

# PERANCANGAN PROSES IMPLEMENTASI *MACHINE LEARNING* DALAM *MAINTENANCE MANAGEMENT* UNTUK MENCEGAH *DERATING*

**Bonifasius Yoga Adi Pratama, Hari Agung Yuniarto\***

*Magister Teknik Industri, Fakultas Teknik, Universitas Gadjah Mada,  
Jl. Grafika No. 2, Yogyakarta, Indonesia 55281*

*(Received: February 25, 2021/ Accepted: May 19, 2021)*

## **Abstrak**

*Derating menjadi masalah yang sering kali muncul pada generator diesel di pembangkit listrik. Derating dapat menyebabkan penurunan kinerja dan produksi listrik pada generator. Kapasitas produksi listrik yang menurun nantinya akan menjadi masalah jika melihat kondisi konsumsi listrik nasional yang terus meningkat dari tahun ke tahun. Derating sering kali terjadi karena suhu yang tinggi pada charge air. Pencegahan derating dapat dilakukan dengan mengimplementasikan strategi maintenance yang mampu memprediksi derating dan mengakomodasi perubahan suhu charge air sebagai penyebab terjadinya derating. Penelitian ini akan memperlihatkan rancangan implementasi strategi maintenance berbasis data untuk memprediksi terjadinya derating dengan menggunakan pendekatan machine learning. Rancangan proses ini akan memberikan gambaran proses seperti apa yang dapat digunakan untuk mencegah derating sehingga membantu menjaga performa generator. Tahapan implementasi machine learning dilakukan dengan mengimplementasikan proses knowledge discovery from data pada proses yang ada dalam maintenance management. Evaluasi terkait proses maintenance management dan machine learning menunjukkan bahwa machine learning dapat diimplementasikan pada tahap controlling. Klasifikasi kondisi generator juga didasarkan pada trend kondisi suhu charge air sehingga prediksi kondisi generator terkait derating tidak dipengaruhi oleh perubahan suhu yang bersifat cepat dan sementara. Penjabaran proses yang ada menunjukkan bahwa implementasi machine learning dalam maintenance management ini mungkin untuk dilakukan.*

**Kata kunci:** *decision tree; derating; machine learning; maintenance management*

## **Abstract**

*[Designing The Implementation Process of Machine Learning in Maintenance Management to Avoid Derating] Derating is problem that often arises in power plant. Derating force generator to work below its optimum performance and resulting low production rate of electricity. Declining of electricity production capacity can be problem when we see condition of national electricity consumption in Indonesia which continues to increase year over year. Derating often occurs due to high temperatures in charge air. Derating prevention can be done by implementing maintenance strategy that is able to predict derating and accommodate changes in charge air temperature. This study designs processes of implementing data-based maintenance strategy to predict occurrence of derating using machine learning approach. Process design will provide overview of what kind of process can be used to avoid derating so that it helps maintain generator performance. Machine learning implementation can be done by implementing process of knowledge discovery from data in existing maintenance management process. Evaluations related to maintenance management and machine learning processes show that machine learning can be implemented at controlling stage. Classification of generator conditions is based on trend of charge air temperature so that prediction of generator conditions will not be affected by temporary changes in temperature. Process overview concludes that it is possible to implement machine learning in maintenance management.*

**Keywords:** *decision tree; derating; machine learning; maintenance management*

---

\*Penulis Korespondensi.

E-mail: h.a.yuniarto@ugm.ac.id

## **1. Pendahuluan**

Listrik menjadi salah satu sumber energi yang sangat penting sekarang ini. Data dari Badan Pusat

Statistik (BPS) menunjukkan bahwa konsumsi listrik per kapita di Indonesia pada tahun 2009 sampai tahun 2018 ada dalam tren yang terus meningkat. Tren peningkatan konsumsi listrik juga akan terus terjadi jika melihat adanya pertambahan jumlah penduduk seperti diproyeksikan oleh Badan Perencanaan Pembangunan Nasional (Bappenas), BPS, dan *United Nations Populations Fund* (UNFPA). Jumlah penduduk Indonesia diprediksi akan berjumlah lebih dari tiga ratus juta penduduk pada tahun 2035. Hal tersebut menunjukkan kemungkinan terjadinya pertambahan konsumen dan konsumsi listrik nasional.

Peningkatan konsumsi listrik ini tentu harus diimbangi juga dengan peningkatan kemampuan produksi listrik nasional. Pemerintah sendiri telah melakukan pembangunan pembangkit listrik baru dan menargetkan penambahan kapasitas daya hingga 35.000 megawatt pada tahun 2024 nanti. Namun usaha peningkatan produksi listrik nasional tidak bisa hanya berhenti pada pembangunan pembangkit listrik baru. Pembangunan pembangkit listrik baru dapat meningkatkan kapasitas produksi listrik, namun kapasitas produksi listrik dapat menurun jika tidak dilakukan *maintenance* yang tepat. Produksi listrik yang tidak optimal pada akhirnya akan tetap menyebabkan terjadinya kesulitan dalam pemenuhan permintaan energi listrik yang terus meningkat.

Salah satu masalah yang berkaitan dengan *maintenance* dan kemampuan produksi listrik adalah *derating*. *Derating* sebenarnya digunakan untuk memperlambat atau menghindari munculnya *breakdown*. Namun usaha tersebut dilakukan dengan cara menurunkan kinerja dari generator. Penurunan kinerja dari generator menjadikan generator bekerja pada kondisi yang lebih aman, tetapi daya yang dapat dihasilkan menjadi lebih rendah. Kondisi tersebut akan menjadikan produksi listrik menurun. Kondisi penurunan produksi listrik ini tentunya nanti dapat menjadi masalah jika melihat konsumsi listrik nasional yang terus meningkat, maka dari itu kondisi *derating* ini perlu untuk dicegah.

Pencegahan kondisi *derating* belum dapat dilakukan dengan baik. Pendekatan yang ada sekarang ini terkait *derating* hanya sebatas mengetahui apakah kondisi generator sekarang ini telah masuk kondisi *derating* atau belum. Kondisi *derating* sendiri bisa saja terjadi untuk waktu yang singkat, karena perubahan kondisi pada generator. Salah satu penyebab sering terjadinya *derating* adalah suhu *charge air* yang tinggi. Perubahan suhu *charge air* yang tidak konsisten, membuat kondisi *derating* sulit ditebak. Kondisi *derating* sendiri bisa saja terjadi untuk waktu yang singkat, karena perubahan suhu *charge air* yang cepat. Kondisi *derating* yang tidak konsisten atau hanya terjadi sesekali bisa saja menunjukkan bahwa tidak ada kerusakan pada generator. Hal yang sama juga dapat terjadi pada kondisi normal, dimana kondisi normal hanya terjadi sementara waktu dan kemudian disusul dengan kondisi *derating* secara terus-menerus, hal tersebut tentu menunjukkan bahwa generator tidak ada

dalam kondisi normal. Kondisi yang tidak pasti seperti itu tentu membutuhkan pendekatan *maintenance* yang dapat melakukan klasifikasi kondisi dengan baik dan juga dapat memastikan bahwa kondisi yang terjadi tidak bersifat sementara. Pendekatan *maintenance* ini akan membantu untuk melihat bagaimana kecenderungan kondisi generator selanjutnya sehingga ada dasar yang tepat dalam pengambilan keputusan terkait *maintenance*.

Ketepatan pengenalan kondisi generator akan mendorong *maintenance* dapat dilakukan pada saat yang tepat. Ketepatan waktu *maintenance* menjadi sangat penting karena dapat mencegah generator terhindar dari kondisi *derating* yang terus-menerus serta menghindarkan dari *maintenance* yang tidak dibutuhkan. *Maintenance* pada waktu yang tepat juga dibutuhkan karena *maintenance* sendiri belum tentu dapat langsung dilakukan. Proses *maintenance* membutuhkan perencanaan terlebih dahulu. *Maintenance* yang terlambat akan menyebabkan generator tidak dapat bekerja secara optimal dalam waktu yang lebih lama. Keterlambatan *maintenance* juga semakin meningkatkan peluang terjadinya *breakdown* pada generator. Kondisi tersebut pada akhirnya mendorong untuk melakukan penelitian yang lebih jauh terkait pendekatan baru untuk mencegah *derating*. *Maintenance* yang dilakukan sebelum terjadinya *derating* dapat membantu menurunkan laju kegagalan dan meningkatkan efektivitas perbaikan mesin sehingga dapat meningkatkan nilai *reliability* dan *availability* dari mesin (Rosihan & Yuniarto, 2019).

*Derating* perlu dicegah dengan strategi *maintenance* baru yang sesuai. Strategi *maintenance* yang banyak dilakukan selama ini adalah *preventive maintenance*. *Preventive maintenance* dilakukan pada rentang waktu yang sangat spesifik, dimana masih ada kemungkinan terjadinya kerusakan di luar rentang waktu tersebut (Possi, 2011). Model *preventive maintenance* biasanya mengasumsikan sistem akan selalu ada dalam kondisi yang sama atau dalam urutan kondisi yang dapat didefinisikan dengan baik. Asumsi tersebut tidak selalu tepat karena terdapat faktor-faktor yang dapat mempengaruhi kondisi dari sistem seperti jadwal kerja, lingkungan dan lain-lain, sehingga menyebabkan interval waktu *maintenance* tidak lagi sesuai dengan kondisi dari sistem (Kothamasu *et al.*, 2006).

Strategi *maintenance* baru dibutuhkan untuk mengakomodasi terjadinya perubahan kondisi sistem dari waktu ke waktu. Bloch dan Geitner (1983) menyebutkan bahwa 99 persen dari semua kerusakan yang terjadi biasanya akan didahului dengan kondisi, tanda, atau indikasi tertentu yang menunjukkan bahwa akan terjadi kerusakan. Hal yang harus dilakukan adalah memahami tanda tersebut sehingga *maintenance* dapat dilakukan sebelum kerusakan terjadi. Proses untuk memahami kondisi dari sistem atau mesin ini dilakukan dalam pendekatan *condition-based maintenance*. *Condition-based maintenance*

sering dihubungkan dengan *predictive maintenance* (Swanson, 2001; Kothamasu *et al.*, 2006; Wang *et al.*, 2015) karena prediksi dilakukan berdasarkan kondisi spesifik sistem. Strategi *maintenance* ini selain mampu mengakomodasi perubahan kondisi sistem dari waktu ke waktu, juga dapat menjadikan *maintenance* yang dilakukan lebih efisien dengan tidak melakukan *maintenance* atau bahkan inspeksi yang tidak dibutuhkan (Alaswad *et al.*, 2017; Jardine *et al.*, 2006). *Maintenance* hanya dilakukan ketika kondisi sistem pada saat itu menunjukkan adanya kebutuhan untuk melakukan *maintenance*.

Strategi *condition-based maintenance* atau *predictive maintenance* ini semakin didukung dengan adanya revolusi industri 4.0. Strategi *maintenance* ini sangat bergantung pada data kondisi dari sistem atau mesin yang jumlahnya bisa sangat besar, karena proses pengawasan dari waktu ke waktu. *Review* terkait strategi *maintenance* ini juga selalu mengarah pada ekstraksi sekumpulan besar data terkait sistem *monitoring* sebagai dasar untuk membangun model (Jardine *et al.*, 2006; Peng *et al.*, 2010; Goyal dan Pabla, 2015; Bousdekis *et al.*, 2018). Revolusi industri 4.0 mengakomodasi penggunaan data yang besar atau sering disebut dengan *big data* dengan pendekatan *artificial intelligence* yang sekarang ini penggunaannya semakin berkembang. *Intelligent method* sendiri menjadi sangat populer sekarang ini karena performansinya yang sangat baik dan tingkat *robustness* yang tinggi (Aribowo, 2017). Salah satu pendekatan *artificial intelligence* yang sangat berhubungan dengan *big data* adalah *machine learning*. Marr (2018) menyebutkan bahwa industri 4.0 akan mengembangkan penggunaan komputer dan otomasi yang dimulai pada revolusi industri ketiga dengan menggunakan *smart system* yang didasarkan pada data (*big data*) dan *machine learning*. *Machine learning* sendiri sebagai bagian dari *artificial intelligence* dapat digunakan untuk menemukan informasi dari sekumpulan data sehingga nantinya dapat membantu dalam prediksi dan pengambilan keputusan (Fernandes *et al.*, 2018). Penggunaan *machine learning* juga menjadi pendekatan baru untuk mencegah *derating* dimana Sakib dan Wuest (2018) dalam penelitiannya menyebutkan bahwa *machine learning* dapat membantu untuk mengembangkan model *maintenance* yang ada sekarang ini.

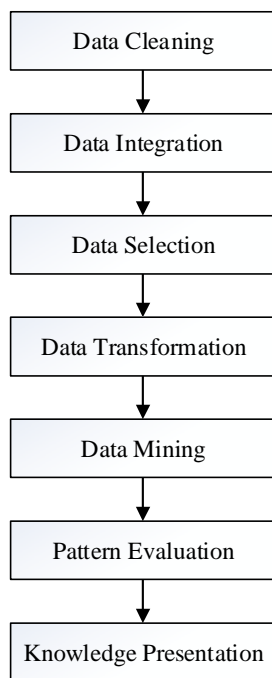
Kondisi *derating* yang dipengaruhi oleh perubahan suhu *charge air* juga membutuhkan proses pengolahan data yang berbeda. Kondisi yang biasanya dilihat adalah kondisi yang konsisten, dimana ketika sistem rusak maka kondisi sistem akan tetap rusak sampai dilakukan perbaikan. Hal tersebut berbeda dengan kondisi *derating* yang terjadi akibat perubahan suhu *charge air*. Perubahan suhu *charge air* yang tidak konsisten, dimana terjadi kenaikan dan penurunan suhu di sepanjang waktu, membutuhkan analisis data yang baru. Model yang dibuat dituntut untuk dapat melakukan klasifikasi kondisi dengan tepat dan tidak melihat kondisi yang hanya muncul sementara saja,

sehingga tidak terjadi kesalahan dalam pengambilan keputusan terkait *maintenance* yang dilakukan.

Penelitian ini selanjutnya akan menjelaskan perancangan implementasi *machine learning* dalam *maintenance management* untuk mencegah *derating*. Penjelasan yang ada akan mengarah pada penjelasan yang lebih mendalam terkait rancangan proses implementasi *machine learning* dan hubungannya dengan proses *maintenance* untuk mencegah *derating*. Penelitian terkait pencegahan *derating* sendiri menjadi topik yang baru untuk dibahas. Penelitian terkait *maintenance* generator selama ini selalu mengarah pada kerusakan generator, padahal untuk tahap *derating* saja sebenarnya telah terjadi penurunan performa dan muncul kondisi yang tidak normal pada generator. Kondisi yang tidak normal ini jika dibiarkan akan mempercepat terjadinya *breakdown* pada generator. Kondisi data suhu *charge air* yang sering berubah juga menjadi hal baru yang perlu dilihat bagaimana penyelesaiannya. Kondisi data yang secara umum digunakan adalah data yang menunjukkan kondisi sistem telah rusak atau tidak, tanpa ada perubahan. Kondisi data tersebut berbeda dengan kondisi dalam penelitian ini, sehingga kondisi *derating* yang disebabkan suhu *charge air* ini butuh pendekatan yang baru.

## 2. Metode Penelitian

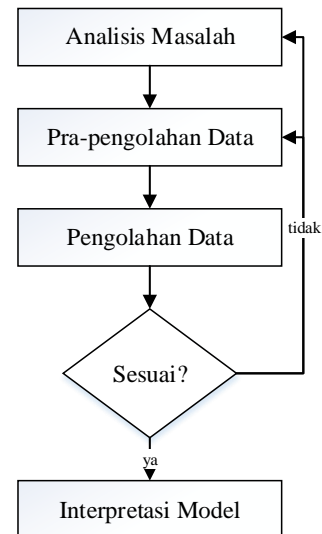
Proses implementasi *machine learning* dalam strategi *maintenance* akan didasarkan pada tahap-tahap dalam *knowledge discovery from data* (KDD) yang dijelaskan oleh Han *et al.* (2012). Tahap-tahap KDD ini sebenarnya didasari oleh proses *data mining* yang bertujuan untuk menemukan pola-pola yang menarik dan informasi dari sekumpulan data yang besar (Han *et al.*, 2012). Tujuan dari *data mining* tersebut memiliki kesamaan dengan tujuan dilakukannya pendekatan *machine learning* yaitu untuk menemukan informasi dari sekumpulan data. *Data mining* sendiri menurut Liao *et al.* (2012) telah menjadi bidang keilmuan yang membentuk cabang pada bidang keilmuan terkait kecerdasan buatan. *Machine learning* memiliki hubungan kuat dengan kecerdasan buatan karena kemampuan untuk belajar (*learning*) yang ada pada *machine learning* merupakan karakteristik utama dari kecerdasan (Kavakiotis *et al.*, 2017). Hubungan tersebut menjadikan pendekatan KDD tersebut dapat diimplementasikan dalam proses pembangunan model *machine learning*. KDD sendiri dibagi ke dalam tujuh tahap seperti dapat dilihat pada **Gambar 1**. Empat tahap pertama merupakan tahap pra-pengolahan data, sedangkan tiga tahap selanjutnya adalah proses pencarian dan penjabaran pola serta informasi yang ada dalam kumpulan data. Tahap-tahap tersebut menjadi dasar untuk membagi proses penelitian ini ke dalam empat tahap utama yaitu tahap analisis masalah, tahap pra-pengolahan data, tahap pengolahan data dan tahap interpretasi model. **Gambar 2** memperlihatkan bagaimana urutan dan proses yang ada pada tahap utama.



**Gambar 1.** Proses *Knowledge Discovery from Data* (Han *et al.*, 2012)

Tahap analisis masalah mencoba untuk melihat masalah yang ada pada tahap *controlling*. Masalah yang ditemukan kemudian dihubungkan dengan ketersediaan data kondisi dan juga proses yang ada pada sistem. Analisis hubungan dari ketiga hal tersebut akan membantu untuk menentukan kondisi seperti apa yang ingin dievaluasi atau diprediksi dari sistem serta data seperti apa yang dapat digunakan untuk membangun model *machine learning* terkait kondisi tersebut. Tahap analisis masalah akan memberikan arahan terkait tujuan serta data yang dibutuhkan untuk model *machine learning* yang dibuat, sehingga pada tahap selanjutnya dapat dimulai proses pengumpulan data. Masalah yang dibahas dalam penelitian ini sendiri adalah munculnya *derating*.

Proses pengumpulan data dilakukan pada tahap pra-pengolahan data. Tahap pra-pengolahan data dapat dibagi menjadi empat tahap utama yaitu tahap pengumpulan dan pemilihan data, tahap penentuan kelas data, tahap *feature selection* serta tahap *resampling data* seperti terlihat pada **Gambar 3**. Aktivitas yang dilakukan pada tahap pengumpulan dan pemilihan data adalah pengambilan data kondisi dari sistem yang telah dikumpulkan. Data kondisi ini dapat memiliki banyak sumber maka dari itu pada tahap ini data-data tersebut akan disatukan menjadi satu kelompok data. Data tersebut juga harus dibersihkan, dimana sering kali data *monitoring* secara *real-time* memiliki data yang tidak konsisten. Data yang tidak konsisten dapat berbentuk data yang kosong atau data yang tidak sesuai dengan proses dari sistem serta tujuan dalam pembangunan model *machine learning*. Tahap pengumpulan dan pemilihan data ini ada dalam tahap *data cleaning* dan *data integration* dalam proses KDD, dimana dilakukan pembersihan data yang tidak relevan



**Gambar 2.** Tahap Utama Implementasi *Machine Learning*

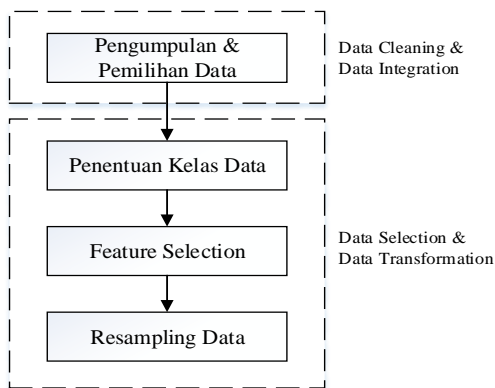
serta integrasi data dari beberapa sumber data *monitoring* kondisi sistem.

Tahap pra-pengolahan data selanjutnya adalah tahap yang menjadi bagian dari proses *data selection and transformation*. Proses ini terdiri dari tiga tahap yaitu tahap penentuan kelas data, *feature selection* dan *resampling data*. Tahap penentuan kelas data bertujuan menambahkan variabel target untuk proses prediksi. Tahap *feature selection* akan membantu pemilihan variabel terbaik untuk proses prediksi. Tahap *resampling data* akan menjadikan komposisi jumlah data antar kelas menjadi lebih proporsional.

Tahap pengolahan data merupakan tahap dimana metode *machine learning* diaplikasikan pada data yang diperoleh dari tahap pra-pengolahan data. Tahap ini bertujuan untuk menemukan pola dari sekumpulan data dan membuat model *machine learning* untuk menggambarkan pola data yang telah ditemukan tersebut.

Proses pengolahan data dapat dibagi menjadi dua tahap yaitu evaluasi parameter dan evaluasi model seperti yang ditampilkan pada **Gambar 4**. Evaluasi parameter dilakukan untuk mengetahui skenario parameter yang dapat menghasilkan model dengan performansi terbaik. Sedangkan evaluasi model dilakukan untuk mengetahui performansi model yang telah dibuat, sehingga dapat diketahui apakah performansi model sudah sesuai dengan standar yang diinginkan atau masih perlu dilakukan perbaikan kembali. Tahap pengolahan data ini ada pada tahap *data mining* dalam proses KDD karena pada tahap ini digunakan *intelligent method* untuk menemukan pola dari data.

Tahap penelitian yang terakhir adalah tahap interpretasi model. Tahap interpretasi model



**Gambar 3.** Tahap Pra-pengolahan Data

merupakan proses untuk menjelaskan informasi yang diperoleh dari model *machine learning* yang telah dibangun. Tahap ini sesuai dengan tahap *pattern evaluation* dan *knowledge representation* dari proses KDD, dimana akan diberikan visualisasi dan penjelasan terkait pola data yang muncul serta informasi yang diperoleh dari pola data yang ditemukan tersebut.

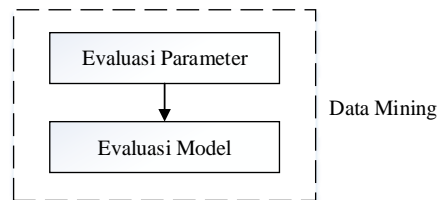
Proses KDD yang telah dijelaskan di atas harus ditempatkan pada ruang lingkup proses *maintenance management* agar implementasinya dapat menyesuaikan dengan strategi *maintenance* yang ada. Proses KDD akan diintegrasikan dengan proses *maintenance management* yang dikembangkan oleh Ben-Daya *et al.* (2009). Proses *maintenance management* tersebut merupakan siklus yang terdiri dari empat tahap yaitu tahap *planning*, *organizing*, *implementing* dan *controlling* seperti digambarkan pada **Gambar 5**. Proses KDD akan diintegrasikan pada tahap yang ada pada proses *maintenance management* ini agar model *machine learning* yang dibangun dapat memberikan informasi dari sekumpulan data terkait proses *maintenance* yang dilakukan.

Informasi yang diperoleh dari model *machine learning* yang dibuat dari proses KDD ini selanjutnya diharapkan dapat membantu untuk melakukan pengambilan keputusan yang lebih tepat. Pengambilan keputusan yang tepat akan mendorong proses yang ada menjadi lebih baik dan pada akhirnya dapat mendorong peningkatan kualitas *maintenance management* yang diterapkan.

### 3. Hasil dan Pembahasan

#### 3.1. *Machine Learning* dalam *Maintenance Management*

Hal pertama yang akan dilakukan untuk mengimplementasikan pendekatan *machine learning* dalam *maintenance management* adalah mengetahui di mana pendekatan ini akan ditempatkan. Proses *maintenance management* memiliki empat tahap yang berbeda sehingga perlu dilihat pada tahap mana pendekatan *machine learning* dapat ditempatkan dan dapat berperan dengan baik. Pendekatan *machine learning* sendiri sangat sesuai dengan pendekatan *condition-based maintenance*, di mana dalam *condition-based maintenance* data-data terkait kondisi



**Gambar 4.** Tahap Pengolahan Data

sistem akan dievaluasi untuk melihat kemunculan tanda-tanda kerusakan.

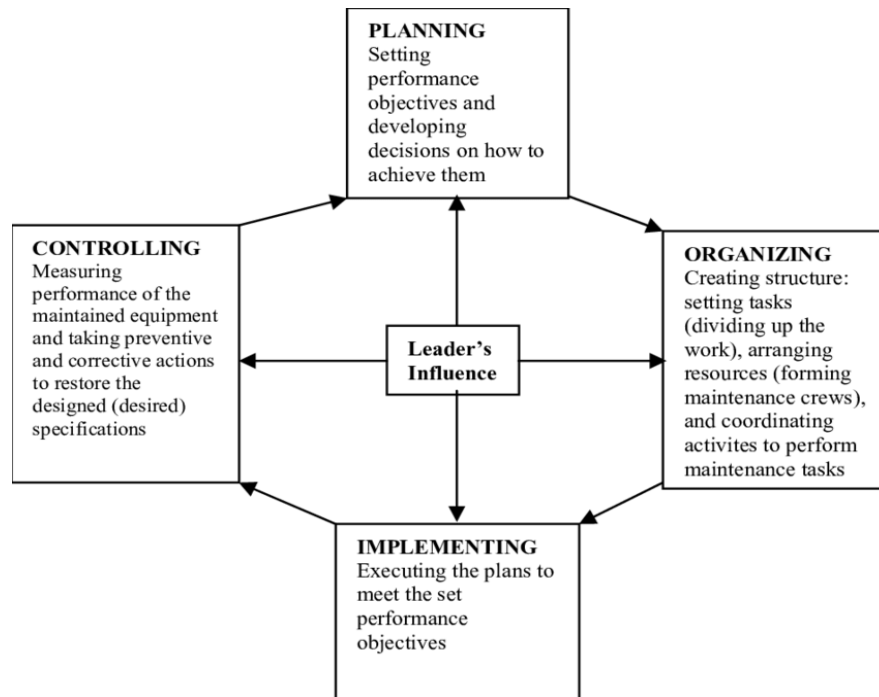
Metode *decision tree* yang digunakan juga dapat melakukan klasifikasi kondisi sistem sehingga dapat diketahui kapan harus dilakukan *maintenance*. Proses evaluasi kondisi sistem seperti ini banyak dilakukan pada tahap *controlling*. Tahap *controlling* bertujuan untuk melakukan pengukuran kondisi sistem sehingga dapat diketahui apakah *maintenance* perlu dilakukan atau tidak.

Karakteristik tahap *controlling* tersebut menjadi sangat sesuai dengan pendekatan *condition-based maintenance* dengan menggunakan *machine learning* untuk mengevaluasi dan mencari informasi dari data-data kondisi sistem yang dikumpulkan. Kesesuaian tersebut menjadi dasar yang tepat untuk mengimplementasikan pendekatan *machine learning* pada tahap *controlling*. **Gambar 6**. memberikan ilustrasi terkait penempatan implementasi *machine learning* dalam proses *maintenance management*.

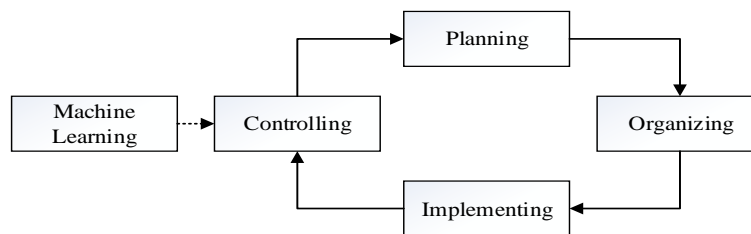
#### 3.2. Rancangan Proses Implementasi *Machine Learning*

##### 3.2.1. *Decision Tree*

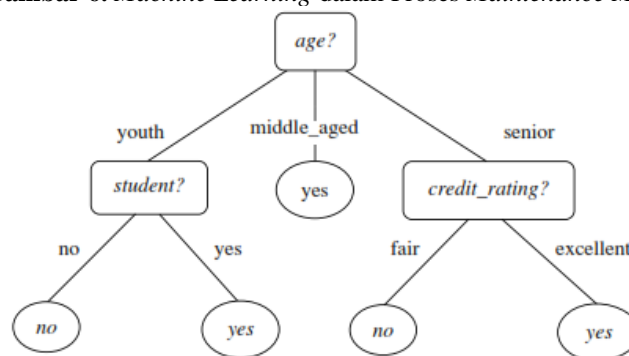
Pendekatan *machine learning* yang akan diimplementasikan pada *maintenance management* dalam penelitian ini adalah metode *decision tree*. Metode *decision tree* dipilih karena dapat membantu untuk melakukan pendekatan prognosis dan juga diagnosis kondisi sistem sekaligus. Hal tersebut dilakukan melihat hasil penelitian Alaswad *et al.* (2017) yang menunjukkan bahwa tren strategi *maintenance* ke depan tidak hanya bertujuan untuk memprediksi (prognosis) saja tetapi mengkombinasikan pendekatan diagnosis dan prognosis. Metode *decision tree* memiliki kelebihan yaitu model yang bersifat *white box*. Model yang bersifat *white box* dapat memperlihatkan struktur algoritma dari analisis data yang dilakukan (Delibasic *et al.*, 2012), sehingga dapat diketahui hubungan antar variabel yang membangun model *machine learning*. Kondisi tersebut akan membantu untuk memberikan informasi terkait hubungan antar bagian pada sistem yang menyebabkan terjadinya kerusakan, sehingga dapat membantu dalam proses diagnosis dan mengarahkan tindakan *maintenance* yang dilakukan.



Gambar 5. Proses *Maintenance Management* (Ben-Daya et al., 2009)



Gambar 6. *Machine Learning* dalam Proses *Maintenance Management*



Gambar 7. Struktur Model Decision Tree (Han et al., 2012)

Penggunaan metode *decision tree* juga didasarkan dari bagaimana performansi metode ini untuk pendekatan *maintenance* pada masalah yang berbeda. Penelitian Gandhi *et al.* (2018) dan Bukhsh *et al.* (2019) menunjukkan bahwa metode *decision tree* ini memiliki performansi yang baik ketika diimplementasikan dalam bidang *maintenance*. Penelitian Bukhsh *et al.* (2019) bahkan secara jelas memperlihatkan akurasi prediksi kondisi sebesar 84,4 persen, yang artinya metode *decision tree* ini memiliki performansi yang tinggi ketika digunakan pada proses *maintenance*.

*Decision tree* sendiri merupakan model dengan struktur berupa diagram alir yang berbentuk seperti pohon, dimana setiap *internal node* menggambarkan tes yang dilakukan pada atribut, setiap cabang menggambarkan hasil dari tes yang dilakukan pada *internal node*, dan bagian daun (bagian akhir) menggambarkan label dari kelas yang ada (Han *et al.*, 2012). Gambar 7 menunjukkan bagian *age*, *student* dan *credit rating* sebagai *internal node*. Bagian cabang digambarkan dengan garis, misalnya cabang untuk *internal node age* adalah *youth*, *middle\_aged* dan *senior*. Bagian daun digambarkan pada bagian



berbentuk oval dengan tulisan *yes* atau *no* untuk menggambarkan label klasifikasi dari data.

### 3.2.2. Pra-pengolahan Data

Tahap pra-pengolahan data adalah tahap di mana data disiapkan agar dapat digunakan untuk membangun model *machine learning*. Salah satu tahap penting agar data dapat digunakan adalah tahap penentuan kelas data. Hal ini penting karena *decision tree* masuk dalam *supervised learning* dimana model klasifikasi dibuat berdasarkan pola kelas data yang telah ada. Tahap ini akan membantu untuk menentukan data yang ada masuk dalam kelas klasifikasi yang mana, karena data yang telah ada belum memiliki kelas. Proses penentuan kelas data didasarkan pada tren data kondisi suhu *charge air*. Penentuan kelas data dengan menggunakan tren data akan membantu melihat konsistensi kondisi suhu dari *charge air*.

Tren kondisi data akan memperlihatkan kecenderungan kondisi sistem pada suatu periode sehingga klasifikasi data menjadi lebih konsisten. Hal ini akan membantu dalam mengatasi kondisi suhu *charge air* yang terus berubah-ubah dari waktu ke waktu. Klasifikasi yang dilakukan nantinya tidak hanya akan melihat kondisi sementara tetapi mengarah pada klasifikasi kondisi berdasarkan tren kondisi suhu *charge air* pada suatu periode.

Tahap pra-pengolahan data tidak hanya menjadikan *dataset* yang ada dapat digunakan, tetapi juga membantu membuat data yang ada dapat menghasilkan model dengan performansi terbaik. Hal tersebut yang akan dilakukan pada tahap-tahap selanjutnya yaitu tahap *feature selection* dan *resampling data*. Proses *feature selection* dapat dilakukan ketika *dataset* yang dibutuhkan telah siap. Proses *feature selection* bertujuan untuk memilih variabel yang memiliki hubungan kuat dengan perubahan kondisi yang akan diprediksi pada sistem, dalam hal ini kondisi *derating*. Pemilihan variabel ini akan membantu untuk meningkatkan performansi model karena variabel yang tidak relevan tidak akan digunakan dalam membangun model. Proses *training* untuk membuat model juga menjadi lebih cepat dan kompleksitas model dapat dikurangi karena lebih sedikit variabel yang digunakan, sehingga model lebih mudah untuk diinterpretasi.

Metode *decision tree* merupakan metode *machine learning* yang mengarah pada klasifikasi sehingga kelas *dataset* biasanya akan berbentuk kategori. Kelas data akan disesuaikan dengan kategori kondisi yang ingin diprediksi. Data-data yang ada dalam kelas ini jumlahnya tidak selalu proporsional antara satu kelas dengan kelas lainnya. Kondisi perbedaan jumlah data antar kelas yang cukup jauh akan menyebabkan model tidak dapat memprediksi kondisi sistem dengan baik. Model akan cenderung mampu untuk memprediksi data dari kelas mayoritas saja dan sulit atau bahkan tidak mampu untuk memprediksi dengan tepat data dari kelas minoritas. Hal itu dapat terjadi karena model *machine learning*

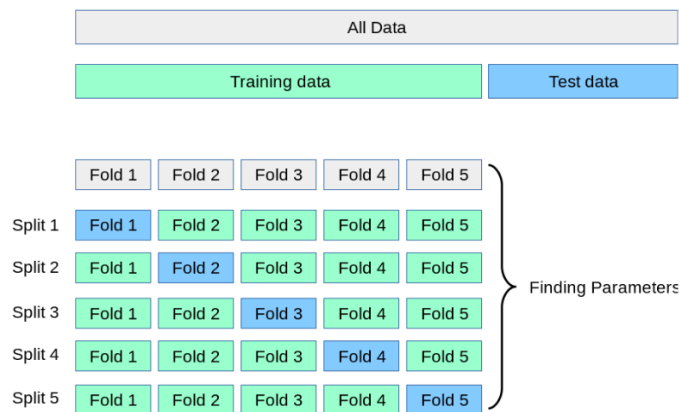
dibangun melalui proses *training* dengan *dataset* yang dipenuhi oleh data dari kelas mayoritas, sehingga model tersebut kesulitan melihat pola yang tepat dari kelas minoritas. Proses *resampling* data perlu dilakukan untuk mengatasi masalah tersebut. *Resampling* data bertujuan untuk menjadikan jumlah data antar kelas menjadi lebih proporsional. Data yang lebih proporsional akan menjadikan model mampu memprediksi semua kelas dengan baik, sehingga pada akhirnya performansi dari model dapat ditingkatkan. Proses *Resampling data* untuk menambahkan data disebut *oversampling*, sedangkan untuk mengurangi jumlah data disebut *undersampling*. *Resampling data* dapat menggunakan salah satu atau bahkan mengkombinasikan kedua pendekatan tersebut sesuai kondisi dari *dataset*. Proses *feature selection* hingga *resampling data* masuk dalam tahap *data selection* dan *data transformation* dari proses KDD karena pada proses ini terjadi pemilihan variabel dan transformasi data untuk menjadikan jumlah data antar kelas lebih proporsional.

### 3.2.3. Pengolahan Data

Tahap pengolahan data dibagi menjadi dua yaitu tahap evaluasi parameter dan tahap evaluasi model. Evaluasi parameter atau sering juga disebut dengan *parameter tuning* dan perlu dilakukan sebelum model *machine learning* dibangun.

Tahap ini akan memastikan bahwa kombinasi nilai yang dipilih dari parameter-parameter yang digunakan merupakan skenario terbaik agar diperoleh performansi model terbaik. Metode-metode yang ada dalam *machine learning* biasanya memiliki beberapa parameter yang dapat diatur nilainya. *Decision tree* sendiri memiliki parameter seperti tingkat kedalaman dari model untuk membatasi kedalaman model yang dibuat, atau parameter jumlah sampel minimum yang harus ada pada setiap cabang agar dapat dilakukan proses *splitting* serta parameter-parameter lainnya. Parameter-parameter yang ada pada *decision tree* ini berfungsi untuk mengatur dan memberi batasan pembangunan *tree model*. *Tree model* yang terlalu dalam dapat menyebabkan *overfitting*, dimana model sulit memprediksi data baru, sedangkan model yang terlalu pendek akan menghasilkan prediksi yang sangat umum sehingga dapat menimbulkan banyak kesalahan prediksi juga. Evaluasi parameter perlu dilakukan untuk menemukan skenario parameter terbaik dimana model tidak menjadi terlalu dalam atau terlalu pendek.

Skenario parameter yang diperoleh akan digunakan untuk membangun model *machine learning* pada tahap evaluasi model. Model *machine learning* yang dibangun adalah model *decision tree* dan pendekatan *decision tree* sendiri termasuk dalam *supervised learning*. Vadim (2018) menjelaskan *supervised learning* sebagai *learning with teacher training*, dimana terdapat *training dataset* (*training set*) yang akan digunakan untuk membangun model dan *test dataset* (*test set*) untuk mengevaluasi performansi model. Model *decision tree* akan dibangun dengan



**Gambar 8.** Data Splitting Metode *K-fold Cross Validation* (scikit-learn.org)

mempelajari pola data yang ada pada *training set* dan model tersebut akan digunakan untuk memprediksi data yang ada pada *test set* untuk melihat seberapa banyak data yang dapat diprediksi dengan tepat. Proses tersebut sesuai dengan tahap *data mining* pada proses KDD karena akan ada proses penemuan pola data pada saat membangun model *decision tree*. Rasio jumlah data antara *training set* dan *test set* juga perlu diatur sebelum proses *training* dilakukan dan rasio yang biasa digunakan adalah 70% untuk *training set* dan 30% untuk *test set*.

Data-data yang masuk dalam *training set* dan *test set* juga dapat diatur menjadi beberapa kombinasi menggunakan metode *k-folds cross validation*. *K-folds cross validation* pertama-tama akan membagi *dataset* menjadi beberapa set data berbeda (*data splitting*) seperti terlihat pada **Gambar 8**. Proses yang selanjutnya dilakukan dalam metode ini adalah melatih serta mengevaluasi model yang dibangun dari semua kombinasi set data yang dapat dibuat. Evaluasi model dapat menjadi lebih baik dengan metode tersebut karena evaluasi performansi model dilakukan dengan lebih banyak kombinasi data yang mungkin muncul. Kombinasi data yang lebih banyak akan memperlihatkan sebaik apa model *decision tree* yang dibangun dengan berbagai set data, sehingga dapat diketahui performansi rata-rata model serta sejauh apa deviasi performansi yang dapat muncul dengan kombinasi *training set* dan *test set* yang berbeda. Hasil evaluasi performansi model tersebut akan digunakan untuk melihat apakah performansi model yang dibuat sudah sesuai dengan standar yang dibutuhkan. Jika performansi model belum sesuai dengan apa yang dibutuhkan maka perlu kembali dilakukan evaluasi pada tahap pra-pengolahan data atau bahkan pada tahap analisis masalah seperti terlihat pada proses yang ada pada **Gambar 2**. Hal tersebut dilakukan untuk mengevaluasi variabel, parameter dan data yang digunakan, sehingga dapat dilakukan perbaikan agar performansi model dapat ditingkatkan.

### 3.2.4. Interpretasi Model

Metode *decision tree* yang bersifat *white box* dapat dengan baik menggambarkan pola data yang ada melalui *tree model*. Hal ini akan membantu dalam

menjelaskan informasi yang diperoleh dari model *decision tree* ini. Model yang dibangun dapat memberikan gambaran kondisi seperti apa yang menandakan bahwa akan terjadi *derating*, serta variabel apa saja yang berhubungan dengan terjadinya kondisi tersebut. Kedua informasi tersebut akan membantu dalam proses interpretasi model yang dibuat, karena pola yang ada dapat ditampilkan dengan jelas. Pemahaman pola data ini akan membuat pendekatan *maintenance* ini menjadi lebih lengkap karena tidak hanya pendekatan prognosis yang dapat dilakukan, tetapi juga diagnosis kondisi yang ada.

## 4. Kesimpulan

Komputerisasi *maintenance management* menghasilkan banyak sekali data, dimana hal ini sangat sesuai dengan pendekatan *machine learning* yang bertujuan untuk mencari informasi dari sekumpulan besar data. Proses untuk membangun model *machine learning* juga sangat mungkin untuk diimplementasikan dalam proses *maintenance management*. Hal tersebut dapat dilihat rancangan implementasi yang telah dibuat, dimana *machine learning* dapat diimplementasikan dalam proses *maintenance management* untuk pencegahan *derating*. Pendekatan *machine learning* tetap dapat mengakomodasi data yang tersedia walaupun memiliki karakteristik yang berbeda dengan data yang biasa. Hasil ini dapat menjadi dasar untuk penelitian ke depan, dimana dapat dilihat sejauh apa performansi dari *machine learning* ini. *Machine learning* memberikan kecepatan pengolahan data, namun dapat dilihat kembali sejauh apa performansi dari *machine learning* dapat memenuhi kebutuhan yang ada dalam *maintenance management*. Hasil performansi ini tentu bisa didapatkan dengan adanya aplikasi langsung dari proses yang telah dijelaskan pada penelitian ini untuk penelitian ke depan.

## 5. Daftar Pustaka

Alaswad, S. & Xiang, Y. (2017). A Review on Condition-based Maintenance Optimization Models for Stochastically Deteriorating System. *Reliability Engineering and System Safety*, 157,



- 54–63.
- Aribowo, W. (2017). An Adaptive Power System Stabilizer Based on Focused Time Delay Neural Network. *Jurnal Teknosains*, Vol. 7 No. 1, 67–73.
- Badan Perencanaan Pembangunan Nasional, Badan Pusat Statistik dan United Nations Population Fund. (2013). *Proyeksi Penduduk Indonesia 2010-2035*. Badan Pusat Statistik. Jakarta.
- Badan Pusat Statistik. (2019). *Konsumsi Listrik per Kapita (MWH/Kapita)*. Diperoleh 18 Juni 2019 dari [https://www.bps.go.id/indikator/indikator/view\\_data/0000/data/1156/sdgs\\_7/1](https://www.bps.go.id/indikator/indikator/view_data/0000/data/1156/sdgs_7/1).
- Ben-Daya, M., Duffuaa, S. O., Raouf, A., Knezevic, J., & Ait-Kadi, D. (2009). *Handbook of Maintenance Management and Engineering*. Springer. London.
- Bloch, H. P., & Geitner, F. K. (1983). *Machinery Failure Analysis and Troubleshooting*. Gulf. Houston.
- Bousdekis, A., Magoutas, B., Apostolou, D., & Mentzas, G. (2018). Review, Analysis and Synthesis of Prognostic-Based Decision Support Methods for Condition Based Maintenance. *Journal Intelligent Manufacturing*, 29 (2018), 1303–1316.
- Bukhsh, Z. A., Saeed, A., Stipanovic, I., & Doree, A. G. (2019). Predictive Maintenance Using Tree-Based Classification Techniques: A Case of Railway Switches. *Transportation Research Part C*, 101, 35–54.
- Delibašić, B., Vukićević, M., Jovanović, M., & Suknović, M. (2013). White-Box or Black-Box Decision Tree Algorithms: Which to Use in Education? *IEEE Transactions on Education*, 56 (3), 287–291.
- Fernandes, M., Canito, A., Bolón-canedo, V., Conceição, L., & Praça, I. (2018). Data Analysis and Feature Selection for Predictive Maintenance: A Case-Study in The Metallurgic Industry. *International Journal of Information Management*.
- Gandhi, K., Schmidt, B., & Ng, A. H. C. (2018). Towards Data Mining Based Decision Support in Manufacturing Maintenance. *Procedia CIRP*, 72 (2018), 261–265.
- Goyal, D., & Pabla, B. S. (2015). Condition Based Maintenance of Machine Tools — A Review. *CIRP Journal of Manufacturing Science and Technology*, 10, 24–35.
- Han, J., M. Kamber, & J. Pei. (2012). *Data mining Concept & Technique*. Morgan Kaufmann. Waltham.
- Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A Review on Machinery Diagnostics and Prognostics Implementing Condition-based Maintenance. *Mechanical Systems and Signal Processing*, 20, 1483–1510.
- Kavakiotis, I., Tsave, O., Salifoglou, A., Maglaveras, N., Vlahavas, I., & Chouvarda, I. (2017). Machine learning and Data mining Methods in Diabetes Research. *Computational and Structural Biotechnology Journal*, 15, 104–116.
- Kothamasu, R., Huang, S. H., & Verduin, W. H. (2006). System Health Monitoring and Prognostics - A Review of Current Paradigms and Practices. *International Journal of Advance Manufacturing Technology*, 28 (2006), 1012–1024.
- Liao, S. H., Chu, P. H., & Hsiao, P. Y. (2012). Data Mining Techniques and Applications - A Decade Review from 2000 to 2011. *Expert Systems with Applications*, 39 (12), 11303–11311.
- Marr, B. (2018). *What is Industry 4.0? Here's A Super Easy Explanation for Anyone*. Diperoleh 24 Agustus 2019 dari <https://www.forbes.com/sites/bernardmarr/2018/09/02/what-is-industry-4-0-heres-a-super-easy-explanation-for-anyone/#4cdd44519788>.
- Peng, Y., Dong, M., & Zuo, M. J. (2010). Current Status of Machine Prognostics in Condition-based Maintenance: A Review. *International Journal of Advance Manufacturing Technology*, 50 (2010), 297–313.
- Possi, M. (2011). Maintenance and Operation. *Elect. Sect. Mag.*, 66 (2011), 32–40.
- Rosihan, R. I., & Yuniarto, H. A. (2019). Analisis Sistem Reliability dengan Pendekatan Reliability Block Diagram. *Jurnal Teknosains*, Vol. 9 No. 1, 57–67.
- Sakib, N., & Wuest, T. (2018). Challenges and Opportunities of Condition-based Predictive Maintenance: A Review. *Procedia CIRP*, 78 (2018), 267–272.
- Scikit-learn Developers. (2019). *Cross-validation: Evaluating Estimator Performance*. Diperoleh 29 November 2019 dari [https://scikit-learn.org/stable/modules/cross\\_validation.html](https://scikit-learn.org/stable/modules/cross_validation.html).
- Swanson, L. (2001). Linking Maintenance Strategies to Performance. *International Journal of Production Economics*, 70, 237–244.
- Vadim, K. (2018). Overview of Different Approaches to Solving Problems of Data Mining. *Procedia Computer Science*, 123, 234–239.
- Wang, K., Li, Z., Braaten, J., & Yu, Q. (2015). Interpretation and Compensation of Backlash Error Data in Machine Centers for Intelligent Predictive Maintenance Using ANNs. *Advances in Manufacturing*, 3 (2), 97–104.
- Wicaksono, P. E. (2019). *Seluruh Pembangkit di Proyek 35 Ribu MW Beroperasi 2024*. Diperoleh 18 Juni 2019 dari <https://www.liputan6.com/bisnis/read/3879693/seluruh-pembangkit-di-proyek-35-ribu-mw-beroperasi-2024>.