

PREDIKSI KEAUSAN MATA BOR MESIN CNC *DRILLING* MENGGUNAKAN PEMODELAN PREDIKTIF BERBASIS *CUTTING SPEED* DAN *FEEDING RATE*

Muhammad Fatih A A, Nur Iksan

Program Studi Teknik Elektro, Fakultas Teknik,

Universitas Negeri Semarang

fatih.azhari@mail.unnes.ac.id

DOI: <https://doi.org/10.15294/bte.v1i1.314>

QRCBN 62-6861-4195-455

ABSTRAK

Proses permesinan dalam industri manufaktur bergantung pada performa alat potong untuk menghasilkan produk dengan dimensi presisi tinggi. Selama proses berlangsung, alat potong mengalami keausan akibat gesekan terus-menerus dengan benda kerja. Keausan tersebut bersifat progresif dan sulit dimonitor secara langsung. Oleh karena itu, pendekatan prediktif dikembangkan sebagai alternatif pemantauan berbasis parameter proses. Dua parameter utama yang memengaruhi keausan adalah *cutting speed* dan *feeding rate*. Pemodelan prediktif dilakukan dengan memanfaatkan pendekatan *Adaptive Neuro-Fuzzy Inference System* (ANFIS) dan *Multiple Linear Regression* (MLR). Keduanya diimplementasikan dalam lingkungan simulasi MATLAB, dengan input berupa *cutting speed* dan *feeding rate*, serta *output* berupa *flank wear*. Model ANFIS menggunakan sistem fuzzy Sugeno berbasis aturan *if-then* yang dikombinasikan dengan jaringan saraf tiruan, sedangkan MLR mengandalkan persamaan regresi linier multipel. Tingkat akurasi evaluasi ditentukan menggunakan nilai *Root Mean Square Error* (RMSE). Hasil simulasi menunjukkan bahwa ANFIS memberikan prediksi yang lebih akurat, namun stabilitasnya lebih rendah dibanding MLR. Sementara itu,

MLR tetap dapat diandalkan untuk data dengan variabel terbatas dan hubungan linier antar parameter. Model prediktif ini menunjukkan bahwa keausan mata bor dapat diestimasi dengan baik hanya melalui dua parameter proses utama.

Kata Kunci: *flank wear*, pemodelan prediktif, regresi linear, simulasi MATLAB, sistem

PENDAHULUAN

Peralatan elektronik sudah banyak digunakan manusia untuk menyelesaikan berbagai pekerjaan sehari-hari. Perkembangan teknologi yang sangat cepat menggantikan sistem kendali peralatan yang dilakukan secara manual oleh pengguna menjadi sistem kendali otomatis dengan komputer terutama pada bidang industri dan manufaktur, salah satu contohnya adalah penggunaan mesin CNC untuk peralatan manufaktur yang dapat bekerja secara otomatis. Pengendalian alat dengan komputer dapat memberikan hasil dengan tingkat presisi yang tinggi. Dengan ditemukannya sistem kendali alat secara otomatis seperti mesin CNC dapat memberikan dampak dalam bidang manufaktur baik dari segi kualitas produksi maupun dari segi ekonomi/biaya.

Mesin CNC bekerja dengan mengendalikan alat menggunakan perangkat komputer yang telah diprogram. Pada industri manufaktur, mesin CNC digunakan untuk membuat suatu bentuk tertentu dengan ukuran yang presisi. Mesin dijalankan dengan suatu prosedur untuk *drilling* (pengeboran), *spreading* (penyebaran), *weaning*, dan *threading* (penguliran). Mesin CNC memiliki berbagai jenis bergantung dengan proses yang dilakukan, salah satunya adalah pengeboran atau *drilling*. Proses permesinan *drilling* memiliki peran penting pada proses manufaktur saat ini untuk membuat banyak lubang dengan ukuran presisi.

Pada proses permesinan, mata bor bergesekan dengan material untuk membuat lubang, gesekan yang terjadi terus menerus dapat menyebabkan kenaikan suhu. Gesekan akan mengubah sifat material dari mata bor dan jika terus menerus akan terjadi keausan pada mata bor. Mata bor yang aus akan

berakibat pada hasil bentuk yang tidak presisi. Keausan pada mata bor perlu untuk dimonitor sehingga dapat dilakukan tindakan ketika nilai keausan mencapai tingkat tertentu. Nilai keausan akan terus berubah seiring dengan proses permesinan yang dilakukan. Keausan ini seringkali sulit untuk dimonitor secara langsung karena terjadi secara bertahap dan juga sulit untuk diamati secara langsung. Sebuah model dapat digunakan untuk memprediksi nilai keausan menggunakan parameter dari proses permesinan.

Pendekatan dengan model prediksi dapat dilakukan dengan menggunakan parameter permesinan yang dapat diukur menggunakan sensor untuk memprediksi keausan berdasarkan pengaruh parameter terhadap keausan. Parameter *cutting speed* yang dapat diukur berdasarkan kecepatan putaran *spindle* menggunakan sensor rotary dan parameter *feeding rate* dapat dihitung berdasarkan gaya tekan *feed (Feed Force)* yang dapat diukur menggunakan *load cell*. Dari data pengukuran oleh sensor dapat di hitung parameter *cutting speed* dan *feeding rate* sebagai input model dalam memprediksi *flank wear*.

Pengalokasian ANFIS dan ANN menghasilkan performa untuk memodelkan hubungan kompleks dan non-linier antara prediksi parameter permesinan dengan tingkat keausan pahat. Permasalahan utama yang sering dihadapi adalah tingkat keausan yang sulit untuk dimonitor secara langsung, sehingga perlu suatu desain model sistem untuk membuat prediksi keausan dari mata bor berdasarkan pada parameter yang umum digunakan pada mesin pengeboran.

A. MESIN CNC DRILLING

Proses permesinan seperti pembubutan (*Turning*), pengeboran (*Drilling*), dan penggerindaan (*Grinding*) adalah metode untuk menghilangkan material yang telah digunakan secara luas dalam industri manufaktur sejak revolusi industri. Dalam waktu terakhir, otomatisasi penuh pada mesin-mesin perkakas menjadi sangat penting di industri manufaktur. Hal ini terjadi karena teknologi permesinan telah berkembang pesat, dari mesin yang dioperasikan secara manual menjadi pusat mesin CNC yang canggih dan modern. Dari beberapa jenis operasi permesinan, pengeboran (*drilling*) adalah salah

satu operasi yang umum di industri manufaktur jika mempertimbangkan pengulangan pada prosesnya. Pengeboran lubang adalah salah satu bagian penting dalam proses manufaktur dan sering menjadi bagian dari rencana produksi. Karena lubang dibuat pada tahap akhir, kualitas yang buruk dapat menyebabkan produk ditolak dan menimbulkan kerugian besar. Selain itu, jika alat bor rusak atau aus, benda kerja sering harus diproses ulang atau bahkan dibuang, yang juga menambah biaya produksi.

Sebagai salah satu upaya agar hasil pengeboran lubang tetap berkualitas baik, perlu dilakukan control dan monitor dari alat bor. Prediksi tingkat keausan menjadi salah satu metode untuk menjaga agar alat bor tetap berfungsi dan menghasilkan kualitas hasil pengeboran yang diinginkan. Parameter *cutting speed*, *feeding rate* dan *flank wear* digunakan untuk menghitung nilai prediksi. *Flank wear* atau keausan tepi terjadi pada bagian pahat yang berkontak langsung dengan benda kerja. Flank merupakan sisi samping muka pahat potong dimana terletak pada tepi sisi potong utama.

B. MODEL PREDIKTIF PEMANTAUAN KEAUSAH

1. Model ANFIS

Sistem inferensi fuzzy ANFIS yang diterapkan dalam kerangka jaringan adaptif merupakan metode yang menggunakan pembelajaran gabungan untuk membangun hubungan antara input dan output. Hubungan input dan output dibangun berdasarkan data yang tersedia, dan aturan fuzzy yang berasal dari pengetahuan manusia. Dalam system kerjanya, ANFIS digunakan untuk memodelkan fungsi non-linier dan memprediksi pola yang kompleks seperti deret waktu *chaos*.

ANFIS digunakan untuk memprediksi *Flank Wear* dengan menggunakan input *cutting speed* dan *feeding rate*, dengan output *flank wear*. Simulasi dilakukan dengan menggunakan *software* MATLAB untuk membuat model dan prediksi dari *flank wear*.

2. Arsitektur ANFIS

ANFIS menggabungkan logika fuzzy dan jaringan syaraf tiruan / neuro network untuk membangun sisten inferensi adaptif. Model sesuai untuk digunakan pada system non-linear

dengan data yang terbatas karena dapat mengekstrak informasi aturan fuzzy secara otomatis dan mengoptimalkan parameter melalui training data.

Pada proses fuzzifikasi, variable angka/numerik diubah menjadi *variable linguistic*. *Variable linguistic* menggunakan kata dalam Bahasa umum (besar, kecil, tinggi, rendah) untuk menjelaskan suatu nilai dan direpresentasikan dalam himpunan fuzzy, dalam satu himpunan hanya termuat anggota yang berasal dari satu parameter yang dibagi menjadi beberapa anggota berdasarkan *range* tertentu.

Model ANFIS menggunakan lima lapisan jaringan untuk melaksanakan interferensi fuzzy, lima lapisan jaringan tersebut adalah 1) fuzzifikasi data input, 2) konstruksi fuzzy database, 3) konstruksi fuzzy *rule base* (peraturan), 4) pengambilan keputusan (*decision making*), 5) defuzzifikasi output.

Tiap lapisan jaringan pada ANFIS terdiri dari beberapa node yang dijelaskan oleh fungsi node. Input pada tiap lapisan didapatkan dari node pada lapisan sebelumnya. Rule base atau peraturan ANFIS terdapat peraturan fuzzy *if-then* tipe sugeno. Sistem interferensi fuzzy orde pertama dapat dimisalkan memiliki 2 input, x dan y, dan satu output, dua aturan dengan fuzzy *if-the* dapat dinyatakan dengan:

$$\begin{aligned} \text{Aturan 1 : } & \text{if}(x \text{ adalah } A_1) \text{ dan } (y \text{ adalah } B_1) \text{ then}(f_1 \\ & = p_1x + q_1y + r_1) \end{aligned}$$

$$\begin{aligned} \text{Aturan 2 : } & \text{if}(x \text{ adalah } A_2) \text{ dan } (y \text{ adalah } B_2) \text{ then}(f_2 \\ & = p_2x + q_2y + r_2) \end{aligned}$$

Dimana x dan y adalah input A1 dan B1 adalah himpunan fuzzy, f adalah output dalam daerah fuzzy yang dipengaruhi oleh aturan fuzzy p, q, dan r adalah parameter yang ditentukan saat proses *training*.

Pada lapisan pertama, semua node adalah adaptif node. Output pada lapisan pertama adalah tingkat keanggotaan fuzzy dari input seperti pada Persamaan (1) dan (2).

$$O_i^1 = \mu_{A_i}(x), \quad i = 3, 4 \quad (1)$$

$$O_i^1 = \mu_{B_{i-2}}(y), \quad i = 3, 4 \quad (2)$$

Dimana x adalah input node i, dan A1 adalah label yang terasosiasi dengan fungsi node. Pada umumnya, $\mu_{A_i}(x)$

dan $\mu_{B_{i-2}}(y)$ dipilih untuk menjadi berbentuk bel/lonceng dengan maksimum nilai 1 dan minimum sama dengan 0.

Jika menggunakan segitiga maka seperti Persamaan (3).

$$\mu_{A_i}(x) = \frac{1}{1 + [(x - c_1/a_1)]^{b_1}} \quad (3)$$

Jika yang digunakan adalah Gaussian maka seperti Persamaan (4).

$$\mu_{A_i}(x) = \exp \left[- \left(\frac{x - c_1}{a_1} \right)^2 \right] \quad (4)$$

Dimana a_i , b_i , c_i merupakan parameter fungsi keanggotaan yang mengatur fungsi sesuai denagn bentuknya.

Pada lapisan ke dua, semua node adalah node tetap, dan diberi tanda Π yang mengindikasikan bahwa node melaksanakan perkalian sederhana. Output pada lapisan ini dapat ditunjukkan pada Persamaan (5)

$$O_i^2 = \mu_{A_i}(x)\mu_{B_i}(y), \quad i = 1, 2 \quad (5)$$

Pada lapisan ke tiga semua node juga node tetap, dan dineri tanda N , yang mengindikasikan bahwa node berperan dalam menormalisasikan *firing strength* lapisan sebelumnya. Output pada lapisan ini dapat ditunjukkan pada Persamaan (6).

$$O_i^3 = w_1 = \frac{w_1}{w_1 + w_2}, \quad i = 1, 2 \quad (6)$$

Pada lapisan ke empat node menjadi adaptif, output tiap node adalah hasil sederhana dari normalisasai *firing strength* dan *polynomial* orde pertama (model sugeno orde pertama). Output pada lapisan ini dapat ditunjukkan pada Persamaan (7).

$$O_i^4 = w_1 f_1 = w_1 (p_1 x + q_1 y + r_1), \quad i = 1, 2 \quad (7)$$

Pada lapisan ke lima, hanya ada node tunggal yang tetap dengan label Σ . Pada node ini semua sinyal yang masuk dijumlahkan. Model output ditunjukkan pada Persamaan (8).

$$O_i^5 = \sum_{i=1}^2 w_1 f_1 = \frac{\sum_{i=1}^2 w_1 f_1}{w_1 + w_2}, \quad i = 1, 2 \quad (8)$$

Dari kelima lapisan ANFIS terdapat dua lapisan adaptif, yaitu lapisan pertama dan keempat. Pada lapisan pertama terdapat tiga parameter yang dapat diubah (a_i, b_i, c_i) yang berhubungan dengan input fungsi keanggotaan. Pada lapisan keempat juga terdapat tiga parameter yang dapat diubah (p_i, q_i, r_i) yang menyinggung orde pertama *polynomial*. Parameter tersebut adalah parameter akibat (*consequent*).

3. Model MLR

Regresi merupakan konsep sederhana untuk meneliti hubungan fungsi antara output dan input berdasarkan variable untuk pemilihan keputusan dari suatu proses dan memungkinkan untuk memproses deskripsi data manufaktur, estimasi dan control. Kriteria untuk menepatkan garis data pada regresi linear sederhana adalah dengan meminimalisasikan jumlah sisa kuadrat antara nilai yang dihitung dari respond dan nilai respon yang dihitung dengan model regresi. Respon linear dapat ditunjukkan pada Persamaan (9).

$$Y = a_0 + a_1 X \quad (9)$$

Dimana Y adalah nilai respon an X adalah nilai variable. *Multiple Linear Regression* yang digunakan jika respon adalah fungsi linear dari dua atau lebih variable independen. Pada umumnya, respon variable y dapat dihubungkan dengan variable regresi k. Model *Multiple Linear Regression* dengan variable regresi k dapat ditunjukkan pada Persamaan (10).

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_\varepsilon x_\varepsilon + \varepsilon \quad (10)$$

Parameter β_j ($j = 0, 1, 2, \dots, k$) disebut koefisien regresi.

C. SIMULASI ANFIS MENGGUNAKAN MATLAB

ANFIS menggabungkan logika fuzzy dan jaringan syaraf tiruan/*neuro network* untuk membangun sisten inferensi adaptif. Dengan menggunakan referensi dataset menggunakan

data *cutting speed*, *feeding rate*, dan *flank wear*, dilakukan pembagian data untuk training 70 % dan testing 30%.

Pada model ANFIS yang digunakan, data input berupa nilai *cutting speed* dan *feeding rate* yang diubah menjadi *variable linguistic* dengan beberapa aturan. Aturan fuzzy direpresentasikan dalam format IF-THEN dengan kombinasi input anggota himpunan.

Dengan x adalah *cutting speed*, y adalah *feeding rate*, dan f adalah *flank wear*, A dan B adalah data input dalam himpunan fuzzy, p, q, r adalah parameter yang didapatkan setelah proses training. Data referensi dapat disajikan dalam range sebagai berikut:

- Cutting speed* (low, med, high) Dapat disimbolkan dengan V_c
- Feeding rate* (low, med, high) Dapat disimbolkan dengan V_f
- Flank wear* (low, med, high) Dapat disimbolkan dengan V_B
- Membership function* akan menghasilkan aturan berdasarkan input *cutting speed* (V_c) yang memiliki tiga jangkauan nilai.

Aturan dalam simulasi ANFIS pada MATLAB ditunjukkan pada Tabel 1 dengan aturan sebagai berikut:

Aturan 1 : *if* (V_c adalah A_1) *dan* (V_f adalah B_1) *then* (V_B
 $= p_1 V_c + q_1 V_f + r_1$)

Aturan 2 : *if* (V_c adalah A_1) *dan* (V_f adalah B_2) *then* (V_B
 $= p_2 V_c + q_2 V_f + r_2$)

Aturan 3 : *if* (V_c adalah A_1) *dan* (V_f adalah B_3) *then* (V_B
 $= p_3 V_c + q_3 V_f + r_3$)

Aturan 4 : *if* (V_c adalah A_2) *dan* (V_f adalah B_1) *then* (V_B
 $= p_4 V_c + q_4 V_f + r_4$)

Aturan 5 : *if* (V_c adalah A_2) *dan* (V_f adalah B_2) *then* (V_B
 $= p_5 V_c + q_5 V_f + r_5$)

Aturan 6 : *if* (V_c adalah A_2) *dan* (V_f adalah B_3) *then* (V_B
 $= p_6 V_c + q_6 V_f + r_6$)

Aturan 7 : *if* (V_c adalah A_3) *dan* (V_f adalah B_1) *then* (V_B
 $= p_7 V_c + q_7 V_f + r_7$)

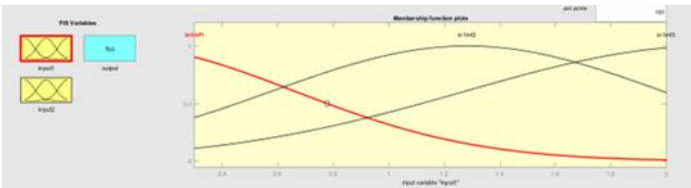
Aturan 8 : *if* (V_c adalah A_3) *dan* (V_f adalah B_2) *then* (V_B
 $= p_8V_c + q_8V_f + r_8$)

Aturan 9 : *if* (V_c adalah A_3) *dan* (V_f adalah B_3) *then* (V_B
 $= p_9V_c + q_9V_f + r_9$)

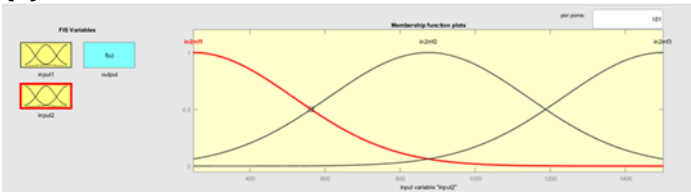
Tabel 1. Aturan yang dari dua parameter dengan 3 range

	Vc1	Vc2	Vc3
Vf1	VB1	VB4	VB7
Vf2	VB2	VB5	VB8
Vf3	VB3	VB6	VB9

Dengan dua input utama, yaitu *feeding rate* dan *cutting speed*, fungsi keanggotaan dapat divisualisasikan seperti pada Gambar 1a dan 1b. Masing-masing input dibagi ke dalam tiga kategori linguistik (rendah, sedang, tinggi) yang direpresentasikan dengan kurva keanggotaan berbentuk Gaussian.



(a)



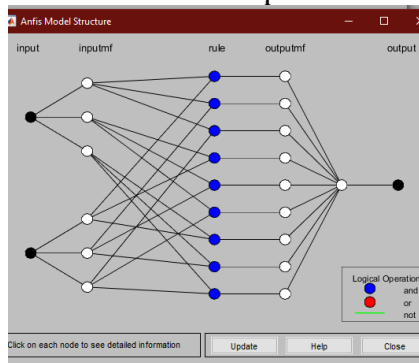
(b)

Gambar 1. Input Keanggotaan (a) *feeding rate*, (b) *cutting rate*

ANFIS menggunakan konsep neuro network dan fuzzy untuk menentukan hubungan antara output dan input berdasarkan input secara non-linear. Pada simulasi digunakan parameter sebagai berikut: Input: *cutting speed* dan *feeding rate*, Output: *Flank Wear*, Training: *Least Square*, Epoch: 100, Validasi: RMSE. Langkah simulasi menggunakan MATLAB seperti berikut:

1. Desain model

Pada model ANFIS yang digunakan, input berupa *cutting speed* dan *feeding rate* dengan input keanggotaan yang berjumlah tiga untuk tiga *range* nilai seperti ditunjukkan pada Gambar 2. Dari dua input yang masing masing memiliki tiga fungsi keanggotaan menghasilkan tiga aturan. Dari aturan tersebut dapat dihasilkan output nilai *flank wear* yang berdasarkan *range* nilai dari kedua input.



Gambar 2. Struktur ANFIS 5 lapisan

2. Pengumpulan data

Data diperoleh melalui referensi data set dengan jumlah data sebanyak total 6000 data yang masing-masing *Cutting Speed*, *Data Feeding Rate*, dan data *Flank Wear* memiliki 2000 nilai. Matriks data training berisi tiga kolom data sebanyak 70 % data (4200 data) yang dibagi menjadi 3 kolom. Dan 30 % (1800) data sisanya digunakan untuk testing/pengujian.

3. Simulasi MATLAB untuk ANFIS

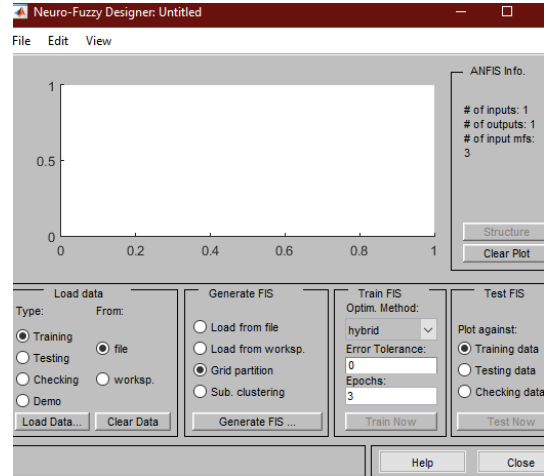
a. Input data

Data input yang digunakan terdiri dari tiga parameter, yaitu *cutting speed*, *feeding rate*, dan *flank wear*. Data disusun dalam bentuk tabel dan disimpan dalam file Excel untuk kemudian dimasukkan ke dalam *workspace* MATLAB. Dataset ini dibagi menjadi dua bagian, yaitu *dataTrain* dan *dataTest*, masing-masing digunakan untuk proses pelatihan dan pengujian model. Visualisasi data dapat dilihat pada Gambar 3.

	1	2	3	4	5
1	0.7000	250	0.3935		
2	0.7000	500	0.4673		
3	0.7000	750	0.5410		
4	0.7000	1000	0.6148		
5	0.7000	1250	0.6885		
6	0.8000	500	0.5355		
7	0.8000	750	0.6093		
8	0.8000	1000	0.6830		
9	0.8000	1250	0.7568		
10	0.8000	250	0.8305		
11	0.9000	750	0.6775		
12	0.9000	1000	0.7513		
13	0.9000	1250	0.8250		
14	0.9000	250	0.9088		

Gambar 3. Data untuk *Training* Dan *Testing*

Selanjutnya, proses pelatihan dan pengujian dilakukan melalui antarmuka Neuro-Fuzzy Designer pada MATLAB. Antarmuka ini memungkinkan pengguna untuk memuat data dari *workspace*, mengatur jenis metode training, serta menentukan struktur dan parameter sistem inferensi fuzzy yang digunakan. Pengaturan awal dilakukan dengan memilih tipe sistem fuzzy (*Sugeno*), metode pelatihan (*Hybrid*), serta jumlah epoch dan fungsi keanggotaan. Tampilan antarmuka ini dapat dilihat pada Gambar 4.

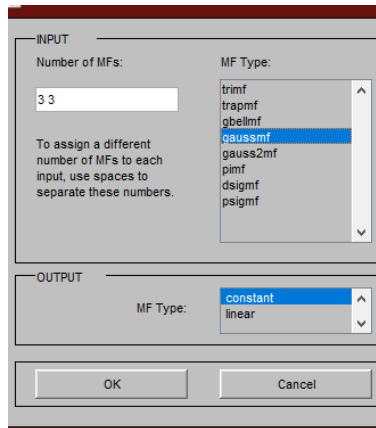


Gambar 4. *Neuro Fuzzy Designer* untuk Pengaturan Data *Training* Dan *Testing*

b. Inisialisasi ANFIS

Struktur awal ANFIS dibuat dengan mengatur jumlah

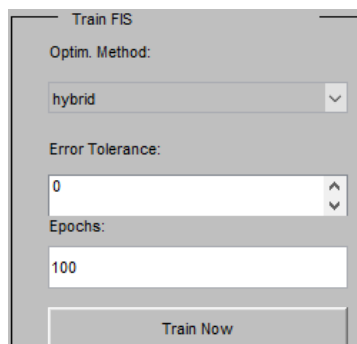
input, banyaknya fungsi keanggotaan dan jenis fungsi keanggotaan. Pada neuro fuzzy designer, data dimasukkan dengan menekan 'load data' dari *workspace* seperti ditunjukkan pada Gambar 5.



Gambar 5. Bagian Generate Fis untuk Setting Jenis Keanggotaan

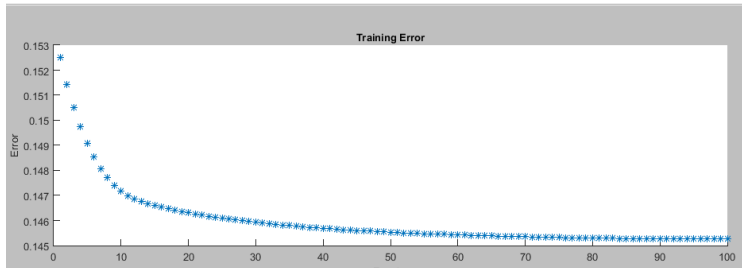
c. *Training data dengan Hybrid least square dan back propagation*

Training dilakukan dengan metode *hybrid* menggunakan *Least square* untuk *training* maju dan *back propagation* untuk *training* mundur. *Training* dilakukan dengan *least square* yang dilanjutkan dengan *back propagation* kemudian dilakukan lagi dengan *least square* secara berulang. Banyaknya pengulangan proses *training* tersebut disebut *Epoch*, banyaknya *epoch* digunakan dalam simulasi adalah 100 ditunjukkan pada Gambar 6.



Gambar 6. Tipe Training Data Epoch

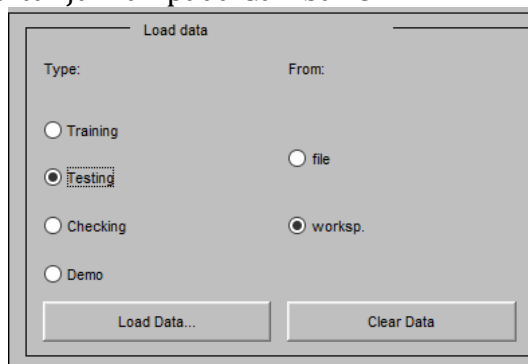
Setelah melakukan training data akan diperoleh grafik seperti pada Gambar 7. Grafik akan cenderung menurun sesuai dengan nilai *error* dengan RMSE yang akan mendekati nilai 0. Nilai *error* akan terus menurun seiring dengan banyaknya *epoch* sampai mendekati nilai yang terus melandai pada grafik atau stabil.



Gambar 7. Hasil Grafik Setelah Training Data

d. *Testing data*

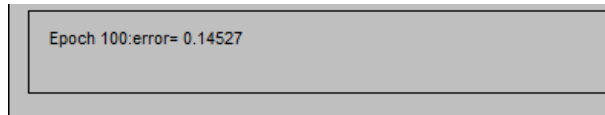
Setelah proses pelatihan selesai, data pengujian (*testing data*) dimasukkan untuk mengevaluasi kinerja model terhadap data yang belum dikenali sebelumnya. Langkah pengujian dilakukan dengan cara yang sama seperti pelatihan, namun dilakukan dengan mengubah pengaturan input dari training menjadi testing pada antarmuka Neuro-Fuzzy Designer, sebagaimana ditunjukkan pada Gambar 8.



Gambar 8. Mengubah *Setting* dari *Training* ke *Testing*

e. *Evaluasi dengan RMSE*

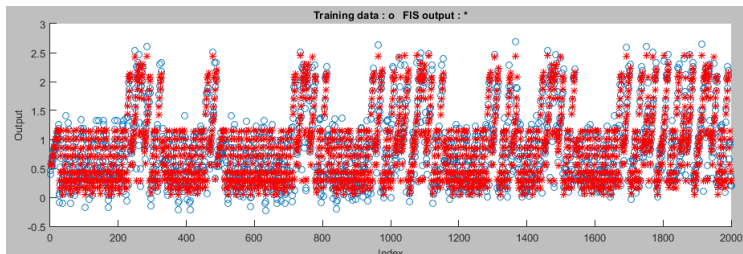
Nilai kesalahan prediksi model dievaluasi menggunakan RMSE. RMSE dihitung secara otomatis setelah proses pelatihan selesai. Semakin kecil nilai RMSE, maka akurasi model dalam memprediksi output akan semakin tinggi. Contoh tampilan hasil RMSE dapat dilihat pada Gambar 9.



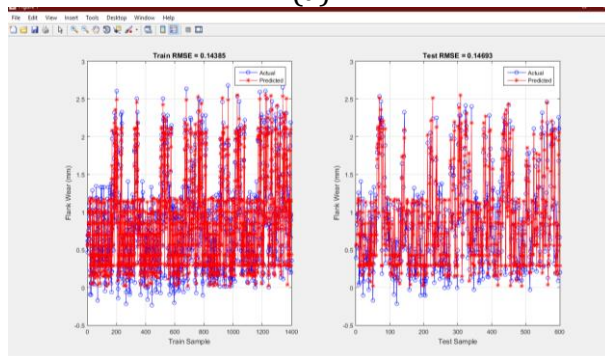
Gambar 9. Banyaknya *Epoch* Dan Nilai *Error*

f. Prediksi dan visualisasi

Hasil prediksi dari data pelatihan dan pengujian dapat divisualisasikan dalam bentuk grafik. Visualisasi ini dapat dilakukan langsung pada *Neuro-Fuzzy Designer* maupun melalui *plotting* manual di MATLAB. Visualisasi ini membantu untuk membandingkan nilai aktual dan hasil prediksi model secara visual. Gambar 10a dan 10b. menampilkan contoh hasil visualisasi.



(a)

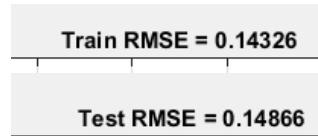


(b)

Gambar 10. Visualisasi hasil prediksi menggunakan (a) *Neuro-Fuzzy Designer*, (b) MATLAB

g. Validasi data

Proses validasi dilakukan dengan membandingkan hasil prediksi antara data pelatihan (*training*) dan data pengujian (*testing*) berdasarkan nilai RMSE. Validasi ini berguna untuk menilai sejauh mana model mampu mempertahankan performa prediksi saat diberikan data baru yang tidak termasuk dalam pelatihan awal. Didapatkan nilai seperti pada Gambar 11.



Gambar 11. Nilai RMSE

D. Simulasi MATLAB untuk MLR

MLR menggunakan konsep sederhana untuk meneliti hubungan fungsi antara output dan input berdasarkan variable untuk pemilihan keputusan. Dengan menggunakan referensi dataset *cutting speed*, *feeding rate*, dan *flank wear*, dilakukan pembagian data untuk training 70 % dan testing 30%. Pada simulasi digunakan parameter sebagai berikut: Input: *cutting speed* dan *feeding rate*, Output : *Flank Wear*, Training :Least Square, Epoch : 100, Validasi : RMSE.

Pada model MLR yang digunakan, data input berupa nilai *cutting speed* dan *feeding rate* untuk mendapatkan data output berupa nilai *flank wear*. Dengan menggunakan dua variable untuk input, model regresi dapat dibuat Persamaan (11).

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon \quad (11)$$

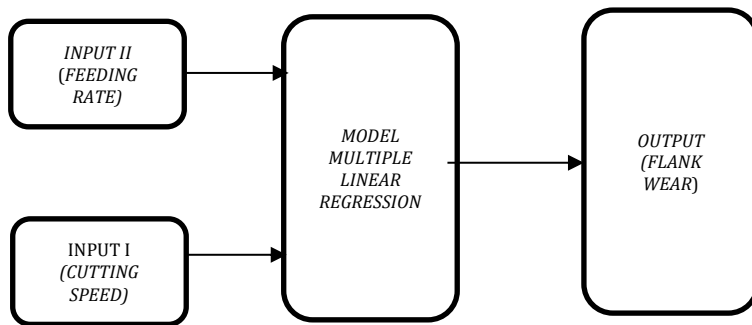
Dengan y: variabel output, x1, x2: variabel input, β_0 : (konstanta), β_1 , β_2 : koefisien regresi yang mempengaruhi variabel input terhadap output, dan ε : error. Langkah simulasi menggunakan MATLAB seperti berikut:

1. Analisis masalah

Masalah utama yang dibahas adalah tentang *flank wear* yang dimonitor untuk dikontrol menggunakan model ANFIS, dalam hal ini input menggunakan nilai *cutting speed* dan *feeding rate*, output berupa nilai *flank wear*.

2. Desain Model MLR

Model MLR digunakan untuk memodelkan hubungan linear antara input dan output. Diguankan dua input *cutting speed* dan *feeding rate*, dengan output *Flank wear* seperti ditunjukkan pada Gambar 12.



Gambar 12. Struktur Model MLR

3. Pengumpulan Data

Data diperoleh melalui referensi data set dengan jumlah data sebanyak total 6000 data yang masing-masing *Cutting Speed*, *Data Feeding Rate*, dan data *Flank Wear* memiliki 2000 nilai. Matriks data *training* berisi tiga kolom data sebanyak 70 % data (4200 data) yang dibagi menjadi 3 kolom. Dan 30 % (1800) data sisanya digunakan untuk *testing*/pengujian.

4. Simulasi MATLAB

a. Input data

Data set dimasukkan ke dalam MATLAB dengan menggunakan fungsi 'xlsread'. Data dibagi menjadi 3 kolom table untuk masing masing; *cutting speed*, *feed rate*, dan *Flank wear*. Penulisan dan struktur kode ditunjukkan pada Gambar 13 untuk memberikan gambaran proses awal pengolahan data.

```

%% === LOAD DATA FROM EXCEL ===
filename = 'flank_wear_dataset_2000.xlsx';
data = xlsread(filename);

Vc = data(:,1); % Cutting speed
f = data(:,2); % Feed rate
VB = data(:,3); % Flank wear

% Combine predictors
X = [Vc f];
  
```

Gambar 13. Penulisan Struktur Kode Simulasi

b. Inisiasi MLR

Dataset diubah menjadi format tabel dapat dilakukan *train* dengan model MLR menggunakan fungsi 'fitlm (tlb_train)' atau menggunakan fit linear model untuk melakukan train pada table dataset seperti ditunjukkan pada Gamba 14.


```

%% === KONVERSI KE FORMAT TABEL UNTUK fitlm ===
tbl_train = array2table([X_train Y_train], 'VariableNames', {'Vc','f','VB'});
tbl_test = array2table([X_test Y_test], 'VariableNames', {'Vc','f','VB'});

%% === TRAINING MODEL MLR ===
mdl = fitlm(tbl_train, 'VB ~ Vc + f');
B = mdl.Coefficients.Estimate; % Koefisien regresi

```

Gambar 14. Inisiasi MLR

c. *Training dan Testing data*

Data *train* dan data *test* dilakukan prediksi berdasarkan input data referensi seperti ditunjukkan Gambar 15.

```

%% === TRAINING MODEL MLR ===
mdl = fitlm(tbl_train, 'VB ~ Vc + f');
B = mdl.Coefficients.Estimate; % Koefisien regresi

%% === PREDIKSI TRAINING DAN TESTING ===
Y_train_pred = predict(mdl, tbl_train);
Y_test_pred = predict(mdl, tbl_test);

```

Gambar 15. *Training dan Testing Data*

d. *Evaluasi*

Evaluasi dilakukan dengan menghitung nilai RMSE untuk menilai seberapa besar galat prediksi model terhadap data sebenarnya seperti ditunjukkan pada Gambar 16.

```

%% === EVALUASI TRAINING ===
rmse_train = sqrt(mean((Y_train - Y_train_pred).^2));

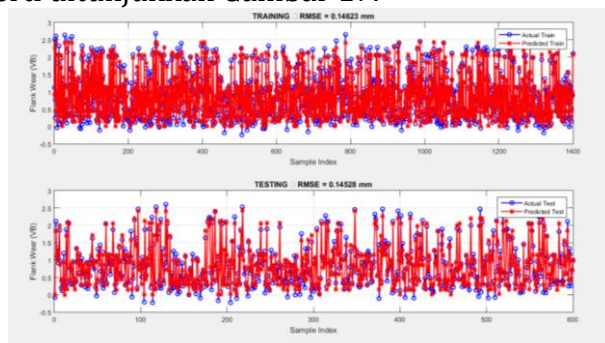
%% === EVALUASI TESTING ===
rmse_test = sqrt(mean((Y_test - Y_test_pred).^2));

```

Gambar 16. *Evaluasi Training dan Testing*

e. *Visualisasi*

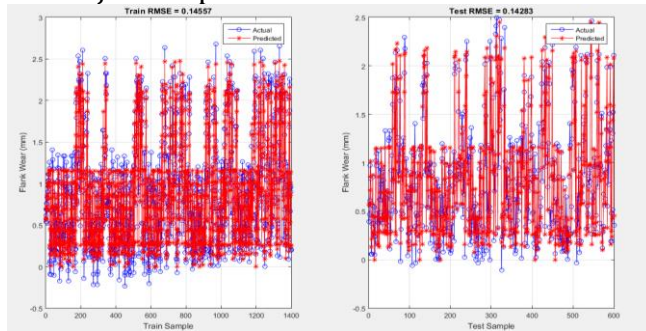
Data disajikan dalam bentuk grafik untuk perbandingan antara dataset dan data prediksi berdasarkan training dan testing seperti ditunjukkan Gambar 17.

Gambar 17. *Grafik Training dan Testing*

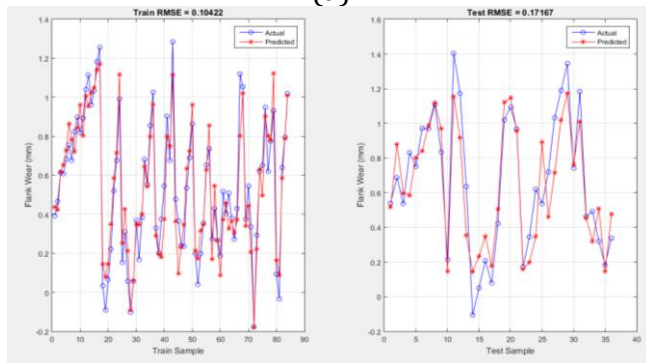
E. VALUASI HASIL SIMULASI MODEL ANFIS DAN MLR

1. MODEL ANFIS

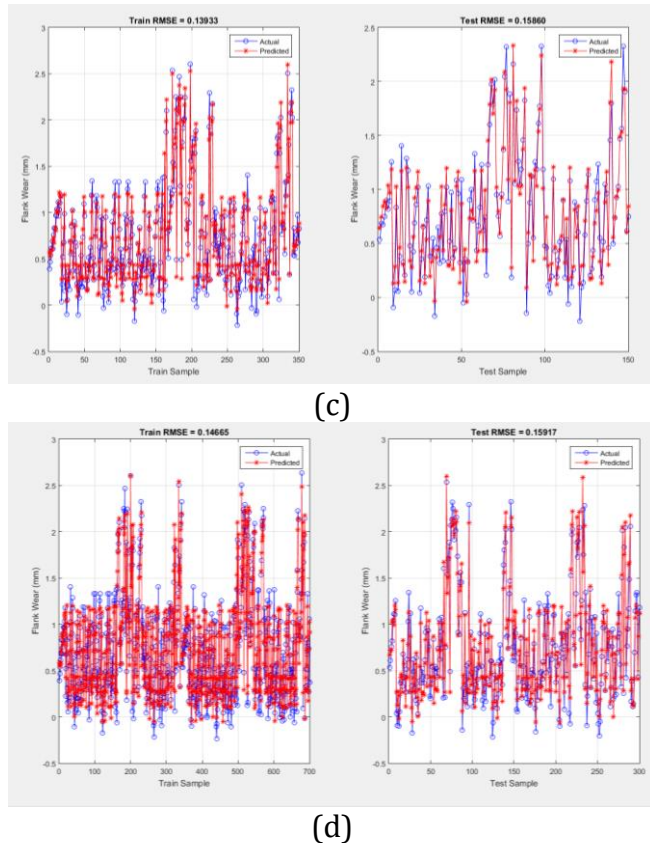
Hasil dari simulasi yang dilakukan dengan menggunakan ANFIS dengan 6000 data memiliki RMSE 0,14557 untuk *train* dan RMSE 0,14283 untuk *test* seperti ditunjukkan pada Gambar 18a. Hasil dari simulasi yang dilakukan dengan menggunakan ANFIS dengan 360 data memiliki RMSE 0,10422 untuk *train* dan RMSE 0,17167 untuk *test* seperti ditunjukkan pada Gambar 18b. Hasil dari simulasi yang dilakukan dengan menggunakan ANFIS dengan 1500 data memiliki RMSE 0,13933 untuk *train* dan RMSE 0,15860 untuk *test* seperti ditunjukkan pada Gambar 18c. Hasil dari simulasi yang dilakukan dengan menggunakan ANFIS dengan 3000 data memiliki RMSE 0,14665 untuk *train* dan RMSE 0,15917 untuk *test* seperti ditunjukkan pada Gambar 18d.



(a)



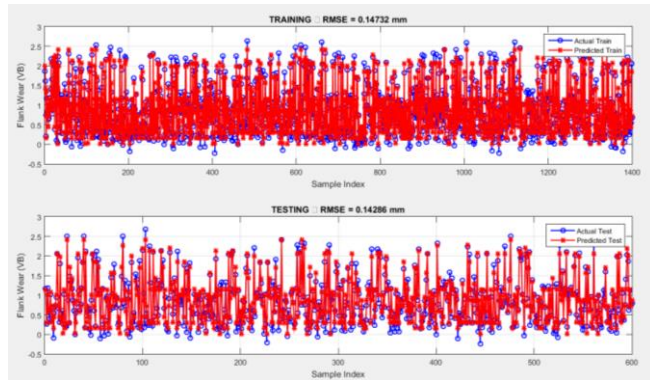
(b)



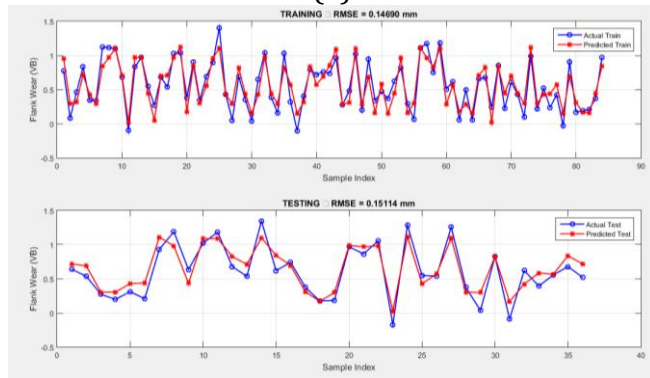
Gambar 18. Grafik Model ANFIS (a) 6000 data, (b) 360 data, (c) 1500 data, (d) 3000 data

2. MODEL MLR

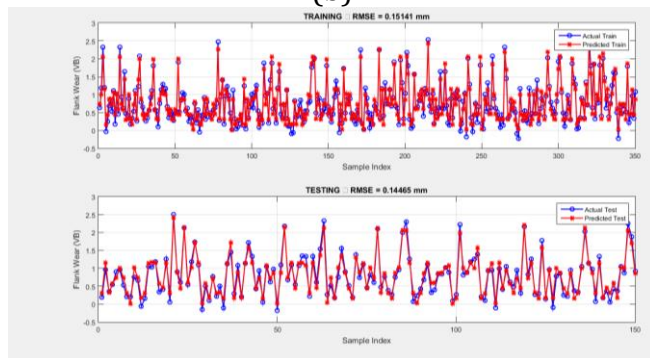
Hasil dari simulasi yang dilakukan dengan menggunakan *Linear Regression* dengan 6000 data menunjukkan RMSE 0, 14732 untuk *train* dan RMSE 0, 14286 untuk *test* seperti ditunjukkan pada Gambar 19a. Hasil dari simulasi yang dilakukan dengan menggunakan *Linear Regression* dengan 360 data menunjukkan RMSE 0, 14690 untuk *train* dan RMSE 0, 15114 untuk *test* seperti ditunjukkan pada Gambar 19b. Hasil dari simulasi yang dilakukan dengan menggunakan *Linear Regression* dengan 1500 data menunjukkan RMSE 0,15141 untuk *train* dan RMSE 0, 14465 untuk *test* seperti ditunjukkan pada Gambar 19c. Hasil dari simulasi yang dilakukan dengan menggunakan *Linear Regression* dengan 3000 data menunjukkan RMSE 0,15298 untuk *train* dan RMSE 0, 14898 untuk *test* seperti ditunjukkan pada Gambar 19d.



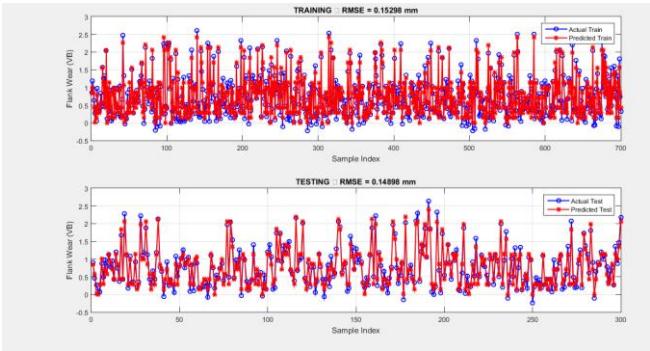
(a)



(b)



(c)



(d)

Gambar 10. Grafik Model MLR (a) 6000 data, (b) 360 data, (c) 1500 data, (d) 3000 data

Berdasarkan simulasi yang telah dilakukan dengan dua model yaitu ANFIS dan *Multiple Linear Regression* dengan *epoch* 100, data simulasi yang telah dilakukan dapat dibuat Tabel 2.

Tabel 2. Perbandingan nilai RMSE antara ANFIS dan MLR

MODEL	Hasil RMSE Train	Hasil RMSE Test
ANFIS (360 data)	0,10422	0,17167
ANFIS (1500 data)	0,13933	0,15860
ANFIS (3000 data)	0,14665	0,15917
ANFIS (6000 data)	0,14557	0,14283
Rata-Rata	0,133943	0,158068
Linear Regression (360 data)	0,14690	0,15114
Linear Regression (1500data)	0,15141	0, 14465
Linear Regression (3000data)	0,15298	0, 14898
Linear Regression (6000data)	0,14732	0,14286
Rata-Rata	0,149653	0,147

ANFIS mampu menangani hubungan non-linear antara input dan output. Model ANFIS menghasilkan akurasi yang lebih tinggi yang di tunjukan dengan nilai RMSE lebih rendah dibandingkan dengan model MLR karena dapat mempelajari pola yang kompleks dalam data. Namun karena sifatnya yang kompleks dan melibatkan parameter tersembunyi, hasil sulit

untuk diinterpretasikan secara langsung. Di lain sisi, MLR dapat dipahami dan dijelaskan dengan bentuk persamaan linear. Model MLR mampu menjelaskan pengaruh masing-masing variable secara eksplisit melalui koefisien regresi.

Dengan menggunakan 2 parameter input, selisih RMSE pada kedua model tidak terlalu jauh. Selisih antara kedua model dan antara training dan testing yang dihasilkan kemungkinan dipengaruhi oleh nilai variable yang digunakan.

Model MLR cenderung lebih mudah diinterpretasikan karena menggunakan persamaan linier dan setiap koefisien menunjukkan pengaruh terhadap masing masing variable. Model ANFIS cenderung sulit di interpretasikan, ANFIS menggunakan aturan *fuzzy* dan juga *neuro network* yang tidak dapat diinterpretasikan secara langsung.

Kedua model memiliki pola hubungan antara input dan output yang berbeda, struktur, kompleksitas dan cara pembelajaran dari tiap model memiliki karakteristik tertentu. *Flank wear* tidak hanya dipengaruhi oleh parameter *cutting speed* dan *feeding rate*, berbagai parameter dalam proses permesinan dapat mempengaruhi baik dari parameter mesin maupun dari bahan yang diproses. Keausan yang terjadi dapat dibengaruhi oleh berbagai faktor tidak hanya *cutting speed* dan *feeding rate*. Factor lainnya diantaranya adalah: suhu, waktu kerja/proses, dan getaran. Hubungan antara factor ini tidak linier, misal kenaikan *cutting speed* dapat mempengaruhi *flank wear* namun pada titik tertentu pengaruhnya akan berkurang bahkan hilang. Contoh lain adalah parameter *cutting speed* dan *feeding rate* tidak memiliki sinergi, kedua parameter menghasilkan nilai keausan yang tidak sebanding satu sama lain atau keausan yang dihasilkan adalah dampak masing masing parameter yang tidak mempengaruhi satu sama lain. Model ANFIS lebih cocok untuk diterapkan dalam proses permesinan *drilling* karena dapat digunakan untuk memprediksi dengan akurasi yang lebih tinggi dibandingkan dengan model MLR. ANFIS dapat mengungkap pola hubungan dari data yang diberikan secara dua arah dengan menggunakan *back propagation*, dibandingkan dengan MLR yang hanya bisa memberikan hubungan satu arah.

Dalam memprediksi flank wear akurasi yang tinggi lebih diutamakan dibandingkan dengan interpretasi model. Nilai

flank wear dapat menentukan hasil proses *drilling*, dan efisiensi dari proses manufaktur.

Model MLR memiliki nilai akurasi yang lebih rendah dari model ANFIS yang dapat dilihat dari nilai RMSE yang lebih tinggi. Perbandingan kedua nilai RMSE antara kedua model tidak jauh berbeda yaitu di antara rentan nilai 0,10 dan 0,17. Dengan selisih antar nilai yang kecil model MLR masih relevan untuk digunakan sebagai model prediksi *flank wear* berdasarkan input nilai *cutting speed* dan *feeding rate*

PENUTUP

Kemampuan model ANFIS dalam penyesuaian parameter yang fleksibel terhadap data kompleks dengan *fuzzy* yang dapat membagi menjadi rentan nilai tertentu dan *neuro network* yang dapat memproses data dengan parameter nilai yang kompleks dan tersembunyi untuk data *non-linear*. Nilai RMSE dari ANFIS lebih rendah ketika menggunakan data yang berbeda. Selisih antar kedua model masih terlihat meskipun menggunakan jumlah data yang berbeda. Nilai ANFIS berkisar antara 0,01 – 0,17, dengan rata rata 0,133943 untuk *training* dan 0,158068 untuk *testing*, sedangkan MLR stabil pada 0,14 - 0,150, dengan rata rata 149653 untuk *training* dan 0,147 untuk *testing*. Hal tersebut mengindikasikan bahwa ANFIS dapat memprediksi dengan baik dalam parameter dan jumlah data tertentu. Dan model MLR memberikan nilai akurasi yang stabil. Meskipun model MLR memiliki nilai akurasi yang lebih rendah, namun model masih relevan digunakan. Dengan 2 parameter input, perbedaan nilai RMSE tidak terlalu jauh terhadap nilai RMSE model ANFIS. Prediksi *Flank Wear* berbasis dua input parameter menunjukkan tingkat keakuratan model ANFIS dan MLR yang mampu memprediksi nilai keausan dengan parameter input sederhana yaitu *cutting speed* dan *feeding rate* yang dapat diukur secara langsung saat proses permesinan terjadi.

DAFTAR PUSTAKA

- Abbas, A. T., Alata, M., Ragab, A. E., El Rayes, M. M., & El Danaf, E. A. (2017). Prediction model of cutting parameters for turning high-strength steel grade-H: Comparative study of regression model versus ANFIS. *Advances in Materials Science and Engineering*, 2017, 1–12.
- Aru, O. (2016). Exploration of the adaptive neuro-fuzzy inference system architecture and its applications. *American Journal of Engineering Research*, 5(10), 181–188.
- Çakan, A., Evrendilek, F., & Ozkaner, V. (2016). Data-driven simulations of flank wear coated cutting tools in hard turning. *Mechanics*, 21(6).
- Çelik, Y. H., & Fidan, Ş. (2022). Analysis of cutting parameters on tool wear in turning of Ti-6Al-4V alloy by multiple linear regression and genetic expression programming methods. *Measurement*, 200, Article 111638.
- Dutta, S., Pal, S. K., & Sen, R. (2016). On-machine tool prediction of flank wear from machined surface images using texture analyses and support vector regression. *Precision Engineering*, 43, 34–42.
- Feng, K., Ji, J. C., Ni, Q., & Beer, M. (2023). A review of vibration-based gear wear monitoring and prediction techniques. *Mechanical Systems and Signal Processing*, 182, Article 109605.
- Imhade, P. O., & Jude, E. S. (2023). Overview study of ANN-GA, ANN-PSO, ANFIS-GA, and ANFIS-PCM prediction analysis on tool wear. *Journal Européen des Systèmes Automatisés*, 56(2), 269–280.
- Jeong, M. J., Lee, S. W., Jang, W. K., Kim, H. J., Seo, Y. H., & Kim, B. H. (2021). Prediction of drill bit breakage using an infrared sensor. *Sensors*, 21(8), 2808.
- Khaled, Z., Mondher, B. A., & Mourad, T. (2018). Modeling of flank wear using ANFIS. *Proceedings of the International Conference on Engineering and MIS (ICEMIS)*.
- Kong, D., Chen, Y., & Li, N. (2018). Gaussian process regression for tool wear prediction. *Mechanical Systems and Signal Processing*, 104, 556–574.
- Lin, C.-J., Jhang, J.-Y., Chen, S.-H., & Young, K.-Y. (2020). Using an interval type-2 fuzzy neural network and tool chips for

- flank wear prediction. *IEEE Access*, 8, 122626–122640.
- Mushtaq, Z., Hanief, M., & Manroo, S. A. (2021). Prediction of friction and wear during ball-on-flat sliding using multiple regression and ANN: Modeling and experimental validation. *Jurnal Tribologi*, 28, 117–128.
- Sarhan, A. (2015). Adaptive neuro-fuzzy approach to predict tool wear accurately in turning operations for maximum cutting tool utilization. *IFAC-PapersOnLine*, 48.
- Sarikaya, M., & Güllü, A. (2015). Taguchi design and response surface methodology-based analysis of machining parameters in CNC turning under MQL. *Journal of Cleaner Production*, 65.
- Senthilkumar, N., & Tamizharasan, T. (2015). Flank wear and surface roughness prediction in hard turning via artificial neural network and multiple regressions. *Australian Journal of Mechanical Engineering*, 13, 31–45.
- Srivastava, A. K., & Islam, M. M. (2024). Prediction of tool wear and surface finish using ANFIS modelling during turning of carbon fiber reinforced plastic (CFRP) composites. *Manufacturing Letters*, 41, 658–669.
- Thangarasu, S. K., Mohanraj, T., Devendran, K., Rajalakshmi, M., Subrata, C., & Saravanakumar, G. (2023). Prediction of flank wear during turning of EN8 steel with cutting force signals using a deep learning approach. *Mathematical Problems in Engineering*.
- Vuong, N. D. Y., Xue, S., Liu, Y., Zhou, Y., & Wu, M. (2020). Predicting surface roughness and flank wear in turning processes. *International Conference on Prognostics and Health Management (ICPHM)*, Detroit, MI, USA.
- Xu, L., Huang, C., Li, C., Wang, J., Liu, H., & Wang, X. (2021). Estimation of tool wear and optimization of cutting parameters based on novel ANFIS-PSO method toward intelligent machining. *Journal of Intelligent Manufacturing*, 32(1), 77–90.
- Zhang, J., Starly, B., Cai, Y., Cohen, P., & Lee, Y.-S. (2017). Particle learning in online tool wear diagnosis and prognosis. *Journal of Manufacturing Processes*, 28, 322–330.