

PREDIKSI *REMAINING USEFUL LIFE* (RUL) PADA JET ENGINE SEBAGAI UPAYA *PREDICTIVE MAINTENANCE* BERBASIS *MACHINE LEARNING*

*Bagus Sajiwo¹, Toni Prahasto², Achmad Widodo²

¹Mahasiswa Jurusan Teknik Mesin, Fakultas Teknik, Universitas Diponegoro

²Dosen Jurusan Teknik Mesin, Fakultas Teknik, Universitas Diponegoro

Jl. Prof. Sudharto, SH., Tembalang-Semarang 50275, Telp. +62247460059

*E-mail: bsajiwo134@gmail.com

Abstrak

Predictive maintenance adalah pendekatan revolusioner dalam pemeliharaan industri yang bertujuan untuk mengurangi kerusakan mesin yang tidak terduga dan mengoptimalkan waktu pemeliharaan. *Jet engine* adalah komponen kunci dalam industri penerbangan yang memerlukan pemeliharaan yang tepat waktu untuk menjaga keandalan dan keamanan operasional.. Kerusakan pada mesin jet adalah isu kritis dalam industri penerbangan.. Salah satu tantangan utama dalam pemeliharaan mesin jet adalah memprediksi kapan komponen atau sistem akan mengalami kerusakan sehingga pemeliharaan dapat dilakukan sebelum terjadinya kegagalan yang tidak terduga. Dalam upaya untuk meningkatkan efisiensi pemeliharaan mesin jet, penelitian ini fokus pada penggunaan teknik *machine learning* untuk memprediksi *Remaining Useful Life* (RUL) *jet engine*. Penelitian ini menggunakan dataset yang mencakup data operasional mesin jet dan pembacaan sensor dari NASA *Turbofan Jet Engine*. Penelitian ini bertujuan untuk mengidentifikasi fitur-fitur yang mempengaruhi RUL *jet engine* dengan metode *feature selection*. Tiga model klasifikasi yang digunakan dalam penelitian ini adalah *K-Nearest Neighbors* (KNN), *Support Vector Machine* (SVM), dan *Random Forest*. Model-model ini dilatih menggunakan data historis dan diuji pada data tes untuk mengetahui seberapa akurat model prediksi RUL terhadap RUL aktual. Evaluasi kinerja model-model ini dilakukan dengan menggunakan metrik R2 dan RMSE. Hasil penelitian ini memberikan wawasan berharga dalam pemeliharaan prediktif mesin jet dan menunjukkan manfaat signifikan dari penerapan *machine learning* dalam meningkatkan efisiensi dan keandalan pesawat.

Kata kunci: *machine learning*; mesin jet; *predictive maintenance*; *rul*

Abstract

Predictive maintenance is a revolutionary approach to industrial maintenance that aims to reduce unexpected machine breakdowns and optimize maintenance time. *Jet engines* are key components in the aviation industry that require timely maintenance to maintain operational reliability and safety. *Jet engine failure* is a critical issue in the aviation industry. One of the main challenges in *jet engine* maintenance is predicting when a component or system will fail. damage so that maintenance can be carried out before an unexpected failure occurs. In an effort to improve *jet engine* maintenance efficiency, this research focuses on using *machine learning* techniques to predict the *Remaining Useful Life* (RUL) of *jet engines*. This research uses a dataset that includes *jet engine* operational data and sensor readings from the NASA *Turbofan Jet Engine*. This research aims to identify features that influence *jet engine* RUL using the *feature selection* method. The three classification models used in this research are *K-Nearest Neighbors* (KNN), *Support Vector Machine* (SVM), and *Random Forest*. These models are trained using historical data and tested on test data to find out how accurate the RUL prediction model is to the actual RUL.. Performance evaluation of these models is carried out using R2 and RMSE metrics. The results of this research provide valuable insights into *predictive maintenance* of *jet engines* and demonstrate the significant benefits of applying *machine learning* in improving aircraft efficiency and reliability.

Keyword: *jet engines*; *machine learning*; *predictive maintenance*; *rul*

1. Pendahuluan

Pemeliharaan merupakan hal yang penting dalam sebuah industri. Pemeliharaan di industri manufaktur memiliki tujuan utama meningkatkan ketersediaan aset. Strategi pemeliharaan yang buruk dapat mengakibatkan produktivitas mesin yang lebih rendah. Kurangnya pemeliharaan pada mesin juga dapat menyebabkan kerugian finansial. Oleh karena itu, diperlukan sebuah strategi pemeliharaan yang baik untuk mencegah terjadinya *downtime* yang tidak direncanakan. *Predictive maintenance* adalah metode yang menggunakan alat prediksi untuk menentukan kapan pemeliharaan harus dilakukan. Teknik ini didasari pada pemantauan (*monitoring*) berkelanjutan pada peralatan sehingga memungkinkan dilakukannya pemeliharaan jika diperlukan [1].

Berkembangnya industri 4.0, data dapat dianggap sebagai aset penting dan krusial untuk struktural pemeriksaan kesehatan peralatan mesin yang digunakan di industri. Pengolahan data yang besar dan struktural sangat diperlukan untuk mendiagnosa ketidaknormalan pada mesin-mesin industri. Hal ini mengarahkan data yang besar tersebut untuk diolah

menjadi sebuah visualisasi data yang akan dilanjutkan ke tahap simpulan hasil diagnosa ketidaknormalan. Pengolahan data ini dapat dilakukan dengan metode machine learning yang sangat membantu untuk melakukan predictive maintenance terhadap seluruh mesin industri dan jet engine.

Pada penelitian ini, beberapa pendekatan dan algoritma akan digunakan pada data yang sama untuk menentukan hasil analisis yang paling optimal. Hasil dari masing-masing pendekatan akan dibandingkan dan akan dipilih yang memiliki nilai akurasi tertinggi.

2. Dasar Teori

2.1 Predictive Maintenance

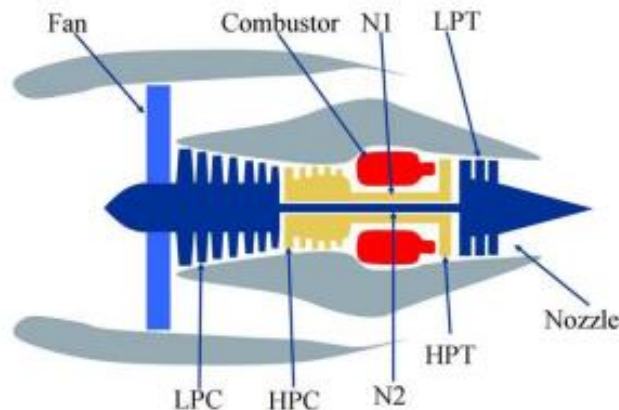
Selama tahun-tahun pertama revolusi industri, umumnya pemeliharaan dilakukan secara reaktif (*Reactive Maintenance*), yang berarti mesin diperbaiki hanya ketika tidak beroperasi. Strategi pemeliharaan ini sangat mempengaruhi produktivitas mesin menjadi tidak aktif selama waktu yang diperlukan untuk memperbaikinya. Menyelesaikan masalah ini, sebuah pendekatan yang dikenal sebagai *Preventive Maintenance* diusulkan, yang melibatkan pemeliharaan mesin secara berkala dan komponen untuk mencegah inefisiensi dalam jangka waktu lama. *Preventive Maintenance* memecahkan masalah pencegahan kerusakan tetapi mempunyai kelemahan yaitu pemeliharaan dilakukan tanpa mempertimbangkan status mesin [2].

Pemeliharaan digambarkan sebagai kumpulan kegiatan untuk mengembalikan kinerja peralatan ke fungsi yang ditentukan. Untuk meminimalkan biaya operasi dan pemeliharaan mesin, diperlukan strategi pemeliharaan yang efisien. Salah satu cara efektif untuk meningkatkan ketersediaan sistem dan mengurangi biaya pemeliharaan adalah dengan bermigrasi dari pemeliharaan berbasis kegagalan dan perbaikan yang bersifat reaktif menjadi pemeliharaan berbasis prediksi (*Predictive Maintenance*) dan pencegahan kondisi [3].

2.2 Jet Engine

Mesin Mesin jet merupakan sebuah macam mesin pembakaran dalam menghirup udara yang sering digunakan dalam pesawat. Prinsip seluruh mesin jet pada dasarnya sama, mereka mempercepat massa (udara dan hasil pembakaran) ke satu arah dan dari hukum gerak Newton ketiga mesin akan mengalami dorongan ke arah yang berlawanan [4].

Turbofan merupakan mesin yang umum dari turunan mesin-mesin turbin gas untuk menggerakkan pesawat terbang komersial maupun pesawat tempur. Mesin ini sebenarnya merupakan sebuah mesin *by-pass* dimana beberapa dari udara dipadatkan dan disalurkan ke ruang pembakaran, sementara sisanya dengan kepadatan rendah disalurkan sekeliling bidang luar ruang pembakaran. Sekaligus udara tersebut berfungsi untuk mendinginkan suhu ruang pembakaran [5].



Gambar 1. Parameter-Parameter Mesin Jet [6]

2.3 Machine Learning

Machine Learning (ML) merupakan bidang studi yang fokus kepada desain dan analisis algoritma sehingga memungkinkan komputer untuk dapat belajar. Menurut Samuel, ML berisi sebuah algoritma yang bersifat *generic* (umum) dimana algoritma tersebut dapat menghasilkan sesuatu yang menarik atau bermanfaat dari sejumlah data tanpa harus menulis kode yang spesifik [7].

Pembagian *Machine Learning* berdasarkan cara belajarnya dibagi menjadi tiga kelompok yaitu *Supervised Learning*, *Unsupervised Learning*, dan *Reinforcement Learning*. *Supervised Learning* menggunakan serangkaian pelatihan untuk mengajarkan model guna menghasilkan output yang diinginkan. *Supervised Learning* dapat dipisahkan menjadi dua jenis masalah saat penggalan data, yaitu klasifikasi dan regresi.

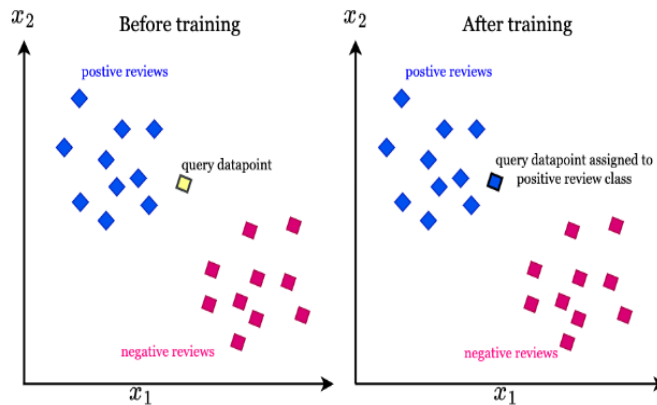
2.4 Klasifikasi

Algoritma Klasifikasi adalah salah satu metode dalam proses KDD (*Knowledge Discovery in Databases*) yang ada pada tahap *data mining* untuk mengidentifikasi setiap kelompok berdasarkan kasus dan kemudian dikelompokkan datanya

berdasarkan atribut. Klasifikasi ini merupakan metode *supervised learning* yang mencoba menemukan hubungan antara atribut *input* dan atribut target. Tujuan klasifikasi ini untuk meningkatkan kehandalan hasil yang diperoleh dari data.

2.4.1 *K-Nearest Neighbors (KNN)*

K-nearest neighbour (KNN) sebagian besar digunakan untuk mengklasifikasikan titik data meskipun dapat melakukan regresi juga. Algoritma *K*-Nearest Neighbors mengklasifikasikan titik data baru ke dalam kategori tertentu berdasarkan kemiripannya dengan titik data lain dalam kategori tersebut. Algoritme KNN memiliki kebebasan untuk memilih bentuk fungsional apa pun dari data pelatihan untuk memetakan data masukan [8].

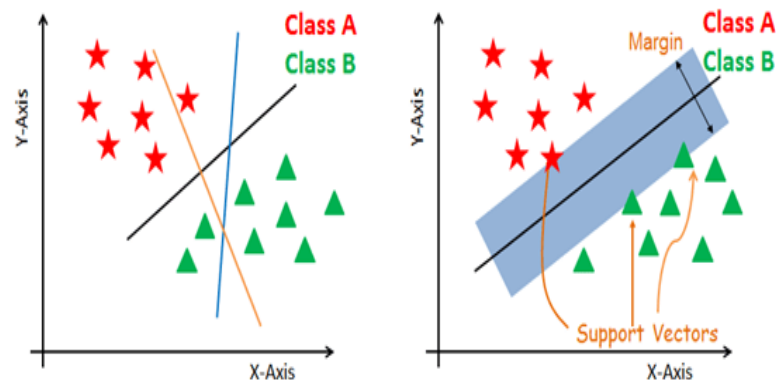


Gambar 2 Klasifikasi Menggunakan *K-Nearest Neighbors* [8]

2.4.2 *Support Vector Machine (SVM)*

Secara umum, Support Vector Machine dianggap sebagai pendekatan klasifikasi, tetapi dapat digunakan dalam kedua jenis masalah klasifikasi dan regresi. Itu dapat dengan mudah menangani beberapa variabel kontinu dan kategorikal. SVM membangun hyperplane dalam ruang multidimensi untuk memisahkan kelas-kelas yang berbeda. SVM menghasilkan hyperplane optimal secara iteratif, yang digunakan untuk meminimalkan kesalahan. Ide inti dari SVM adalah untuk menemukan Maximum Marginal Hyperplane (MMH) yang paling baik membagi kumpulan data ke dalam kelas-kelas [9].

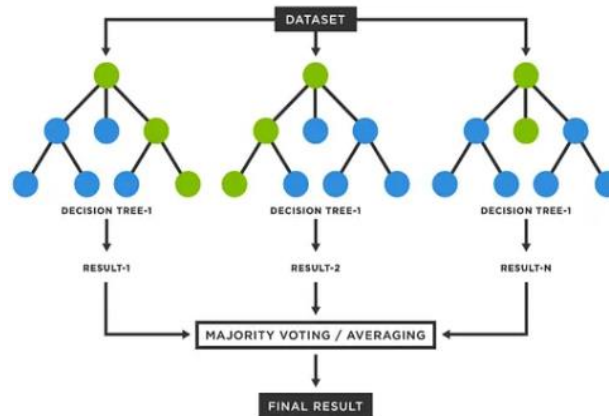
Tujuan utama SVM adalah memisahkan kumpulan data tertentu dengan cara terbaik. Jarak antara titik terdekat disebut margin. Tujuannya adalah untuk memilih hyperplane dengan margin maksimum yang mungkin antara vektor pendukung dalam kumpulan data tertentu.



Gambar 3 Cara Kerja SVM [9]

2.4.3 *Random Forest*

Random Forest adalah algoritma machine learning ansambel lain yang mengikuti teknik bagging. Ini merupakan perpanjangan dari algoritma bagging estimator. Bagging estimator adalah salah satu jenis ensemble learning yang melibatkan penggunaan model dasar yang serupa untuk melakukan prediksi dan kemudian menggabungkan hasil prediksi tersebut. Penduga dasar di random forest adalah decision tree. Tidak seperti penduga meta bagging, random forest secara acak memilih serangkaian fitur yang digunakan untuk menentukan pemisahan terbaik di setiap simpul decision tree [10].



Gambar 4 Random Forest Classification [10]

2.5 Prediction Error

Prediction error dalam konteks *machine learning*, merujuk pada selisih antara nilai yang diamati (*actual values*) dan nilai yang diprediksi (*predicted values*) oleh sebuah model prediksi atau algoritma. Ini menggambarkan perbedaan antara hasil aktual dan hasil yang diprediksi untuk sekumpulan variabel input tertentu. *Prediction error* yang lebih rendah menunjukkan bahwa model membuat prediksi yang lebih akurat, sementara *prediction error* yang lebih tinggi mengindikasikan bahwa prediksi model kurang akurat. Meminimalkan *prediction error* merupakan tujuan utama dalam pengembangan dan penyempurnaan model prediksi guna meningkatkan kinerja dan keandalan model tersebut.

2.5.1 R Squared (R^2)

R^2 juga dikenal sebagai koefisien determinasi, adalah ukuran yang digunakan dalam regresi untuk mengevaluasi sejauh mana variabilitas dalam variabel dependen dapat dijelaskan oleh variabel independen dalam model regresi. R^2 memiliki nilai antara 0 dan 1, dimana nilai 1 menunjukkan bahwa semua variasi dalam variabel dependen dapat dijelaskan oleh variabel independen dan nilai 0 menunjukkan bahwa variabel independen tidak dapat menjelaskan variasi apa pun dalam variabel dependen. Maka semakin tinggi nilai R^2 , semakin baik model regresi dalam menjelaskan variasi dalam data. Nilai R^2 dapat dicari menggunakan formula sebagai berikut :

$$R^2 = \frac{SSR}{SST} = 1 - \frac{(y_1 - \hat{y})^2}{(y_1 - \bar{y})^2}$$

Keterangan :

SSR (*Sum of Squared Residuals*) : jumlah kuadrat dari selisih antara nilai prediksi model (\hat{y}) dan nilai observasi yang diamati (y) untuk setiap pengamatan.

SST (*Total Sum of Squares*) : jumlah kuadrat dari selisih antara nilai observasi yang diamati (y) dan rata-rata data (\bar{y}) untuk setiap pengamatan.

2.5.2 Root Mean Squared Error (RMSE)

RMSE adalah metrik evaluasi yang digunakan untuk mengukur sejauh mana kesalahan prediksi model regresi terhadap nilai sebenarnya. RMSE menghitung akar rata-rata dari selisih kuadrat antara nilai prediksi dan nilai sebenarnya. RMSE memberikan gambaran tentang seberapa dekat prediksi model dengan nilai sebenarnya. Semakin rendah nilai RMSE, semakin baik model dalam menghasilkan prediksi yang akurat. Nilai RMSE dapat dicari menggunakan formula sebagai berikut :

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_1 - \hat{y})^2}$$

Keterangan :

n : jumlah sampel atau pengamatan.

Σ (sigma) : simbol untuk menggambarkan penjumlahan (dalam hal ini, penjumlahan dari 1 hingga n).

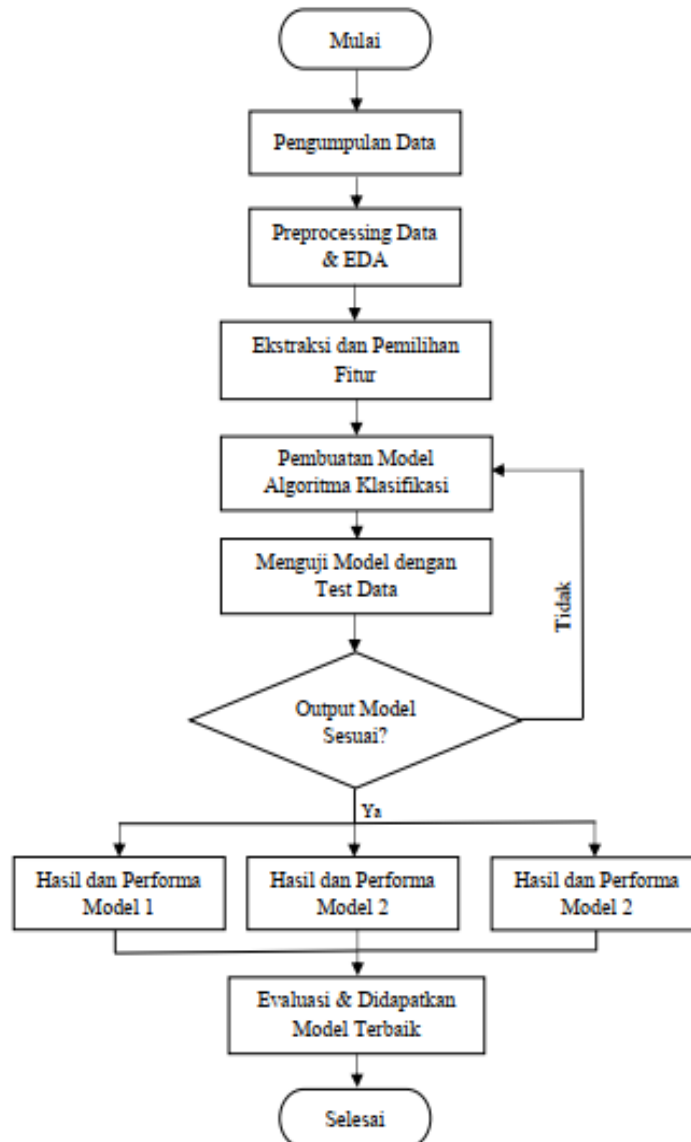
y : nilai aktual atau observasi yang sebenarnya.

\hat{y} : adalah nilai yang diprediksi oleh model.

3. Metode Penelitian

3.1 Diagram Alir Penelitian

Diagram Alir penelitian dapat dilihat pada Gambar 5 dibawah ini.



Gambar 5 Diagram alir penelitian

3.2 Software yang Digunakan Dalam Penelitian

Dalam melakukan penelitian, penulis menggunakan beberapa software yang membantu untuk mengolah data-data yang tersedia dan mengambil kesimpulan. Software yang digunakan dalam penelitian adalah sebagai berikut.

3.2.1 Python

Python adalah bahasa pemrograman tingkat tinggi yang populer dalam pengembangan perangkat lunak, analisis data, kecerdasan buatan (AI), dan machine learning. Dalam bahasa pemrograman Python, modul adalah file yang berisi definisi variabel, fungsi, dan kelas yang dapat digunakan kembali. Modul memungkinkan untuk mengorganisasi kode Python menjadi unit yang lebih kecil dan terstruktur. Beberapa modul Python yang digunakan untuk menunjang penelitian ini yaitu:

1. NumPy (Numerical Python)
NumPy digunakan untuk manipulasi data, termasuk data dalam bentuk matriks atau *array*.
2. Pandas
Pandas menyediakan struktur data tingkat tinggi (*dataframe*) yang berguna dalam analisis dan manipulasi data.
3. Scikit-learn
Scikit-learn yang menyediakan algoritma *machine learning*, pemrosesan data, evaluasi model, dan banyak lagi.
4. Matplotlib

5. Matplotlib adalah modul untuk visualisasi data dan pembuatan grafik.
5. Seaborn
Seaborn menyediakan palet warna yang estetik, digunakan untuk membuat visualisasi data yang menarik.
6. Statsmodels
Statsmodels menyediakan alat untuk analisis statistik dan pemodelan data. Berguna untuk analisis regresi, uji hipotesis, dan pemodelan statistik.

3.2.2 Google Colab

Google Colab adalah layanan komputasi awan berbasis Python yang disediakan oleh Google. Ini memungkinkan pengguna untuk menulis dan mengeksekusi kode Python di lingkungan berbasis web tanpa perlu menginstal atau mengonfigurasi perangkat lunak di komputer lokal.

3.2.3 Kaggle

Kaggle adalah komunitas *online data scientist* dan praktisi *machine learning*. Kaggle memungkinkan pengguna untuk menemukan dan menerbitkan data, menjelajahi dan membuat model dalam lingkungan *data science* berbasis *web*, bekerja dengan *data scientist* dan insinyur *machine learning* lainnya, dan mengikuti kompetisi untuk memecahkan tantangan *data science*.

3.3 Pengumpulan Data

Kumpulan data ini adalah kumpulan data versi Kaggle yang dapat diakses oleh publik untuk pemodelan degradasi aset mesin turbojet dari NASA. Ini mencakup data simulasi Run-to-Failure dari mesin jet turbofan. Simulasi degradasi mesin dilakukan dengan menggunakan C-MAPSS. Empat set yang berbeda disimulasikan di bawah kombinasi kondisi operasional dan mode kesalahan yang berbeda. Beberapa saluran sensor digunakan untuk mencatat karakterisasi evolusi kerusakan. Kumpulan data disediakan oleh Prognostics CoE di NASA Ames. Saluran sensor yang digunakan berjumlah 21 unit yang berfungsi untuk mengukur data-data fitur.

3.4 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) merupakan langkah awal pada penelitian ini dengan tujuan untuk memahami apa isi dari data yang digunakan dan bagaimana pola maupun hubungan antar variabel di dalamnya. Berikut alur-alur EDA yang diterapkan untuk memahami dataset mesin jet NASA ini:

1. Membaca Dokumentasi Data
Ini dapat dilakukan dengan cara membaca dokumen atau deskripsi data yang biasanya disertakan dengan *dataset*. Dokumentasi ini dapat berisi informasi tentang variabel-variabel yang ada dalam data, jenis data, dan keterangan tentang setiap kolom.
2. Melihat Sampel
Periksa beberapa baris pertama dari data dengan menggunakan perintah seperti *head()*. Ini akan memberikan gambaran awal tentang bagaimana data terstruktur dengan melihat beberapa sampel data yang ditampilkan pada *notebook*.
3. Statistik Deskriptif
Tujuan dari melakukan statistik deskriptif adalah memberikan gambaran yang jelas dan ringkas tentang data yang sedang dianalisis. Tahap ini juga dapat digunakan untuk menemukan data yang bernilai konstan.
4. Korelasi Data
Korelasi data adalah pengukuran hubungan statistik antara dua atau lebih variabel dalam dataset. Ini mengidentifikasi apakah terdapat hubungan positif, negatif, atau tidak ada hubungan sama sekali antara variabel-variabel tersebut. Untuk membuat *heatmap* korelasi dapat menggunakan fungsi *corr()*.
5. Pembersihan Data
Data yang akan digunakan untuk membuat model estimasi RUL perlu dibersihkan dari variabel yang bernilai konstan, dan variabel-variabel yang berkorelasi tinggi dikarenakan variabel ini tidak signifikan untuk prediksi.

4. Hasil dan Pembahasan

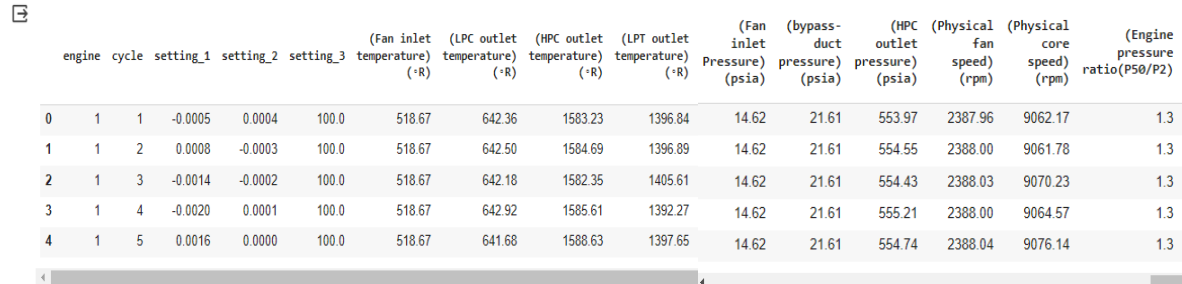
4.1 Penyajian Data Hasil Pengukuran Sensor Operasi Mesin Jet

Dari hasil pengukuran sensor measurement pada Tabel 3.1, didapat *dataset* yang dinamakan FD001, FD002, FD003, FD004. Masing-masing *dataset* ini dibedakan berdasarkan kondisi operasional dan mode kerusakan yang diuji. Pada studi analisis data disini penulis menggunakan *dataset* FD003 sebagai *dataset* yang akan dianalisis mengenai keterhubungan antar *sensor output*-nya. Berikut merupakan kriteria-kriteria dari dataset mesin jet FD003:

- *Data Set* : FD003
- *Train trajectories*: 100

- *Test trajectories*: 100
- *Conditions*: ONE (Sea Level)
- *Fault Modes*: TWO (HPC Degradation, Fan Degradation)

Setiap *dataset* memiliki 3 jenis *dataframe* yang perlu dianalisis keterkaitannya. Tiga jenis *dataframe* tersebut adalah *train dataframe*, *test dataframe*, dan RUL aktual mesin. *Train dataframe* adalah data-data yang memiliki *track record* data yang berakhir pada *failure threshold*. Sedangkan, *test dataframe* mencatat data yang berhenti sesaat sebelum *failure threshold* tercapai. Berikut adalah *sensor output sample* dari *train dataframe* FD003 yang dapat dilihat pada Gambar 6.



engine	cycle	setting_1	setting_2	setting_3	(Fan inlet temperature) (°R)	(LPC outlet temperature) (°R)	(HPC outlet temperature) (°R)	(LPT outlet temperature) (°R)	(Fan inlet Pressure) (psia)	(bypass-duct pressure) (psia)	(HPC outlet pressure) (psia)	(Physical fan speed) (rpm)	(Physical core speed) (rpm)	(Engine pressure ratio(P50/P2))	
0	1	1	-0.0005	0.0004	100.0	518.67	642.36	1583.23	1396.84	14.62	21.61	553.97	2387.96	9062.17	1.3
1	1	2	0.0008	-0.0003	100.0	518.67	642.50	1584.69	1396.89	14.62	21.61	554.55	2388.00	9061.78	1.3
2	1	3	-0.0014	-0.0002	100.0	518.67	642.18	1582.35	1405.61	14.62	21.61	554.43	2388.03	9070.23	1.3
3	1	4	-0.0020	0.0001	100.0	518.67	642.92	1585.61	1392.27	14.62	21.61	555.21	2388.00	9064.57	1.3
4	1	5	0.0016	0.0000	100.0	518.67	641.68	1588.63	1397.65	14.62	21.61	554.74	2388.04	9076.14	1.3

Gambar 6 Sample Data Pembacaan Sensor dari *Dataframe Train* FD003

4.2 Analisis Statistik

Dari sampel *train dataframe* FD003, selanjutnya dilakukan perhitungan terhadap parameter-parameter statistik yaitu *count*, *mean*, *standard deviation*, *minimum value*, *percentile value*, dan *maximum value* menggunakan *describe()* *function* yang disediakan pada *library* *pandas* untuk Python. Hasilnya dapat dilihat pada Gambar 7 dibawah ini.

	count	mean	std	min	25%	50%	75%	max
(Required fan conversion speed)	24720.0	100.000000	0.000000e+00	100.0000	100.0000	100.0000	100.0000	100.0000
setting_3	24720.0	100.000000	0.000000e+00	100.0000	100.0000	100.0000	100.0000	100.0000
(Fan inlet temperature) (°R)	24720.0	518.670000	0.000000e+00	518.6700	518.6700	518.6700	518.6700	518.6700
(Required fan speed)	24720.0	2388.000000	0.000000e+00	2388.0000	2388.0000	2388.0000	2388.0000	2388.0000
(Burner fuel-air ratio)	24720.0	0.030000	1.734759e-17	0.0300	0.0300	0.0300	0.0300	0.0300
(Fan inlet Pressure) (psia)	24720.0	14.620000	3.552786e-15	14.6200	14.6200	14.6200	14.6200	14.6200
setting_2	24720.0	0.000005	2.940429e-04	-0.0006	-0.0002	-0.0000	0.0003	0.0007
setting_1	24720.0	-0.000024	2.193545e-03	-0.0086	-0.0015	-0.0000	0.0015	0.0086

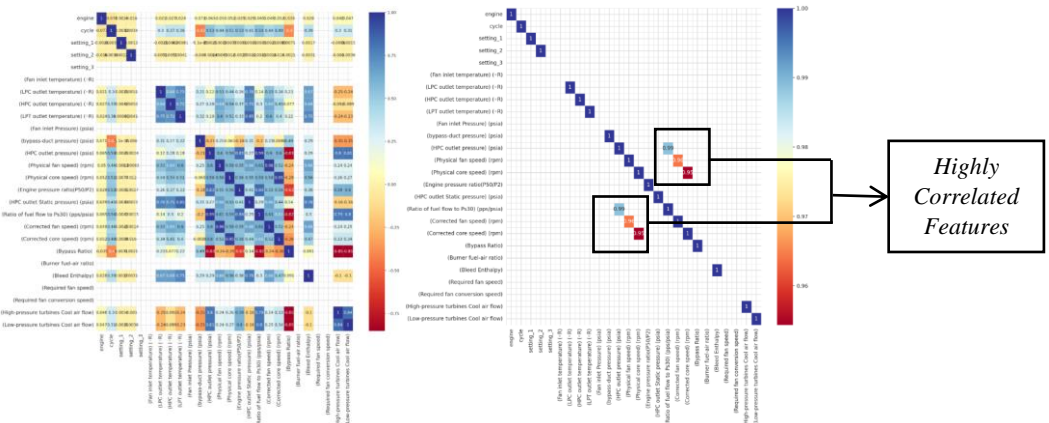
Gambar 7 Deskripsi Statistik *Dataframe* FD003

Pada *dataframe* FD003 terlihat bahwa ada fitur-fitur yang ternyata memiliki nilai konstan. Hal itu dapat dibuktikan pada hasil deskripsi statistik yang memperlihatkan nilai maksimum dan minimum yang sama, dan juga dengan nilai standar deviasinya sebesar 0. Fitur-fitur ini bisa disebut *outlier* karena tidak memiliki pengaruh langsung terhadap proses estimasi RUL jet engine. Terdapat 6 fitur yang bernilai konstan, fitur-fitur tersebut yaitu : Setting 3, Fan Inlet Temperature, Fan Inlet Pressure Burner Fuel-Air Ratio, Required Fan Speed, dan Required Fan Conversion Speed.

4.3 Korelasi Fitur

Fitur berperan penting untuk mengestimasi RUL. Semakin besar selisih nilai data pada masing-masing fitur maka akan semakin mudah untuk melihat tren atau perubahan nilai data seiring bertambahnya siklus operasi. Namun, jika fitur terlalu banyak justru malah akan membuat prediksi RUL menjadi semakin tidak akurat, apalagi jika ternyata ada fitur-fitur yang nilainya sama. Maka dari itu perlu dilihat korelasi antar masing-masing siklus. Tujuannya adalah untuk melihat apakah ada fitur yang mempunyai kemiripan nilai data dilihat dari nilai korelasinya. Jika ada beberapa fitur yang memiliki nilai yang sama, maka lebih baik salah satu dari fitur tersebut didrop agar estimasi RUL lebih terfokus pada fitur-fitur yang nilainya bervariasi.

Nilai korelasi antar fitur berkisar di angka -1 hingga 1. Diantara seluruh nilai korelasi, terdapat fitur yang tergolong fitur-fitur yang memiliki nilai korelasi paling tinggi atau *the highly correlated features*, yaitu fitur-fitur yang nilai korelasinya lebih dari 0.9. Untuk memudahkan pembacaan fitur yang berkorelasi tinggi, maka plot disederhanakan seperti pada Gambar 8 di bawah ini.



Gambar 8 Korelasi Fitur Data Train FD003

Pada Gambar 8 terlihat bahwa ada beberapa fitur yang berkorelasi tinggi yaitu :

- Fitur HPC Outlet Pressure terhadap Ratio Of Fuel Flow To Ps30, dengan nilai korelasi = 0,99
- Fitur Physical Fan Speed terhadap Corrected Fan Speed, dengan nilai korelasi = 0,96
- Fitur Physical Core Speed terhadap Corrected Core Speed, dengan nilai korelasi = 0,95

4.4 Pembersihan Data

Dalam konteks pemodelan statistik, variabel yang tidak berpengaruh mengacu pada variabel yang tidak memiliki dampak signifikan secara statistik terhadap variabel target. Dengan kata lain, ketika menganalisis hubungan antara prediktor (variabel independen) dan hasil (variabel dependen), variabel yang tidak signifikan adalah variabel yang tidak memberikan kontribusi signifikan untuk menjelaskan variasi target.

Seperti yang sudah disebutkan sebelumnya, terdapat fitur yang memiliki nilai konstan pada setiap *cycle*-nya dan berkorelasi tinggi terhadap fitur lain. Fitur-fitur yang tidak signifikan ini dieliminasi dari pemodelan analisis, sehingga tersisa fitur-fitur yang signifikan dan berpengaruh terhadap pemodelan analisis RUL.

4.5 Seleksi Fitur

Seleksi fitur dilakukan untuk mengeliminasi fitur-fitur yang tidak memiliki atau sedikit korelasi dengan fitur yang lain. Metode ini dilakukan dengan cara mencari nilai *p-value* dari tiap fitur yang akan dianalisis keterkaitannya. *P-value* adalah nilai yang menunjukkan *P-value* dari tiap fitur untuk melakukan estimasi. *P-value* dari tiap fitur untuk melakukan estimasi RUL jet engine dapat dilihat pada Gambar 9.

cycle	4.455429545804e-140	(Physical fan speed) (rpm)	1.926592728701e-55
setting_1	3.925181749864e-01	(Physical core speed) (rpm)	0.000000000000e+00
setting_2	8.513792655112e-01	(Engine pressure ratio(P50/P2))	6.765353808479e-116
(LPC outlet temperature) (°R)	1.601015180787e-48	(HPC outlet Static pressure) (psia)	1.765459810702e-209
(HPC outlet temperature) (°R)	1.358333924648e-38	(Bypass Ratio)	1.624240272666e-18
(LPT outlet temperature) (°R)	8.183625454447e-113	(Bleed Enthalpy)	3.693200952449e-39
(bypass-duct pressure) (psia)	7.284711153017e-109	(High-pressure turbines Cool air flow)	2.567917121174e-25
(HPC outlet pressure) (psia)	3.916523970358e-02	(Low-pressure turbines Cool air flow)	3.091274669583e-18

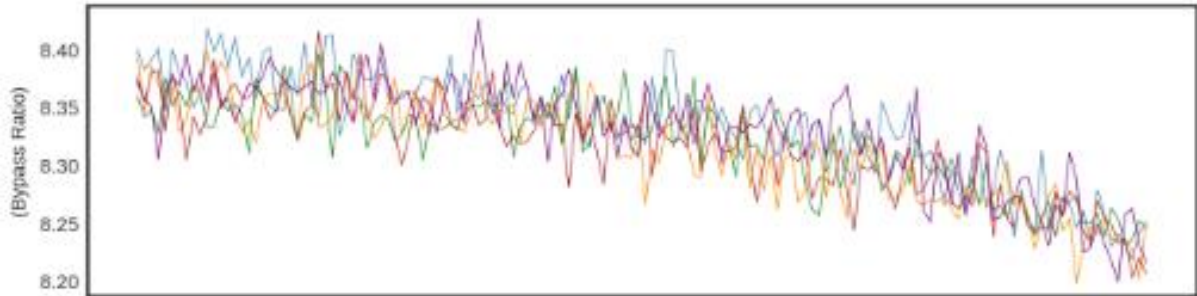
Gambar 9 P-Value Setiap Variabel

Pada Gambar 9 diatas, terlihat bahwa *p-value* yang berada diatas nilai 0.05 adalah fitur Setting 1 dan Setting 2. Masing-masing memiliki nilai *p-value* sebesar 0,39 dan 0,85. Nilai ini jauh melebihi angka 0,05, yang merupakan angka signifikansi yang sudah ditentukan. Sehingga kedua fitur tersebut akan dieliminasi dari daftar fitur yang akan dianalisis untuk melakukan estimasi RUL jet engine karena memiliki signifikansi yang kecil terhadap variabel RUL. Sehingga untuk fitur-fitur yang akan masuk kedalam fitur prediktor dari target setelah melewati tahap pembersihan data dan seleksi fitur adalah sebagai berikut:

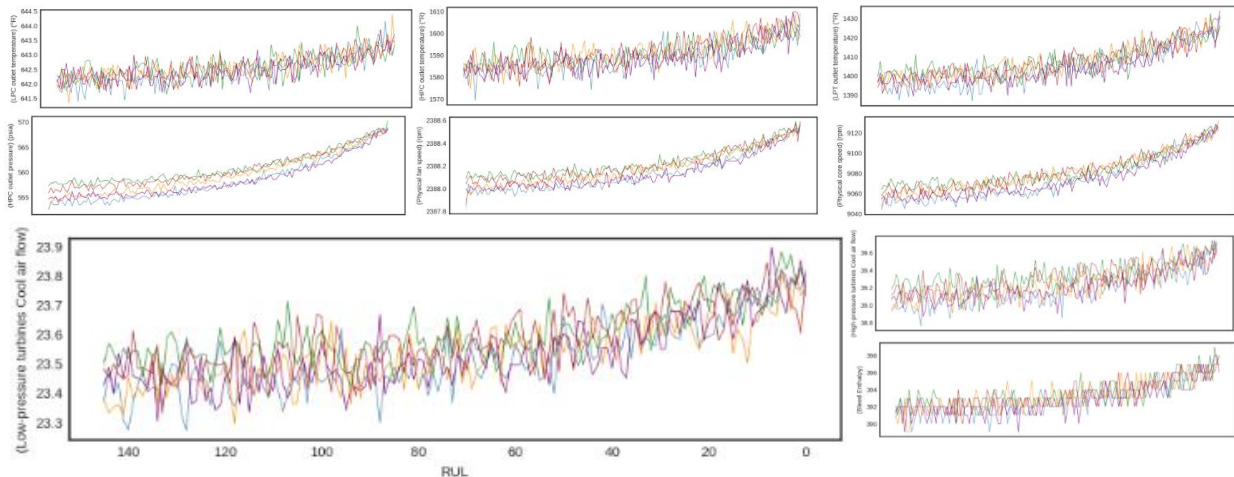
- Cycle
- LPC Outlet Temperature
- HPC Outlet Temperature
- LPT Outlet Temperature
- Bypass-Duct Pressure
- HPC Outlet Pressure
- Physical Fan Speed
- Physical Core Speed
- Engine Pressure Ratio (P50/P2)
- HPC Outlet Static Pressure
- Bypass Ratio
- Bleed Enthalpy
- HPT Cool Air Flow
- LPT Cool Air Flow

4.6 Visualisasi Fitur

Ketika siklus mulai mendekati akhir dari masa pakainya, maka tren perubahan data akan semakin ekstrim. Oleh karena itu diperlukan limit RUL yang sesuai yang cenderung mendekati akhir dari masa pakai. Dari 100 mesin jet, terdapat mesin yang memiliki siklus maksimal paling kecil, yaitu 145 siklus. Oleh karena itu, 145 ini dipilih sebagai limit RUL untuk pembuatan model pada tahap berikutnya. Data yang dipakai untuk prediksi RUL hanyalah data ketika RUL tersisa 145 siklus hingga 0 atau hingga mesin berakhir masa pakainya. Berikut ini merupakan visualisasi dari data setiap fitur terhadap RUL dengan limit 145 siklus. Visualisasi diwakili oleh 5 mesin dengan label warna yang berbeda.



Gambar 10 Grafik Data Fitur Semakin Menurun Ketika RUL Semakin Kecil



Gambar 11 Grafik Data Fitur Semakin Meningkatkan Ketika RUL Semakin Kecil

4.7 Preprocessing Model Klasifikasi

Untuk melakukan prediksi RUL mesin jet, diperlukan sebuah model klasifikasi yang dibuat dengan melihat hubungan antara variabel prediktor dan variabel target dalam data latih (*train data*). Untuk memudahkan pengklasifikasian, setiap variabel dibuatkan *dataframe* baru yang mewakili variabel target atau prediktor dan mewakili *train data* atau *test data*. Keterangan lebih rinci mengenai pembagian *dataframe* adalah sebagai berikut:

- X_{train} : Fitur-fitur terseleksi pada train data
- y_{train} : RUL berdasarkan siklus mesin pada train data
- X_{test} : Fitur-fitur terseleksi pada test data
- y_{test} : RUL aktual FD003
- y_{pred} : RUL prediksi

Dengan menggunakan suatu metode klasifikasi, pola atau hubungan dari X_{train} dan y_{train} dianalisa untuk membuat sebuah model untuk prediksi RUL. Setelah model dibuat, prediksi RUL dilakukan dengan memasukkan data-data pada X_{test} ke dalam model untuk mendapatkan *output* berupa RUL prediksi (y_{pred}). Untuk mengetahui keakuratan dari model, RUL prediksi (y_{pred}) dibandingkan dengan RUL aktual mesin (X_{test}) menggunakan matriks evaluasi.

4.8 Hasil Pemodelan Klasifikasi

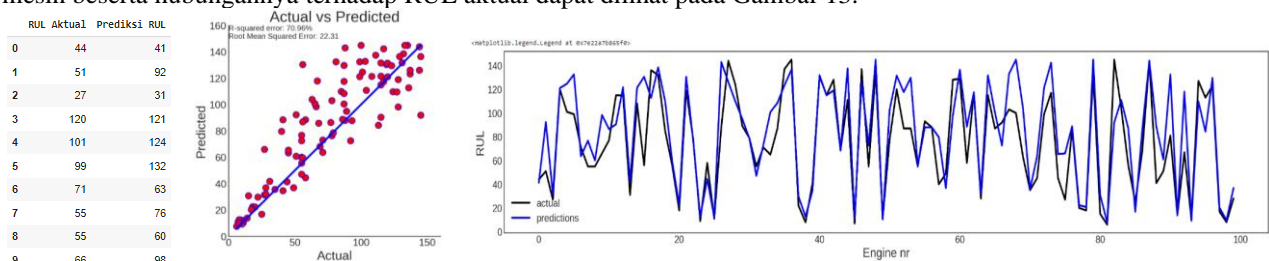
Tujuan dari algoritma klasifikasi adalah membuat model yang dapat memprediksi RUL ketika dimasukkan sebuah variabel prediktor. Model klasifikasi dapat dibuat menggunakan beberapa metode. Metode yang digunakan ialah *K-Nearest Neighbors* (KNN), *Support Vector Machine* (SVM), dan *Random Forest*. Setiap hasil dari pemodelan dievaluasi menggunakan matriks evaluasi R2 dan RMSE. Setelah melakukan *running* model, didapatkan nilai keakuratan dari setiap model pada Gambar 12.

	R2	RMSE	time to train	time to predict	total time
KNN	70.96%	22.31	0.048	0.020	0.068
SVM	67.13%	23.73	40.598	0.152	40.750
Random Forest	72.61%	21.66	48.485	0.057	48.541

Gambar 12 Evaluasi Performa Model

4.8.1 Pemodelan K-Nearest Neighbors (KNN)

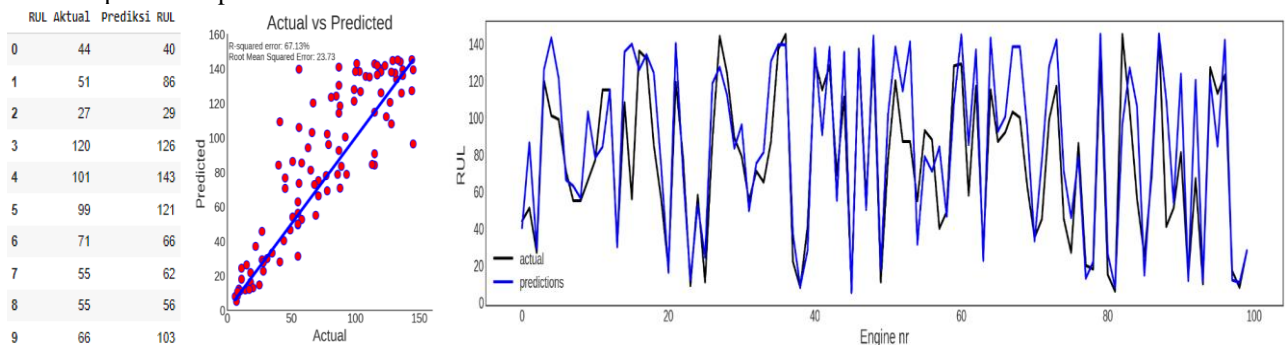
Dalam pemodelan KNN pada dataframe Train FD001 yang sudah terseleksi fitur-fiturnya, proses klasifikasi menggunakan *nearest neighbor* (NN) dengan nilai 6. Karena setelah dilakukan sejumlah eksperimen dihasilkan bahwa NN = 6 memiliki *Root Mean Squared Error* (RMSE) dan R2 Error yang lebih kecil dan fit dengan nilai data-data fitur yang sudah terseleksi dibandingkan dengan NN lain. Hasil prediksi RUL dengan model KNN berupa sampel beberapa mesin beserta hubungannya terhadap RUL aktual dapat dilihat pada Gambar 13.



Gambar 13 Hasil Prediksi RUL dengan Metode KNN dan Hubungannya Terhadap RUL Aktual

4.8.2 Pemodelan Support Vector Machine (SVM)

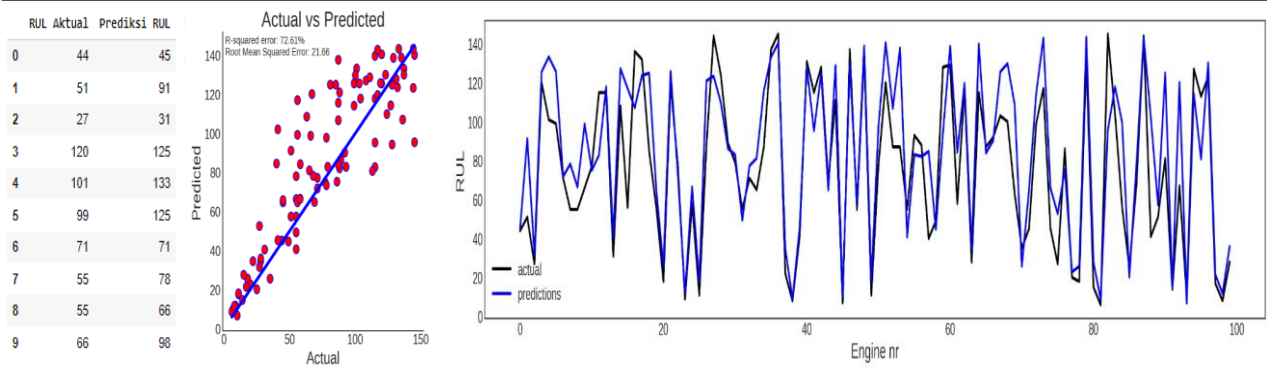
Pada pemodelan algoritma SVM disini menggunakan fungsi *kernel Radial Basis Function* (RBF). Kernel RBF digunakan untuk mengubah input data menjadiruang berdimensi lebih tinggi, sehingga menjadi lebih mudah untuk memisahkan kelas atau *cluster* yang berbeda. Sebelum menjalankan model, kita perlu menentukan nilai C, gamma dan epsilon terbaik menggunakan modul *GridSearchCV*. Didapatkan *hyperparameter* terbaik untuk model SVM, yaitu C bernilai 10; gamma bernilai 1; dan epsilon bernilai 0,01. Parameter ini dimasukkan ke dalam model klasifikasi SVM kernel RBF. Hasil prediksi RUL dengan model SVM berupa sampel beberapa mesin beserta hubungannya terhadap RUL aktual dapat dilihat pada Gambar 14.



Gambar 14 Hasil Prediksi RUL dengan Metode SVM dan Hubungannya Terhadap RUL Aktual

4.8.3 Pemodelan Random Forest

Random Forest dibangun di atas konsep pohon keputusan (*decision trees*) dan menggabungkan beberapa pohon keputusan untuk membuat prediksi yang lebih akurat. Untuk menentukan nilai *n_estimator*, *min_sample leaf*, *nax_features* dan *n_jobs* perlu dilakukan beberapa percobaan untuk menemukan nilai terbaik dilihat dari seberapa bagus performa model yang dihasilkan. Berdasarkan beberapa percobaan nilai parameter, ditetapkan nilai nilai *n_estimator* = 500, *min_sample_leaf* = 1, *n_jobs* = -1, dan *max_features* = 'sqrt'. Hasil prediksi RUL dengan model *Random Forest* berupa sampel beberapa mesin beserta hubungannya terhadap RUL aktual dapat dilihat pada Gambar 15.



Gambar 15 Hasil Prediksi RUL dengan Metode *Random Forest* dan Hubungannya Terhadap RUL Aktual

5. Kesimpulan

Dari semua hal yang telah dilakukan dalam penelitian, maka dapat disimpulkan bahwa :

1. Dari hasil penelitian didapatkan fitur-fitur yang signifikan dan tidak signifikan terhadap prediksi RUL. Fitur yang tidak signifikan adalah fitur bernilai konstan dan fitur yang berkorelasi tinggi terhadap fitur lain. Fitur yang tidak signifikan juga memiliki p-value lebih dari 0,05. Oleh karena itu, fitur-fitur yang signifikan terhadap prediksi RUL mesin jet didapatkan setelah membuang fitur yang tidak signifikan tersebut.
2. Metode Klasifikasi KNN, SVM, dan *Random Forest* dapat digunakan untuk membuat model prediksi RUL *jet engine*, dengan nilai keakuratan sebagai berikut:
 - KNN, dengan nilai $R^2 = 70,96\%$, $RMSE = 22,31$
 - SVM, dengan nilai $R^2 = 67,13\%$, $RMSE = 23.73$
 - *Random Forest*, dengan nilai $R^2 = 72,61\%$, $RMSE = 21.66$
3. Kriteria model terbaik adalah model dengan nilai R^2 tertinggi, dan nilai RMSE terendah. Diantara ketiga metode klasifikasi yang digunakan pada penelitian ini, model terbaik untuk prediksi RUL NASA *Jet Engine dataset* ketiga (FD003) adalah model *Random Forest*.

Ucapan Terima Kasih

Terima kasih kepada Bapak Toni Prahasto dan Bapak Achmad Widodo atas dukungan dan bimbingannya hingga selesai tugas akhir ini dan juga rekan-rekan mahasiswa Teknik Mesin Angkatan 2019 Universitas Diponegoro yang senantiasa memberikan bantuan dan dukungannya terhadap penulis.

DAFTAR PUSTAKA

- [1] S. Lourensius, N. H. Djunggu dan Y. E. Prawatya, "IMPLEMENTASI PREDICTIVE MAINTENANCE UNTUK MESIN PENGUPAS BUAH PINANG DENGAN MIKROKONTROLLER," *INTEGRATE: Industrial Engineering and Management System Volume 7, No. 2*, pp. 1-6, 2023.
- [2] J. Hurtado, D. Salvati, R. Semola, M. Bosio dan V. Lomonaco, "Continual learning for predictive maintenance: Overview and challenges," *Intelligent Systems with Applications 19 (2023) 200251*, pp. 1-18, 2023.
- [3] M. Tahan, M. Muhammad dan Z. A. A. Karim, "A Framework for Intelligent Condition-based Maintenance of Rotating Equipment," *MATEC Web of Conferences*, pp. 1-5, 2014.
- [4] P. M. Sforza, *Theory Of Aerospace Propulsion (2nd Edition)*, Oxford: Butterworth-Heinemann, 2017.
- [5] U. Hartomo, "Mempersiapkan Mesin Turbofan Untuk Pesawat Komersial Masa Depan (Bagian 1)," 17 Juni 2015. [Online]. Available: <https://www.kompasiana.com/uhartomo/54f34c967455139e2b6c702f/mempersiapkan-mesin-turbofan-untuk-pesawat-komersial-masa-depan-bagian-1>.
- [6] A. Saxena, K. Goebel, D. Simon dan N. Eklund, "Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation," *2008 International Conference on Prognostics and Health Management*, pp. 1-9, 2008.
- [7] I. Daqiqil, *MACHINE LEARNING : Teori, Studi Kasus dan Implementasi Menggunakan Python*, Pekanbaru: UR PRESS, 2021.
- [8] Roshna, "K-Nearest Neighbors Algorithm," 7 April 2023. [Online]. Available: <https://intuitivetutorial.com/2023/04/07/k-nearest-neighbors-algorithm/>.
- [9] A. Navlani, "Support Vector Machines with Scikit-learn Tutorial," 15 Desember 2019. [Online]. Available: <https://www.datacamp.com/tutorial/svm-classification-scikit-learn-python>.

-
- [10] D. Kumar, "How did The Random Forest Algorithm work in Machine Learning?," 30 September 2021. [Online]. Available: <https://divakarkumarp.medium.com/how-did-the-random-forest-algorithm-work-in-machine-learning-9e044573898b>.