

Predictive Maintenance untuk Kendaraan Bermotor dengan Menggunakan Support Vector Machine (SVM)

Sindy Caesar Rusnanda Himatul Haliza¹, Anita Qoiriah²

¹Jurusan Teknik Informatika/Teknik Informatika, Universitas Negeri Surabaya

²Jurusan Teknik Informatika/Teknik Informatika, Universitas Negeri Surabaya

¹sindyhaliza16051204037@mhs.unesa.ac.id

²anitaqoiriah@unesa.ac.id

Abstrak— Seiring dengan bertambahnya kebutuhan hidup masyarakat, kebutuhan transportasi juga semakin bertambah. Kendaraan bermotor adalah salah satu dari sekian banyak alat transportasi yang digunakan untuk memindahkan manusia dan barang, kendaraan bermotor saat ini banyak digunakan dan menunjang kehidupan masyarakat masa kini. Selama penggunaan, kendaraan bermotor dapat mengalami kerusakan sehingga perlu diadakan pemeliharaan dan perbaikan. Dalam hal ini proses pemeliharaan merupakan peranan penting dari kendaraan bermotor. Pemeliharaan (*maintenance*) dilakukan untuk mencegah terjadinya kegagalan pada sistem dan juga untuk membuat sistem berfungsi kembali apabila kegagalan telah terjadi. Proses *maintenance* bertujuan untuk menjaga keandalan mesin (*reliability*) sehingga mesin akan berfungsi dengan normal. *Maintenance* adalah sebuah kegiatan wajib dan rutin yang dilakukan untuk kendaraan bermotor. Pada penelitian ini akan dilakukan *predictive maintenance* untuk kendaraan dinas roda-4 dari DPU Bina Marga Jatim menggunakan algoritma *support vector machine* (SVM) yang menggunakan metode *one-against-all* dan metode *one-against-one*. Kesimpulan yang didapat dari penelitian ini adalah pengujian dari penelitian ini mendapatkan hasil akurasi yang tinggi, *predictive maintenance* untuk kendaraan bermotor ini menghasilkan nilai akurasi yang mencapai 92.92% dengan menggunakan algoritma *support vector machine* baik yang menggunakan metode *one-against-all* maupun yang menggunakan metode *one-against-one*. Kernel *polynomial* dan kernel *radial basis function* sama-sama menghasilkan akurasi tertinggi 92.92%, akan tetapi kernel *polynomial* menghasilkan akurasi tertinggi lebih banyak dibandingkan dengan kernel *radial basis function* dalam beberapa percobaan.

Kata Kunci— *predictive maintenance*, *support vector machine*, kendaraan bermotor

I. PENDAHULUAN

Seiring dengan bertambahnya kebutuhan hidup masyarakat, kebutuhan transportasi juga semakin bertambah. Alat transportasi atau kendaraan dibutuhkan untuk memudahkan kehidupan manusia. Kendaraan bermotor adalah salah satu dari sekian banyak alat transportasi yang digunakan untuk memindahkan manusia dan barang, kendaraan bermotor saat ini banyak digunakan dan menunjang kehidupan masyarakat masa kini. Selama penggunaan, kendaraan bermotor dapat mengalami kerusakan sehingga perlu diadakan pemeliharaan dan perbaikan. Dalam hal ini proses pemeliharaan merupakan

peranan penting dari kendaraan bermotor. Pemeliharaan (*maintenance*) dilakukan untuk mencegah terjadinya kegagalan pada sistem dan juga untuk membuat sistem berfungsi kembali apabila kegagalan telah terjadi [1]. Proses *maintenance* bertujuan untuk menjaga keandalan mesin (*reliability*) sehingga mesin akan berfungsi dengan normal. *Maintenance* adalah sebuah kegiatan wajib dan rutin yang dilakukan untuk kendaraan bermotor, terutama untuk kendaraan bermotor yang menunjang pekerjaan seperti kendaraan dinas. *Maintenance* dilakukan untuk memperpanjang umur kendaraan dinas dan untuk menyediakan alat transportasi yang *reliable* bagi penggunanya.

Prediksi atau *forecasting* dihasilkan dari kegiatan memprediksi kemungkinan yang terjadi di masa depan. Prediksi dapat dilakukan dengan menggunakan data masa lampau yang telah ada, lalu diproyeksikan dalam bentuk matematis untuk memprediksi masa yang akan datang. Prediksi memperlihatkan kejadian yang akan terjadi pada keadaan-keadaan tertentu di masa depan, dan prediksi adalah masukan untuk proses perencanaan dan pengambilan keputusan.

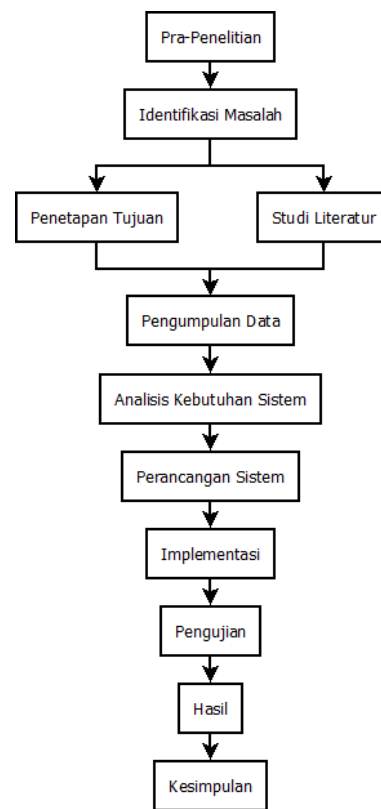
Predictive maintenance adalah pemeliharaan yang menggunakan alat prediksi untuk menentukan kapan tindakan pemeliharaan diperlukan. Hal ini didasarkan pada pemantauan mesin secara terus-menerus, yang memungkinkan pemeliharaan hanya dilakukan saat diperlukan. Selain itu, *predictive maintenance* juga memungkinkan deteksi dini kegagalan berdasarkan data historis [2].

SVM (*Support Vector Machine*) adalah salah satu metode dari beberapa metode *machine learning* atau pembelajaran mesin yang termasuk dalam kategori *supervised learning*. SRM (*Structural Risk Minimization*) adalah prinsip kerja dari metode *support vector machine*, SRM memiliki tujuan untuk menemukan pemisah kelas yang juga disebut sebagai *hyperplane* yang paling baik yang digunakan untuk memisahkan dua buah kelas dengan cara memaksimalkan jarak antar kelas. Penelitian yang dilakukan oleh Darmawan, Kustian, & Rahayu pada tahun 2018 untuk prediksi kepuasan pengunjung taman Tabebuya dengan menggunakan metode *support vector machine* menghasilkan akurasi dan rata-rata kinerja sebanyak 86,00% dan nilai AUC atau area di bawah kurva yang dihasilkan adalah 0,947 [3]. Hasil penelitian yang dilakukan oleh Apriana & Handayani pada tahun 2017 untuk

perkiraan beban daya listrik jangka pendek menggunakan algoritma *neural network* dan *support vector machine* menyimpulkan bahwa nilai RMSE (*Root Mean Square Error*) yang diperoleh dari pengujian menggunakan algoritma *neural network* adalah 0,035, sedangkan *root mean square error* yang diperoleh dari pengujian menggunakan algoritma *support vector machine* lebih rendah yaitu sebesar 0,034 [4]. Hasil penelitian dari Pratama, Wihandika, & Ratnawati pada tahun 2018 untuk prediksi ketepatan waktu kelulusan mahasiswa dengan menggunakan metode *support vector machine* memperoleh rata-rata akurasi sebesar 80,55% [5]. Hasil penelitian yang dilakukan oleh Jain, Patel, Bhambure, Sen, & Shaikh pada tahun 2019 tentang pendekatan *predictive maintenance* untuk mobil menyimpulkan bahwa *predictive maintenance* akan menghindarkan kendaraan dari kerusakan dan memperkecil biaya perawatan karena pengguna dapat mengetahui kesalahan yang terjadi pada kendaraan sehingga pengguna tidak bisa dibodohi oleh mekanik [6]. Sebuah tinjauan yang dilakukan oleh Carvalho, Soares, Vita, & Francisco pada tahun 2019 merangkum metode-metode yang biasanya dilakukan untuk *predictive maintenance* yaitu *support vector machine*, *random forest*, *artificial neural network*, dan *k-means* [7]. Selanjutnya adalah penelitian dari Chaudhuri pada tahun 2018 tentang *predictive maintenance* untuk armada kendaraan dengan menggunakan *hierarchical modified fuzzy support vector machine* menyimpulkan bahwa *predictive maintenance* akan memperpanjang usia kendaraan dan mengoptimalkan produktivitas kendaraan [8]. Dalam penelitian ini akan dilakukan *predictive maintenance* untuk kendaraan dinas roda-4 DPU Bina Marga Jatim dengan menggunakan algoritma *support vector machine*. Diharapkan dengan dilakukan penelitian "Predictive Maintenance untuk Kendaraan Bermotor dengan Menggunakan Support Vector Machine (SVM)" ini dapat memberikan nilai akurasi dengan nilai error terkecil dibandingkan dengan metode lain.

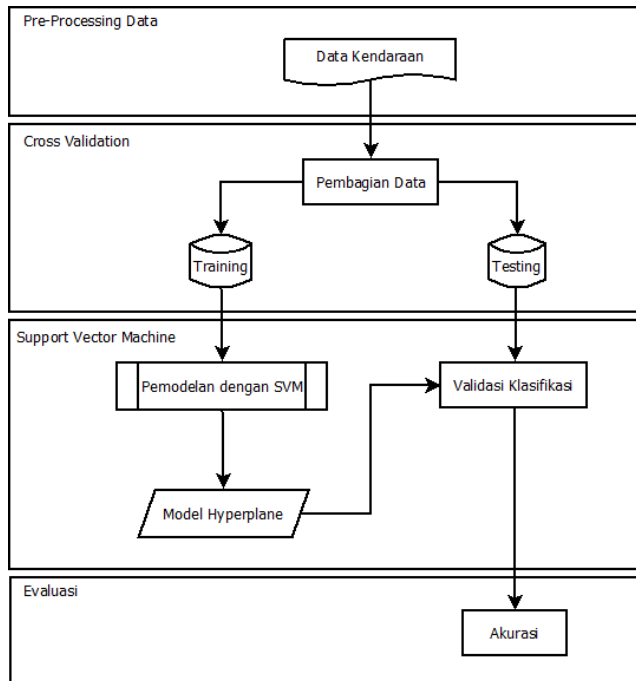
II. METODOLOGI PENELITIAN

Pada bab ini akan dibahas mengenai metode penelitian yang meliputi langkah-langkah penelitian, langkah-langkah pengumpulan data, langkah-langkah pengolahan data, algoritma yang digunakan dalam penelitian, dan rancangan pengujian. Gambar 1 memperlihatkan langkah-langkah dari penelitian.



Gbr. 1 Langkah Penelitian

Secara garis besar, proses pengolahan data dalam penelitian ini dibagi menjadi tahap *pre-processing* data, pengolahan data dengan *support vector machine*, dan evaluasi. Gambar 2 menunjukkan alur dari pengolahan data.



Gbr. 2 Alur pengolahan data

Tahap pertama adalah *preprocessing* data, *preprocessing* data adalah sebuah proses yang dijalankan untuk mengubah data asli menjadi data yang lebih berkualitas dan sesuai untuk menjadi *input* yang baik bagi sistem. Data yang digunakan dalam penelitian ini berasal dari DPU Bina Marga Jatim. Pada tahapan *pre-processing*, akan dilakukan normalisasi data, data mentah yang diperoleh lalu diubah menjadi bentuk data yang dibutuhkan dan dikelompokkan untuk diolah. Tahap *pre-processing* juga mencakup proses seleksi atau pemilihan data yang selanjutnya akan digunakan untuk *input* pada pemrosesan data.

Tahapan selanjutnya adalah mengimplementasikan algoritma *support vector machine* dan *cross validation*. *Cross validation* bisa juga disebut estimasi rotasi, *cross validation* akan membagi data ke dalam dua kelompok, yaitu kelompok data *training* atau data latih dan kelompok data *testing* atau data uji. Data *training* adalah data yang akan digunakan dalam melakukan *training* atau pelatihan algoritma, sedangkan data *testing* adalah data yang akan digunakan untuk *testing* atau proses pengujian algoritma. Awalnya, data yang masuk akan diproses dengan melakukan pelatihan atau *training* data. Setelah data melalui proses *training*, dilakukan pertukaran dimana data *testing* akan menjadi data *training* dan begitu pula sebaliknya, data yang awalnya merupakan data *training* kini akan menjadi data *testing*. Data *training* akan melalui proses pemodelan klasifikasi menggunakan *support vector machine* sampai mendapatkan model *hyperplane*. Setelah model *hyperplane* didapatkan, model tersebut lalu diterapkan pada data *testing*. Tahapan yang terakhir adalah evaluasi yang dilakukan dengan cara melakukan penghitungan akurasi.

A. Pre-Processing Data

Data yang digunakan penulis dalam penelitian ini didapatkan dari DPU Bina Marga Jatim, penulis mengambil data kendaraan dinas roda-4 yang berjumlah 300 kendaraan. Selain kendaraan dinas roda-4 DPU Bina Marga Jatim juga mempunyai kendaraan dinas roda-2 dan alat-alat berat. Bina marga bertugas dalam perencanaan pembangunan, pemeliharaan jalan, jembatan dan sebagainya. Bina marga memiliki banyak alat berat dan truk-truk besar karena tugasnya yang berkaitan dengan pembangunan ini. Bidang aset di DPU Bina Marga Jatim adalah bidang yang menangani masalah kendaraan dinas dan sebagainya yang ada pada Bina Marga di kota-kota seluruh Jawa Timur.

Dua jenis variabel dibutuhkan dalam penelitian ini, yaitu variabel prediktor atau *input* (X) dan variabel respon atau *output* (Y). Untuk variabel respon yang digunakan adalah kategori keadaan kendaraan dinas roda-4 yang terdiri dari 3 kategori keadaan kendaraan yaitu:

Y = (1) *Immediate Risk*

Y = (2) *Short Term Risk*

Y = (3) *Long Term Risk*

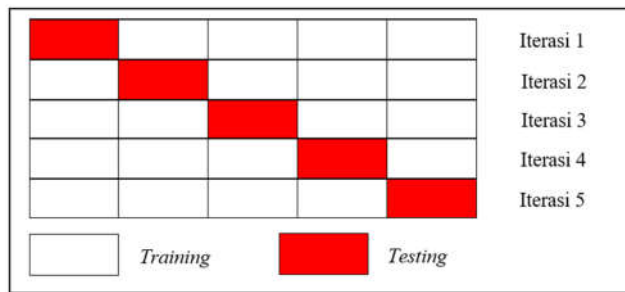
Tabel I dibawah ini merupakan variabel prediktor kategori keadaan dinas roda-4.

TABEL I
VARIABEL PREDIKTOR

Variabel	Kategori
Usia Kendaraan (X ₁)	0=kurang dari 6 tahun; 1=6-10 tahun; 2=lebih dari 10 tahun
Jumlah Perbaikan Kendaraan (X ₂)	0=kurang dari 8 perbaikan; 1=8-15 perbaikan; 2=lebih dari 15 perbaikan
Rata-rata Waktu Kerja (X ₃)	0=kurang dari 4 jam/hari; 1=4-12 jam/hari; 2=lebih dari 12 jam/hari
Rata-rata Biaya Perbaikan (X ₄)	0=kurang dari 1 juta; 1=1-2 juta; 2=lebih dari 2 juta

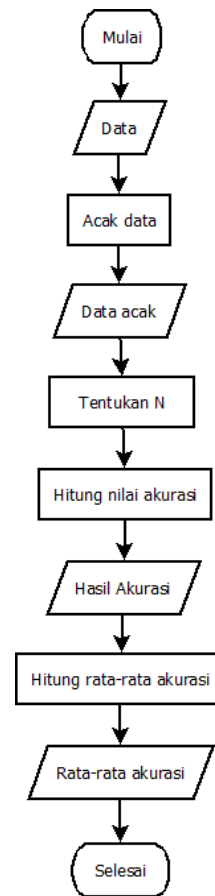
B. K-folds Cross Validation

Cross validation atau bisa juga disebut validasi silang adalah sebuah metode untuk membandingkan algoritma pembelajaran dan mengevaluasinya. Langkah awal dari *cross validation* adalah melakukan pembagian data menjadi dua bagian data, yaitu data latih atau *training* untuk pelatihan dan data uji atau *testing* untuk pengujian [9]. Salah satu bentuk dari *cross validation* adalah *k-folds cross validation*. Gambar 3 merupakan ilustrasi dari *k-folds cross validation*.



Gbr. 3 Ilustrasi *k-folds cross validation*

Cara kerja dari *k-folds cross validation* adalah dengan cara membagi data (D) menjadi k bagian data seperti $D_1, D_2, D_3, \dots, D_k$ dengan jumlah data yang sama pada masing-masing bagian. Proses pelatihan dan pengujian data akan dilaksanakan sebanyak k kali. Pada proses pertama, bagian D_1 digunakan untuk data *testing* dan bagian-bagian data lainnya digunakan untuk data *training*, pada proses kedua, bagian data $D_1, D_3, \dots, D_k$ digunakan untuk data *training* dan bagian D_2 digunakan untuk data *testing*, dan seterusnya sampai semua bagian data sudah pernah digunakan menjadi data *testing* [10]. Permasalahan yang memiliki jumlah sampel terbatas sangat cocok untuk menggunakan metode *k-folds cross validation*. Tahapan dalam *k-fold cross validation* ditunjukkan pada gambar 4.



Gbr. 4 Tahapan *k-folds cross validation*

Ada beberapa tahapan dalam *k-folds cross validation*, yaitu menentukan jumlah K , mengacak data, membagi data menjadi *training* dan *testing*, dan menghitung akurasi. Penelitian ini menggunakan 3 jenis K dengan jumlah berbeda, yaitu $K=3$, $K=5$, dan $K=10$. Pembagian ini dilakukan untuk melihat jumlah K mana yang menghasilkan akurasi lebih tinggi.

C. Support Vector Machine

Support vector machine atau SVM adalah salah satu dari banyak teknik pembelajaran mesin atau *machine learning* yang bisa terbelang baru dalam bidang prediksi. *Support vector machine* termasuk dalam *supervised learning*. *Supervised learning* adalah algoritma yang dilatih terlebih dahulu untuk bisa melakukan klasifikasi dan prediksi. *Support vector machine* berfungsi untuk mencari *hyperplane* dengan cara mencari jarak terbesar antara kelas. *Hyperplane* merupakan sebuah fungsi. *Support vector machine* dapat melakukan klasifikasi data yang bersifat *non-linear separable* dan *linearly separable* [11]. Data yang bisa dipisahkan secara linier disebut *linearly separable*. Contohnya seperti dataset $\{x_1, \dots, x_n\}$ dan $x_i \in \mathbb{R}^d$, serta $y_i \in \{+1, -1\}$ adalah label kelas dari data x_i . x yang berada pada *hyperplane* akan memenuhi persamaan $w \cdot x + b = 0$ apabila ada beberapa *hyperplanes* yang bisa memisahkan sampel positif dan sampel negatif. Algoritma

support vector machine hanya mencari *hyperplane* yang memiliki margin yang paling besar untuk mengatasi permasalahan *linear*. Margin adalah jarak antara dua kelas. *Hyperplane* yang terbaik adalah *hyperplane* yang bisa menjadi pemisah data yang baik dan memiliki ukuran margin yang paling besar. *Support vector* adalah data dalam bidang pembatas.

Melakukan pemetaan data ke ruang yang berdimensi lebih tinggi yaitu *feature space* atau ruang fitur adalah cara untuk menyelesaikan permasalahan data *non-linear* pada *support vector machine* [11], dengan menggunakan transformasi $\Phi: \mathbb{R}^d \mapsto H$ data yang ada pada ruang ini bisa dipisahkan secara linier.

Maka, dengan demikian *dot product* dalam H akan mempengaruhi algoritma pelatihan. Sebagai contoh $\Phi(x_i) \cdot \Phi(x_j)$. Algoritma untuk *training* hanya membutuhkan fungsi *kernel* K , dan tidak harus mengetahui transformasi Φ secara pasti apabila ada fungsi *kernel* K seperti $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$.

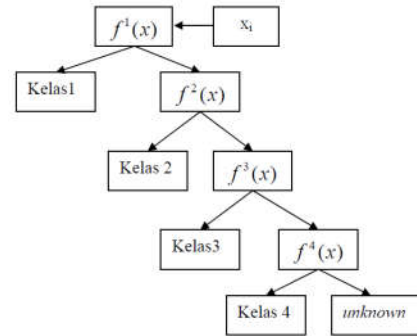
Performa dari *support vector machine* sangat bergantung dari nilai parameter-parameternya, seperti *kernel*. Apabila terdapat data yang susah atau tidak bisa diklasifikasikan secara linier, *support vector machine* menggunakan metode *kernel* sebagai solusinya. *Kernel linear*, *gaussian*, dan *polynomial* adalah beberapa *kernel* yang paling sering digunakan.

Support vector machine hanya bisa mengelompokkan data yang ada ke dalam dua buah kelas atau bisa disebut klasifikasi biner saat pertama kali dikenalkan. Berkat penelitian pengembangan lebih lanjut, sekarang *support vector machine* dapat melakukan klasifikasi untuk data yang mempunyai kelas lebih dari dua atau biner. *Support vector machine* yang mempunyai kelas lebih dari dua atau bisa juga disebut *multiclass* memiliki dua buah cara untuk pengimplementasiannya, yaitu dengan membuat sebuah bentuk permasalahan optimasi dengan cara menggabungkan semua data yang ada pada beberapa kelas atau dengan cara membuat penggabungan dari beberapa *support vector machine* biner. Metode *one-against-one* dan metode *one-against-all* adalah dua cara yang biasanya digunakan untuk mengimplementasikan *multiclass support vector machine*.

Pada metode *one-against-all* akan membangun model *support vector machine* biner sebanyak k buah. Contohnya adalah seperti masalah klasifikasi dengan 4 buah kelas. 4 buah *support vector machine* biner digunakan untuk pelatihan seperti yang terlihat pada tabel II dan gambar 5 adalah penggunaan *one-against-all* dalam mengklasifikasikan data.

TABEL II
CONTOH SUPPORT VECTOR MACHINE DENGAN METODE ONE-AGAINST-ALL

$y_i = 1$	$y_i = -1$	Hipotesis
Class 1	Bukan Class 1	$f^1(x) = (w^1)x + b^1$
Class 2	Bukan Class 2	$f^2(x) = (w^2)x + b^2$
Class 3	Bukan Class 3	$f^3(x) = (w^3)x + b^3$
Class 4	Bukan Class 4	$f^4(x) = (w^4)x + b^4$

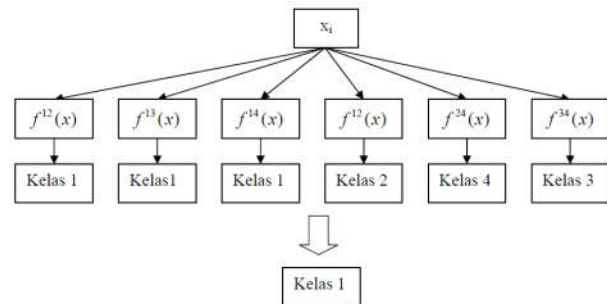


Gbr. 5 Contoh metode *One-against-all*

Pada metode *one-against-one* akan dibangun sejumlah $\left(\frac{k(k-1)}{2}\right)$ buah model klasifikasi biner (k adalah jumlah kelas). Data yang diambil dari dua *class* digunakan untuk melatih setiap model klasifikasi. Ada beberapa metode yang bisa diterapkan untuk melangsungkan proses pengujian setelah semua dari $\left(\frac{k(k-1)}{2}\right)$ model klasifikasi telah usai dibangun, salah satu contoh metode yang bisa digunakan untuk pengujian adalah metode *voting*. Contohnya dapat dilihat pada tabel III dan gambar 6.

TABEL III
CONTOH SUPPORT VECTOR MACHINE DENGAN METODE ONE-AGAINST-ONE

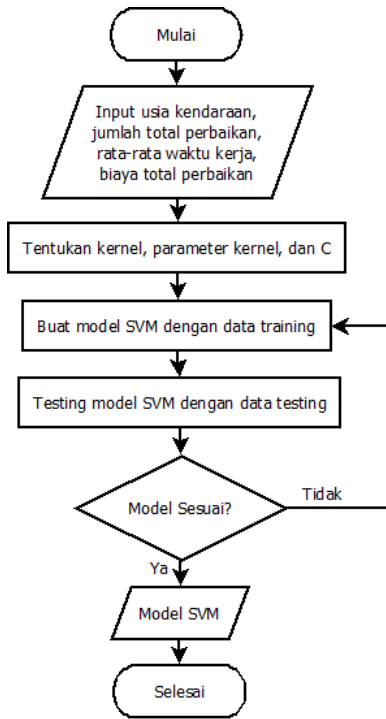
$y_i = 1$	$y_i = -1$	Hipotesis
Class 1	Class 2	$f^{12}(x) = (w^{12})x + b^{12}$
Class 1	Class 3	$f^{13}(x) = (w^{13})x + b^{13}$
Class 1	Class 4	$f^{14}(x) = (w^{14})x + b^{14}$
Class 2	Class 3	$f^{23}(x) = (w^{23})x + b^{23}$
Class 2	Class 4	$f^{24}(x) = (w^{24})x + b^{24}$
Class 3	Class 4	$f^{34}(x) = (w^{34})x + b^{34}$



Gbr. 6 Contoh metode *One-against-one*

Langkah pertama dalam proses *support vector machine* adalah melakukan *input* data kendaraan dinas roda-4. Langkah selanjutnya adalah menentukan parameter *kernel* dan C , penelitian ini menggunakan dua buah *kernel* yang ada pada *support vector machine* untuk dibandingkan, yaitu *kernel*

polynomial dan kernel radial basis function (RBF). Langkah selanjutnya adalah melakukan *training* dan *testing*. Gambar 7 menunjukkan alur dari SVM.



Gbr. 7 Alur Algoritma SVM

Dataset yang telah ada akan dibagi menjadi dua bagian data, yaitu data *training* dan data *testing*, lalu *support vector machine* melakukan proses *training* terhadap data *training* tersebut. Sebelum melakukan proses *training*, tentukan dulu jenis *kernel*, nilai parameter *kernel*, dan C. Pemilihan nilai parameter dan C yang tepat dapat meningkatkan akurasi *support vector machine*. Proses *training* akan menghasilkan model *support vector machine* terbaik untuk melakukan proses testing apabila sudah menemukan parameter terbaik. Persamaan (1) berikut adalah bentuk penulisan model dari *support vector machine*.

$$f(\phi(x)) = \text{sign}(w \cdot \phi(x) + b) \quad (1)$$

Keterangan:

$f(\phi(x))$ = fungsi keputusan

w = bobot

$\phi(x)$ = fungsi kernel

b = bias

D. Evaluasi

Melakukan proses evaluasi memiliki tujuan untuk mencari tahu tingkat akurasi yang dilakukan dengan cara menghitung jumlah data yang sudah diprediksi yang mengeluarkan hasil yang akurat. Evaluasi dilakukan dengan cara melakukan perhitungan nilai *accuracy*.

Accuracy atau akurasi adalah ukuran ketepatan dari kejadian yang diinginkan, *accuracy* berfungsi untuk mengukur

tingkat persamaan dari nilai sesungguhnya apabila dibandingkan dengan nilai prediksi. Untuk menghitung nilai *accuracy* dibutuhkan perhitungan dengan rumus yang ada pada persamaan (5).

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

Keterangan:

1. TP atau *True Positive* merupakan total dari nilai positif yang juga diklasifikasikan sebagai nilai positif.
2. TN atau *True Negative* merupakan total dari nilai negatif yang juga diklasifikasikan sebagai nilai negatif.
3. FP atau *False Positive* merupakan total dari nilai negatif yang diklasifikasikan sebagai nilai positif.
4. FN atau *False Negative* merupakan total dari nilai positif yang diklasifikasikan sebagai nilai negatif.

III. HASIL DAN PEMBAHASAN

A. Pre-Processing Data

Penelitian ini menggunakan data yang diambil dari data kendaraan dinas roda-4 yang ada pada DPU Bina Marga Jatim yang berjumlah 300 kendaraan. Data terdiri dari 5 atribut, dengan 4 atribut sebagai prediktor dan 1 atribut sebagai label. Tabel IV adalah informasi mengenai atribut dan nilainya.

TABEL IV
DAFTAR ATRIBUT DAN NILAINYA

No.	Atribut	Nilai Atribut
1.	Usia Kendaraan	≤ 5 tahun 6-10 tahun > 10 tahun
2.	Jumlah Perbaikan Kendaraan	≤ 5 perbaikan 6-10 perbaikan > 10 perbaikan
3.	Rata-rata Waktu Kerja	≤ 4 jam/hari 5-12 jam/hari > 12 jam/hari
4.	Total Biaya Perbaikan	< 1.000.000 1.000.000 – 5.000.000 > 5.000.000
5.	Keadaan Kendaraan	<i>Immediate Risk</i> <i>Short Term Risk</i> <i>Long Term Risk</i>

Atribut data yang diperoleh akan diubah menjadi numerik. Setiap nilai dikonversi menjadi angka numerik berbeda yang berupa bilangan bulat yang dimulai dari nol. Konversi nilai atribut ditunjukkan pada tabel V.

TABEL V
KONVERSI NILAI ATRIBUT

No.	Atribut	Nilai Atribut	Nilai Numerik
1.	Usia Kendaraan	≤ 5 tahun 6-10 tahun > 10 tahun	0 1 2
2.	Jumlah Perbaikan Kendaraan	≤ 7 perbaikan 8-15 perbaikan	0 1

		> 15 perbaikan	2
3.	Rata-rata Waktu Kerja	≤ 4 jam/hari	0
		5-12 jam/hari	1
		> 12 jam/hari	2
4.	Rata-rata Biaya Perbaikan	< 1.000.000	0
		1.000.000 – 2.000.000	1
		> 2.000.000	2
5.	Keadaan Kendaraan	Long Term Risk	0
		Short Term Risk	1
		Immediate Risk	2

Tabel VI berikut ini merupakan dataset keadaan kendaraan dinas Bina Marga Jawa Timur.

TABEL VI
DATASET

Usia	Jumlah Perbaikan	Rata-rata Waktu Kerja	Rata-rata Biaya Perbaikan	Keadaan Kendaraan
1	1	0	1	1
0	1	0	0	0
1	1	1	1	1
1	2	0	1	1
1	2	1	1	1
2	2	1	1	2
0	0	1	0	0
0	0	0	0	0
2	2	0	2	2
2	2	1	2	2

B. Implementasi Support Vector Machine

Penelitian ini menggunakan bahasa *python* dalam mengimplementasikan metode *support vector machine*. Metode *Support vector machine* berguna untuk melakukan proses *training* dan *testing* terhadap *dataset*.

1. Menginputkan nilai parameter seperti C, γ, d, λ
2. Perhitungan kernel
3. Perhitungan matriks *Hessian* dengan rumus seperti persamaan (6).
$$D_{ij} = y_i y_j (K(x_i x_j) + \lambda^2) \quad (6)$$
4. Melakukan training SVM:
 - a. Menghitung nilai E_i dengan rumus seperti persamaan (7).
$$E_i = \sum_{j=1}^1 \alpha_i D_{ij} \quad (7)$$
 - b. Menghitung nilai $\delta \alpha_i$ dengan rumus seperti persamaan (8).
$$\delta \alpha_i = \min\{\max[\gamma(1 - E_i), -\alpha_i], C - \alpha_i\} \quad (8)$$
 - c. Menentukan nilai α_i baru. Nilai α_i paling besar pada kelasnya dijadikan support vector (x^+ dan x^-). x^+ adalah data latih yang mempunyai nilai α_i paling besar di kelas positif. x^- adalah data latih yang mempunyai nilai α_i paling besar di kelas negatif.
 - d. Melakukan perhitungan $k(x_i, x^+)$ dan $k(x_i, x^-)$.
5. Melakukan testing SVM:
 - a. Menghitung nilai b (*bias*).
 - b. Menghitung nilai $f(x)$.
 - c. Menghitung nilai *one-against-one* dan *one-against-all*.

Perhitungan ini menghasilkan bidang pemisah yang memisahkan dataset menjadi tiga kelas berbeda, yaitu *Long Term Risk*, *Short Term Risk*, dan *Immediate Risk*. Berikut adalah contoh hasil dari uji coba:

Contoh dibawah adalah data kendaraan:

1. Usia kendaraan : 1 tahun
2. Jumlah perbaikan kendaraan : 2 perbaikan
3. Rata-rata waktu kerja : 2 jam/hari
4. Rata-rata biaya perbaikan : Rp 0,-

Proses yang telah dilakukan menghasilkan hasil bahwa kendaraan tersebut berada pada keadaan *Long Term Risk*.

Tabel VII berikut merupakan perbandingan hasil *output* dari metode *support vector machine* dan dataset yang menunjukkan bahwa keduanya sesuai. Hal ini menunjukkan bahwa sistem *predictive maintenance* untuk kendaraan bermotor dengan metode SVM berjalan dengan baik.

TABEL VII
PERBANDINGAN OUTPUT

Usia Kendaraan	Jumlah Perbaikan Kendaraan	Rata-rata Waktu Kerja	Rata-rata Biaya Perbaikan	Keadaan Kendaraan	
				Dataset	SVM
8 (1)	16 (2)	2 (0)	1.200.000 (1)	Short Term Risk (1)	Short Term Risk (1)
8 (1)	16 (2)	5 (1)	1.350.000 (1)	Short Term Risk (1)	Short Term Risk (1)
15 (2)	30 (2)	6 (1)	1.500.000 (1)	Immediate Risk (2)	Immediate Risk (2)
3 (0)	6 (0)	10 (1)	500.000 (0)	Long Term Risk (0)	Long Term Risk (0)
3 (0)	5 (0)	2 (0)	300.000 (0)	Long Term Risk (0)	Long Term Risk (0)

C. Evaluasi

Selanjutnya adalah proses evaluasi dari *support vector machine*. Teknik *k-folds cross validation* yang digunakan dalam penelitian ini menggunakan jumlah $K = 3, 5$, dan 10 untuk mengetahui pengaruh jumlah K terhadap nilai akurasi. Data (D) dibagi menjadi k bagian data yaitu seperti $D_1, D_2, \dots, D_k$. Pada setiap bagian data memiliki jumlah ukuran data yang sama antara satu dengan yang lain. Pada percobaan pertama, D_1 menjadi data *testing* dan bagian data selain D_1 menjadi data *training*. Pada percobaan kedua, D_2 menjadi data *testing* dan bagian data selain D_2 menjadi data *training*, dan seterusnya sampai D_k menjadi data *testing* yang terakhir.

Predictive maintenance untuk kendaraan bermotor dengan menggunakan *support vector machine* pada penelitian ini menggunakan dua buah metode untuk *multiclass support vector machine*, metode yang digunakan adalah metode *one-against-one* dan metode *one-against-all*. Pada penelitian ini juga digunakan dua buah fungsi *kernel*, yaitu *kernel polynomial* dan *kernel radial basis function* (RBF). Parameter d yang digunakan pada *kernel polynomial* berjumlah sebanyak 3 jenis, yaitu $d=1$, $d=2$, dan $d=3$. Lalu untuk parameter γ yang digunakan pada *kernel radial basis function* juga berjumlah

sebanyak 3 jenis, yaitu $\gamma=1$, $\gamma=2$, dan $\gamma=3$. Penelitian ini juga membandingkan parameter C, yaitu C=1, C=10, C=100, dan C=1000. C (*Cost*) merupakan sebuah parameter yang berguna untuk menentukan besar penalti apabila ada kesalahan dalam pengklasifikasian data. Pada *predictive maintenance* untuk kendaraan bermotor ini membandingkan empat nilai C yang berbeda, dan membandingkan tiga nilai d pada fungsi kernel *polynomial* dan tiga nilai γ pada fungsi kernel *radial basis function* (RBF) untuk mengetahui pengaruh kombinasi nilai *hyperparameter* terhadap nilai akurasi *support vector machine*.

TABEL VIII
TINGKAT AKURASI SVM ONE-AGAINST-ALL DENGAN C=1

K	Polynomial			RBF		
	d=1	d=2	d=3	$\gamma=1$	$\gamma=2$	$\gamma=3$
3	92.92	92.92	92.08	92.92	87.5	87.5
5	92.5	84.58	84.58	84.58	84.58	83.33
10	92.92	90.83	90.83	90.83	90.83	89.58

TABEL IX
TINGKAT AKURASI SVM ONE-AGAINST-ONE DENGAN C=1

K	Polynomial			RBF		
	d=1	d=2	d=3	$\gamma=1$	$\gamma=2$	$\gamma=3$
3	92.92	92.92	92.08	92.92	87.5	87.5
5	92.5	84.58	84.58	84.58	84.58	83.33
10	92.92	90.83	90.83	90.83	90.83	89.58

Tabel VIII dan IX menunjukkan bahwa nilai akurasi tertinggi pada C=1 terdapat pada pengelompokan SVM yang menggunakan jumlah K=3 dan K=10 dengan fungsi kernel *polynomial* d=1, d=2, dan K=3 dengan fungsi kernel RBF $\gamma=1$. Pada pengelompokan ini tingkat akurasi tertinggi mencapai 92.92%. Kedua tabel tersebut menunjukkan bahwa tingkat akurasi SVM *one-against-all* dan SVM *one-against-one* adalah sama.

TABEL X
TINGKAT AKURASI SVM ONE-AGAINST-ALL DENGAN C=10

K	Polynomial			RBF		
	d=1	d=2	d=3	$\gamma=1$	$\gamma=2$	$\gamma=3$
3	92.92	92.92	92.08	92.92	87.5	87.5
5	84.17	84.58	84.58	84.58	84.58	83.33
10	90.83	90.83	90.83	90.83	90.83	89.58

TABEL XI
TINGKAT AKURASI SVM ONE-AGAINST-ONE DENGAN C=10

K	Polynomial			RBF		
	d=1	d=2	d=3	$\gamma=1$	$\gamma=2$	$\gamma=3$
3	92.92	92.92	92.08	92.92	87.5	87.5
5	84.17	84.58	84.58	84.58	84.58	83.33
10	90.83	90.83	90.83	90.83	90.83	89.58

Berdasarkan tabel X dan XI, nilai akurasi tertinggi pada C=10 terdapat pada pengelompokan SVM yang menggunakan jumlah K=3 dengan fungsi kernel *polynomial* yang menggunakan nilai d=1, d=2 dan kernel RBF yang menggunakan $\gamma=1$. Tingkat akurasi tertinggi mencapai 92.92% pada pengelompokan ini. Kedua tabel tersebut menunjukkan bahwa tingkat akurasi SVM *one-against-all* dan SVM *one-against-one* adalah sama.

TABEL XII
TINGKAT AKURASI SVM ONE-AGAINST-ALL DENGAN C=100

K	Polynomial			RBF		
	d=1	d=2	d=3	$\gamma=1$	$\gamma=2$	$\gamma=3$
3	92.92	92.92	92.08	92.92	87.5	87.5
5	84.17	84.58	84.58	84.58	84.58	83.33
10	90.83	90.83	90.83	90.83	90.83	89.58

TABEL XIII
TINGKAT AKURASI SVM ONE-AGAINST-ONE DENGAN C=100

K	Polynomial			RBF		
	d=1	d=2	d=3	$\gamma=1$	$\gamma=2$	$\gamma=3$
3	92.92	92.92	92.08	92.92	87.5	87.5
5	84.17	84.58	84.58	84.58	84.58	83.33
10	90.83	90.83	90.83	90.83	90.83	89.58

Dapat dilihat dari tabel XII dan XIII bahwa tingkat akurasi tertinggi pada C=100 terdapat pada pengelompokan SVM yang menggunakan jumlah K=3 dengan fungsi kernel *polynomial* yang menggunakan nilai d=1, d=2 dan kernel RBF yang menggunakan $\gamma=1$. Tingkat akurasi tertinggi mencapai 92.92%. Kedua tabel tersebut menunjukkan bahwa tingkat akurasi SVM *one-against-all* dan SVM *one-against-one* adalah sama.

TABEL XIV
TINGKAT AKURASI SVM ONE-AGAINST-ALL DENGAN C=1000

K	Polynomial			RBF		
	d=1	d=2	d=3	$\gamma=1$	$\gamma=2$	$\gamma=3$
3	92.92	92.92	92.08	92.92	87.5	87.5
5	84.17	84.58	84.58	84.58	84.58	83.33
10	90.83	90.83	90.83	90.83	90.83	89.58

TABEL XV
TINGKAT AKURASI SVM ONE-AGAINST-ONE DENGAN C=1000

K	Polynomial			RBF		
	d=1	d=2	d=3	$\gamma=1$	$\gamma=2$	$\gamma=3$
3	92.92	92.92	92.08	92.92	87.5	87.5
5	84.17	84.58	84.58	84.58	84.58	83.33
10	90.83	90.83	90.83	90.83	90.83	89.58

Berdasarkan tabel XIV dan XV, nilai akurasi tertinggi pada C=1000 terdapat pada pengelompokan SVM yang menggunakan jumlah K=3, dengan fungsi kernel *polynomial* yang menggunakan nilai d=1, d=2 dan kernel RBF yang menggunakan $\gamma=1$. Tingkat akurasi pada pengelompokan ini mencapai 92.92%. Kedua tabel tersebut menunjukkan bahwa tingkat akurasi SVM *one-against-all* dan SVM *one-against-one* adalah sama.

IV. KESIMPULAN

Langkah awal dari implementasi SVM adalah mengumpulkan data dari DPU Bina Marga Jatim. Langkah selanjutnya adalah pembagian data menjadi dua bagian data, yaitu data *training* untuk pelatihan dan data *testing* untuk pengujian yang dilakukan dengan menggunakan teknik *k-folds cross validation*. Kemudian melakukan perhitungan *kernel*, matriks *hessian*, dan *training* SVM. Setelah *training* selesai lalu dilanjutkan dengan proses *testing* yang menghasilkan klasifikasi *class* dari data *testing*. Hasil akhir dari sistem berupa akurasi yang diperoleh dari perbandingan hasil klasifikasi pada *class* di data sebenarnya dengan *class* hasil prediksi.

Berdasarkan hasil perhitungan akurasi yang telah dilakukan sebelumnya, dapat disimpulkan bahwa *kernel polynomial* menghasilkan akurasi tertinggi yang sama dibandingkan dengan *kernel radial basis function* yaitu sebesar 92.92%, namun *kernel polynomial* menghasilkan akurasi tertinggi lebih banyak jika dibandingkan dengan *kernel radial basis function*. *Predictive maintenance* untuk kendaraan bermotor ini menghasilkan nilai akurasi yang mencapai 92.92% dengan menggunakan algoritma *support vector machine*, baik dengan yang menggunakan *one-against-all* maupun dengan yang menggunakan *one-against-one*.

UCAPAN TERIMA KASIH

Puji syukur terimakasih Alhamdulillah penulis panjatkan kepada Allah SWT yang memberikan rahmat dan hidayah sehingga penulis dapat melakukan penelitian dan penulisan artikel ilmiah. Penulis juga mengucapkan terimakasih kepada semua pihak yang telah mendukung, membantu, dan membimbing penulis dalam jalannya penelitian ini.

REFERENSI

- [1] F. Kurniawan, Manajemen Perawatan Industri: Teknik dan Aplikasi Implementasi Total Productive Maintenance (TPM), Preventive Maintenance dan Reliability Centered (RCM), Yogyakarta: Graha Ilmu, 2013.
- [2] R. Goriveau, K. Medjaher and N. Zerhouni, From Prognostics and Health Systems Management to Predictive Maintenance 1: Monitoring and Prognostics, 2016.
- [3] A. Darmawan, N. Kustian and W. Rahayu, "Implementasi Data Mining Menggunakan Model SVM untuk Prediksi Kepuasan Pengunjung Taman Tabebuaya," *Jurnal String Vol. 2 No. 3*, pp. 299-307, 2018.
- [4] V. Apriana and R. I. Handayani, "Analisis Algoritma Prediksi untuk Menghasilkan Prediksi Beban Listrik Jangka Pendek," *Jurnal Ilmu Pengetahuan dan Teknologi Komputer*, pp. 73-78, 2017.
- [5] A. Pratama, R. C. Wihandika and D. E. Ratnawati, "Implementasi Algoritme Support Vector Machine (SVM) untuk Prediksi Ketepatan Waktu Kelulusan Mahasiswa," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, pp. 1704-1708, 2018.
- [6] P. Jain, P. Patel, S. Bhambure, Y. Sen and D. N. Shaikh, "Predictive Maintenance Approach for Automobiles," *International Journal of Recent Scientific Research*, pp. 32777-32782, 2019.
- [7] T. P. Carvalho, F. A. A. M. N. Soares, R. Vita and R. d. P. Francisco, "A Systematic Literature Review of Machine Learning Methods Applied to Predictive Maintenance," *Computers & Industrial Engineering* 137, pp. 1-10, 2019.
- [8] A. Chaudhuri, "Predictive Maintenance for Industrial IoT of Vehicle Fleets using Hierarchical Modified Fuzzy Support Vector Machine," 2018.

- [9] P. Refaeilzadeh, L. Tang and H. Liu, Cross-Validation. Encyclopedia of Database Systems, 2009.
- [10] M. Bramer, Principles of Data Mining, London: Springer, 2007.
- [11] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, no. 2, pp. 955-947, 1998.