusan sesuai data yang ada dengan menghubungkan parameter input dan respon. Setiap neuron akan terhubung satu sama lain dimana terdapat suatu nilai tertentu yang informasinya tersimpan dalam bobot atau *weight*. Sebuah jaringan dikatakan ANN apabila memenuhi tiga syarat yaitu arsitektur jaringan sebagai pola penghubung antar neuron, metode untuk menentukan bobot pada setiap neuron, serta fungsi aktivasi yang digunakan.

### 2.6.1 Fungsi Aktivasi

Fungsi aktivasi dalam *artificial neural network* digunakan untuk menentukan output sebuah neuron. Beberapa jenis fungsi aktivasi memiliki fungsi dan tujuan yang berbeda dalam menghasilkan output. Berikut merupakan jenis fungsi aktivasi:

#### 1. Linear Activation Function

Fungsi aktivasi linier digunakan agar nilai gradien tidak bernilai nol, sehingga dalam sebuah fungsi harus terdapat sebuah konstanta  $a$  yang didefinisikan dengan

$$f(x) = ax \dots \dots \dots (2.6)$$

Dari nilai fungsi tersebut, nilai konstanta tidak bergantung terhadap nilai input  $x$ . sehingga nilai bobot dan bias tidak akan berubah selama proses kalibrasi dikarenakan nilai eror tidak mengalami perubahan sehingga jaringan tidak dapat melakukan proses pelatihan dengan akurat. Karena itu, fungsi ini umumnya hanya digunakan untuk melakukan tugas yang sederhana.

#### 2. Sigmoid

Fungsi *sigmoid* bersifat nonlinier dan akan merubah nilai input dalam output analog yaitu bilangan antara 0 dan 1. Jika nilai input lebih besar dari 1 maka akan dirubah menjadi nilai 1, sebaliknya jika nilai input lebih kecil dari 0 maka nilai akan dirubah menjadi 0. Persamaan dasar pada fungsi aktivasi *sigmoid* yaitu

$$\phi(x) = \frac{1}{1+e^{-x}} \dots \dots \dots (2.7)$$

Fungsi *sigmoid* sering digunakan karena memiliki interpretasi yang baik karena merubah angka negatif menjadi 0 dan angka positif sebesar apapun menjadi 1. Akan tetapi, fungsi ini memiliki keterbatasan dikarenakan meniadakan nilai gradien sehingga jika nilai bobot awal terlalu besar maka sebagian neuron tidak dapat mempelajari apapun karena berubah menjadi jenuh.

#### 3. Rectifier Linear Unit (RELU)

Fungsi aktivasi RELU memiliki perhitungan sederhana yaitu menggunakan proses *forward* dan *backward* dengan kondisi *if*. Jika elemen bernilai negatif maka nilai akan diset menjadi 0 sehingga tidak bisa dilakukan operasi perkalian, pembagian maupun eksponensial. Persamaan pada fungsi RELU yaitu

$$f(x) = \max(0, x) \dots \dots \dots (2.8)$$

Kelebihan dari fungsi aktivasi ini akan mempercepat nilai konvergensi karena meniadakan nilai negatif. Keterbatasan fungsi aktivasi ini yaitu proses pelatihan akan cepat berhenti sehingga bobot akan ter-*update* secara tidak baik dikarenakan neuron yang terdapat pada nilai negatif tidak akan pernah aktif pada *datapoint* manapun.

4. *Hyperbolic Tangent Function*

Fungsi tan hiperbolik atau TANH merupakan fungsi aktivasi yang nilai outputnya antara -1 dan 1. pada umumnya lebih cepat mencapai nilai konvergen dibandingkan dengan fungsi aktivasi *sigmoid* dikarenakan memiliki performa dan prediksi yang hamper sama dengan fungsi aktivasi RELU. Neuron dari fungsi aktivasi TANH didefinisikan dengan

$$\tanh(x) = 2\Sigma(2x) - 1.....(2.9)$$

Dalam penggunaannya terdapat beberapa fungsi aktivasi yang tersedia dalam perangkat lunak MATLAB. Fungsi aktivasi pada MATLAB memiliki jenis fungsi aktivasi yang sama namun memiliki perbedaan nama. Beberapa contoh fungsi aktivasi pada MATLAB seperti tabel 2.6 berikut.

**Tabel 2. 6** Fungsi aktivasi pada MATLAB

Name	Input/Output Relation	Icon	MATLAB Function
Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$		hardlim
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$		hardlims
Linear	$a = n$		purelin
Saturating Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$		satlin
Symmetric Saturating Linear	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = 1 \quad n > 1$		satlins

Log-Sigmoid	$a = \frac{1}{1 + e^{-n}}$		logsig
Hyperbolic Tangent Sigmoid	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$		tansig
Positive Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$		poslin
Competitive	$a = 1 \quad \text{neuron with max } n$ $a = 0 \quad \text{all other neurons}$		compet

### 2.6.2 Bias

Bias merupakan neuron spesial yang ditambahkan pada setiap *layer* di *neural network*. Fungsi utama dari bias sebagai pemindah fungsi aktivasi ke kanan ataupun ke kiri. Neuron yang dilengkapi dengan bias akan memiliki persamaan

$$Y = \Sigma(\text{weight} \times \text{input}) + \text{bias} \dots \dots \dots (2.10)$$

Nilai bias secara sederhana bernilai 1, tanpa adanya nilai bias maka setiap neuron akan merupakan perkalian antara bobot dengan nilai inputnya sehingga nilai output yang didapatkan tidak sesuai dengan nilai yang dimasukkan. Meskipun sebuah *neural network* dapat bekerja tanpa membutuhkan neuron bias, tetapi nilai bias akan selalu ditambahkan untuk memperkirakan nilai dari keseluruhan model.

### 2.6.3 Pelatihan Jaringan Syaraf Tiruan

Pada jaringan syaraf tiruan terdapat proses kalibrasi yang disebut dengan *training*. Berdasarkan cara mengatur nilai bobotnya, terdapat dua metode *training* dalam jaringan syaraf tiruan yaitu:

#### 1. Supervised Learning

Pelatihan pada jenis ini menggunakan pasangan data input dan output untuk melatih jaringan sehingga mendapat nilai output yang bernilai sama dengan inputnya. Perbedaan nilai output dengan input disebut dengan error. nilai error akan dikalibrasi sehingga didapat nilai error yang mendekati nol. Contoh jaringan yang menggunakan *supervised learning* yaitu *perceptron* dan *multilayer perceptron* (BPNN).

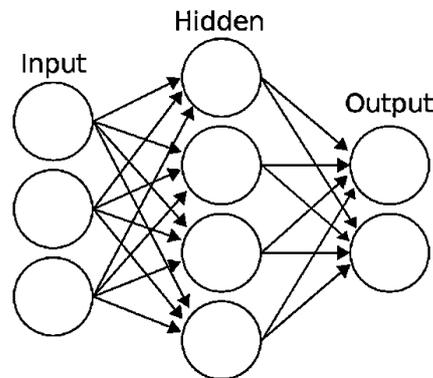
#### 2. Unsupervised Learning

Pada jenis pelatihan *unsupervised*, perubahan pada nilai bobot didasarkan parameter tertentu tanpa meninjau target data. Dengan kata lain pada jenis jaringan ini tidak memerlukan adanya *training* dalam perubahan nilai bobot. Contoh *unsupervised learning* yaitu *clustering*.

### 2.6.4 Backpropagation Neural Network

BPNN atau *multilayer perceptron* merupakan salah satu jaringan syaraf tiruan dengan cara meninjau nilai error antara input dan output kemudian mengevaluasi setiap nilai bobot atau weight berdasarkan nilai error yang didapat. Nilai error pada algoritma BPNN didapat menggunakan perambatan maju (*forward propagation*) terlebih dahulu yang kemudian dievaluasi sehingga tergolong *supervised learning* dengan arah evaluasi mundur (*backward*). Setiap nilai input yang dihubungkan ke neuron pada *hidden layer* dan dihubungkan ke output terdapat penghubung yang disebut dengan *weight*. Nilai *weight* akan mengalami perubahan

sesuai dengan berjalannya proses BPNN. Pada gambar 2.11 merupakan contoh BPNN yang terdiri dari tiga lapisan utama yaitu lapisan input, *hidden layer*, dan output dengan konfigurasi 3 input, 1 *hidden layer* dengan 4 neuron, dan 2 output.



**Gambar 2. 11** Contoh jaringan BPNN

Kelebihan dari algoritma BPNN yaitu dapat digunakan untuk menyelesaikan masalah prediksi dan peramalan dengan tingkat akurasi yang baik. Adanya perbaikan nilai eror pada BPNN membuat prediksi yang dihasilkan akan memiliki output yang akurat. Sedangkan keterbatasan dari algoritma ini membutuhkan waktu yang lama dalam proses pelatihan data dikarenakan parameter mengalami perubahan eror pada setiap iterasi. Dalam melakukan proses pelatihan data pada BPNN diperlukan penggunaan data pelatihan yang banyak, dikarenakan hasil prediksi dipengaruhi oleh banyaknya jumlah data yang digunakan. Jika data yang digunakan sedikit, maka hasil prediksi yang dihasilkan BPNN akan semakin tidak akurat.

### 2.7 *Artificial Bee Colony Algorithm*

Pada kehidupan nyata, terdapat individu-individu khusus yang melakukan tugas untuk koloni lebah. Tugas dari lebah khusus ini untuk memaksimalkan jumlah nektar yang diperoleh sumber makanan dan menyimpan didalam sarang dengan prinsip pembagian kerja dan pengorganisasian yang efisien. Algoritma *artificial bee colony* merupakan algoritma optimasi yang mencerminkan perilaku kelompok lebah dalam mencari sumber makanan. Pemodelan dari algoritma ABC dibagi menjadi tiga jenis lebah: lebah pekerja (*employed bees*), lebah penonton (*onlooker bees*), dan lebah pengintai (*scout bee*).

Dalam sebuah koloni lebah, hampir lima puluh persen terdiri dari lebah pekerja dan separuh sisanya menjadi lebah penonton. Tugas utama dari lebah pekerja yaitu untuk mengeksploitasi sumber makanan berupa nektar dari sumber makanan yang sudah ditemukan sebelumnya dan memberikan informasi tentang kondisi sumber makanan terhadap lebah penonton. Lebah penonton akan memberikan rekomendasi tentang sumber makanan mana yang akan diambil berdasarkan informasi dari lebah pekerja. Sedangkan lebah pengintai bertugas untuk mencari calon sumber makanan baru pada lingkungan baru berdasarkan petunjuk dari luar.

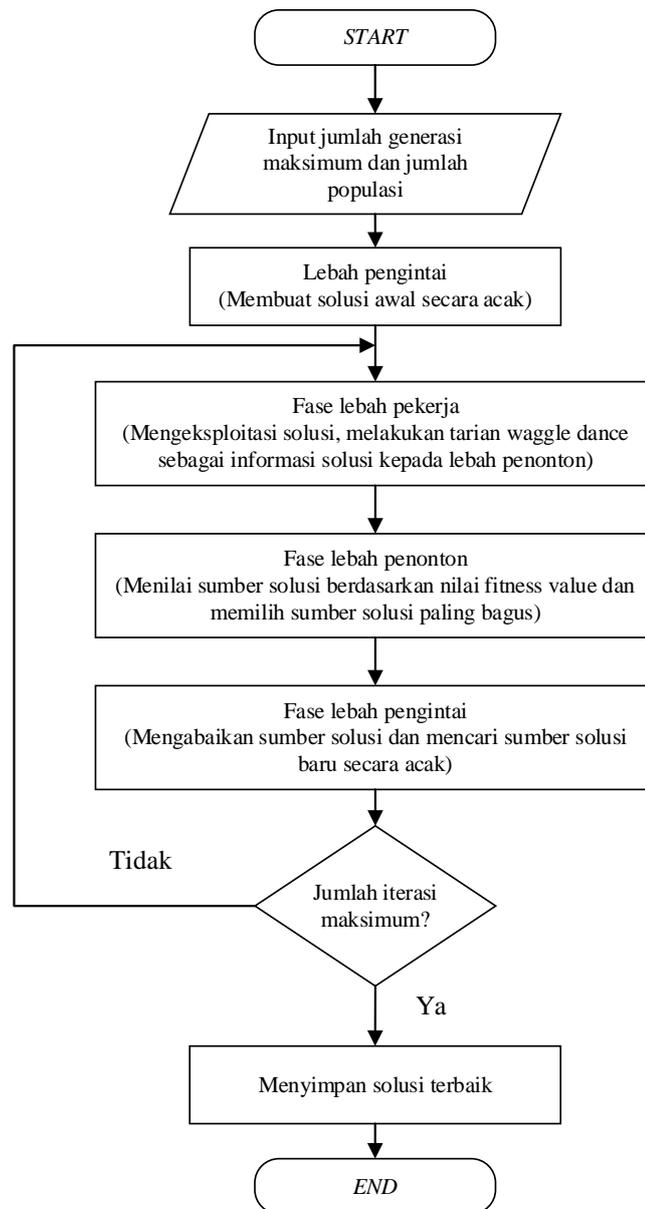
Secara umum, perilaku lebah dalam mencari makanan diterapkan kedalam algoritma lebah melalui skema pada gambar 2.12 berikut:

1. Tahap inialisasi awal, yaitu para lebah akan mencari sumber makanan di lingkungan baru secara acak.
2. Ketika sumber makanan sudah ditemukan, para lebah pekerja akan mengeksploitasi sumber makanan tersebut dan membawa nektar ke sarang. Informasi tentang adanya

sumber makanan didasarkan pada adanya tarian (*waggle dance*) yang dilakukan para lebah pekerja di area tarian. Jika sumber makanan sudah habis dieksploitasi, lebah pekerja akan berubah menjadi lebah pengintai dan memulai tugasnya untuk mencari sumber makanan baru secara acak.

3. Lebah penonton bertugas melihat tarian dari lebah pekerja dan memilih sumber makanan berdasarkan frekuensi tarian yang memiliki kualitas paling bagus.

Posisi sumber makanan merupakan solusi paling optimal dimana jumlah nektar dari suatu sumber makanan merupakan profitabilitas dari solusi yang direncanakan. Sehingga setiap sumber makanan akan dieksploitasi oleh satu lebah pekerja saja sampai sumber makanan tersebut habis dan lebah pekerja akan menjadi lebah pengintai. Dengan kata lain, jumlah sumber makanan yang dieksploitasi akan sama dengan jumlah lebah pekerja.



**Gambar 2. 12** Prosedur algoritma *artificial bee colony* (ABC)

Berdasarkan analogi kecerdasan seperti pada gambar 2.12 mengenai prosedur algoritma *artificial bee colony* terhadap perilaku lebah dalam mencari sumber makanan, maka unit-unit algoritma ABC dapat dijelaskan sebagai berikut:

1. Produksi Sumber Makanan Awal

Penerapan algoritma dimulai dengan menentukan sumber makanan pada solusi ruang pencarian secara acak. Sumber makanan awal akan diproduksi dalam batasan parameter yang telah ditentukan.

$$X_{ij} = X_{j \min} + rand(0,1)(X_{j \max} - X_{j \min}) \dots\dots\dots(2.11)$$

nilai  $i = 1 \dots SN = 1 \dots D$ . dimana SN merupakan jumlah sumber makanan dan D merupakan karakter yang akan dilakukan optimasi. Setelah proses pencarian sumber makanan selesai, populasi sumber makanan sebagai solusi akan mengalami siklus pengulangan pada lebah pekerja, penonton, dan pengintai. Proses pengulangan siklus akan berhenti ketika tercapai jumlah siklus maksimum.

2. Pengiriman Lebah Pekerja ke Lokasi Sumber Makanan

Berdasarkan penjelasan sebelumnya bahwa setiap sumber makanan hanya dipasang dengan satu lebah pekerja yang berarti jumlah sumber makanan sama dengan jumlah lebah pekerja. Posisi dari sumber makanan akan dimodifikasi oleh lebah pekerja berdasarkan informasi visual untuk menentukan sumber makanan tetangga, dan menentukan tingkat kualitasnya. Pada algoritma ABC, sumber makanan tetangga didefinisikan sebagai,

$$V_{ij} = X_{ij} + \theta_{ij}(X_{ij} - X_{kj}) \dots\dots\dots(2.12)$$

nilai  $V_{ij}$  menunjukkan posisi terbaru sumber makanan pertama pada komponen j, sedangkan  $X_{ij}$  merupakan posisi awal dari makanan pertama pada komponen j. Nilai  $i = 1,2, \dots, SN; j = 1,2, \dots, n; k = 1,2, \dots, SN$  dimana SN menunjukkan jumlah lebah pekerja.

Apabila nilai parameter yang dihasilkan melebihi batas dari ruang solusi, parameter akan diatur ulang ke nilai yang diterima diantara batas boundary-nya. Jika  $x_i > x_i^{max}$ , maka nilai  $x_i = x_i^{max}$ . Dan jika nilai  $x_i < x_i^{min}$ , maka  $x_i = x_i^{min}$ . Setelah mendapat posisi sumber makanan yang paling optimum, nilai *fitness* untuk solusi posisi  $V_i$  didefinisikan dengan,

$$fitness_i = \begin{cases} 1/(1 + f_i) , f_i \geq 0 \\ 1 + abs(f_i) , f_i < 0 \end{cases} \dots\dots\dots(2.13)$$

nilai  $f_i$  adalah nilai *fitness* dari solusi  $V_i$ . Setiap sumber makanan akan memiliki nilai  $x_i$  dan  $V_i$  yang berbeda. Jika sumber makanan  $V_i$  lebih tinggi dibandingkan dengan nilai  $x_i$ , lebah pekerja akan mengingat dan mengganti posisi yang lama. Jika nilai  $V_i$  rendah, maka lebah pekerja akan mengingat posisi yang lama.

3. Perhitungan Nilai Profitabilitas

Ketika para lebah mencari sumber makanan baru dan berbagi informasi beserta jumlah nektar kepada lebah penonton di area tarian, lebah penonton akan mengamati dan melakukan seleksi lokasi sumber makanan berdasarkan solusi nilai *fitness* populasi tersebut. Nilai *fitness* dari pencarian sebelumnya akan diseleksi nilai probabilitasnya menggunakan persamaan

$$P_i = \frac{fitness_i}{\sum_{i=1}^{SN} fitness_i} \dots\dots\dots(2.14)$$

Dalam skema seleksi nilai probabilitas ini, nilai jumlah nektar pada sumber makanan berbanding lurus dengan jumlah lebah yang akan mengunjungi sumber makanan tersebut.

#### 4. Pemilihan Sumber Makanan oleh Lebah Penonton

Setiap sumber makanan memiliki nilai bilangan acak dari  $[0,1]$ . Apabila nilai probabilitas lebih besar dari batas nilai acak, maka lebah penonton akan memodifikasi lokasi sumber makanan menggunakan persamaan pencarian sumber makanan tetangga. Setelah sumber makanan dievaluasi, lebah penonton akan mengingat lokasi sumber makanan yang baru ataupun menyimpan lokasi yang lama bergantung terhadap nilai fitness suatu sumber makanan. Proses ini mengalami pengulangan sampai semua lebah penonton dapat terdistribusi secara merata ke setiap sumber makanan.

#### 5. Kriteria Limitasi dan Pembentukan Lebah Pengintai

Jika semua tugas dari lebah pekerja dan lebah penonton telah diselesaikan, algoritma akan menentukan apakah terdapat sumber yang sudah tereksploitasi habis. Sumber makanan akan diputuskan untuk ditinggalkan atau tidak berdasarkan nilai *counter* yang selalu diperbarui selama proses pencarian. Jika nilai *counter* melebihi nilai parameter limit algoritma ABC, maka dapat diasumsikan sumber sudah habis dan harus ditinggalkan.

Dalam algoritma ABC, terdapat *pseudo-code* yang digunakan sebagai berikut:

1. Inisialisasi penentuan parameter limit algoritma ABC berdasarkan besar koloni, jumlah sumber makanan, dan iterasi maksimum.
2. Penentuan posisi
3. Penerapan kriteria seleksi
4. Pembagian informasi dari lebah pekerja ke lebah penonton
5. Penentuan lokasi sumber makanan berdasarkan perhitungan probabilitas
6. Menerapkan kriteria seleksi
7. Jika terdapat solusi berlimpah, maka dilanjutkan dengan inisiasi posisi sumber makanan baru. Jika tidak, berarti mengikuti langkah selanjutnya.
8. Mempertahankan solusi terbaik.
9. Melakukan pengulangan (siklus 4-8) sampai dihentikan oleh lebah pengintai.

## 2.8 *Differential Evolution*

Salah satu metode optimasi yang menggunakan algoritma genetika dalam menyelesaikan permasalahan yaitu *Differential Evolution* (DE). Algoritma DE mensimulasikan individu dalam suatu populasi yang dipilih secara acak dan akan mengalami diferensiasi dan *crossover* yang terjadi secara berurutan. Hasil proses tersebut akan menghasilkan individu yang akan menggantikan induk apabila memiliki kecocokan keturunan yang lebih baik dari induknya dan dianggap sebagai solusi terbaik dari proses optimasi yang dilakukan. Kelebihan metode DE dibandingkan metode algoritma optimasi yang lainnya yaitu adanya evolusi atau perubahan terhadap individu dalam mencari solusi optimalnya sehingga dapat menemukan global optimum (nilai terbaik dari lokal optimum) dengan probabilitas yang lebih baik.

Tahapan dalam melakukan optimasi menggunakan DE dibagi menjadi proses inisialisasi, mutasi, *crossover*, dan seleksi. Keempat tahapan dalam melakukan optimasi DE tersebut akan dijelaskan sebagai berikut:

1. Inisialisasi

Tahapan pertama dalam melakukan metode DE yaitu menentukan populasi awal yang akan digunakan pada proses-proses selanjutnya untuk mendapatkan solusi yang optimal. Pada langkah ini nilai batas atas (ub) dan bawah (lb) ditentukan dengan persamaan sebagai berikut:

$$x_{j,i,0} = lb_j + rand.(ub_j - lb_j) \dots\dots\dots(2.15)$$

dimana fungsi *rand* digunakan untuk menyatakan bilangan acak yang tersebar dalam batas bawah 0 dan batas atas 1 dan akan digunakan untuk menentukan nilai baru pada perhitungan berikutnya.

2. Mutasi

Proses mutasi merupakan perubahan individu dengan nilai yang acak. Mutasi pada algoritma DE berfungsi untuk menghasilkan vektor ujicoba untuk setiap individu yang berada dalam suatu populasi dengan cara mengambil 3 vektor dari populasi (vektor r1, r2, dan r0). Perbedaan nilai vektor antara r1 dan r2 akan menghasilkan vektor baru yang dipengaruhi nilai konstanta F. Persamaan mutasi pada algoritma DE dinyatakan sebagai berikut:

$$v_{i,g} = x_{r0,g} + F.(x_{r1,g} - x_{r2,g}) \dots\dots\dots(2.16)$$

dimana nilai r0, r1, dan r2 merupakan indeks acak dengan nilai bilangan integer yang berbeda.

3. Crossover

Tahapan *crossover* merupakan persilangan antara dua vektor yaitu vektor awal  $x_{i,g}$  dan mutan  $v_{i,g}$ . Persilangan tersebut akan membentuk vektor baru dengan notasi  $u_{i,g}$ . Persamaan umum crossover dalam algoritma DE adalah sebagai berikut:

$$u_{i,g} = u_{j,i,g} = \begin{cases} v_{j,i,g} & \text{if } (rand_j(0,1) \leq Cr \text{ atau } j = j_{rand}) \\ x_{j,i,g} & \text{if } (rand_j(0,1) > Cr \text{ atau } j \neq j_{rand}) \end{cases} \dots\dots\dots(2.17)$$

Probabilitas nilai *crossover*, *Cr* dengan nilai range antara [0,1] merupakan nilai yang digunakan untuk mengontrol pembagian parameter yang digandakan dari mutan. Nilai  $rand_j$  akan menentukan apakah sebuah vektor akan dilakukan crossover atau tidak. Jika nilai  $rand_j$  dari suatu vektor kurang dari nilai *Cr* maka nilai vektor mutan yang akan digandakan. Sebaliknya jika nilai  $rand_j$  dari suatu vektor melebihi nilai *Cr* maka vektor yang akan digandakan adalah vektor awal.

4. Seleksi

Proses terakhir dalam algoritma DE yaitu melakukan seleksi terhadap individu yang akan digunakan dalam populasi berikutnya. Syarat agar individu agar bisa menggantikan induknya yaitu memiliki kecocokan dari keturunan yang lebih baik daripada induk. Jika digunakan dalam notasi vektor, vektor ujicoba  $u_{i,g}$  yang mempunyai nilai *fitness* value lebih kecil dibandingkan dengan vektor target atau vektor awal  $x_{i,g}$  akan menggantikan posisi vektor awal tersebut pada generasi selanjutnya. Dan jika terjadi sebaliknya, maka posisi vektor awal akan tetap pada populasinya dalam populasi. Persamaan seleksi pada algoritma DE dapat dinyatakan sebagai berikut:

$$x_{i,g} = \begin{cases} u_{i,g} & \text{jika } f(u_{i,g}) \leq f(x_{i,g}) \\ x_{j,i,g} & \text{jika } f(u_{i,g}) > f(x_{i,g}) \end{cases} \dots\dots\dots(2.18)$$

Dalam algoritma DE, terdapat *pseudo-code* yang digunakan sebagai berikut:

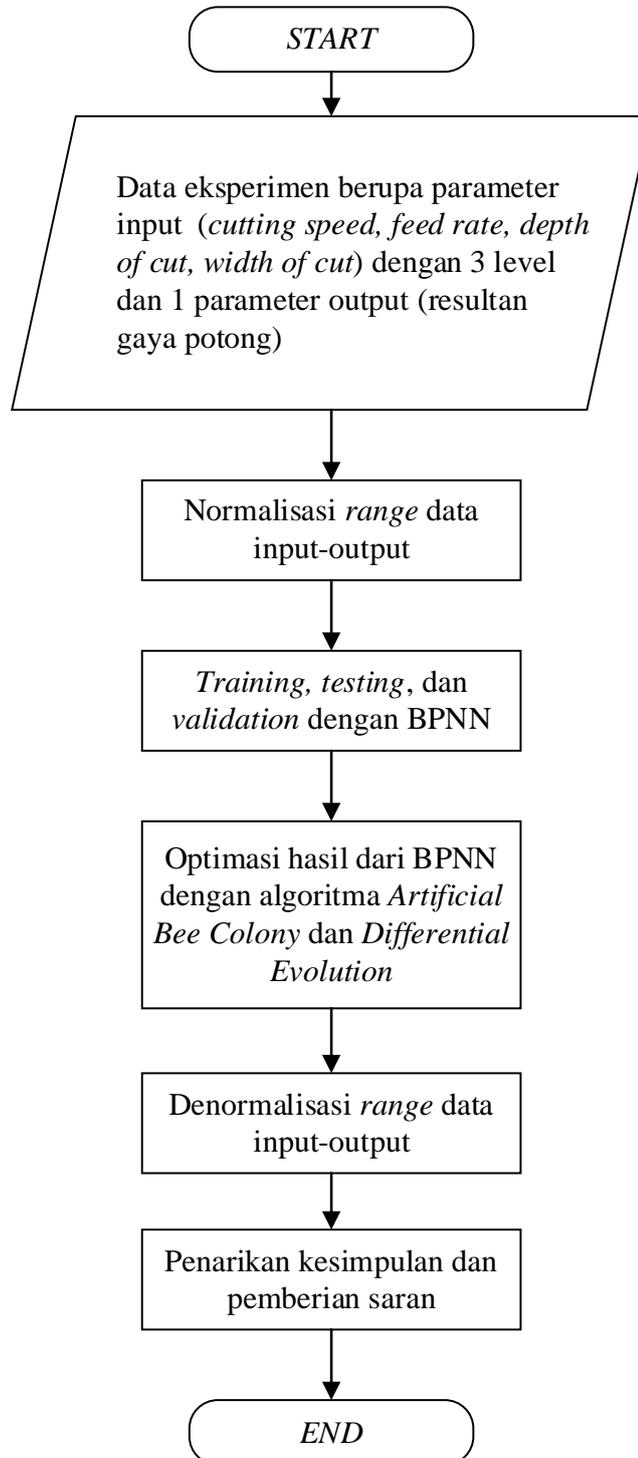
1. Inisialisasi fungsi yang digunakan
2. Penentuan posisi batas atas (*upper bound*) dan batas bawah (*batas atas*)

3. Inisialisasi setiap variabel  $j$  dari individu  $i$
4. Menentukan batas nilai  $rand$  yang digunakan
5. Menentukan parameter algoritma DE seperti jumlah populasi, jumlah iterasi, jenis DE, dan faktor skala untuk mutasi
6. Pada tahap mutasi, memilih vektor target dari populasi secara acak
7. Melakukan perhitungan nilai vektor ujicoba menggunakan konstanta faktor skala
8. Pada tahap *crossover*, menentukan nilai konstanta probabilitas *crossover*
9. Menentukan apakah vektor ujicoba pada *crossover*, berada pada batas atas atau batas bawah
10. Pada tahap seleksi, menentukan apakah vektor keturunan/vektor ujicoba  $u_{i,g}$  atau vektor induk  $x_{i,g}$  yang akan bertahan
11. Melakukan update pada simulasi menggunakan vektor hasil seleksi
12. Mengulangi langkah 6-11 untuk setiap generasi

## BAB III METODOLOGI

### 3.1 Diagram Alir Penelitian

Tahapan yang akan dilakukan dalam penelitian ini ditunjukkan dalam diagram alir seperti pada gambar 3.1 berikut



**Gambar 3. 1** Diagram alir penelitian

### 3.2 Observasi

Observasi yang dilakukan oleh penulis dalam penelitian ini bertujuan untuk menentukan batasan ruang lingkup permasalahan yang akan dijadikan objek penelitian. Dalam penelitian ini dilakukan observasi pada proses pemesinan frais dengan tipe *end milling* dengan optimasi menggunakan perangkat lunak MATLAB R2020b.

### 3.3 Perumusan Permasalahan

Permasalahan pada penelitian ini untuk menentukan parameter optimal pada proses *end milling* untuk mendapatkan hasil proses pemesinan *end milling* yang menghasilkan resultan gaya potong paling minimum dengan menggunakan optimasi algoritma ABC dan DE. Permasalahan yang digunakan pada penelitian ini sesuai dengan kondisi proses manufaktur dimana pembentukan geometri sebuah produk akan semakin baik jika pembentukan produk dilakukan dengan proses pemesinan yang optimal.

### 3.4 Studi Literatur

Langkah yang mengawali penelitian ini adalah melakukan studi literatur. Studi literatur merupakan rangkaian proses yang bertujuan untuk memperdalam dasar teori yang digunakan pada penelitian ini. Peneliti melakukan studi literatur dengan mencari jurnal dari penelitian sebelumnya dengan topik yang sama mengenai optimasi pada proses *end milling* pada inconel 718, *Backpropagation Neural Network* (BPNN), *Artificial Bee Colony Algorithm* (ABC), dan *Differential Evolution* (DE).

### 3.5 Pengumpulan Data

#### 3.5.1 Dimensi dan Material

Data yang dibutuhkan pada penelitian ini yaitu dimensi jenis pahat *ball end mill*, material pahat *end mill*, dan material benda kerja. Data-data tersebut didapat dari penelitian sebelumnya dengan topik yang serupa seperti tertera pada tabel 3.1 berikut.

**Tabel 3. 1** Spesifikasi pahat (Kasim et al., 2015)

Jenis	<i>Ball end mill</i>
<i>Radial rake angle</i>	0°
<i>Axial rake angle</i>	-3°
Diameter	16 mm
<i>Relief angle</i>	11°

Spesifikasi dari pahat yang digunakan didapat melalui penelitian dari jurnal penelitian sebelumnya (Kasim et al. 2015). Pada pahat digunakan *cutter* berupa tungsten karbida berdiameter 10 mm yang dilapisi menggunakan TiAlN/AlCrN grade ACK 300. Pada *tes end milling* digunakan clamping menggunakan BIG Hi-Power DV40-HMC20-85 dan akan diuji pada CNC vertikal berjenis DMC 635 V Eco dengan spesifikasi pada lampiran 1. Pengukuran dilakukan menggunakan dynamometer dan *charge amplifier* Kistler 5070A dengan spesifikasi pada lampiran 2 dan 3. Konfigurasi pemasangan *cutter* pada pahat terdapat pada gambar 3.2 dimana *cutter* berupa tungsten karbida dipasang pada bagian ujung pahat *ball nose end mill*.



**Gambar 3. 2** Pahat *ball nose end mill* (Kasim et al., 2015)

### 3.5.2 Desain Eksperimen Penelitian

Pada penelitian ini digunakan parameter input dengan 3 level dengan input yang berbeda seperti pada tabel 3.2 yaitu kecepatan potong, kecepatan makan, kedalaman potong, dan lebar pemotongan. Variasi ditentukan dari jurnal penelitian sebelumnya dimana disebutkan bahwa variasi dari parameter input memiliki pengaruh terhadap hasil simulasi dari proses *end milling*. Desain eksperimen dari penelitian ini akan menggunakan data berdasarkan desain *Box-Behnken* (Kasim et al., 2015) dimana terdapat 29 data untuk mendapatkan hubungan setiap parameter input terhadap parameter output seperti pada tabel 3.3 berikut.

**Tabel 3. 2** Data input proses *end milling*

Parameter Input	Level 1	Level 2	Level 3
Kecepatan potong, $V_c$ (mm/min)	100	120	140
Kecepatan makan, $f_z$ (mm/gigi)	0.1	0.15	0.2
Kedalaman potong, $a_p$ (mm)	0.5	0.75	1
Lebar pemotongan, $a_e$ (mm)	0.2	1	1.8

**Tabel 3. 3** Hasil eksperimen *end milling* (Kasim et al., 2015)

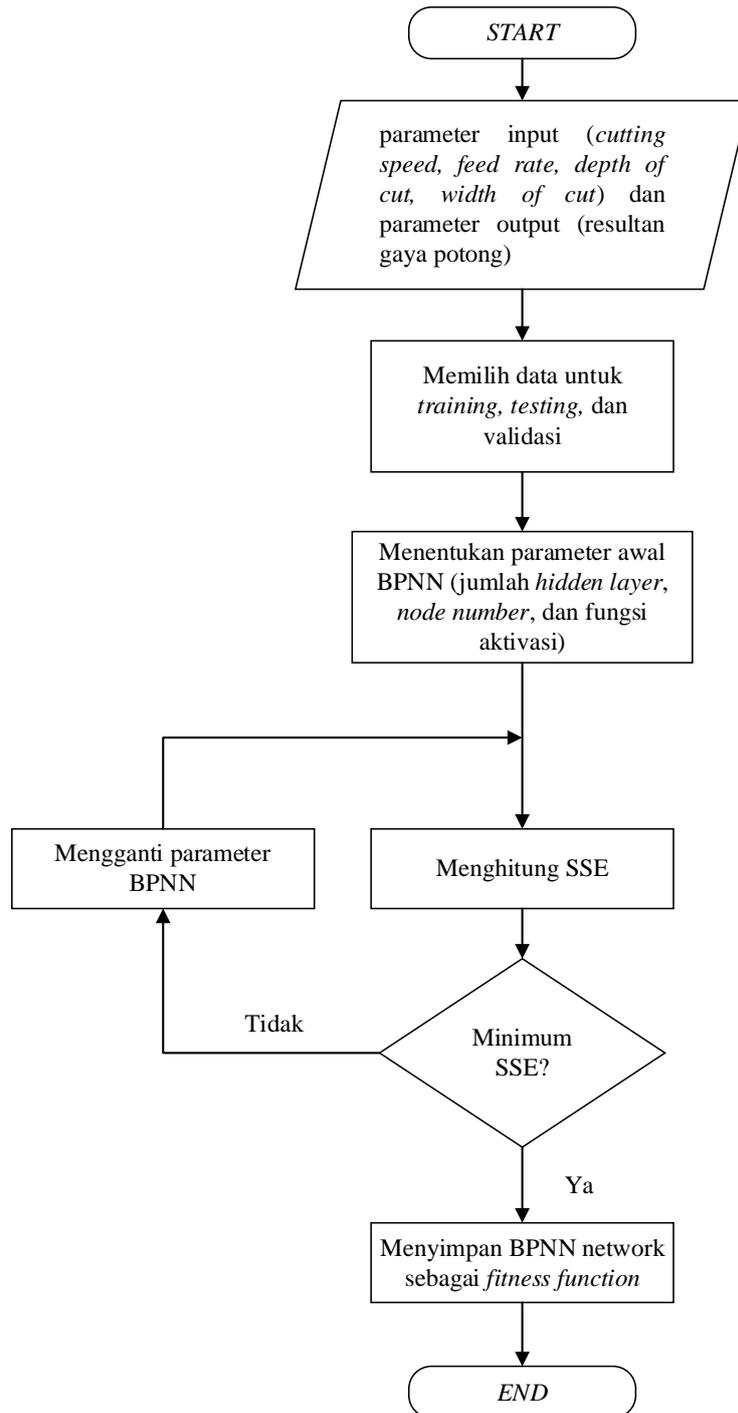
No.	$V_c$ (m/min)	$f_z$ (mm/gigi)	$a_p$ (mm)	$a_e$ (mm)	$F_x$ (N)	$F_y$ (N)	$F_z$ (N)	$F_r$ (N)
1	100	0.15	0.75	0.2	185.5	54.23	57.13	201.53
2	120	0.15	1	0.2	238.91	78.8	72.66	261.85
3	120	0.1	1	1	293.03	89.95	65.95	313.54
4	120	0.15	0.75	1	301.47	85.53	75.68	322.38
5	100	0.15	0.75	1.8	321	105.51	63.14	343.75

6	120	0.15	0.5	0.2	141.11	54.34	44.07	157.5
7	140	0.15	0.75	0.2	170.36	54.66	61.07	189.05
8	120	0.2	0.5	1	274.75	69.96	85.85	296.23
9	120	0.2	0.75	0.2	208.79	64.33	86.36	234.92
10	120	0.15	0.75	1	309.37	92	75.68	331.52
11	120	0.1	0.75	0.2	158.63	52.02	52.86	175.11
12	120	0.1	0.5	1	195.97	68.02	53.07	214.12
13	140	0.15	0.75	1.8	345.7	103.79	72.54	368.17
14	120	0.15	0.75	1	288.07	73.85	70.8	305.7
15	120	0.1	0.75	1.8	247.74	73.29	60.85	265.42
16	120	0.2	0.75	1.8	416.58	122.45	98.69	445.28
17	100	0.2	0.75	1	351.53	83.47	93.66	373.25
18	140	0.15	0.5	1	243.97	74.2	58.14	261.55
19	120	0.15	0.5	1.8	250.78	82.31	47.61	268.2
20	120	0.15	1	1.8	418.4	105.99	96.8	442.33
21	100	0.15	0.5	1	233.17	67.03	61.62	250.32
22	140	0.15	1	1	376.68	103.55	78.77	398.51
23	100	0.1	0.75	1	229.22	71.4	59.11	247.25
24	140	0.1	0.75	1	222.21	67.6	60.09	239.91
25	120	0.2	1	1	440.26	98.21	115.78	465.7
26	100	0.15	1	1	358.93	82.99	94.06	380.22
27	140	0.2	0.75	1	367.34	119.81	71.08	392.87
28	120	0.15	0.75	1	290.21	76.05	83.77	311.48
29	120	0.15	0.75	1	281.04	66.88	83.68	300.76

Dari parameter input tabel diatas merupakan data eksperimen yang diambil dari referensi penelitian yang dilakukan oleh (Kasim et al., 2015). Dari semua parameter input yang digunakan termasuk dalam data yang bersifat kontinu. Data kecepatan potong bersifat kontinu karena data dapat diperoleh dari level 1 sampai level 3 diantara 100 m/min hingga 140 m/min. Data kecepatan makan juga tergolong data kontinu, karena dapat diperoleh data yang berada pada level 1 sampai level 3 diantara 0.1 mm/gigi hingga 0.2 mm/gigi. Data kedalaman potong bersifat kontinu dan dapat diperoleh dari level 1 sampai level 3 diantara 0.5 mm sampai 1 mm. Sedangkan data lebar pemotongan bersifat kontinu dan dapat diperoleh dari level 1 sampai dengan level 3 dengan data diantara 0.2 sampai 1.8 mm.

### 3.6 Pemodelan BPNN

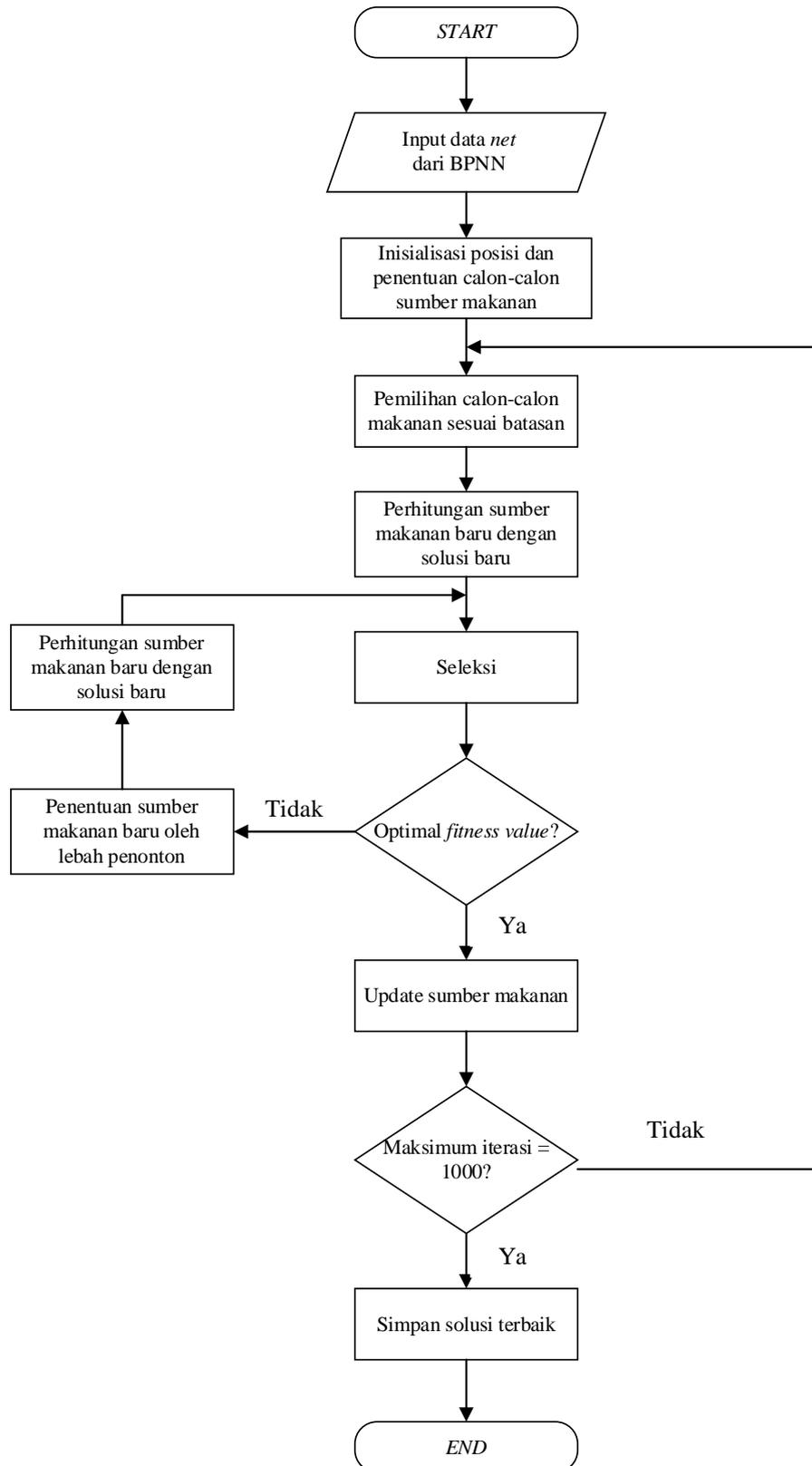
Dalam pemodelan BPNN untuk mendapatkan hubungan antara parameter input dan parameter output yang memiliki nilai MSE terkecil untuk dilakukan proses optimasi menggunakan metode BPNN-ABC dan BPNN-DE, maka ditunjukkan dengan diagram alir BPNN pada gambar 3.3 berikut:



**Gambar 3. 3** Diagram alir BPNN

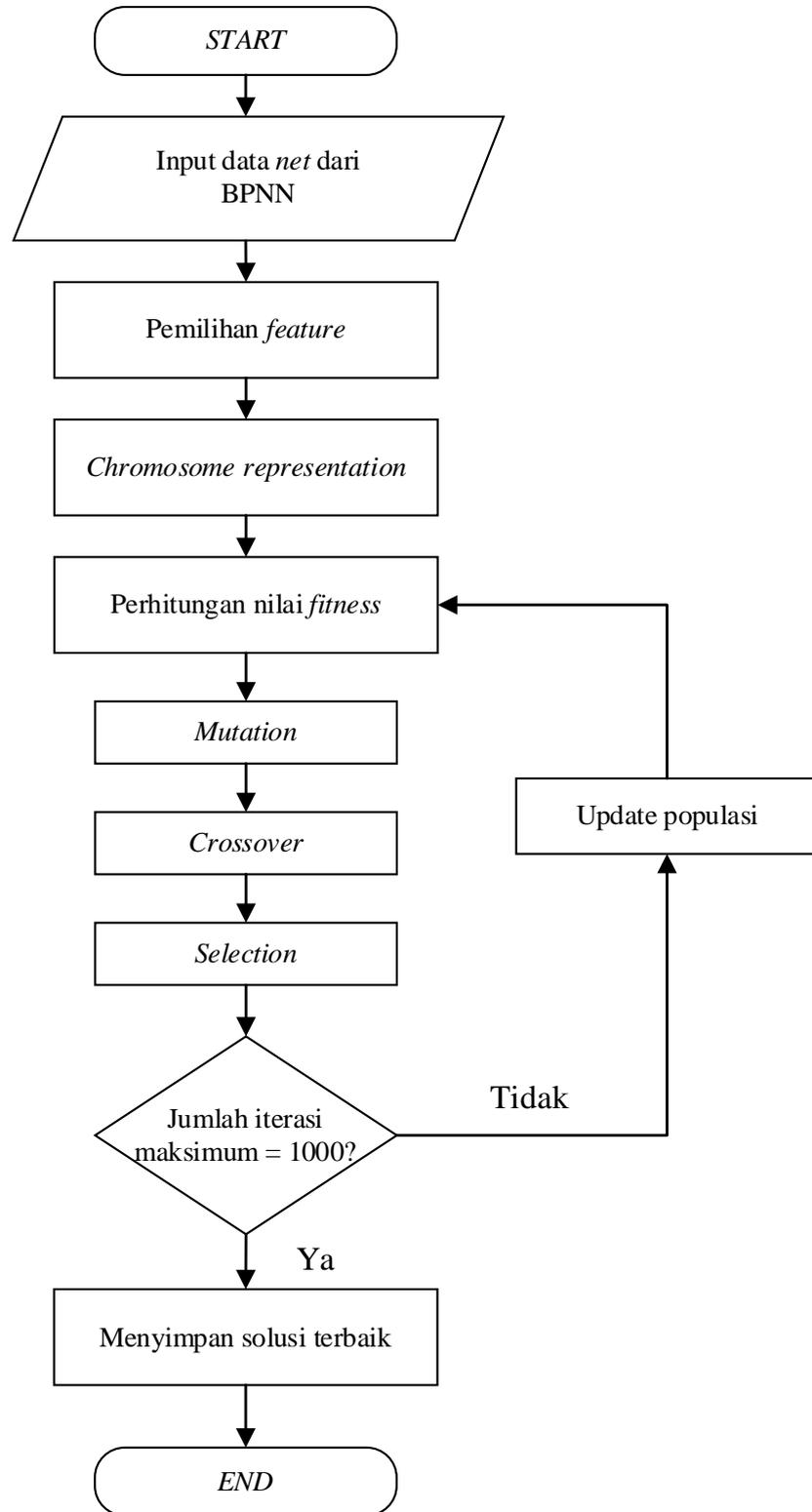
Dari diagram alir diatas, jumlah *hidden layer* yang digunakan pada penelitian ini yaitu 1 sampai 5 *hidden layer* dengan jumlah neuron pada setiap *layer* menggunakan 2 sampai 10 neuron. Fungsi aktivasi yang digunakan yaitu *hardlim*, *hardlims*, *purelin*, *satlin*, *logsig*, dan *tansig*. Proses *running* untuk setiap konfigurasi dilakukan dalam sepuluh kali percobaan, dan dicari nilai MSE (*Mean Square Error*) terkecil. Setelah melakukan semua konfigurasi percobaan, nilai MSE yang paling rendah digunakan untuk proses optimasi menggunakan algoritma *Artificial Bee Colony Algorithm* (ABC) dan *Differential Evolution* (DE).

Pencarian nilai minimum menggunakan metode *Artificial Bee Colony Algorithm* (ABC) yang digambarkan dengan diagram alir optimasi ABC pada gambar 3.4 sebagai berikut.



**Gambar 3. 4** Diagram alir algoritma *Artificial Bee Colony*

Sedangkan pencarian nilai minimum menggunakan metode *Differential Evolution* (DE) yang digambarkan dengan diagram alir optimasi DE pada gambar 3.5 sebagai berikut.



**Gambar 3. 5** Diagram alir algoritma *Differential Evolution*

### **3.7 Komparasi Hasil**

Hasil yang diperoleh dari proses optimasi menggunakan metode ABC dan DE digunakan untuk mencari nilai konfigurasi proses *end milling* yang menghasilkan resultan gaya potong paling minimum dan dibandingkan dengan hasil penelitian dengan judul “*Prediction of Cutting Force in End Milling of Inconel 718* “ oleh (Kasim et al., 2015) yang menggunakan metode optimasi *Response Surface Methodology* (RSM).

### **3.8 Penarikan Kesimpulan dan Saran**

Pada tahap ini akan dibahas mengenai hasil dari proses penelitian dan pemberian saran untuk membantu penelitian yang akan dilakukan selanjutnya.

## BAB IV HASIL DAN PEMBAHASAN

### 4.1 Parameter Input dan Respon

Data parameter input dan parameter output (respon) yang digunakan berdasarkan pada data eksperimen yang telah dilakukan pada jurnal terdahulu dengan judul “*Prediction of Cutting Force in End Milling of Inconel 718*” oleh Kasim et al. pada tahun 2015. Data parameter input yang digunakan pada penelitian ini yaitu kecepatan potong, kecepatan makan, kedalaman potong, dan lebar pemotongan berdasarkan metode *Box-Behnken* dengan 29 variasi eksperimen. Sedangkan data parameter output yang digunakan adalah resultan gaya potong.

### 4.2 Pemodelan BPNN

Proses *training*, *testing*, dan validasi dilakukan menggunakan metode *backpropagation neural network* (BPNN) dengan variasi jumlah *hidden layer*, jumlah neuron setiap *hidden layer*, dan fungsi aktivasi untuk mendapatkan nilai *Mean Square Error* (MSE) yang paling kecil. Sebelum melakukan proses *training neural network*, data hasil simulasi dinormalisasikan terlebih dahulu pada menu MATLAB menggunakan perintah *mapminmax*. Fungsi dari perintah *mapminmax* yaitu mengubah nilai parameter menjadi suatu nilai yang memiliki nilai minimum sebesar -1 dan nilai maksimum +1. Persamaan perintah *mapminmax* dapat dituliskan sebagai berikut.

$$x_j = \frac{x_i - \min_a}{\max_a - \min_a} (\text{newmaks}_a - \text{newmin}_a) + \text{newmin}_a \dots \dots \dots (4.1)$$

dimana

- $x_j$  = nilai  $x$  yang ternormalisasi
- $x_i$  = nilai  $x$  awal
- $\min_a$  = nilai minimum dari keseluruhan data
- $\max_a$  = nilai maksimum dari keseluruhan data
- $\text{newmin}_a$  = nilai minimum dari *range* yang ditentukan
- $\text{newmaks}_a$  = nilai maksimum dari *range* yang ditentukan

Dengan contoh perhitungan menggunakan data eksperimen pertama dengan variabel kecepatan potong ( $v$ ) bernilai  $x_1 = 100$ , dan *range* dari data yaitu  $\min_a = 100$ ,  $\max_a = 140$ ,  $\text{newmin}_a = -1$ ,  $\text{newmaks}_a = 1$ , sehingga diperoleh nilai

$$x_j = \frac{100 - 100}{140 - 100} (1 - (-1)) + (-1)$$

$$x_j = \frac{0}{40} (2) - 1$$

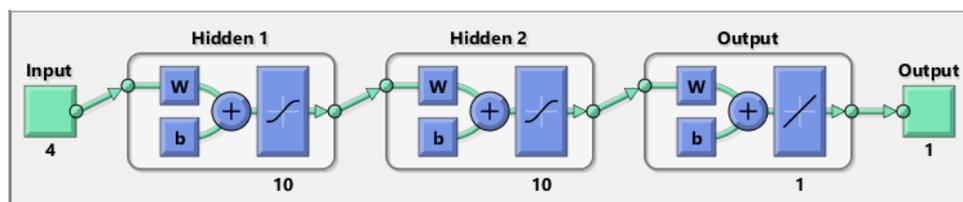
$$x_j = -1$$

Normalisasi nilai diperlukan dikarenakan pada parameter input kecepatan potong 100, 120, dan 140 memiliki nilai yang berbeda jauh dengan parameter input kecepatan makan, kedalaman potong, dan lebar pemotongan. Perbedaan nilai dari parameter input yang digunakan merupakan sebab dilakukannya normalisasi agar nilai setiap parameter yang digunakan tidak terlalu jauh sehingga hasil prediksi yang didapatkan semakin akurat. Parameter input kecepatan

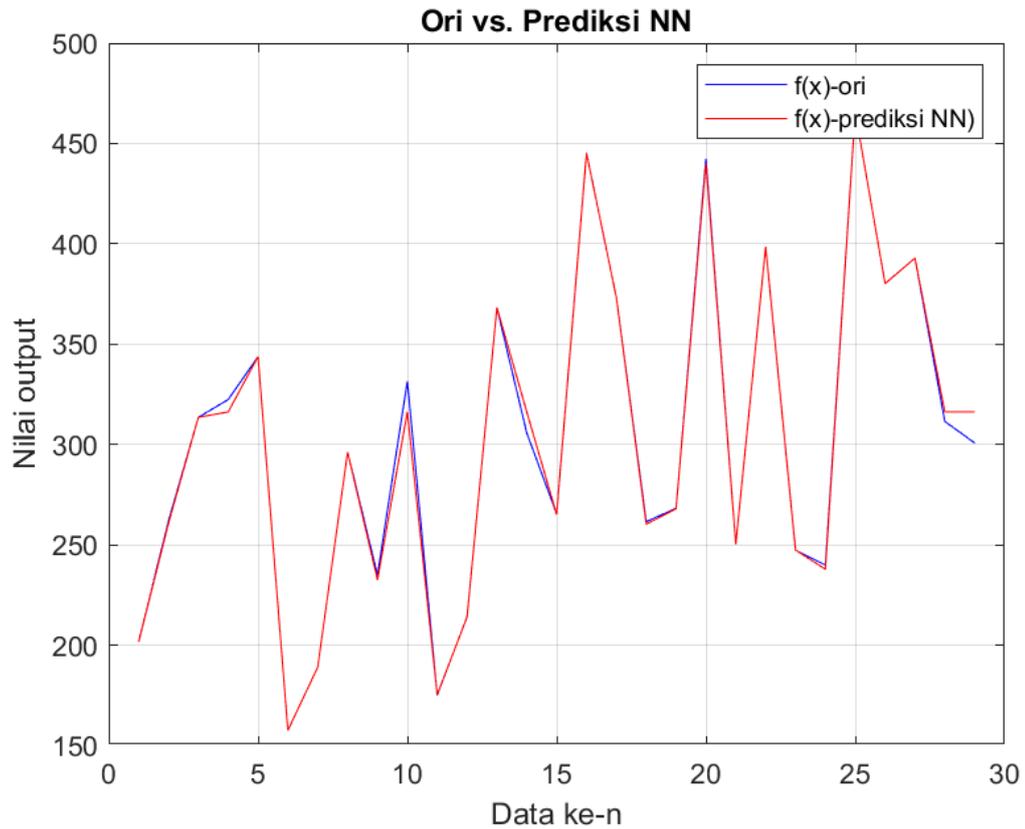
potong akan dinormalisasi menggunakan persamaan 4.1 dari nilai 100 hingga 140 m/menit menjadi -1 hingga 1, kecepatan makan akan dinormalisasi dari 0.1 hingga 0.2 mm/gigi menjadi -1 hingga 1, kedalaman potong akan dinormalisasi dari 0.5 sampai 1 mm menjadi -1 hingga 1, dan lebar pemotongan akan dinormalisasi dari 0.2 hingga 1.8 mm menjadi -1 hingga 1. Sedangkan parameter output yang akan dinormalisasi yaitu gaya potong maksimum dari 157.5 sampai 465.7 N menjadi -1 hingga 1.

Konfigurasi pada BPNN terdapat pada lampiran 4 menggunakan variasi 1 sampai 5 *hidden layer* dimana setiap *hidden layer* terdiri dari 2 sampai 10 neuron dengan kombinasi fungsi aktivasi *hardlim*, *hardlims*, *purelin*, *satlin*, *logsig*, dan *tansig* dengan 10 kali pengulangan setiap variasi konfigurasi BPNN yang dilakukan. Proses *training* dilakukan menggunakan metode *trainlm* (*Lavenberg-Marquardt*) dimana model yang digunakan yaitu model yang memiliki nilai MSE paling kecil dengan parameter input berupa kecepatan potong, kecepatan makan, kedalaman potong, dan lebar pemotongan sedangkan parameter output yang dicari berupa resultan gaya potong. Jumlah *epoch* yang digunakan pada BPNN yaitu 1000 dimana dari total 29 data dibagi menjadi 3 kategori yaitu 70% untuk *training* dengan 21 data, 15% untuk *testing* dengan 4 data, dan 15% untuk validasi dengan 4 data.

Hasil pemodelan BPNN yang akan digunakan merupakan kombinasi parameter BPNN yang menunjukkan nilai MSE terkecil pada hasil *training* untuk setiap variasinya. Pada gambar 4.1 merupakan konfigurasi BPNN dengan 2 *hidden layer* dengan struktur jaringan 4-10-10-1 dimana terdapat 4 input yaitu kecepatan potong, kecepatan makan, kedalaman potong, dan lebar pemotongan. Pada *hidden layer* pertama menggunakan fungsi aktivasi *tansig* dengan jumlah neuron 10 dan pada *hidden layer* kedua menggunakan fungsi aktivasi *tansig* dengan jumlah neuron 10 dimana terdapat 1 output berupa resultan gaya potong. Nilai MSE yang didapat menggunakan konfigurasi ini yaitu 0.00096269. Output *net* dari BPNN disimpan kedalam file “bestfunction.mat” seperti pada lampiran 5 yang akan digunakan dalam optimasi menggunakan algoritma *Artificial Bee Colony* (ABC) dan *Differential Evolution* (DE).



**Gambar 4. 1** Struktur jaringan 4-10-10-1



**Gambar 4. 2** Hasil prediksi BPNN

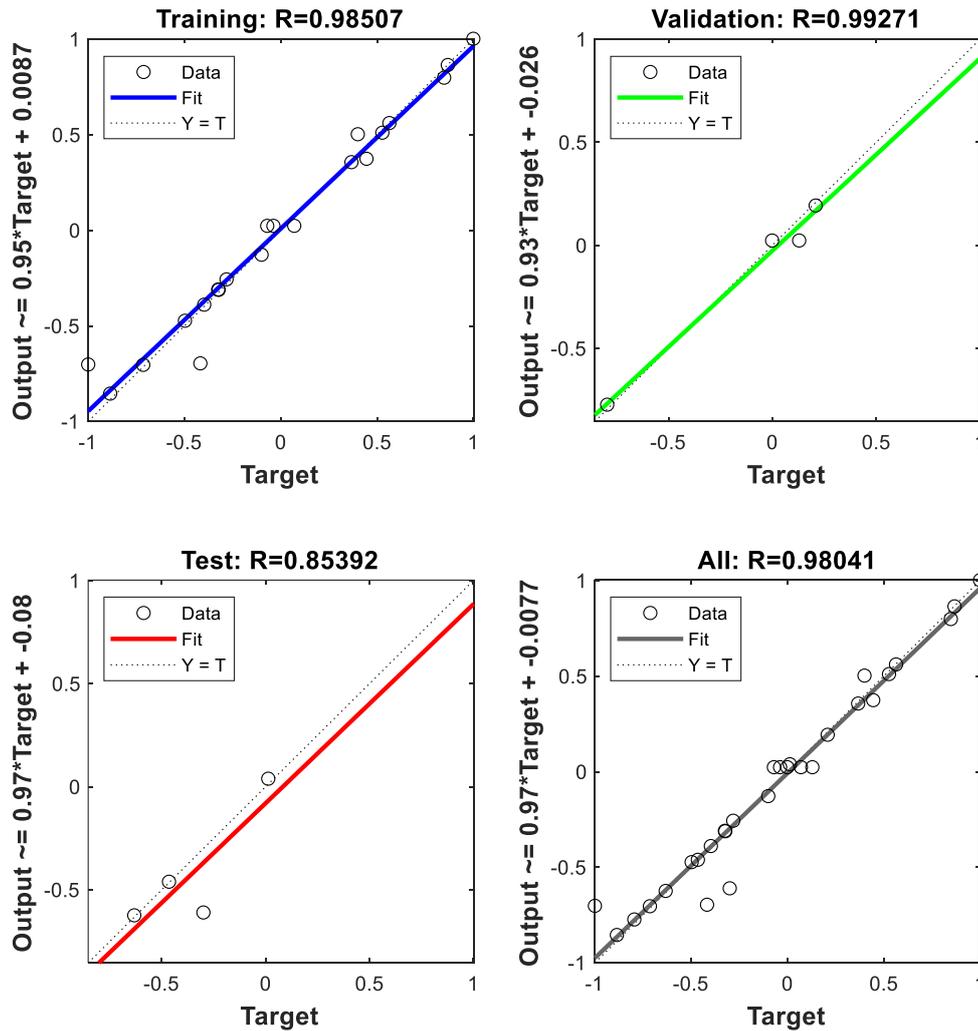
**Tabel 4. 1** Perbandingan nilai target dan nilai hasil prediksi BPNN

No.	$V_c$ (m/min)	$f_z$ (mm/gigi)	$a_p$ (mm)	$a_e$ (mm)	$F_r$ Target (N)	$F_r$ Prediksi (N)	Error (%)
1	100	0.15	0.75	0.2	201.53	201.5212	0.004385
2	120	0.15	1	0.2	261.85	260.6547	0.456495
3	120	0.1	1	1	313.54	313.533	0.002243
4	120	0.15	0.75	1	322.38	316.2224	1.910054
5	100	0.15	0.75	1.8	343.75	343.8038	0.015641
6	120	0.15	0.5	0.2	157.5	157.5616	0.039093
7	140	0.15	0.75	0.2	189.05	189.0318	0.009618
8	120	0.2	0.5	1	296.23	296.2066	0.007895
9	120	0.2	0.75	0.2	234.92	232.5326	1.016279
10	120	0.15	0.75	1	331.52	316.2224	4.614392
11	120	0.1	0.75	0.2	175.11	175.3467	0.135167
12	120	0.1	0.5	1	214.12	213.9879	0.061686
13	140	0.15	0.75	1.8	368.17	368.2667	0.026271
14	120	0.15	0.75	1	305.7	316.2224	3.442057
15	120	0.1	0.75	1.8	265.42	265.142	0.104722
16	120	0.2	0.75	1.8	445.28	445.1867	0.020954

17	100	0.2	0.75	1	373.25	373.2603	0.002768
18	140	0.15	0.5	1	261.55	260.2404	0.500694
19	120	0.15	0.5	1.8	268.2	268.0017	0.073921
20	120	0.15	1	1.8	442.33	440.073	0.510261
21	100	0.15	0.5	1	250.32	250.2539	0.026398
22	140	0.15	1	1	398.51	398.5115	0.000381
23	100	0.1	0.75	1	247.25	247.4335	0.074219
24	140	0.1	0.75	1	239.91	237.7808	0.887519
25	120	0.2	1	1	465.7	465.7261	0.005603
26	100	0.15	1	1	380.22	380.145	0.019723
27	140	0.2	0.75	1	392.87	392.8822	0.003105
28	120	0.15	0.75	1	311.48	316.2224	1.522527
29	120	0.15	0.75	1	300.76	316.2224	5.141099

Grafik pada gambar 4.2 menunjukkan variasi data dari hasil prediksi BPNN (output) terhadap data eksperimen (target). Perbedaan antara nilai output dan target dapat disebabkan beberapa faktor diantaranya kurangnya data yang digunakan pada proses *training* sehingga hasil prediksi yang dilakukan BPNN belum akurat. Hasil BPNN pada gambar 4.2 memiliki tren yang sama dengan hasil eksperimen dimana tidak terdapat data yang memiliki nilai eror lebih dari 10% dari prediksi yang dihasilkan oleh BPNN. Pada tabel 4.4 hasil prediksi yang dihasilkan dari BPNN memiliki tingkat akurasi yang baik dengan nilai rerata eror sebesar 0.71% sehingga dapat dikatakan BPNN dapat memodelkan prediksi resultan gaya potong dengan baik.

Grafik pada *nntraintool* menghasilkan grafik pada gambar 4.3 yang menunjukkan grafik *regression* pada data *training*, *testing*, dan validasi dimana grafik tersebut menunjukkan nilai hubungan antara target dengan respon dengan konstanta R (*correlation coefficient*). Nilai R merupakan konstanta statistik yang menunjukkan seberapa dekat antara dataset ke garis regresi. Nilai target pada grafik regresi menunjukkan nilai yang terukur pada dataset (eksperimen) sedangkan output merupakan hasil prediksi dari BPNN. Untuk mencapai kesesuaian data (*perfect fit*), nilai suatu data harus berada pada garis dengan gradien 1 atau disepanjang garis 45 derajat. Hasil dari regresi untuk data *training* yaitu memiliki nilai R sebesar 0.98507 pada data *training*, 0.85932 pada data *testing*, 0.99271 pada data validasi, dan 0.98041 untuk nilai R total. Hal tersebut menunjukkan data yang digunakan terdapat kesesuaian dari data yang digunakan (*perfect fit*) dengan nilai R total sebesar 0.98041 dimana kesesuaian data untuk respon resultan gaya potong dapat dinyatakan baik.



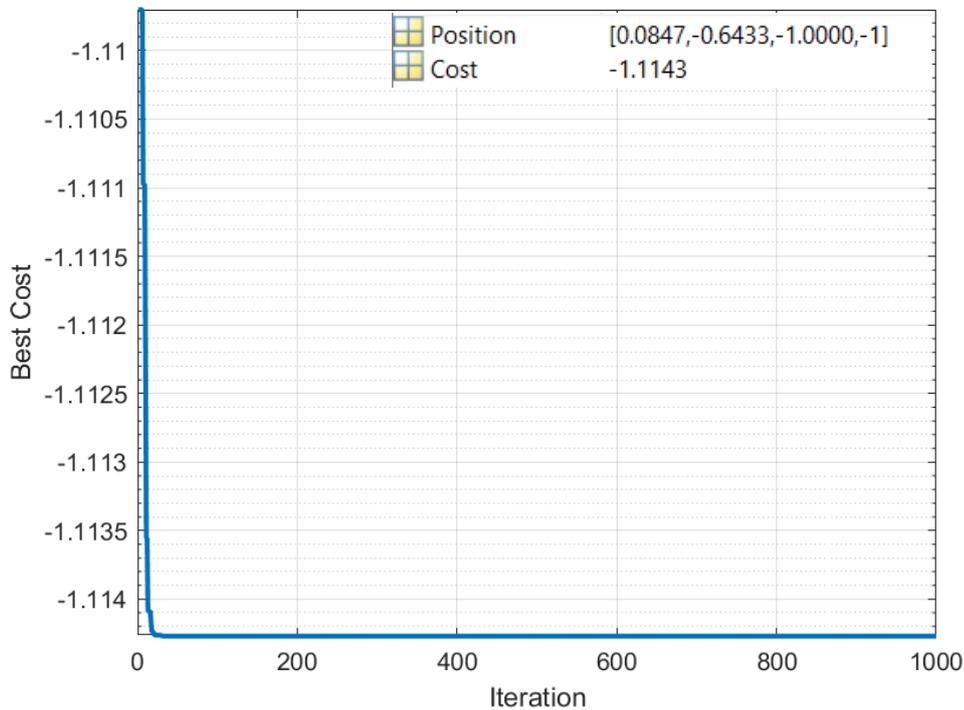
**Gambar 4. 3** Nilai konstanta R untuk data *training*, *testing*, dan *validasi*

#### 4.3 Optimasi dengan Algoritma *Artificial Bee Colony*

Proses optimasi menggunakan algoritma *Artificial Bee Colony* (ABC) pada penelitian ini menggunakan bantuan perangkat lunak MATLAB R2020b. Data *net* paling optimum dari BPNN yang didapat dengan cara mencari nilai MSE paling kecil pada setiap konfigurasi digunakan untuk mencari input paling optimal menggunakan algoritma ABC. Input pada data *net* berupa informasi mengenai parameter-parameter yang digunakan pada BPNN seperti jumlah *hidden layer*, jumlah neuron setiap *hidden layer*, fungsi aktivasi, *bias*, *weight* dan *stopping criteria*. Sedangkan input optimal yang ingin dicari menggunakan algoritma ABC berupa parameter kecepatan potong, kecepatan makan, kedalaman potong, dan lebar pemotongan yang paling optimal sehingga menghasilkan nilai resultan gaya potong paling kecil. Nilai input parameter optimal pada algoritma ABC disimbolkan dengan “Bestsol” yang menunjukkan posisi dari sumber makanan pada koloni lebah yang memiliki nilai *fitness value* terkecil. Sedangkan nilai output optimal pada algoritma ABC disimbolkan dengan “Bestcost” yang menunjukkan nilai dari *fitness value* yang terkecil.

Langkah dalam melakukan optimasi menggunakan metode ABC yaitu menentukan parameter optimasi seperti jumlah iterasi maksimum, jumlah populasi koloni lebah, faktor *trial limit*, dan koefisien percepatan. Pada penelitian ini jumlah iterasi maksimum (MaxIt) yang

digunakan yaitu 1000 iterasi, jumlah populasi koloni lebah (nPop) 100 populasi, konstanta trial limit bernilai 0.6, sedangkan nilai koefisien percepatan yaitu 1. Hasil dari optimasi algoritma *artificial bee colony* berupa nilai *fitness value* yang menunjukkan parameter input kecepatan potong, kecepatan makan, kedalaman potong, dan lebar pemotongan paling optimal setelah nilai *fitness value* tersebut dinormalisasi.



**Gambar 4. 4** Hasil optimasi ABC

Pada gambar 4.4 hasil optimasi algoritma ABC didapat nilai solusi terbaik pada *fitness value* sebesar -1.11427 dan mencapai konvergensi pada iterasi ke-54. Hasil dari *best sol* dari algoritma ABC yaitu didapat parameter input paling optimal kecepatan potong dengan nilai 0.08472, kecepatan makan dengan nilai -0.64331, kedalaman potong dengan nilai -1, dan lebar pemotongan dengan nilai -1. Nilai dari *best sol* dan *best cost* yang didapat merupakan nilai dalam kondisi ternormalisasi, sehingga diperlukan denormalisasi menggunakan fungsi *reverse mapminmax*. Fungsi *reverse mapminmax* merupakan kebalikan dari fungsi *mapminmax* dimana tujuan dari fungsi ini untuk mengubah nilai yang ternormalisasi menjadi nilai awal/aktual dengan rentang sesuai data awal sebelum dilakukannya normalisasi. Persamaan pada fungsi *reverse mapminmax* dapat ditunjukkan sebagai berikut:

$$x_i = \frac{x_j - \text{newmin}_a}{\text{newmaks}_a - \text{newmin}_a} (\text{maks}_a - \text{min}_a) + \text{min}_a \dots \dots \dots (4.2)$$

Dimana

$x_i$  = nilai  $x$  awal

$x_j$  = nilai  $x$  yang ternormalisasi

$\text{newmin}_a$  = nilai minimum normalisasi dari *range* yang ditentukan

$\text{newmaks}_a$  = nilai maksimum normalisasi dari *range* yang ditentukan

$\text{min}_a$  = nilai minimum awal dari keseluruhan data

$maks_a$  = nilai maksimum awal dari keseluruhan data

Hasil dari proses denormalisasi menggunakan fungsi *reverse mapminmax* terhadap parameter input dan parameter output paling optimal pada proses *end milling* adalah sebagai berikut:

1. Kecepatan Potong

Nilai *best sol* dari proses optimasi menggunakan metode ABC didapatkan nilai untuk parameter input kecepatan potong yaitu sebesar 0.08472. Pada parameter ini nilai ternormalisasi antara -1 hingga 1 akan didenormalisasi menjadi 100 hingga 140. Dengan menggunakan persamaan 4.2 untuk nilai ternormalisasi sebesar 0.08472 didapatkan nilai aktual sebagai berikut:

$$x_i = \frac{0.08472 - (-1)}{1 - (-1)} (140 - 100) + 100$$
$$x_i = \frac{1.08472}{2} (40) + 100$$
$$x_i = 121.6944$$

sehingga untuk metode optimasi menggunakan ABC didapat parameter input optimal untuk kecepatan potong sebesar 121.6944 m/min.

2. Kecepatan Makan

Nilai *best sol* dari proses optimasi menggunakan metode ABC didapatkan nilai untuk parameter input kecepatan makan yaitu sebesar -0.64331. Pada parameter ini nilai ternormalisasi antara -1 hingga 1 akan didenormalisasi menjadi 0.1 hingga 0.2. Dengan menggunakan persamaan 4.2 untuk nilai ternormalisasi sebesar -0.64331 didapatkan nilai aktual sebagai berikut:

$$x_i = \frac{-0.64331 - (-1)}{1 - (-1)} (0.2 - 0.1) + 0.1$$
$$x_i = \frac{0.35669}{2} (0.1) + 0.1$$
$$x_i = 0.11783$$

sehingga untuk metode optimasi menggunakan ABC didapat parameter input optimal untuk kecepatan makan sebesar 0.11783 mm/gigi.

3. Kedalaman Potong

Nilai *best sol* dari proses optimasi menggunakan metode ABC didapatkan nilai untuk parameter input kedalaman potong yaitu sebesar -1. Pada parameter ini nilai ternormalisasi antara -1 hingga 1 akan didenormalisasi menjadi 0.5 hingga 1. Dengan menggunakan persamaan 4.2 untuk nilai ternormalisasi sebesar -1 didapatkan nilai aktual sebagai berikut:

$$x_i = \frac{-1 - (-1)}{1 - (-1)} (1 - 0.5) + 0.5$$
$$x_i = \frac{0}{2} (0.5) + 0.5$$
$$x_i = 0.5$$

sehingga untuk metode optimasi menggunakan ABC didapat parameter input optimal untuk kedalaman potong sebesar 0.5 mm.

#### 4. Lebar Pemotongan

Nilai *best sol* dari proses optimasi menggunakan metode ABC didapatkan nilai untuk parameter input kedalaman potong yaitu sebesar -1. Pada parameter ini nilai ternormalisasi antara -1 hingga 1 akan didenormalisasi menjadi 0.2 hingga 1.8. Dengan menggunakan persamaan 4.2 untuk nilai ternormalisasi sebesar -1 didapatkan nilai aktual sebagai berikut:

$$x_i = \frac{-1 - (-1)}{1 - (-1)} (1.8 - 0.2) + 0.2$$

$$x_i = \frac{0}{2} (1.6) + 0.2$$

$$x_i = 0.2$$

sehingga untuk metode optimasi menggunakan ABC didapat parameter input optimal untuk lebar pemotongan sebesar 0.2 mm

#### 5. Resultan Gaya Potong

Nilai *best cost* dari proses optimasi menggunakan metode ABC didapatkan nilai untuk parameter output resultan gaya potong yaitu sebesar -1.11427. Pada parameter ini nilai ternormalisasi antara -1 hingga 1 akan didenormalisasi menjadi 157.5 sampai 465.7. Dengan menggunakan persamaan 4.2 untuk nilai ternormalisasi sebesar -1.11427 didapatkan nilai aktual sebagai berikut:

$$x_i = \frac{-1.11427 - (-1)}{1 - (-1)} (465.7 - 157.5) + 157.5$$

$$x_i = \frac{-0.11427}{2} (308.2) + 157.5$$

$$x_i = 139.8909$$

sehingga untuk metode optimasi menggunakan ABC didapat parameter output optimal untuk resultan gaya potong sebesar 139.8909 N.

Hasil dari proses optimasi menggunakan algoritma ABC terdapat pada tabel 4.2 dimana parameter input optimal didapat dengan kecepatan potong sebesar 121.6944 m/min, kecepatan makan 0.11783 mm/gigi, kedalaman potong 0.5 mm, dan lebar pemotongan 0.2 mm. Sedangkan parameter output optimal resultan gaya potong berdasarkan hasil optimasi ABC didapat sebesar 139.8909 N.

**Tabel 4. 2** Nilai optimal hasil optimasi ABC

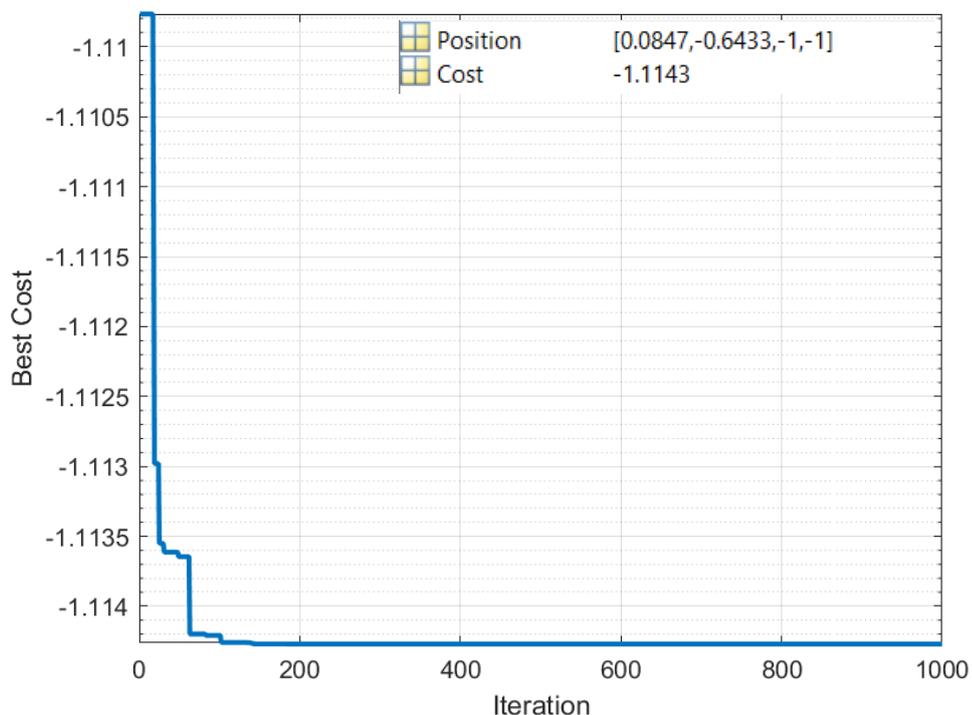
Variabel End Milling	Satuan	Nilai normalisasi	Nilai denormalisasi
Kecepatan potong ( $V_c$ )	m/min	0.08472	121.6944
Kecepatan makan ( $f_z$ )	mm/gigi	-0.64331	0.11783
Kedalaman potong ( $a_p$ )	Mm	-1	0.5
Lebar pemotongan ( $a_e$ )	Mm	-1	0.2

Parameter respon	Satuan	Nilai normalisasi	Nilai denormalisasi
Resultan gaya potong ( $F_r$ )	N	-1.11427	139.8909

#### 4.4 Optimasi dengan Algoritma *Differential Evolution*

Proses optimasi menggunakan algoritma *Differential Evolution* (DE) pada dasarnya memiliki kesamaan dengan proses optimasi menggunakan algoritma ABC. Nilai MSE terkecil dalam suatu konfigurasi BPNN digunakan untuk membentuk *net* yang berisikan informasi mengenai bagaimana konfigurasi tersebut dilakukan. Parameter input dari BPNN yang digunakan pada penelitian ini yaitu kecepatan potong, kecepatan makan, kedalaman potong, dan lebar pemotongan dimana parameter output yang akan diprediksi yaitu resultan gaya potong. Hasil prediksi dari BPNN disimpan dalam suatu *file (net)* yang berisikan jumlah *hidden layer*, jumlah neuron setiap *hidden layer*, fungsi aktivasi, *bias*, *weight* dan *stopping criteria*. Nilai input parameter optimal pada algoritma DE disimbolkan dengan “Bestsol” yang menunjukkan populasi vektor yang memiliki nilai *fitness value* terkecil. Sedangkan nilai output optimal pada algoritma DE disimbolkan dengan “Bestcost” yang menunjukkan nilai dari *fitness value* yang terkecil.

Sedangkan langkah dalam melakukan optimasi menggunakan metode DE yaitu menentukan parameter optimasi seperti jumlah iterasi maksimum, jumlah populasi, batas atas dan bawah faktor skala, dan probabilitas *crossover*. Pada penelitian ini jumlah iterasi maksimum yang digunakan yaitu 1000 iterasi, jumlah populasi 100, batas bawah faktor skala bernilai 0, batas atas faktor skala 10, sedangkan probabilitas *crossover* bernilai 0.2. Optimasi algoritma DE memiliki kesamaan dengan algoritma ABC dimana hasil optimasi berupa nilai *fitness value* yang menunjukkan parameter input kecepatan potong, kecepatan makan, kedalaman potong, dan lebar pemotongan paling optimal dengan parameter output resultan gaya potong paling minimum.



**Gambar 4. 5** Hasil optimasi DE

Grafik pada gambar 4.4 menunjukkan hasil optimasi algoritma DE pada parameter *end milling* dan didapat nilai solusi terbaik pada *fitness value* sebesar -1.11427 dan konvergen pada iterasi ke-243. Hasil dari *best sol* dari algoritma DE yaitu didapat parameter input paling optimal kecepatan potong dengan nilai 0.08472, kecepatan makan dengan nilai -0.64331,

kedalaman potong dan lebar pemotongan dengan nilai sama sebesar -1. Nilai dari *best sol* dan *best cost* perlu dilakukan denormalisasi menggunakan fungsi *reverse mapminmax* dikarenakan hasil parameter merupakan nilai dalam kondisi ternormalisasi. Persamaan pada fungsi *reverse mapminmax* dapat ditunjukkan pada persamaan 4.2. Hasil dari parameter input dan parameter output yang didenormalisasi menggunakan fungsi *reverse mapminmax* pada algoritma DE adalah sebagai berikut:

#### 1. Kecepatan Potong

Nilai *best sol* dari proses optimasi menggunakan metode DE didapatkan nilai untuk parameter input kecepatan potong yaitu sebesar 0.08472. Pada parameter ini nilai ternormalisasi antara -1 hingga 1 akan didenormalisasi menjadi 100 hingga 140. Dengan menggunakan persamaan 4.2 untuk nilai ternormalisasi sebesar 0.08472 didapatkan nilai aktual sebagai berikut:

$$x_i = \frac{0.08472 - (-1)}{1 - (-1)} (140 - 100) + 100$$

$$x_i = \frac{1.08472}{2} (40) + 100$$

$$x_i = 121.6944$$

sehingga untuk metode optimasi menggunakan DE didapat parameter input optimal untuk kecepatan potong sebesar 121.6944 m/min.

#### 2. Kecepatan Makan

Nilai *best sol* dari proses optimasi menggunakan metode DE didapatkan nilai untuk parameter input kecepatan makan yaitu sebesar -0.64331. Pada parameter ini nilai ternormalisasi antara -1 hingga 1 akan didenormalisasi menjadi 0.1 hingga 0.2. Dengan menggunakan persamaan 4.2 untuk nilai ternormalisasi sebesar -0.64331 didapatkan nilai aktual sebagai berikut:

$$x_i = \frac{-0.64331 - (-1)}{1 - (-1)} (0.2 - 0.1) + 0.1$$

$$x_i = \frac{0.35569}{2} (0.1) + 0.1$$

$$x_i = 0.11783$$

sehingga untuk metode optimasi menggunakan DE didapat parameter input optimal untuk kecepatan makan sebesar 0.11783 mm/gigi.

#### 3. Kedalaman Potong

Nilai *best sol* dari proses optimasi menggunakan metode DE didapatkan nilai untuk parameter input kedalaman potong yaitu sebesar -1. Pada parameter ini nilai ternormalisasi antara -1 hingga 1 akan didenormalisasi menjadi 0.5 hingga 1. Dengan menggunakan persamaan 4.2 untuk nilai ternormalisasi sebesar -1 didapatkan nilai aktual sebagai berikut:

$$x_i = \frac{-1 - (-1)}{1 - (-1)} (1 - 0.5) + 0.5$$

$$x_i = \frac{0}{2} (0.5) + 0.5$$

$$x_i = 0.5$$

sehingga untuk metode optimasi menggunakan DE didapat parameter input optimal untuk kedalaman potong sebesar 0.5 mm.

#### 4. Lebar Pemotongan

Nilai *best sol* dari proses optimasi menggunakan metode DE didapatkan nilai untuk parameter input lebar pemotongan yaitu sebesar -1. Pada parameter ini nilai ternormalisasi antara -1 hingga 1 akan didenormalisasi menjadi 0.2 hingga 1.8. Dengan menggunakan persamaan 4.2 untuk nilai ternormalisasi sebesar -1 didapatkan nilai aktual sebagai berikut:

$$x_i = \frac{-1 - (-1)}{1 - (-1)} (1.8 - 0.2) + 0.2$$

$$x_i = \frac{0}{2} (1.6) + 0.2$$

$$x_i = 0.2$$

sehingga untuk metode optimasi menggunakan DE didapat parameter input optimal untuk lebar pemotongan sebesar 0.2 mm.

#### 5. Resultan Gaya Potong

Nilai *best cost* dari proses optimasi menggunakan metode DE didapatkan nilai untuk parameter output resultan gaya potong dengan nilai *fitness value* sebesar -1.11427. Pada parameter ini nilai ternormalisasi antara -1 hingga 1 akan didenormalisasi menjadi 157.5 sampai 465.7. Dengan menggunakan persamaan 4.2 untuk nilai ternormalisasi sebesar -1.11427 didapatkan nilai aktual sebagai berikut:

$$x_i = \frac{-1.11427 - (-1)}{1 - (-1)} (465.7 - 157.5) + 157.5$$

$$x_i = \frac{-0.11427}{2} (308.2) + 157.5$$

$$x_i = 139.8909$$

sehingga untuk metode optimasi menggunakan DE didapat parameter output optimal untuk resultan gaya potong sebesar 139.8909 N.

Hasil dari proses optimasi menggunakan algoritma DE terdapat pada tabel 4.3 dimana parameter input optimal didapat dengan kecepatan potong sebesar 121.8122 m/min, kecepatan makan 0.11801 mm/gigi, kedalaman potong 0.5 mm, dan lebar pemotongan 0.2 mm. Sedangkan parameter output optimal resultan gaya potong berdasarkan hasil optimasi DE didapat sebesar 139.9003 N.

**Tabel 4. 3** Nilai optimal hasil optimasi DE

Variabel End Milling	Satuan	Nilai normalisasi	Nilai denormalisasi
Kecepatan potong ( $V_c$ )	m/min	0.08472	121.6944
Kecepatan makan ( $f_z$ )	mm/gigi	-0.64331	0.11783
Kedalaman potong ( $a_p$ )	mm	-1	0.5
Lebar pemotongan ( $a_e$ )	mm	-1	0.2

Parameter respon	Satuan	Nilai normalisasi	Nilai denormalisasi
Resultan gaya potong ( $F_r$ )	N	-1.11427	139.8909

#### 4.5 Penentuan Parameter Berdasarkan Kondisi Sebenarnya

Dikarenakan parameter input kecepatan potong dan kecepatan makan hasil optimasi menggunakan metode ABC dan DE memiliki keakuratan 4 angka dibelakang koma, maka perlu dilakukan pendekatan ke nilai yang lebih rendah maupun lebih tinggi berdasarkan spesifikasi mesin CNC yang digunakan. Berdasarkan mesin CNC DMC 635 V Eco parameter input kecepatan potong hasil metode optimasi ABC dan DE dibulatkan keatas dari 121.6944 menjadi 122 m/min, kecepatan makan dibulatkan dari 0.1178 menjadi 0.12 mm/gigi. Sedangkan pembulatan kebawah dari 121.6944 menjadi 121 m/min, kecepatan makan dibulatkan dari 0.1178 menjadi 0.11 dimana parameter kedalaman potong dan lebar pemotongan bernilai tetap yaitu 0.5 mm dan 0.2 mm.

Hasil dari perubahan nilai terhadap parameter input perlu dilakukan *training* menggunakan data *net* dari hasil prediksi BPNN yang sudah dilakukan pada subbab sebelumnya untuk mendapatkan prediksi parameter output berupa resultan gaya potong. Dikarenakan pada data *net* digunakan input yang ternormalisasi, maka parameter input hasil pembulatan perlu dilakukan normalisasi menggunakan persamaan 4.1 sebagai berikut:

##### 1. Kecepatan Potong

Nilai input kecepatan potong aktual untuk pembulatan keatas yaitu 122 sedangkan pembulatan kebawah yaitu 121. Pada parameter ini nilai aktual antara 100 hingga 140 akan dinormalisasi menjadi -1 hingga 1. Dengan menggunakan persamaan 4.1 untuk nilai aktual sebesar 122 dan 121 didapatkan nilai normalisasi sebagai berikut:

$$\begin{aligned}x_{122} &= \frac{122 - 100}{140 - 100} (1 - (-1)) + (-1) \\x_{122} &= \frac{22}{40} (2) - 1 \\x_{122} &= 0.1\end{aligned}$$

$$\begin{aligned}x_{121} &= \frac{121 - 100}{140 - 100} (1 - (-1)) + (-1) \\x_{121} &= \frac{21}{40} (2) - 1 \\x_{121} &= 0.05\end{aligned}$$

##### 2. Kecepatan Makan

Nilai input kecepatan makan aktual yaitu untuk pembulatan keatas yaitu 0.12 sedangkan pembulatan kebawah yaitu 0.11. Pada parameter ini nilai aktual antara 0.1 hingga 0.2 akan dinormalisasi menjadi -1 hingga 1. Dengan menggunakan persamaan 4.1 untuk nilai aktual sebesar 0.12 dan 0.11 didapatkan nilai normalisasi sebagai berikut:

$$\begin{aligned}x_{0.12} &= \frac{0.12 - 0.1}{0.2 - 0.1} (1 - (-1)) + (-1) \\x_{0.12} &= \frac{0.02}{0.1} (2) - 1 \\x_{0.12} &= -0.6\end{aligned}$$

$$\begin{aligned}x_{0.11} &= \frac{0.11 - 0.1}{0.2 - 0.1} (1 - (-1)) + (-1) \\x_{0.11} &= \frac{0.01}{0.1} (2) - 1 \\x_{0.11} &= -0.8\end{aligned}$$

### 3. Kedalaman Potong

Nilai input kedalaman potong aktual yaitu 0.5. Pada parameter ini nilai aktual antara 0.5 hingga 1 akan dinormalisasi menjadi -1 hingga 1. Sehingga untuk nilai aktual 0.5 didapat nilai normalisasi -1.

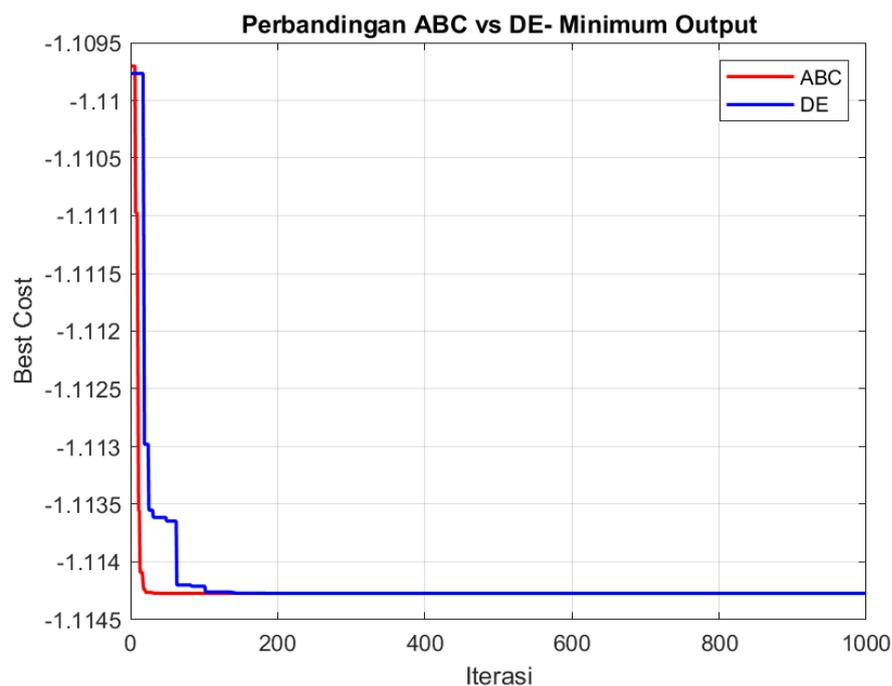
### 4. Lebar Pemotongan

Nilai input lebar pemotongan aktual yaitu 0.2. Pada parameter ini nilai aktual antara 0.2 hingga 1.8 akan dinormalisasi menjadi -1 hingga 1. Sehingga untuk nilai aktual 0.2 didapat nilai normalisasi -1.

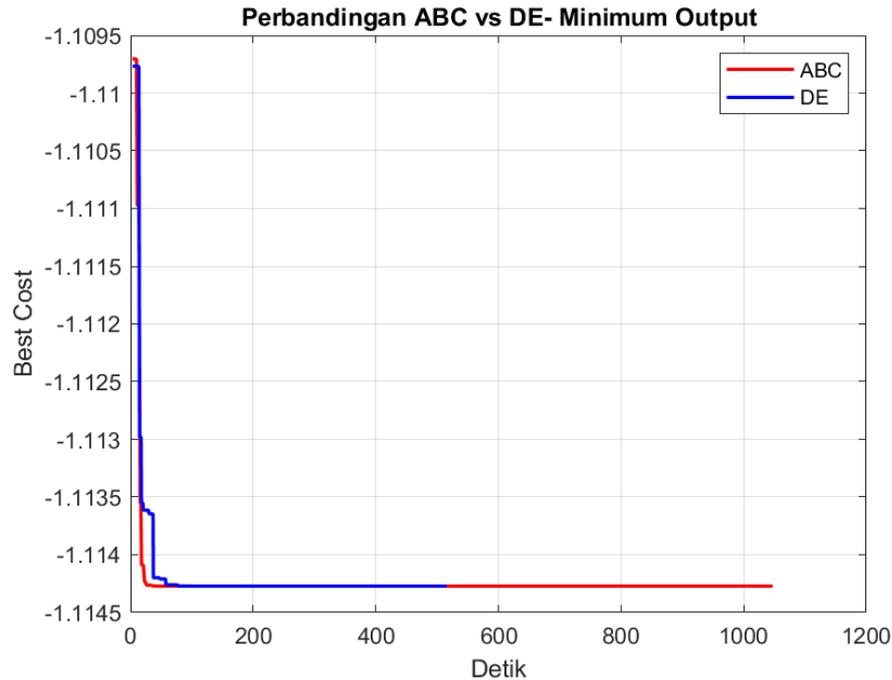
## 4.6 Perbandingan Hasil Optimasi Metode ABC dan DE

Untuk mendapatkan nilai parameter *end milling* yang paling optimal, maka diperlukan untuk mencari parameter input yang menghasilkan nilai resultan gaya potong paling kecil. Grafik pada gambar 4.6 menunjukkan hasil optimasi menggunakan metode ABC dan DE dengan nilai jumlah iterasi 1000 dan jumlah populasi 100. Pada metode ABC mencapai nilai konvergen pada *best cost* -1.1143 pada iterasi ke-54 sedangkan pada metode DE mencapai nilai konvergen pada *best cost* -1.1143 pada iterasi ke-243. Metode optimasi menggunakan ABC memiliki hasil nilai *best cost* yang sama dengan metode DE dan lebih cepat mencapai nilai konvergen terhadap waktu optimasi dan jumlah iterasi dibandingkan metode DE.

Hasil grafik pada gambar 4.7 menunjukkan hasil optimasi menggunakan metode ABC dan DE terhadap waktu optimasi pada perangkat lunak MATLAB R2020b. Pada metode ABC mencapai jumlah iterasi 1000 dalam waktu 1044.698 detik dan mencapai konvergensi pada detik ke-61 sedangkan pada metode DE mencapai jumlah iterasi 1000 dalam waktu 517.445 detik dan mencapai konvergensi pada detik ke-129.6. Untuk jumlah iterasi sebesar 1000, metode optimasi DE memiliki waktu yang lebih cepat dalam melakukan proses optimasi dibandingkan dengan metode ABC. Hal tersebut dikarenakan pada metode DE pencarian solusi yang dilakukan lebih menyebar dibandingkan dengan metode ABC sehingga proses untuk mencapai nilai solusi terbaik lebih cepat dilakukan.



**Gambar 4. 6** Perbandingan hasil optimasi ABC vs DE terhadap jumlah iterasi



**Gambar 4. 7** Perbandingan hasil optimasi ABC vs DE terhadap waktu

#### 4.7 Perbandingan dengan Penelitian Terdahulu

Hasil perbandingan antar metode optimasi yang digunakan untuk mendapat resultan gaya potong paling minimum terdapat pada tabel 4.4. Pada eksperimen yang menggunakan metode *Box-Behnken* dengan 29 data, nilai resultan gaya potong paling minimum didapat dengan konfigurasi parameter input kecepatan potong 120 m/min, kecepatan makan 0.15 mm/gigi, kedalaman potong 0.5 mm, dan lebar pemotongan 0.2 mm dimana gaya potong hasil eksperimen dengan parameter input tersebut didapatkan nilai 157.50 N. Pada metode RSM, parameter input paling optimal terdapat pada kecepatan potong 125.7 m/min, kecepatan makan 0.12 mm/gigi, kedalaman potong 0.52 mm, dan lebar pemotongan 0.22 mm dimana hasil prediksi resultan gaya potong didapatkan nilai 150.23 N (Kasim et al., 2015). Pada metode optimasi menggunakan ABC dan DE, parameter input yang menghasilkan output paling optimal bernilai sama dengan konfigurasi kecepatan potong 121.6944 m/min, kecepatan makan 0.1178 mm/gigi, kedalaman potong 0.5 mm, dan lebar pemotongan 0.2 mm dimana hasil prediksi resultan gaya potong didapatkan nilai 139.8909 N.

Hasil pembulatan optimasi metode ABC dan DE untuk parameter kecepatan potong sesuai keadaan sebenarnya yaitu 122 m/min untuk pembulatan keatas dan 121 m/min untuk pembulatan kebawah, parameter kecepatan makan 0.12 mm/gigi untuk pembulatan keatas dan 0.11 mm/gigi untuk pembulatan kebawah, kedalaman potong 0.5 mm, dan lebar pemotongan 0.2 mm. Berdasarkan nilai dari tabel 4.4 untuk mendapatkan parameter resultan gaya potong paling minimum sesuai dengan kondisi sebenarnya diperoleh dengan menggunakan parameter input kecepatan potong sebesar 122 m/min, kecepatan makan sebesar 0.12 mm/gigi, kedalaman potong 0.5 mm, dan lebar pemotongan 0.2 mm dengan nilai resultan gaya potong sebesar 139.9599 N.

**Tabel 4. 4** Perbandingan hasil optimasi dengan jurnal (Kasim et al., 2015)

Metode	Parameter				Output
	Kecepatan potong, $V_c$ (mm/min)	Kecepatan makan, $f_z$ (mm/gigi)	Kedalaman potong, $a_p$ (mm)	Lebar pemotongan, $a_e$ (mm)	Resultan gaya potong, $F_r$ (N)
Eksperimen ( <i>Box-Behnken</i> )	120	0.15	0.5	0.2	157.50
RSM	125.7	0.12	0.52	0.22	150.23
ABC	121.6944	0.1178	0.5	0.2	139.8909
DE	121.6944	0.1178	0.5	0.2	139.8909
Pembulatan ABC dan DE	122	0.12	0.5	0.2	139.9599
	121	0.12	0.5	0.2	140.1051
	122	0.11	0.5	0.2	140.9553
	121	0.11	0.5	0.2	141.0043

Hasil prediksi optimasi resultan gaya potong menggunakan metode optimasi BPNN-ABC dan BPNN-DE ketika dibulatkan sesuai keadaan sebenarnya didapat nilai parameter resultan gaya potong sebesar 139.9599 N, sedangkan pada penelitian sebelumnya menggunakan metode RSM didapat nilai resultan gaya potong sebesar 150.23 N (Kasim et al., 2015). Hasil tersebut menunjukkan nilai prediksi resultan gaya potong berdasarkan metode optimasi BPNN-ABC dan BPNN-DE lebih kecil 6.8% dibandingkan dengan metode RSM pada penelitian sebelumnya.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil yang telah didapatkan dari penelitian tugas akhir ini, kesimpulan yang dapat diambil adalah sebagai berikut:

1. Berdasarkan hasil prediksi menggunakan metode *Backpropagation Neural Network* (BPNN) konfigurasi yang menghasilkan kesalahan prediksi paling kecil yaitu BPNN dengan jumlah *hidden layer* 2, jumlah neuron 10 pada *hidden layer* pertama dengan fungsi aktivasi *tansig*, jumlah neuron 10 pada *hidden layer* kedua dengan fungsi aktivasi *tansig* dengan nilai MSE sebesar 0.00096269.
2. Berdasarkan hasil optimasi metode *Artificial Bee Colony* (ABC) dan *Differential Evolution* (DE), parameter yang menghasilkan nilai optimal bernilai sama yaitu kecepatan potong 122 m/min, kecepatan makan 0.12 mm/gigi, kedalaman potong 0.5 mm, dan lebar pemotongan 0.2 mm dengan output resultan gaya potong sebesar 139.9599 N.
3. Berdasarkan hasil optimasi resultan gaya potong menggunakan metode optimasi BPNN-ABC dan BPNN-DE sebesar 139.9599 N, lebih kecil 6.8% dibandingkan dengan penelitian sebelumnya yang menggunakan metode RSM sebesar 150.23 N.

#### **5.2 Saran**

Adapun saran yang dapat diberikan setelah melakukan penelitian tugas akhir ini sebagai evaluasi pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Diperlukan adanya uji validasi yang sesuai dengan data eksperimen dengan perangkat lunak ANSYS ataupun ABAQUS sebagai pembanding dari hasil yang didapatkan.
2. Diperlukan adanya uji konfirmasi terhadap hasil optimasi yang didapatkan sehingga bisa dibandingkan dengan data yang lain.
3. Untuk penelitian menggunakan BPNN diperlukan untuk menggunakan data *training* dalam jumlah yang besar agar hasil prediksi semakin akurat.
4. Untuk penelitian selanjutnya, dapat menggunakan bentuk metode optimasi selain BPNN-ABC dan BPNN-DE sehingga mendapatkan kualitas perbandingan hasil optimasi.

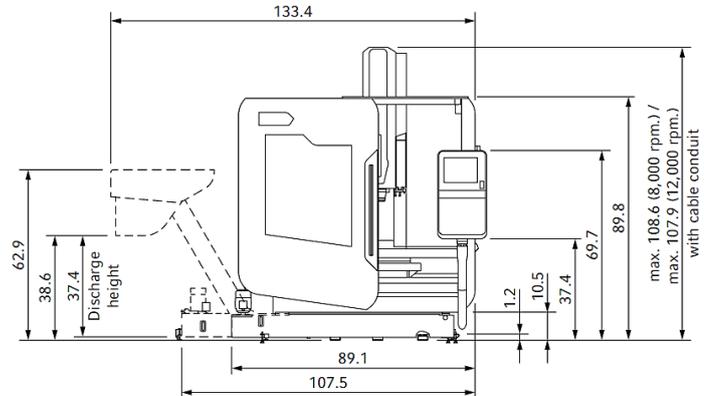
## DAFTAR PUSTAKA

- Akhyar, G., Hamni, A., & Akmal, J. (2016). Keausan Mata Pahat Karbida pada Pemesinan Inconel 718 Menggunakan Pelumas Berkwantitas Minimum. *Seminar Nasional Tahunan Teknik Mesin XV (SNTTM XV)*, TP-002.
- Dikshit, M. K., Puri, A. B., Maity, A., & Banerjee, A. J. (2014). Analysis of Cutting Forces and Optimization of Cutting Parameters in High Speed Ball-end Milling Using Response Surface Methodology and Genetic Algorithm. *Procedia Materials Science*, 5, 1623-1632.
- Hossain, M. S. (2017). Cutting Parameter Optimization for End Milling Operation Using Advanced Metaheuristic Algorithms. *IJARA*, 2, 1-12.
- Kalpakjian, S., Schmid, S., & Vijay. (2013). *Manufacturing Engineering and Technology*.
- Kamaruddin, S., Nor, A., Saad, M., Zakaria, M., & Baharudin, M. (2019). Optimization of surface roughness on CNC milling machining using differential evolution (DE) method. *AIP Conference Proceedings*, 2129.
- Kasim, M., Haron, C., Ghani, J., Mohamad, N., Izamshah, R., Minhat, M., & Saad, N. (2015). Prediction of Cutting Force in End Milling of Inconel 718. *Journal of Engineering and Technology*, 63-70.
- Lestari, E., & Isnina, W. (2017). Analisis Kinerja Manufaktur di Indonesia. *Jurnal Riset Ekonomi dan Manajemen*, 17.
- Luo, J., Hongtao, S., & Albert. (2005). Induction-Heated Tool Machining of Elastomers— Part 2: Chip Morphology, Cutting Forces, and Machined Surfaces. *Machining Science and Technology*.
- Man, X., Ren, D., Usui, S., Johnson, C., & Marusich, T. D. (2012). Validation of Finite Element Cutting Force Prediction for End Milling. *Procedia CIRP*, 1, 663-668.
- Öztürk, O., Kalyoncu, M., & Ünüvar, A. (2018). Multi Objective Optimization of Cutting Parameters in a Single Pass Turning Operation Using the Bees Algorithm.
- Rao, K. M., Kumar, V. D., Shekar, C. K., & Singaravel, B. (2021). Optimization of EDM process parameters using TOPSIS for machining AISI D2 steel material. *Materials Today: Proceedings*, 46, 701-706.
- Rochim, T. (1993). *Proses Permesinan*. Bandung: Institut Teknologi Bandung.
- Sodhi, H. (2014). Optimization of end milling process for d2 (die steel) by using Response surface methodology. *17*, 73-78.
- Yang, L., Sun, X., & Li, Z. (2019). An Efficient Framework for Remote Sensing Parallel Processing: Integrating the Artificial Bee Colony Algorithm and Multiagent Technology. *Remote Sensing*.

- Yusup, N., Sarkheyli, A., Zain, A. M., Hashim, S. Z., & Ithnin, N. (2014). Estimation of optimal machining control parameters using artificial bee colony. *J Intell Manuf*, 25, 1463-1472.
- Zenghui, A., Fu, X., Pan, Y., & Tang, A. (2014). An Experiment-Based Investigation on Characteristic and Model of Milling Forces during End-Milling Aluminum Alloy. *Applied Mechanics and Materials*, 494-495.

## LAMPIRAN

### Lampiran 1 : Spesifikasi CNC



<b>DMC 635 V ecoline</b>		
<b>Work area</b>		
Travel in the X-axis	in.	25.0
Travel in the Y-axis	in.	20.1
Travel in the Z-axis	in.	18.1
<b>Spindle</b>		
Speed range, max.	rpm.	8,000 / 12,000*
Drive power (40 / 100 % DC) 8,000 / 12,000*	hp.	17.4 / 12.1
Torque (40 / 100 % DC) 8,000 / 12,000*	ft./lbs.	61.2 / 42.0
Rapid traverse in X / Y / Z	ipm.	1,181.1
Feed force, max.	lbf.	1,124.0
Feed rate	ipm.	944.9
<b>Positioning accuracy</b>		
Resolution in X / Y / Z (in- / direct measuring system)	in.	0.00004
P <sub>max</sub> according to DGQ 3441 (in- / direct measuring system)	in.	0.0008 / 0.0003
P <sub>max</sub> according to JIS B6330-1980 (in- / direct measuring system)	in.	0.0004 / 0.0002
<b>Tool magazine / changer</b>		
Number of tools		20 / 30*
Tool weight, max.	lbs.	13.2
Tool length, max.	in.	11.8
Tool diameter, max.	in.	ø 3.1
Tool diameter with vacant adjacent pockets, max.	in.	ø 5.1
Chip-to-chip time	sec.	5
<b>Table design</b>		
Setup area with T-grooves	in.	31.1 × 22.0
Loading height (upper edge of table)	in.	28.3
Table load (assembly weight)	lbs.	1,322.8
<b>Machine weight / connection values</b>		
Machine weight with 20-pocket tool magazine and a chip tray	lbs.	8,487.8
I <sub>n</sub> max. by 100 % DC	a	28
Power consumption at 100 % DC	kVA	17
Backup fuse, max.	a	35
<b>Controls</b>		
SLIMline® with Operate 4.5 on a SIEMENS 840D solutionline control with ShopMill		•
SLIMline® with a HEIDENHAIN TNC 620 control and plain text dialogue		•
MAPPS IV on a MITSUBISHI control		◦

\* Optional, • Standard, ◦ Optional

Lampiran 2 : Spesifikasi *Dynamometer*



**Technical data**

Range	$F_x, F_y, F_z$	kN	$-5 \dots 5^{1)}$
$F_z$ for $F_x$ and $F_y \leq 0,5 F_z$	$F_z$	kN	$-5 \dots 10^{2)}$
Calibrated partial range 1	$F_x, F_y$	N	0 ... 500
	$F_z$	N	0 ... 1 000
Calibrated partial range 2	$F_x, F_y$	N	0 ... 50
	$F_z$	N	0 ... 100
Overload	$F_x, F_y, F_z$	kN	$-7,5/7,5$
	$F_z$ for $F_x$ and $F_y \leq 0,5 F_z$	kN	$-7,5/15$
Threshold		N	$<0,01$
Sensitivity	$F_x, F_y$	pC/N	$\approx -7,5$
	$F_z$	pC/N	$\approx -3,7$
Linearity, all ranges		%FSO	$\leq \pm 1$
Hysteresis, all ranges		%FSO	$\leq 0,5$
Cross talk		%	$\leq \pm 2$
Rigidity	$c_x, c_y$	kN/ $\mu$ m	$>1$
	$c_z$	kN/ $\mu$ m	$>2$
Natural frequency	$f_n(x, y, z)$	kHz	$\approx 3,5^{4)}$
Natural frequency (mounted on flanges)	$f_n(x, y)$	kHz	$\approx 2,3^{4)}$
	$f_n(z)$	kHz	$\approx 3,5^{4)}$
Operating temperature range		$^{\circ}$ C	0 ... 70
Capacitance	$F_x, F_y, F_z$	pF	$\approx 220$
Insulation resistance (20 $^{\circ}$ C)		$\Omega$	$>10^{13}$
Ground insulation		$\Omega$	$>10^8$
Protection class EN60529		–	IP67 <sup>3)</sup>
Weight		kg	7,3
Clamping area	mm	100x170	
Connection		Fischer flange, 9 pol. neg.	

Lampiran 3 : Spesifikasi *Charge Amplifier*



**Technical Data**

**Charge Amplifier**

Number of channels		4
Option		8
Connector type		BNC neg.
Option		Fischer 9-pole neg.
Measuring range FS	pC	±200 ... 200 000
Option	pC	±600 ... 600 000
Measurement uncertainty (0 ... 50 °C) typ./max.	%	<±0,3/<±1
Drift, measuring mode DC (Long) at 25 °C, max. relative humidity RH of 60 % (non-condensing)	pC/s	<±0,05 (typ. <±0,03)
at 25 °C, max. relative humidity RH of 70 % (non-condensing)	pC/s	typ. <±0,05
at 50 °C, max. relative humidity RH of 50 % (non-condensing)	pC/s	<±0,2
Frequency range (20 Vpp)	kHz	=0 ... >45
Group delay	µs	<9

**Voltage Output**

Connector type		D-Sub 15f
Output voltage	V	±10
Output current	mA	<±2
Output resistance	Ω	10
Reset-measure transition	pC	<±2
Zero point error (Reset)	mV	<±10
Output interference signal (0,1 Hz ... 1 MHz)	mVpp	<10

Lampiran 4 : Konfigurasi BPNN

layer	neuron	fungsi	Min. MSE	layer	neuron	fungsi	Min. MSE
1	2	hardlim	0.25097274	2	2	hardlim	0.26438577
1	2	hardlims	0.25688153	2	2	hardlims	0.26523936
1	2	purelin	0.0272604	2	2	purelin	0.02796305
1	2	satlin	0.02361127	2	2	satlin	0.00535229
1	2	logsig	0.02182133	2	2	logsig	0.00381665
1	2	tansig	0.00323875	2	2	tansig	0.00193951
1	3	hardlim	0.20019511	2	3	hardlim	0.25697648
1	3	hardlims	0.20504721	2	3	hardlims	0.23204488
1	3	purelin	0.02953093	2	3	purelin	0.02735677
1	3	satlin	0.04972751	2	3	satlin	0.00495775
1	3	logsig	0.00280755	2	3	logsig	0.00327656
1	3	tansig	0.00196457	2	3	tansig	0.00280302
1	4	hardlim	0.24570015	2	4	hardlim	0.26333728
1	4	hardlims	0.15519698	2	4	hardlims	0.16829442
1	4	purelin	0.02771582	2	4	purelin	0.02846432
1	4	satlin	0.0027476	2	4	satlin	0.01242257
1	4	logsig	0.00238836	2	4	logsig	0.00489525
1	4	tansig	0.00312662	2	4	tansig	0.00225816
1	5	hardlim	0.13142966	2	5	hardlim	0.2504483
1	5	hardlims	0.15204857	2	5	hardlims	0.22868922
1	5	purelin	0.02872628	2	5	purelin	0.02760858
1	5	satlin	0.00295334	2	5	satlin	0.0093087
1	5	logsig	0.00461949	2	5	logsig	0.00293087
1	5	tansig	0.00206123	2	5	tansig	0.06548666
1	6	hardlim	0.08630667	2	6	hardlim	0.22395751
1	6	hardlims	0.1340423	2	6	hardlims	0.17925214
1	6	purelin	0.02868259	2	6	purelin	0.02739721
1	6	satlin	0.0039921	2	6	satlin	0.00293087
1	6	logsig	0.00129431	2	6	logsig	0.00122698
1	6	tansig	0.00139652	2	6	tansig	0.01056768
1	7	hardlim	0.11489651	2	7	hardlim	0.09009788
1	7	hardlims	0.15392517	2	7	hardlims	0.195078
1	7	purelin	0.02789841	2	7	purelin	0.02783269
1	7	satlin	0.00165285	2	7	satlin	0.00150506
1	7	logsig	0.00422137	2	7	logsig	0.01214021
1	7	tansig	0.0041997	2	7	tansig	0.00393648
1	8	hardlim	0.08193325	2	8	hardlim	0.13501075
1	8	hardlims	0.11743953	2	8	hardlims	0.18318782
1	8	purelin	0.02749797	2	8	purelin	0.02714672
1	8	satlin	0.01050933	2	8	satlin	0.00294117
1	8	logsig	0.0233053	2	8	logsig	0.00486247

1	8	tansig	0.00294419	2	8	tansig	0.00119609
1	9	hardlim	0.10772024	2	9	hardlim	0.21830126
1	9	hardlims	0.16424561	2	9	hardlims	0.09882661
1	9	purelin	0.02934422	2	9	purelin	0.02721817
1	9	satlin	0.00360525	2	9	satlin	0.00265584
1	9	logsig	0.00679879	2	9	logsig	0.00348419
1	9	tansig	0.01080725	2	9	tansig	0.00673035
1	10	hardlim	0.05389477	2	10	hardlim	0.13844156
1	10	hardlims	0.10093923	2	10	hardlims	0.13682787
1	10	purelin	0.02724273	2	10	purelin	0.02858095
1	10	satlin	0.00157364	2	10	satlin	0.00274007
1	10	logsig	0.00155648	2	10	logsig	0.00414749
1	10	tansig	0.010087	2	10	tansig	0.00096269

layer	neuron	fungsi	Min. MSE	layer	neuron	fungsi	Min. MSE
3	2	hardlim	0.27504775	4	2	hardlim	0.27379947
3	2	hardlims	0.27711386	4	2	hardlims	0.27706153
3	2	purelin	0.02704231	4	2	purelin	0.02822793
3	2	satlin	0.04226071	4	2	satlin	0.09777798
3	2	logsig	0.02121825	4	2	logsig	0.00248394
3	2	tansig	0.08259017	4	2	tansig	0.00413293
3	3	hardlim	0.24597755	4	3	hardlim	0.2115063
3	3	hardlims	0.22745326	4	3	hardlims	0.27453188
3	3	purelin	0.02752509	4	3	purelin	0.02946877
3	3	satlin	0.14630793	4	3	satlin	0.01469397
3	3	logsig	0.00510665	4	3	logsig	0.01454007
3	3	tansig	0.01127297	4	3	tansig	0.03349936
3	4	hardlim	0.27494007	4	4	hardlim	0.27707314
3	4	hardlims	0.19537312	4	4	hardlims	0.24793502
3	4	purelin	0.02819222	4	4	purelin	0.02889997
3	4	satlin	0.02368813	4	4	satlin	0.09880172
3	4	logsig	0.00468619	4	4	logsig	0.01138581
3	4	tansig	0.00348563	4	4	tansig	0.00351732
3	5	hardlim	0.26449854	4	5	hardlim	0.24216148
3	5	hardlims	0.15349648	4	5	hardlims	0.25281092
3	5	purelin	0.02759788	4	5	purelin	0.0272776
3	5	satlin	0.0068649	4	5	satlin	0.0143167
3	5	logsig	0.0060417	4	5	logsig	0.00167247
3	5	tansig	0.00759516	4	5	tansig	0.01429541
3	6	hardlim	0.22339223	4	6	hardlim	0.11151315
3	6	hardlims	0.2271962	4	6	hardlims	0.17275743
3	6	purelin	0.02895242	4	6	purelin	0.02824295
3	6	satlin	0.00537593	4	6	satlin	0.00794334
3	6	logsig	0.01305943	4	6	logsig	0.01084223
3	6	tansig	0.00251595	4	6	tansig	0.01218426

3	7	hardlim	0.14959217	4	7	hardlim	0.18598608
3	7	hardlims	0.17404636	4	7	hardlims	0.27096151
3	7	purelin	0.02852737	4	7	purelin	0.02903701
3	7	satlin	0.01439357	4	7	satlin	0.00188402
3	7	logsig	0.00903273	4	7	logsig	0.01766245
3	7	tansig	0.00701277	4	7	tansig	0.00373633
3	8	hardlim	0.16786935	4	8	hardlim	0.19414318
3	8	hardlims	0.22831294	4	8	hardlims	0.26121475
3	8	purelin	0.02741554	4	8	purelin	0.0281036
3	8	satlin	0.00162005	4	8	satlin	0.00217855
3	8	logsig	0.0063718	4	8	logsig	0.00379072
3	8	tansig	0.0448221	4	8	tansig	0.01562263
3	9	hardlim	0.13832666	4	9	hardlim	0.27198491
3	9	hardlims	0.18870357	4	9	hardlims	0.26909714
3	9	purelin	0.02761756	4	9	purelin	0.02785898
3	9	satlin	0.00590899	4	9	satlin	0.00285094
3	9	logsig	0.02349699	4	9	logsig	0.01010176
3	9	tansig	0.00580811	4	9	tansig	0.00212288
3	10	hardlim	0.18068036	4	10	hardlim	0.18879325
3	10	hardlims	0.10019758	4	10	hardlims	0.17411676
3	10	purelin	0.02749603	4	10	purelin	0.0273618
3	10	satlin	0.00266604	4	10	satlin	0.00455603
3	10	logsig	0.00216006	4	10	logsig	0.03073462
3	10	tansig	0.00223276	4	10	tansig	0.01069889

layer	neuron	fungsi	Min. MSE
5	2	hardlim	0.27724383
5	2	hardlims	0.24226281
5	2	purelin	0.0278167
5	2	satlin	0.27757863
5	2	logsig	0.01340006
5	2	tansig	0.00364506
5	3	hardlim	0.16903861
5	3	hardlims	0.27440212
5	3	purelin	0.02765802
5	3	satlin	0.114122
5	3	logsig	0.02052193
5	3	tansig	0.0226872
5	4	hardlim	0.20992254
5	4	hardlims	0.27702905
5	4	purelin	0.0283315
5	4	satlin	0.10084869
5	4	logsig	0.03079095
5	4	tansig	0.0063941
5	5	hardlim	0.2160322

5	5	hardlims	0.27855333
5	5	purelin	0.02719146
5	5	satlin	0.277096
5	5	logsig	0.00294708
5	5	tansig	0.06372537
5	6	hardlim	0.23227415
5	6	hardlims	0.17752449
5	6	purelin	0.0292728
5	6	satlin	0.00636956
5	6	logsig	0.01633247
5	6	tansig	0.00393887
5	7	hardlim	0.24127065
5	7	hardlims	0.26989363
5	7	purelin	0.0276463
5	7	satlin	0.00430122
5	7	logsig	0.00478694
5	7	tansig	0.01304107
5	8	hardlim	0.28012736
5	8	hardlims	0.19903949
5	8	purelin	0.03004459
5	8	satlin	0.00321722
5	8	logsig	0.01301323
5	8	tansig	0.00273085
5	9	hardlim	0.20968596
5	9	hardlims	0.21311699
5	9	purelin	0.02878133
5	9	satlin	0.00109505
5	9	logsig	0.03404666
5	9	tansig	0.00102241
5	10	hardlim	0.34859044
5	10	hardlims	0.14220638
5	10	purelin	0.02746889
5	10	satlin	0.00179828
5	10	logsig	0.0042844
5	10	tansig	0.01079007

## Lampiran 5 : Nilai *Best Function*

```
function [Y,Xf,Af] = fungsibpnn(X,~,~)
%FUNGSIBPNN neural network simulation function.
%
% Auto-generated by MATLAB, 21-Jul-2022 00:41:49.
%
% [Y] = fungsibpnn(X,~,~) takes these arguments:
%
%   X = 1xTS cell, 1 inputs over TS timesteps
%   Each X{1,ts} = 4xQ matrix, input #1 at timestep ts.
%
% and returns:
%   Y = 1xTS cell of 1 outputs over TS timesteps.
%   Each Y{1,ts} = 1xQ matrix, output #1 at timestep ts.
%
% where Q is number of samples (or series) and TS is the number of
timesteps.

%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
x1_step1.xoffset = [-1;-1;-1;-1];
x1_step1.gain = [1;1;1;1];
x1_step1.ymin = -1;

% Layer 1
b1 = [-
2.785745713856684791;1.8687721162491450766;1.6119266934669083113;0.4
5393783360992345788;-0.11956666090027992522;-
0.33078037549294919684;-
0.59739715701534812364;1.3691495347148570083;-
1.9437938499068028175;2.4147260713897056306];
IW1_1 = [1.1023642121495276225 -0.54726837406552675258
1.6957388404081090716 0.10269681478615420855;-1.3091935446038673874
0.47931928141044038982 -1.3912189418034559552
1.6068176024473588281;-1.2837790091190526276 -1.0818669553076925016
0.57997839171867304753 1.4570559433068261956;-1.8543359356852553432
1.1127925868802948095 0.9762585589940769859
0.92214757943045466693;1.7522066257210853202 1.3501892726003503764
0.81190209120454370773 -0.90510719527069860568;-
0.28893234089079317117 -0.47847577187536033394 -
1.4583905146086983962 -1.8884394311431105251;-0.37235695826240922113
-1.4032598929624158224 -0.71422077557063423647
1.9262007486625321029;1.4227784170202479874 -1.4832365607180177935
1.1676079695086736265 1.0910033188404510973;-1.3945931990479265572
0.54814155089278437938 -1.4574174356758893278 -
1.3351100608777834999;1.485340642196322225 -0.96058778588189375203
1.9083430330248718487 0.45840424240770843989];

% Layer 2
```

```

b2 = [1.740551514696056401;-
1.4837953068427531811;1.2733422791358692816;0.68022268542577746242;-
0.11534264068897293054;0.0084329001205619754516;0.397322889906380927
18;-0.94448165409639095458;-
1.2424156778836361603;1.8303251645957852034];
LW2_1 = [-0.56478239497811699721 -0.30388946156424623268
0.95633601567136516941 -0.33808102943318318889
0.55586126743179697929 0.20776532506374156828 0.80063419268014757169
-0.55458584507746377934 0.57082710407011338916
0.57028221066901130243;0.76558060297297980679 -
0.89376785727049923302 0.1613806828051642106 -0.29675248220265976196
-0.68643006828111929885 0.020951968354169080849 -
0.38406794931259075065 0.61478247239241012867 -0.4467414696970083976
-0.06764631947342490681;-0.3800177053930565152 -
0.65254488341508698479 -0.20781044142561447274 -
0.4432117341203162808 -0.83789417196003113109 0.84941246625375987911
-0.054955178166212440005 -0.38877742779336338375 -
0.71159989597300365993 -0.27412019379698659982;-
0.74121374017913488341 0.037256430703194710896
0.78076736203970398265 0.10855633884386471644 0.66560013624522018638
0.66549195140364003631 -1.0301433230596657165 0.41813390222575058663
0.30788999513373471739 0.18882962615603501866;0.58753626662026836591
1.2700894650112473094 0.36941625108008441236 -
0.0072847198510349216172 0.96857780341226629162 -
0.04866672826575415417 0.073392756139607953636
0.36719680327944070886 -0.33445877586906425094 -
0.65572004338776035404;0.90056730932306006121 -
0.41746538861225457318 0.23974793246816530901 0.524503251338619636
0.86085869078555010692 0.29618080374863287041 -
0.26249790208906725875 1.0039363755687096269 -
0.0019801485665011360829 -
0.39553573934643837529;0.64558256864023710886 -
0.29609314874596037725 -0.37977809727621392177 -
0.095302798098869490051 -0.87624374927265991087 -
1.1078779693098443904 0.13477676722263193243 -0.50239051322365224639
0.43178200370125202623 0.30469942442942715699;-
0.11877162738190354119 0.29967850169409704364 -
0.13865636938566372471 -0.98444131995506956034
0.54543273353377097479 -0.48459321918737707779 -
0.6293034609396975787 0.60783064260098085274 0.59025205502230959631
0.86043377693369860193;-0.16086814414505201665 -
0.6032403629376140719 -0.50008880209969941433 0.80403708606968848915
-0.67607361961144163409 -0.90655226165150437634
0.1696035308090243221 -0.51468837505054276793 -
0.12269405518118707554 -
0.24103401673105182001;0.47076115475502150476 0.4916711061516352288
-0.078284133109865566835 0.19512098612351319593 -
0.13373354543165091868 -0.7862052609674989867 0.72913571955054545803
-0.57492247859127476595 -0.6499957158932210266
0.85173068016926656298];

```

```

% Layer 3

```

```

b3 = 1.0481546336686520249;
LW3_2 = [-0.75502789629898281465 0.2274721225580623174 -
0.51369840868183469507 0.60197389232996911801 0.42148785893005563663

```

```

0.24719284252537188085 0.18108538819910091622 -
0.17683332290074560422 0.64578074233056192988
0.21289752614570200961];

% Output 1
y1_step1.ymin = -1;
y1_step1.gain = 1;
y1_step1.xoffset = -1;

% ===== SIMULATION =====

% Format Input Arguments
isCellX = iscell(X);
if ~isCellX
    X = {X};
end

% Dimensions
TS = size(X,2); % timesteps
if ~isempty(X)
    Q = size(X{1},2); % samples/series
else
    Q = 0;
end

% Allocate Outputs
Y = cell(1,TS);

% Time loop
for ts=1:TS

    % Input 1
    Xp1 = mapminmax_apply(X{1,ts},x1_step1);

    % Layer 1
    a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*Xp1);

    % Layer 2
    a2 = tansig_apply(repmat(b2,1,Q) + LW2_1*a1);

    % Layer 3
    a3 = repmat(b3,1,Q) + LW3_2*a2;

    % Output 1
    Y{1,ts} = mapminmax_reverse(a3,y1_step1);
end

% Final Delay States
Xf = cell(1,0);
Af = cell(3,0);

% Format Output Arguments
if ~isCellX
    Y = cell2mat(Y);
end

```

```

end

% ===== MODULE FUNCTIONS =====

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings)
    y = bsxfun(@minus,x,settings.xoffset);
    y = bsxfun(@times,y,settings.gain);
    y = bsxfun(@plus,y,settings.ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n,~)
    a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y,settings)
    x = bsxfun(@minus,y,settings.ymin);
    x = bsxfun(@rdivide,x,settings.gain);
    x = bsxfun(@plus,x,settings.xoffset);
end

```

## BIODATA PENULIS



**Najibul Kaffin**, anak ketujuh dari tujuh bersaudara, lahir di Gresik, 29 Oktober 1998. Penulis telah menempuh pendidikan formal di RA YKUI Sambogunung, MI YKUI Sambogunung, MTs. YKUI Maskumambang, dan MAN Insan Cendekia Gorontalo. Setelah lulus dari jenjang SMA pada tahun 2017, penulis mengikuti SBMPTN dan diterima di jurusan Teknik Mesin FTIRS ITS pada tahun 2018 dan terdaftar dengan NRP 02111840000130.

Semasa kuliah, penulis mengambil bidang studi Manufaktur. Penulis sempat aktif dalam berbagai kegiatan berskala departemen yang diselenggarakan oleh Departemen Teknik Mesin FTIRS ITS. Penulis pernah menjadi staf bidang kaderisasi serta wakil kepala bidang pelayanan umat lembaga dakwah LKKI Ash-Shaff Teknik Mesin. Apabila terdapat pesan atau informasi yang hendak disampaikan kepada penulis, dapat melalui email: [najibulkaffin2@gmail.com](mailto:najibulkaffin2@gmail.com)