

## Penerapan Metode *K-Nearest Neighbors* (kNN) pada Bearing

Anggi Priliani Yulianto\*, Sutawanir Darwis

Prodi Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Islam Bandung, Indonesia.

\*anggipy@gmail.com, sutawanir.darwis@unisba.ac.id

**Abstract.** Monitoring the condition of the engine is a top priority to avoid damage. To know the condition of the bearing, it is important to know the remaining useful life of the machine. In the IEEE PHM 2012 Prognostic Challenge platform provides real data related to accelerated bearing degradation carried out under constant operating conditions and online controlled variables of temperature and vibration (with horizontal and vertical accelerometers). In this platform, the data used is bearing2\_3 data in the horizontal direction which has a duration of about 2 hours, calculated RMS every 1/10 second (2560 data). In this study machine learning based modeling will be done using the k-nearest neighbor (kNN) method to determine the prediction of RMS bearings. The kNN method is based on the classification of objects based on training data that is the closest distance to the object. kNN is a nonparametric machine learning algorithm which is a model that does not assume distribution. The advantage is that the class decision line produced by the model can be very flexible and very nonlinear. The smallest MSE value was obtained at  $k = 16$  with MSE value = 0.157579. After getting the optimum  $k$  value, proceed with predicting a RMS of 97 lags and identifying bearing performance in several phases.

**Keywords:** Remaining useful life, Prognostic Challenge, Bearing

**Abstrak.** Pemantauan kondisi mesin menjadi prioritas utama untuk menghindari adanya kerusakan. Untuk mengetahui kondisi bantalan, penting untuk mengetahui sisa masa manfaat dari mesin tersebut. Dalam platform IEEE PHM 2012 Prognostic Challenge ini menyediakan data nyata terkait dengan degradasi bantalan yang dipercepat yang dilakukan di bawah kondisi operasi konstan dan variabel yang dikendalikan secara online berupa suhu dan getaran (dengan akselerometer horizontal dan vertikal). Dalam platform ini, data yang digunakan adalah data bearing2\_3 pada arah horizontal yang berdurasi sekitar 2 jam ini dihitung RMS setiap 1/10 detik (2560 data). Dalam penelitian ini akan dilakukan pemodelan berbasis machine learning menggunakan metode k-nearest neighbor (kNN) untuk mengetahui prediksi RMS bearing. Metode kNN didasarkan pada klasifikasi terhadap objek berdasarkan data pelatihan yang jaraknya paling dekat dengan objek tersebut. kNN merupakan salah satu algoritma pembelajaran mesin yang bersifat nonparametrik yakni model yang tidak mengasumsikan distribusi. Kelebihannya adalah garis keputusan kelas yang dihasilkan model tersebut bisa jadi sangat fleksibel dan sangat nonlinier. Nilai MSE terkecil diperoleh pada  $k = 16$  dengan nilai MSE = 0,157579. Setelah mendapatkan nilai  $k$  optimum, dilanjutkan dengan memprediksi RMS sebanyak 97-lag serta mengidentifikasi performance kinerja bearing dalam beberapa fase.

**Kata Kunci:** Sisa Masa Manfaat, Tantangan Prognostic, Bearing

## A. Pendahuluan

*Bearing* atau bantalan merupakan salah satu bagian dari elemen mesin yang memegang peranan penting untuk menjaga kinerja sistem dalam perangkat agar tetap dalam kondisi baik. Jika bantalan tidak berfungsi dengan baik, maka seluruh sistem dalam perangkat tidak dapat bekerja secara semestinya. Oleh karena itu, untuk mengetahui kondisi bantalan masih dalam kondisi baik ataupun tidak, penting untuk mengetahui sisa masa manfaat dari mesin tersebut.

Dalam suatu mesin, mengetahui sisa manfaat dari mesin tersebut adalah hal yang penting karena sebagian besar kegagalan dalam mesin tersebut berasal dari sisa masa manfaat yang tidak diketahui. Dalam IEEE PHM 2012 *Prognostic Challenge*, difokuskan pada prognostik dari sisa manfaat dari bantalan, ini termasuk masalah yang cukup kritis karena sebagian besar kegagalan rotasi mesin terkait dengan komponen-komponen ini sangat mempengaruhi ketersediaan, keamanan dan efektifitas biaya industri mekanik maupun listrik.

Dalam gelombang sinusoidal, mencari rata-rata atau nilai tegangan rata-rata, tidak dapat menggunakan *average* atau rata-rata biasa. Untuk gelombang, area diatas sumbu horizontal adalah positif sedangkan area dibawah sumbu horizontal adalah negatif. Hasilnya adalah bahwa nilai rata-rata sama dengan nol (0) karena area atas sama dengan area dibawah sumbu horizontal. Dengan demikian akan membatalkan satu sama lain. Cara lain untuk mencari rata-rata ini adalah menghitung RMS (*Root Mean Square*). RMS ini menghitung akar dari nilai rata-rata dari suatu fungsi yang dikuadratkan. RMS ini umumnya digunakan pada teknik kelistrikan (*electrical engineer*).

Dalam penelitian ini akan dilakukan pemodelan berbasis machine learning menggunakan metode *k-nearest neighbor* (kNN) untuk mengetahui prediksi terkait RMS *bearing*. Metode kNN didasarkan pada klasifikasi terhadap objek berdasarkan data pelatihan yang jaraknya paling dekat dengan objek tersebut. kNN merupakan salah satu algoritma pembelajaran mesin yang bersifat nonparametrik. Model nonparametrik adalah model yang tidak mengasumsikan distribusi di dalam data. Kelebihannya adalah garis keputusan kelas yang dihasilkan model tersebut bisa jadi sangat fleksibel dan sangat nonlinier. Dalam menentukan k optimum, seringkali mengalami kesulitan. Walaupun sebenarnya algoritma kNN adalah algoritma sederhana yang mudah dilakukan, tidak sedikit peneliti yang meragukan nilai dari k yang dihasilkan. Nilai k yang tinggi akan mengurangi efek *noise*, tetapi akan membuat hasil prediksi semakin kabur, sedangkan jika nilai k terlalu kecil atau 1, akan mengakibatkan hasil prediksi terasa kaku. Nilai k optimum dapat dipilih dengan menggunakan *cross validation* (validasi silang). *Cross validation* (validasi silang) adalah salah satu metode statistik yang dapat digunakan untuk mengevaluasi model atau algoritma dimana data dipisahkan menjadi dua subset, yaitu data pelatihan dan data evaluasi. *K-fold cross validation* diawali dengan membagi data sejumlah *k-fold* yang diinginkan. Dalam proses *cross validation*, data akan dibagi dalam k buah partisi dengan ukuran yang sama  $D_1, D_2, \dots, D_k$ , selanjutnya proses uji dan latih dilakukan sebanyak k kali. Dalam iterasi ke-*i* partisi  $D_i$  akan menjadi data uji dan sisanya akan menjadi data latih.

Berdasarkan dari uraian latar belakang diatas, tujuan dari penelitian ini adalah;

1. Mendapatkan k optimum dalam kNN.
2. Mengetahui prediksi RMS bearing menggunakan kNN.
3. Mengidentifikasi hasil (performance) kinerja bearing dalam beberapa fase.

## B. Landasan Teori

*Bearing* (bantalan) adalah elemen mesin yang menumpu proses yang mempunyai beban, digunakan untuk memungkinkan gerakan poros dalam perangkat komersial sederhana seperti sepeda, sepeda motor dan lain sebagainya. Jika *bearing* tidak berfungsi dengan baik maka seluruh sistem dalam perangkat tidak dapat bekerja secara semestinya. Bukti arkeologis menunjukkan bahwa sekitar tahun 2400 SM penduduk Mesir menggunakan kereta luncur sederhana menggunakan pelumasan air atau lemak hewan yang digunakan untuk membawa batu dan patung besar. Penggunaan bantalan yang lebih maju pada 1100 SM di dekat sungai Tigris dengan menggunakan rol dibawah kereta luncur untuk mencapai hasil yang sama dengan lebih sedikit pekerja.

Vibrasi atau getaran didefinisikan oleh Kamus *Webster's New World* sebagai “ayunan yang terjadi secara terus menerus; berosilasi”. Sedangkan bagi para teknisi yang berkecimpung di bidang industri, mendefinisikan vibrasi sebagai gerakan bolak-balik yang terdapat pada bagian sebuah mesin dari tempat awal kedudukannya. Menteri Negara Lingkungan Hidup dalam surat keputusannya mencantumkan bahwa getaran adalah gerakan bolak-balik suatu massa melalui keadaan setimbang terhadap suatu titik acuan. Terdapat tiga parameter utama dalam pengukuran vibrasi terhadap sebuah mesin, yaitu perpindahan atau jarak (*displacement*), kecepatan (*velocity*), dan percepatan (*acceleration*) yang harus diperhatikan sebelum menganalisa penyebab terjadinya vibrasi yang tidak normal. Perpindahan merupakan adalah jarak yang ditempuh oleh gerakan bolak-balik (getaran) pada suatu periode waktu tertentu. Pada [3], perpindahan mempunyai persamaan :

$$x(t) = A \sin \omega t \quad \dots(1)$$

dengan:  $x(t)$  : perpindahan (m);  $A$  : amplitudo;  $\omega$  : kecepatan sudut (rad/s);  $t$  : waktu (s)

Besaran  $\omega$  biasanya diukur dalam radian ( $\pi$ ) per detik dan disebut frekuensi lingkaran. Karena gerak berulang dalam  $2\pi$  radian, maka didapat hubungan :

$$\omega = \frac{2\pi}{T} = 2\pi f \quad \dots(2)$$

Dengan  $T$  dan  $f$  masing-masing adalah periode (s) dan frekuensi (Hz).

Data *mining* adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual. Data *mining* merupakan proses pencarian pola dan relasi-relasi yang tersembunyi dalam sejumlah data yang besar dengan tujuan untuk melakukan klasifikasi, estimasi, prediksi, *association rule* (aturan asosiasi), *clustering* (pengelompokkan), deskripsi dan visualisasi. Data *mining* merupakan pencarian pengetahuan dalam database yang digunakan untuk mendapatkan informasi yang dapat menjelaskan masa lalu maupun dipakai untuk memprediksi masa depan. Dalam memprediksi membutuhkan model, jika model yang digunakan mempunyai tujuan pengelompokkan data berupa atribut target maka termasuk dalam jenis pemodelan prediktif. Jika hasil pemodelan prediktif menghasilkan pengelompokan data dengan nilai diskrit maka disebut klasifikasi. Dalam mengklasifikasi dan meregresikan data jika menggunakan fungsi berdasarkan tingkat kemiripan maka digunakan k-NN.

*K-Nearest Neighbor* atau k-tetangga terdekat adalah salah satu metode menerapkan algoritma *supervised*, algoritma tersebut dibagi menjadi dua jenis yaitu *supervised learning* dan *unsupervised learning*. *Supervised learning* adalah suatu metode pengklasifikasian dimana setiap pengamatan memiliki variabel prediktor yang berkaitan dengan variabel respon dengan tujuan untuk mendapatkan pola baru, sedangkan tujuan dari *unsupervised learning* yaitu untuk mendapatkan pola dalam sebuah data. Cara kerja algoritma kNN ini adalah dengan mencari jarak terdekat antara data yang akan dievaluasi dengan k-tetangga terdekatnya didalam data pelatihan. Jarak yang biasa digunakan adalah *Euclidean distance*, dengan persamaan sebagai berikut :

$$D_{ij} = |x_i - x_j|, \quad \dots(3)$$

dengan :  $D_{i,j}$  = jarak Euclidean;  $x_i$  = titik pada data pelatihan;  $x_j$  = titik pada data pengujian;  $i$  = indeks data pelatihan; dan  $j$  = indeks data pengujian.

Estimasi kNN dari  $y$  pada  $x$  didefinisikan sebagai :

$$\hat{f}_k(x) = \frac{1}{k} \sum_{i=1}^N W_{ki}(x) y_i, \quad \dots(4)$$

dengan  $\{W_{ki}(x)\}_{i=1}^N$  adalah urutan bobot didefinisikan sebagai :

$$W_{ki}(x) = \begin{cases} 1 & , x_i \in S_k \\ 0 & , x_i \notin S_k \end{cases}, \quad \dots(5)$$

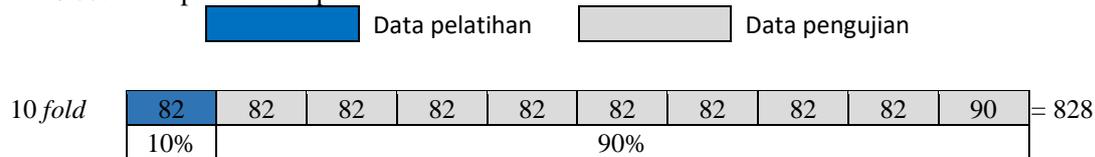
dengan  $S_k$  adalah data terdekat sebanyak k.

Dalam algoritma kNN sebuah data baru diklasifikasikan berdasarkan jarak data baru tersebut dengan tingkat kemiripan data baru terdekat terhadap data yang tersedia. Salah satu metode untuk mengetahui nilai k optimal pada kNN adalah validasi silang atau *cross validation*.

Validasi silang (*Cross Validation*) adalah suatu metode statistik yang digunakan untuk mengetahui rata-rata keberhasilan dari suatu sistem dengan cara melakukan perulangan dengan mengacak atribut masukan sehingga sistem tersebut teruji untuk beberapa atribut input yang acak. Dalam penelitian ini, yang digunakan adalah validasi silang b-fold, dimana data hasil observasi dipartisi menjadi sebanyak b sub-bagian. Dari b sub-bagian tersebut, sebuah sub-bagian digunakan untuk pengujian dan sub-bagian yang lain digunakan untuk pelatihan. Proses validasi silang ini kemudian diulangi sebanyak b kali, dengan setiap sub-bagian digunakan hanya sekali saja. Hasil dari validasi tersebut kemudian dihitung nilai MSE.

**C. Hasil Penelitian dan Pembahasan**

Setelah membagi data menjadi 10 fold, proses kNN dimulai. Menurut pembagian sesuai gambar 3.1, secara tidak langsung, data tersebut sudah terbagi menjadi 10% merupakan data pengujian dan 90% merupakan data pelatihan.



**Gambar 3.1** Skema Pembagian 10 fold dengan Data Sebanyak 828 dan Pembagian Data Pelatihan Sebesar 90% serta Data Pengujian Sebesar 10%.

Mulai dari *fold* pertama atau  $b = 1$  dan  $k = 1$  yang artinya nilai tetangga terdekat yang diambil adalah 1, atau jarak paling dekat dengan data pengujian adalah 1 data pelatihan. Pada *fold* pertama, 82 data pertama digunakan untuk pengujian dan data lainnya sebanyak 746 digunakan untuk pelatihan. Selanjutnya menghitung jarak Euclidean sesuai persamaan (3), untuk jarak antara data pelatihan pertama dengan jarak pengujian pertama, dan seterusnya sebanyak data pengujian.

**Tabel 1.** Nilai Jarak Euclidean pada Fold Pertama dan  $k = 1$

RMS	Waktu	k	$D_{i,j}$
0,43057165	10	1	
0,44867262	20		
...	...		
0,56603513	810		
0,55452201	820		
0,54377114	830		$D_{1,1} = 0,54377114 - 0,43057165 = 0,1131995$
0,55477509	840		$D_{2,1} = 0,55477509 - 0,43057165 = 0,1242034$
0,50407095	850		0,0734993
...	...		...
0,28632120	8270		0,1442505
0,29836365	8280		0,1322080
...	...		...
0,54377114	830		$D_{1,82} = 0,54377114 - 0,55452201 = 0,0107509$
0,55477509	840		$D_{2,82} = 0,55477509 - 0,55452201 = 0,0002531$
0,50407095	850		0,0504511
...	...		...
0,28632120	8270		0,2682008
0,29836365	8280		0,2561584

Dalam tabel tersebut ditulis  $D_{1,1}$ , artinya jarak yang dihitung pada fold pertama dengan nilai  $k = 1$  adalah data pelatihan pertama terhadap data pengujian pertama, sampai  $D_{1,82}$  yang artinya jarak yang dihitung adalah data pelatihan pertama terhadap data pengujian ke-82. Selanjutnya jarak tersebut diurutkan dari yang terkecil sampai terbesar, seperti pada tabel 3.2.

**Tabel 2.** Nilai Jarak (D) yang Diurutkan dari yang Terkecil sampai Terbesar pada Fold Pertama dan k = 1

RMS	Waktu	K	D <sub>i,j</sub>	D_Sort
0,43095729	3180	1	D <sub>318,1</sub>	0,00038564
0,43098414	3610		D <sub>361,1</sub>	0,00041249
0,43012502	3110		D <sub>311,1</sub>	0,00044663
...	....		...	....
0,94881654	3000		D <sub>300,1</sub>	0,51824489
...	....		...	....
0,55440738	1710		D <sub>171,82</sub>	0,00011463
0,55466817	1410		D <sub>141,82</sub>	0,00014616
0,55477509	840		D <sub>84,82</sub>	0,00025308
...	....		...	....
0,94881654	3000		D <sub>300,82</sub>	0,39429453

Langkah selanjutnya yaitu menghitung  $W_{ki}(x)$  sesuai dengan persamaan (5). Dalam menghitung  $W_{ki}(x)$ , diberi nilai 1 jika data terdekat dengan data pengujian sebanyak 1, karena k = 1, dan diberi nilai 0 untuk lainnya, seperti pada tabel 3.3

$$W_{ki}(x) = \begin{cases} 1 & , x_i \in S_k \\ 0 & , x_i \notin S_k \end{cases}, \text{ dengan } S_k \text{ adalah data terdekat sebanyak k.}$$

**Tabel 3.** Nilai  $W_{ki}(x)$  pada Fold Pertama dan k = 1

RMS	Waktu	k	D <sub>i,j</sub>	D_Sort	$W_{ki}(x)$
0,43095729	3180	1	D <sub>318,1</sub>	0,00038564	1
0,43098414	3610		D <sub>361,1</sub>	0,00041249	0
0,43012502	3110		D <sub>311,1</sub>	0,00044663	0
...	....		...	....	...
0,94881654	3000		D <sub>300,1</sub>	0,51824489	0
...	....		...	....	...
0,55440738	1710		D <sub>171,82</sub>	0,00011463	1
0,55466817	1410		D <sub>141,82</sub>	0,00014616	0
0,55477509	840		D <sub>84,82</sub>	0,00025308	0
...	....		...	....	...
0,94881654	3000		D <sub>300,82</sub>	0,39429453	0

Selanjutnya menghitung estimasi kNN  $\hat{f}_k(x)$  sesuai dengan persamaan (4),  
 $\hat{f}_k(x) = \frac{1}{1} \times ((1 \times 0,43095729) + (0 \times 0,43098414) + \dots + (0 \times 0,94881654))$   
 $= 0,43095729$

**Tabel 4.** Nilai Estimasi kNN  $\hat{f}_k(x)$  pada Fold Pertama dan k = 1

RMS	Waktu	k	D <sub>i,j</sub>	D_Sort	$W_{ki}(x)$	$\hat{f}_k(x)$
0,43095729	3180	1	D <sub>318,1</sub>	0,00038564	1	0,43095729
0,43098414	3610		D <sub>361,1</sub>	0,00041249	0	
0,43012502	3110		D <sub>311,1</sub>	0,00044663	0	
...	....		...	....	...	
0,94881654	3000		D <sub>300,1</sub>	0,51824489	0	
...	....		...	....	...	
0,55440738	1710		D <sub>171,82</sub>	0,00011463	1	0,55440738
0,55466817	1410		D <sub>141,82</sub>	0,00014616	0	
0,55477509	840		D <sub>84,82</sub>	0,00025308	0	
...	....		...	....	...	
0,94881654	3000		D <sub>300,82</sub>	0,39429453	0	

Nilai estimasi kNN  $\hat{f}_k(x)$  ini adalah nilai hasil latih data pengujian, dimana untuk k = 1 hasilnya sama dengan nilai data asli karena nilai tetangga yang digunakan adalah 1. Langkah

selanjutnya adalah menghitung MSE untuk setiap  $\hat{f}_k(x)$  yang selanjutnya dihitung rata-rata dari MSE.

$$MSE_{1,1,1} = \frac{1}{746} \times ((0,43095729 - 0,43095729) + (0,43095729 - 0,0,43098414) + \dots + (0,43095729 - 0,94881654)) = 0,17823573$$

**Tabel 5.** Nilai MSE pada Fold Pertama dan k = 1

RMS	Waktu	K	D <sub>i,j</sub>	D_Sort	W <sub>ki</sub> (x)	$\hat{f}_k(x)$	MSE <sub>k,b,d</sub>
0,43095729	3180	1	D <sub>318,1</sub>	0,00039	1	0,43095729	0,17823573
0,43098414	3610		D <sub>361,1</sub>	0,00041	0		
0,43012502	3110		D <sub>311,1</sub>	0,00045	0		
...	....		...	....	...		
0,94881654	3000		D <sub>300,1</sub>	0,51824	0		
...	....		...	....	...		
0,55440738	1710		D <sub>171,82</sub>	0,00011	1	0,55440738	0,17807266
0,55466817	1410		D <sub>141,82</sub>	0,00015	0		
0,55477509	840		D <sub>84,82</sub>	0,00025	0		
...	....		...	....	...		
0,94881654	3000		D <sub>300,82</sub>	0,39429	0		

Lalu dihitung rata-rata dari MSE, menjadi MSE<sub>1,1</sub> yang artinya MSE untuk fold pertama dan k = 1, seperti berikut.

$$MSE_{1,1} = \frac{1}{82} \times (0,17823573 + 0,17821457 + \dots + 0,17807266) = 0,1781792936$$

Jadi, dengan ini didapatkan nilai MSE pertama saat fold pertama dan k = 1. Semua langkah diatas terus diulangi sampai k = 25 dan hasilnya seperti pada tabel 3.6 berikut.

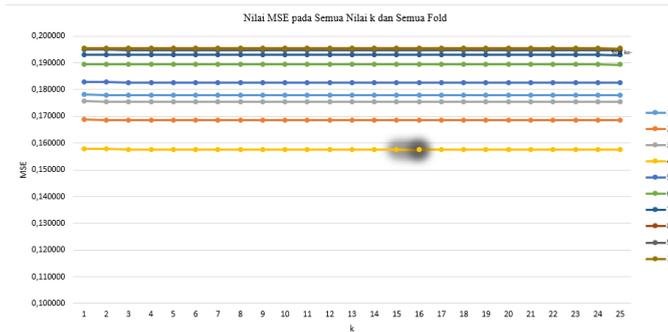
**Tabel 6.** Nilai MSE pada Fold Pertama (b = 1)

k	MSE <sub>k,1</sub>
1	0,1781792936
2	0,1780265965
...	...
24	0,1778874130
25	0,1778870188

Tabel 3.6 diatas menunjukkan hasil nilai MSE pada k mulai dari k = 1 sampai k = 25 pada fold pertama (b = 1). Setelah itu, melakukan semua langkah diatas sampai fold terakhir (b = 10) dan hasilnya seperti pada tabel 3.7 berikut.

**Tabel 7.** Nilai MSE pada Semua Nilai k dan Semua Fold

Fold/k	1	2	3	4	5	6	7	8	9	10
1	0,178179	0,168834	0,175650	0,157918	0,182833	0,189493	0,192989	0,195253	0,194858	0,195520
2	0,178027	0,168626	0,175482	0,157714	0,182709	0,189409	0,192926	0,195203	0,194806	0,195464
...	...	...	...	...	...	...	...	...	...	...
15	0,177894	0,168448	0,175336	0,157580	0,182601	0,189335	0,192871	0,195159	0,194760	0,195415
16	0,177893	0,168447	0,175335	<b>0,157579</b>	0,182600	0,189335	0,192871	0,195158	0,194760	0,195415
17	0,177892	0,168446	0,175334	0,157580	0,182599	0,189334	0,192870	0,195158	0,194759	0,195414
...	...	...	...	...	...	...	...	...	...	...
25	0,177887	0,168442	0,175327	0,157586	0,182594	0,189331	0,192868	0,195156	0,194757	0,195412



**Gambar 2.** Nilai MSE pada Semua Nilai k dan Semua Fold

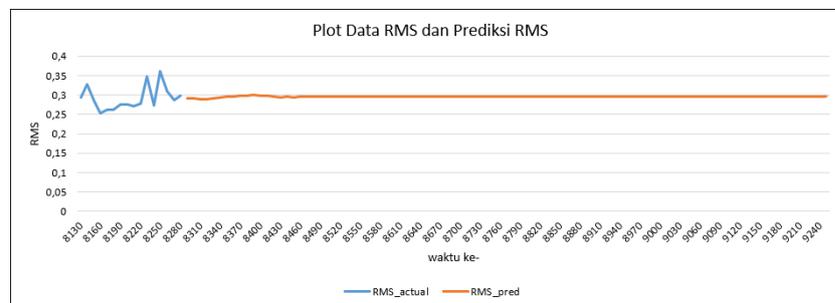
Setelah mendapatkan nilai MSE pada semua nilai k dan semua fold, mencari nilai MSE yang terkecil. Nilai MSE terkecil diperoleh pada nilai k = 16 dengan nilai MSE = 0,157579. Jadi dapat disimpulkan nilai k optimum yang didapatkan adalah 16. Setelah mendapatkan nilai k optimum, selanjutnya adalah memprediksi RMS untuk waktu seterusnya. Setelah mengetahui nilai k yang akan digunakan yakni 16, selanjutnya akan memprediksi RMS pada waktu selanjutnya. Karena diperoleh nilai k = 16, maka ketika akan memprediksi RMS sebanyak 97-lag kedepan, yaitu pada data 829 sampai 925, menggunakan sebanyak 16 data terdekat dari data yang akan diprediksi, seperti berikut.

$$RMS_{8290} = \frac{1}{16} \times (RMS_{8280} + RMS_{8270} + \dots + RMS_{8130})$$

$$RMS_{8290} = \frac{1}{16} \times (0,29836365 + 0,28632120 + 0,31043948 + 0,36081668 + 0,27269707 + \dots + 0,29414655) = 0,29113264$$

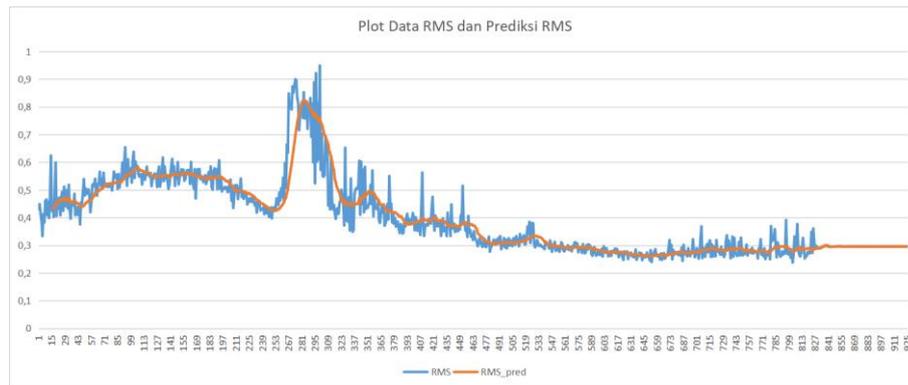
**Tabel 8.** Prediksi RMS 97-lag Kedepan pada RMS ke-829 sampai RMS ke-925

waktu	RMS_actual	RMS_pred
10	0,43057165	
20	0,44867262	
...	...	
8260	0,31043948	
8270	0,2863212	
8280	0,29836365	
8290		0,291132643
8300		0,290944273
...		...
9240		0,295925922
9250		0,295925892

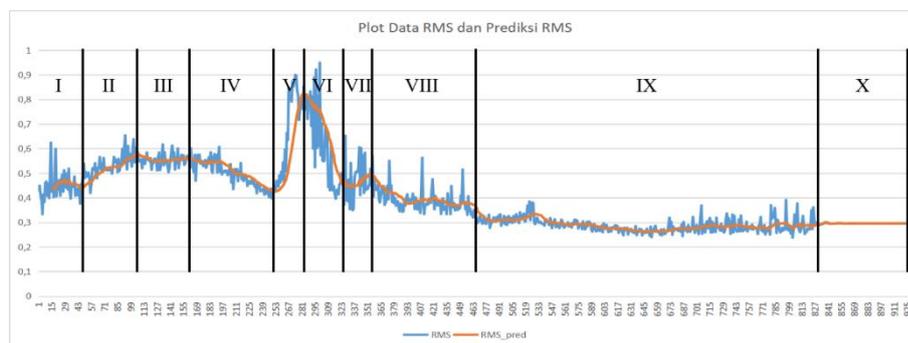


**Gambar 3.** Plot Data RMS dan Prediksi RMS

Gambar 4. dibawah adalah plot data RMS dan prediksi RMS sebanyak 925 dengan menggunakan k = 16. Dapat dilihat dari gambar tersebut bahwa, prediksi RMS mengalami smoothing atau pelembutan yang sesuai dengan data aktualnya.



**Gambar 4.** Plot Data RMS dan Prediksi RMS sebanyak 925



**Gambar 5.** Pembagian Kelompok pada Plot RMS dan Prediksi RMS.

Disini akan dijelaskan bagaimana keadaan mesin bearing berdasarkan plot RMS beserta prediksi RMS yang sudah terbagi menjadi 10 kelompok. Kelompok pertama pada romawi I, keadaan mesin normal karena masih dalam tahap awal mesin dijalankan, pada kelompok kedua mulai ada kenaikan nilai RMS, kenaikan RMS pada kelompok kedua ini tidak ekstrim sehingga dapat disebutkan keadaan mesin masih normal. Pada kelompok ketiga, keadaan menunjukkan normal karena RMS pada saat itu terlihat konstan, sedangkan pada kelompok keempat terlihat ada penurunan yang cukup mencurigakan, akan tetapi mesin masih bekerja dalam keadaan normal. Selanjutnya pada kelompok kelima terlihat ada kenaikan yang cukup ekstrim, disini diduga bahwa mesin telah mengalami kerusakan dan pengoperasian mesin akan dihentikan, akan tetapi tidak berselang lama pada kelompok selanjutnya yaitu kelompok keenam, RMS mengalami penurunan kembali dan nyaris sama nilainya dengan kelompok keempat. Keadaan ini bisa terjadi dan diduga keadaan mesin saat itu sudah mengalami kerusakan (pada kelompok kelima) akan tetapi dapat berjalan normal kembali ada kelompok selanjutnya (kelompok keenam). Pada kelompok ketujuh terlihat ada kenaikan kembali, akan tetapi tidak berselang lama, keadaan kelompok selanjutnya menurun kembali seperti pada kelompok kedelapan disertai dengan sedikit penurunan, disini terlihat bahwa mesin terlihat mengalami penurunan pada pengoperasiannya. Pada kelompok selanjutnya yaitu kelompok kesembilan, RMS terlihat konstan, namun dengan sedikit fluktuasi, disini keadaan mesin seperti normal akan tetapi, pada kelompok selanjutnya prediksi RMS konstan.

#### D. Kesimpulan

Kesimpulan dari penelitian yang sudah dilakukan dalam skripsi ini adalah bahwa metode *k-Nearest Neighbor* berhasil diaplikasikan pada data bearing2\_3. Nilai *k* optimum yang didapatkan dalam proses *k-Nearest Neighbor* ini dibantu dengan RStudio sebesar 16 yang memiliki nilai MSE terkecil yaitu sebesar 0,157579. Dari hasil *k* optimum yang sudah didapatkan, kemudian melakukan prediksi RMS, prediksi berhasil dilakukan dan dihitung sebanyak 97 titik, dari RMS ke-829 sampai RMS ke-925. Model degradasi dapat

diinterpretasikan dengan baik, sesuai dengan data aktual, keadaan sebenarnya dan juga proses *smoothing*.

#### Daftar Pustaka

- [1] Harris T.A dan Kotzalas M.N. 2007. *Rolling Bearing Analysis 5rd Edition*. USA: Taylor and Francis Group, LLC.
- [2] Keputusan Menteri Negara Lingkungan Hidup Nomor : Kep-49/Menlh/11/1996 Tentang Baku Tingkat Getaran, diakses di <http://www.cets-iii.org/BML/Getaran/kepmen4996/> pada 18 Juli 2019.
- [3] Nurhadiyanto, Didik. (2015). *Getaran Struktur*. Yogyakarta: K-Media
- [4] Han, J. Kamber, M. 2001. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers: San Francisco.
- [5] Bode, A. (2017). K-Nearest Neighbor dengan Feature Selection Menggunakan Backward Elimination untuk Prediksi Harga Komoditi Kopi Arabika. *ILKOM Jurnal Ilmiah Volume 9 Nomor 2 Agustus 2017*. Universitas Ichsan Gorontalo
- [6] Renaud, Simon Petit, Thierry Dencœux. (2003). Nonparametric Regression Analysis of Uncertain and Imprecise Data Using Belief Functions. *International Journal of Approximate Reasoning 35 (2004) 1–28*. France
- [7] Pandie, E. S. Y. (2012). *Sistem Informasi Pengambilan Keputusan Pengajuan Kredit dengan Algoritma K-Nearest Neighbour (Studi Kasus: Koperasi Simpan Pinjam)*. Tesis S2 Program Studi Magister Sistem Informasi, Universitas Diponegoro
- [8] Rochmatullah. (2009). *Modifikasi Fuzzy Neuro Learning Vector Quantitation Menggunakan Particle Swarm Optimization untuk Sistem Penciuman Elektronik*. Tesis S2. Fakultas Ilmu Komunikasi. Universitas Indonesia