Predictive Maintenance on Paving Machine Using Support Vector Machine (SVM)
[*Predictive Maintenance* Pada Mesin Paving Menggunakan *Support Vector Machine* (SVM)]

Aidah Fidayanti[1)], Tedjo Sukmono [*,2)]

[1)]Program Studi Teknik Industri, Universitas Muhammadiyah Sidoarjo, Indonesia
[2)] Program Studi Teknik Industri, Universitas Muhammadiyah Sidoarjo, Indonesia
*Email Penulis Korespondensi: thedjoss@umsida.ac.id[2)]

Abstract. *Predictive maintenance is a strategy that uses machine operational data to predict potential damage before it occurs, thereby reducing downtime and maintaining the availability of production equipment. In 2024, PT XYZ experienced 342.75 hours of downtime with 82.30% availability and an output of 3,950,661 units, requiring more structured maintenance planning to minimize disruptions. This study aims to utilize predictions of the downtime duration of a paving machine's hydraulic system to develop a more efficient preventive maintenance schedule. The methods used were linear Support Vector Machine (SVM) and Multinomial Naive Bayes to predict downtime duration, with SVM achieving an accuracy of 89,1% with an MSE of 0,109 and Multinomial Naive Bayes achieving 85,5% with 0,145.*

Keywords – *Predictive maintenance, Support Vector Machine (SVM), Naive Bayes, Paving machine*

Abstrak. *Pemeliharaan prediktif adalah strategi yang menggunakan data operasional mesin untuk memperkirakan potensi kerusakan sebelum terjadi, sehingga dapat mengurangi downtime dan menjaga ketersediaan peralatan produksi. Pada tahun 2024, PT XYZ mengalami downtime 342,75 jam dengan ketersediaan 82,30% dan output 3.950.661 unit, sehingga dibutuhkan perencanaan pemeliharaan yang lebih terstruktur untuk meminimalkan gangguan. Penelitian ini bertujuan memprediksi durasi downtime sistem hidrolik mesin paving untuk menyusun jadwal perawatan preventif yang lebih efisien. Metode yang digunakan adalah Support Vector Machine (SVM) linear dan Multinomial Naive Bayes untuk memprediksi durasi downtime, dengan hasil akurasi SVM mencapai 89,1% dengan MSE 0,109 dan Multinomial Naive Bayes 85,5% dengan 0,145.*

Kata Kunci – *Predictive maintenance, Support Vector Machine (SVM), Naive Bayes, Mesin paving*

I. PENDAHULUAN

PT XYZ merupakan perusahaan yang bergerak di industri konstruksi beton, khususnya dalam produksi paving berbahan beton *masonry*. Peningkatan kapasitas produksi mesin paving untuk memenuhi permintaan pasar tidak didukung oleh keandalan mesin yang menyebabkan berbagai masalah operasional. Salah satu cabang perusahaan ini hanya memiliki satu *batching plant* yang sering mengalami *downtime* akibat peningkatan permintaan, kurangnya keterampilan teknisi, terbatasnya suku cadang, dan proses pengadaan yang lambat karena harus mendapatkan persetujuan dari kantor pusat. Selain itu, belum adanya sistem penjadwalan pemeliharaan yang terstruktur dan berbasis data histrois semakin memperburuk keadaan, sehingga perawatan yang dilakukan cenderung bersifat reaktif. Penjadwalan pemeliharaan yang efektif sangat penting untuk mencegah kerusakan mesin, meningkatkan waktu operasional, dan efisiensi produksi [1].

Permasalahan yang dihadapi perusahaan ini adalah kerusakan sistem hidrolik pada mesin *wet mixing* paving yang menyebabkan *downtime* dan gangguan dalam proses produksi. Jumlah jam kerja yang tersedia adalah 8 jam per hari, sedangkan *downtime* tertinggi pada tahun 2024 mencapai 342,75 jam sehingga *availability* mencapai 82,30% dan menghasilkan produksi sebanyak 3.950.661 *unit*. Oleh karena itu, diperlukan pendekatan pemeliharaan prediktif untuk mendeteksi kerusakan sejak dini dan memperkirakan waktu kegagalan mesin untuk menjaga kelancaran operasional [2]. Metode *Support Vector Machine* (SVM) dipilih sebagai solusi untuk membangun model pemeliharaan prediktif yang dapat memprediksi kerusakan pada sistem hidrolik sehingga dapat mengurangi *downtime* dan meningkatkan efisiensi produksi mesin *wet mixing* paving [3].

Support Vector Machine (SVM) adalah algoritma dalam pembelajaran terawasi (*supervised learning*) yang dikenal memiliki kinerja unggul dibandingkan dengan algoritma lainnya sehingga sering digunakan di berbagai penelitian [4]. Penerapan SVM dalam industri bertujuan mengurangi waktu henti, meningkatkan efisiensi biaya operasional, dan mengoptimalkan jadwal perawatan, sehingga menjadikannya pilihan andal untuk sistem pemeliharaan berbasis data

[5]. Pada penelitian ini menerapkan SVM dengan *hyperplane linear* untuk secara efisien memisahkan dua kelas data dalam ruang fitur berdimensi tinggi [6].

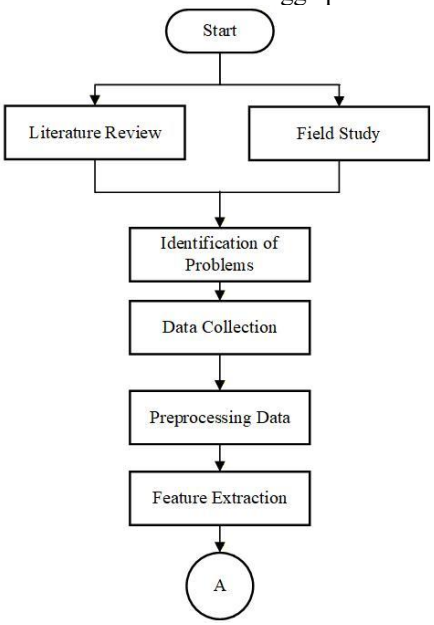
SVM efektif dalam mengolah data berdimensi tinggi dan menghasilkan hasil yang akurat. Di sisi lain, *Naive Bayes* lebih sederhana dan cepat dalam proses klasifikasi [7]. *Naive Bayes* adalah algoritma dalam *data mining* dan *machine learning* yang berlandaskan pada teorema *Bayes* [8]. Penelitian ini menggunakan *Multinomial Naive Bayes* sebagai model perbandingan karena algoritma tersebut efektif dalam klasifikasi teks berdasarkan frekuensi kata. Algoritma ini mengasumsikan independensi antar fitur dan menghitung probabilitas kata pada setiap kelas sentimen sesuai dengan pendekatan TF-IDF [9].

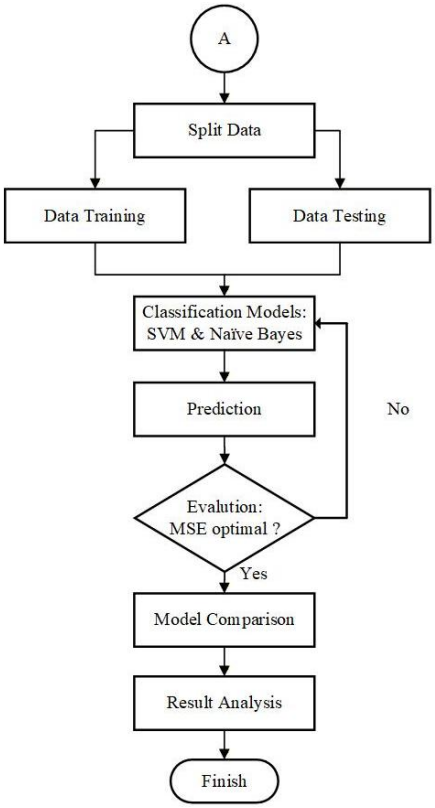
Penelitian sebelumnya telah mengembangkan sistem pemeliharaan prediktif untuk *batching plant* dengan menggunakan *Support Vector Machine* (SVM) untuk memprediksi waktu kerusakan mesin [10]. Penelitian lain menerapkan pemeliharaan prediktif pada lini mesin produksi *Dry* 8 dengan SVM yang menggunakan kernel *linear* untuk meramalkan kegagalan mesin [11]. Perbandingan antara SVM dan *Naive Bayes* dalam pemeliharaan prediktif mesin industri menggunakan data sensor menunjukkan validitas pendekatan klasifikasi *downtime* [12]. Temuan penelitian mengindikasikan bahwa *Naive Bayes* lebih cepat untuk pemantauan *real-time,* sedangkan SVM lebih akurat dalam klasifikasi *downtime* [13]. Selain itu, evaluasi dan perbandingan SVM dengan *Naive Bayes* sebagai model pembelajaran mesin digunakan untuk memprediksi dan mengurangi *downtime* mesin melalui pemeliharaan prediktif pada *dataset* sensor industri [14].

Penelitian ini berbeda dari lima penelitian sebelumnya karena menggunakan jenis *dataset* dan peralatan manufaktur yang berbeda. Dalam penelitian ini, data durasi *downtime* pada sistem hidrolik mesin paving diklasifikasikan secara *multiclass* bukan *binary*. Tujuan penelitian ini adalah untuk memprediksi kategori durasi *downtime* sistem hidrolik mesin paving sebagai dasar dalam menyusun jadwal perawatan preventif yang lebih terstruktur, sehingga dapat mengendalikan *downtime* dan meminimalkan penurunan *output* produksi.

II. METODE

Penelitian ini dilaksanakan di PT XYZ di Sumbersuko, Kecamatan Gempol, Kabupaten Pasuruan, Jawa Timur selama enam bulan dari September 2025 hingga Februari 2026. Metode yang diterapkan terdiri dari observasi, wawancara, dan pengumpulan data historis mengenai pemeliharaan dan produksi mesin paving. Data tersebut dianalisis menggunakan metode *Support Vector Machine* (SVM) untuk mengklasifikasikan durasi *downtime* mesin berdasarkan aktivitas pemeliharaan dan kondisi produksi. Selain itu, *Naive Bayes* digunakan sebagai model perbandingan dalam analisis *downtime* yang berbasis teks aktivitas pemeliharaan. Kedua metode ini dipilih untuk memberikan pemahaman yang lebih baik mengenai pola *downtime* mesin paving sehingga dapat dirumuskan rekomendasi pemeliharaan yang lebih akurat. Proses penelitian dijelaskan secara sistematis dalam diagram alir yang ditampilkan dalam gambar 1, mulai dari identifikasi masalah hingga penarikan kesimpulan.





Gambar 1. Diagram Alir Penelitian

Pada gambar 1 menunjukkan diagram alir penelitian yang dijelaskan secara rinci sebagai berikut:

1. *Field Study and Literature Review*

Field study atau studi lapangan dilakukan melalui observasi langsung pada saat pelaksanaan kegiatan produksi dan pemeliharaan mesin paving di salah satu cabang PT XYZ, yaitu di Pandaan. Observasi dilakukan dengan mengamati proses produksi paving, perawatan mesin, kondisi terbatasnya stok *spare part* di gudang, dan pencatatan data *downtime* dalam sistem *Enterprise Resource Planning* (ERP). *Literature review* atau studi literatur dilakukan dengan menganalisis metode yang akan diterapkan berdasarkan artikel, jurnal, buku, dan karya ilmiah lainnya yang relevan dengan topik penelitian yang sedang dibahas.

2. *Identification of Problems*

Identifications of problems atau identifikasi masalah dilakukan setelah studi literatur dan lapangan, diikuti dengan perumusan masalah berdasarkan hasil observasi.

3. *Data Collection*

Data collection atau pengumpulan data pada penelitian ini menggunakan data primer dan data sekunder. Data primer diperoleh dari wawancara kepala produksi, kepala divisi pemeliharaan, dan seorang manajer untuk mengidentifikasi penyebab *downtime* pada mesin *wet mixing* paving dan informasi untuk pengelolaan pemeliharaan. Data sekunder diambil dari *database* perusahaan yang terdiri dari data historis *downtime*, catatan aktivitas pemeliharaan sistem hidrolik, dan data produksi paving dari Oktober 2022 hingga Oktober 2025. Data historis selama tiga tahun mengenai *downtime* sistem hidrolik dan produksi paving mendukung keberhasilan dalam pemodelan prediktif serta klasifikasi durasi *downtime* [15]. Data yang dikumpulkan mencakup 550 informasi mengenai *downtime* dan terdiri dari berbagai atribut yang dijelaskan dalam tabel 1.

Tabel 1. *Atribute Dataset*

No	*Atribute*	*Description*	*Type Data*
1	Tanggal	Penanda tanggal pencatatan aktivitas pemeliharaan dan produksi	*Object*
2	Jenis	Jenis aktivitas pemeliharaan seperti preventif atau kuratif	*Object*
3	Aktivitas	Detail aktivitas pemeliharaan seperti servis, perbaikan, atau cek rutin	*Object*
4	Komponen	Komponen mesin yang mengalami perawatan atau perbaikan	*Object*

5	Durasi *Downtime* (Jam)	Lama waktu mesin berhenti beroperasi akibat pemeliharaan	*Numeric*
6	Kategori *Downtime*	Kategori lama *downtime* seperti singkat, sedang, lama	*Object*
7	Total Produksi	Jumlah hasil produksi paving per hari	*Numeric*
8	Jenis Kode	Kode numerik hasil pengubahan atribut jenis ke bentuk kategori numerik	*Numeric*
9	Rasio *Downtime* Produksi	Perbandingan durasi *downtime* terhadap total produksi pada hari yang sama	*Numeric*

4. *Preprocessing Data*

Preprocessing data adalah tahap persiapan data sebelum analisis yang mencakup pembersihan, transformasi, dan pengintegrasian data [16]. Dalam penelitian ini, tahap *preprocessing data* terdiri dari empat langkah utama, yaitu *data cleaning* (pembersihan data), *text processing* (pengolahan teks), *labeling* (pelabelan 3 kelas), dan *feature engineering* (rekayasa fitur).

5. *Feature Extraction*

Feature extraction adalah proses konversi kata menjadi angka dalam bentuk vektor dilakukan karena komputer hanya dapat mengenali dan memproses angka, selama angka tersebut masih memiliki makna yang sesuai dengan kata tersebut [17]. Pada penelitian ini, *feature extraction* terdiri dari dua langkah, yaitu TF-IDF (*text*) dan *numerical features*.

6. *Split Data*

Split data adalah teknik untuk memisahkan *dataset* dan memengaruhi performa model klasifikasi dalam algoritma pembelajaran mesin [18]. Dalam penelitian ini menggunakan metode pembagian data tipe *stratified split* sehingga menghasilkan data *training* dan data *testing*.

7. *Classification Models*

Model klasifikasi ini mencakup SVM dengan kernel linier, serta data teks dan angka. Sementara itu, *Naive Bayes* terdiri dari model *multinomial* dengan parameter *default* dan hanya menggunakan data teks.

8. *Prediction*

Pada tahap prediksi ini akan mengklasifikasikan kategori *downtime* sistem hidrolik ke dalam tiga kategori, yaitu singkat (< 1 jam), sedang (1-3 jam), dan lama (> 3 jam) dengan menggunakan model SVM dan *Naive Bayes* pada data uji.

9. *Evalution*

Pada tahap evaluasi, model dinilai dengan *confusion matrix* dan MSE. *Confusion matrix* adalah matriks untuk menunjukkan hasil klasifikasi dan mengevaluasi kinerja model klasifikasi dengan menggunakan metrik seperti akurasi, presisi, *recall,* dan F1-*score* [19]. Adapun rumus *confusion matrix* yang digunakan dalam penelitian ini adalah sebagai berikut.

$$Accuracy = \frac{\text{TPP} + \text{TNtNt} + \text{TNN}}{\text{Total data yang diprediksi}} \quad (1)$$

$$Precision = \frac{\text{TPP}}{\text{TPP} + \text{NtFP} + \text{NFP}} \quad (2)$$

$$Recall = \frac{\text{TPP}}{\text{TPP} + \text{PFNt} + \text{PFN}} \quad (3)$$

$$\text{F1-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

Sumber: [20]

Keterangan:

TPP (*True Positive Positive)* = Data positif yang diprediksi benar sebagai positif
TNtNt (*True Neutral Neutral*) = Data netral yang diprediksi benar sebagai netral
NtFP (*Neutral False Positive*) = Data netral yang terklasifikasi sebagai positif
NFP (*Negative False Positive*) = Data negatif yang terklasifikasi sebagai positif
PFNt (*Postive False Neutral*) = Data positif yang terklasifikasi sebagai netral
PFN (*Positive False Negative*) = Data positif yang terklasifikasi sebagai negatif

Langkah berikutnya adalah menghitung nilai *Mean Squared Error* (MSE) untuk mengukur tingkat kesalahan prediksi model. MSE merupakan metrik yang menghitung rata-rata dari selisih kuadrat antara nilai prediksi dan nilai aktual [21]. MSE dipilih sebagai metrik utama karena nilai yang lebih rendah merepresentasikan kinerja yang lebih baik. Sebuah nilai MSE sebesar 0 menunjukkan model yang sempurna, sedangkan nilai terkecil dari MSE menggambarkan kinerja terbaik, semakin mendekati nol, semakin baik kinerja model tersebut [22][23]. Rumus MSE adalah sebagai berikut.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2 \quad \text{.....(5)}$$

Sumber: [24]

Keterangan:
n = Jumlah keseluruhan data
y_i = Nilai aktual pada data ke-i
$\tilde{y}_i$ = Nilai hasil prediksi pada data ke-i

10. *Model Comparison*

Model perbandingan ini menganalisis kinerja *LinearSVC* pada SVM dan *Multinomial Naive Bayes* dalam klasifikasi *downtime*. Analisis dilakukan dengan menggunakan metrik *accuracy, precision, recall,* F1-*score,* dan MSE, serta menyajikan visualisasi grafik batang untuk membandingkan kinerja kedua model.

11. *Result Analysis*

Pada tahap terakhir ini dilakukan penarikan kesimpulan berdasarkan hasil pengolahan data yang telah dilakukan, yaitu model SVM dan *Naive Bayes* dengan akurasi tinggi terutama dalam mengenali kategori *downtime* pada data.

A. Metode *Support Vector Machine* (SVM)

Support Vector Machine (SVM) adalah metode klasifikasi dalam *machine learning* (*supervised learning*) yang memprediksi kelas berdasarkan pola yang dihasilkan dari proses pelatihan [25]. Metode ini digunakan untuk menentukan *hyperplane* optimal yang memisahkan berbagai kelas dalam ruang vektor [26]. SVM memiliki dua jenis *hyperplane linear* dan *non-linear*. Jika data dapat dipisahkan sepenuhnya oleh *hyperplane linear*, maka disebut SVM *linear*. Sebaliknya, jika data tidak dapat dipisahkan secara *linear*, SVM akan menerapkan teknik transformasi kernel untuk memindahkan data ke dimensi fitur yang lebih tinggi agar *hyperplane linear* dapat terbentuk [27]. Rumus *Support Vector Machine* (SVM) yang dapat dipisahkan menggunakan fungsi linier adalah sebagai berikut.

$$f(x) = w^T + b \quad \text{.....(6)}$$
$$g(x) := \text{sgn}(f(x)) \quad \text{.....(7)}$$

Sumber: [28]

Keterangan:
w = Vektor bobot (arah *hyperplane*)
x = Vektor fitur (data *input*)
b = Bias (menggeser posisi *hyperplane*)

B. Metode *Naive Bayes*

Naive Bayes merupakan metode klasifikasi probabilitas yang sederhana dan mengandalkan asumsi independensi tinggi, serta memanfaatkan teorema *Bayes* [29]. *Naive Bayes* terdiri dari beberapa jenis, antara lain *gaussian, multinomial,* dan *bernoulli bayes*. Pada penelitian ini menggunakan metode *Multinomial Naive Bayes,* yaitu pendekatan yang menghitung frekuensi istilah dalam dokumen [30]. Berikut adalah rumus untuk metode *Multinomial Naive Bayes*.

Klasifikasi diawali dengan menghitung probabilitas prior untuk setiap kelas C_k menggunakan estimator maksimum *likelihood*.

$$p(C_k) = \pi k = \frac{N_k}{N} \quad \text{.....(8)}$$

Sumber: [31]

Keterangan:
C_k = Kelas ke-k
$p(C_k)$ = Probabilitas prior kelas C_k
πk = Parameter prior untuk kelas C_k
N_k = Jumlah data yang termasuk kelas C_k
N = Total jumlah data

Pada model *Naive Bayes* digunakan asumsi *conditional independence* antar fitur terhadap kelas C_k, sehingga probabilitas bersama fitur x_1 dan x_B dinyatakan sebagai berikut:

$$p(x_1, x_B | C_k) = p(x_1 | C_k)\, p(x_B | C_k) \quad \text{.....(9)}$$

Sumber: [31]

Keterangan:
x_1 = Fitur pertama
x_B = Fitur kedua
$p(x_1, x_B | C_k)$ = Probabilitas bersama fitur x_1 dan x_B jika diketahui kelasnya C_k
$p(x_1 | C_k)$ = Probabilitas kemunculan fitur x_1 pada data yang termasuk kelas C_k
$p(x_B | C_k)$ = Probabilitas kemunculan fitur x_B pada data yang termasuk kelas C_k

Berdasarkan asumsi *conditional independence* pada model *Naive Bayes,* probabilitas posterior kelas C_k setelah mengamati vektor fitur x dapat dinyatakan sebanding dengan hasil kali antara probabilitas bersama fitur dan prior kelas yang dinyatakan sebagai berikut:

$$p(C_k \mid x_1, x_B) \propto p(x_1, x_B \mid C_k)\, p(C_k) \quad \text{.....(10)}$$

Sumber: [31]

Keterangan:

$p(C_k \mid x_1, x_B)$ = Probabilitas posterior, yaitu peluang data termasuk kelas C_k setelah melihat fitur x_1 dan x_B

$p(x_1, x_B \mid C_k)$ = Probabilitas bersama fitur x_1 dan x_B jika diketahui kelas C_k

$p(C_k)$ = Probabilitas prior kelas C_k

Pada klasifikasi generatif, skor untuk setiap kelas C_k dapat dinyatakan sebagai log dari hasil kali antara *likelihood* dan prior kelas yang dirumuskan sebagai berikut.

$$a_k = \ln p(x \mid C_k) p(C_k) \quad \text{.....(11)}$$

Sumber: [31]

Keterangan:

$p(x \mid C_k)$ = Probabilitas fitur x jika berasal dari kelas C_k

$p(C_k)$ = Probabilitas awal kelas C_k

III. HASIL DAN PEMBAHASAN

A. *Data Collection*

Data yang digunakan adalah hasil penggabungan dari dua *file*, yaitu data sistem hidrolik dan data produksi yang diimpor ke *Phyton. Dataset* sistem hidrolik memiliki 550 data dengan 6 atribut data, sedangkan *dataset* produksi memiliki 864 data dengan 2 atribut data. Tabel 2 menjelaskan fitur data yang digunakan dalam penelitian ini.

Tabel 2. Fitur *Data Collection*

No	Tanggal	Jenis	Aktivitas	Keterangan	Durasi *Downtime* (Jam)
1	2022-10-01	Kuratif	Perbaikan	Penggantian seal hidrolis gate	4
2	2022-10-10	Kuratif	Perbaikan	Perbaikan aktuator hidrolis	1
3	2022-10-11	Preventif	Cek Rutin	chain kopling hydrolis pump lepas (ganti ex chain kopling mg007)	3
4	2022-10-12	Kuratif	Perbaikan	Hidrolis tumbuk rusak	2
…					
547	2025-10-02	Kuratif	Perbaikan	Ganti baut boom hidrolis moulding	1
548	2025-10-09	Kuratif	Perbaikan	Ganti aktuator hidrolis ex spare	2
549	2025-10-16	Kuratif	Perbaikan	Repair aktuator hidrolis ex mg007	2
550	2025-10-28	Kuratif	Perbaikan	Perbaikan mesin hidrolis	3

Pada tabel 2 fitur *data collection*, data mengenai waktu henti sistem hidrolik dan produksi harian telah tercatat secara menyeluruh melalui variabel tanggal, jenis perawatan, aktivitas, keterangan, dan durasi *downtime.* Struktur data tersebut menjadi landasan untuk analisis pemodelan kategori durasi *downtime* sistem hidrolik.

B. *Preprocessing Data*

Pada penelitian ini, *preprocessing data* dilakukan melalui empat tahapan, yaitu *data cleaning, text processing, labeling,* dan *feature engineering*. Tahap *data cleaning* meliputi konversi format tanggal, penggabungan *dataset*, normalisasi durasi *downtime*, dan penanganan *missing values*. Selanjutnya, fitur teks dibentuk dengan menggabungkan kolom ‘aktivitas’ dan ‘keterangan’ untuk ekstraksi fitur TF-IDF. Label kelas dibagi menjadi tiga kategori durasi *downtime*, yaitu singkat (<1 jam), sedang (1-3 jam), dan lama (>3 jam). Fitur numerik tambahan dihasilkan dari *encoding* jenis pemeliharaan dan perhitungan rasio *downtime* terhadap produksi. Berikut adalah tabel hasil *feature engineering*.

Tabel 3. Hasil *Feature Engineering*

No	Tanggal	Jenis	Durasi (Jam)	Total Produksi	Jenis Kode	Rasio *Downtime*	Kategori *Downtime*
0	2022-10-01	Kuratif	4	2150	0	0,001860	Lama
1	2022-10-10	Kuratif	1	2666	0	0,000375	Sedang
2	2022-10-11	Preventif	3	1376	1	0,002180	Sedang

3	2022-10-12	Kuratif	2	2580	0	0,000775	Sedang

Pada tabel 3 hasil *feature engineering*, data *downtime* telah diproses dengan menyesuaikan format tanggal, mengonversi durasi ke dalam angka, menambahkan variabel teks gabungan, serta mengkategorikan durasi menjadi 'Singkat, Sedang, dan Lama'. Selanjutnya, rasio *downtime* terhadap total produksi telah dihitung agar data siap digunakan sebagai *input* pemodelan.

C. *Feature Extraction*

Fitur teks diubah menjadi representasi numerik menggunakan TF-IDF, kemudian digabungkan dengan fitur numerik untuk model SVM. Gambar 2 menunjukkan *feature extraction*.

```
X_text_train_tfidf = tfidf.fit_transform(X_text_train)
X_text_test_tfidf  = tfidf.transform(X_text_test)
```

Gambar 2. *Source Code Feature Extraction* TF-IDF

Pada gambar 2, *source code feature extraction* TF-IDF berfungsi untuk mengonversi teks menjadi bentuk numerik melalui metode TF-IDF sehingga kalimat dapat diolah sebagai fitur oleh model *machine learing*. Hasilnya adalah matriks nilai TF-IDF yang merepresentasikan tingkat kepentingan masing-masing kata dalam data latih dan data uji.

D. *Split Data*

Split data adalah proses pembagian antara data latih dan data uji. Data dibagi menjadi data latih 90% dan data uji 10% dengan *stratified sampling* untuk menjaga proporsi kelas. Sebelum pembagian, fitur (x) dan label (y) dipisahkan. Fitur terdiri dari dua tipe, yaitu fitur tekstual (x_*text*) yang mencakup kolom teks, serta fitur numerik (x_num) yang meliputi total produksi, jenis kode, dan rasio *downtime* produksi. Label (y) adalah kategori *downtime* yang menjadi target prediksi. Parameter yang digunakan dalam proses pembagian data dapat dilihat pada tabel 4.

Tabel 4. Data Parameter

Parameter	Nilai
test_size	0,1
random_state	42
Stratify	y (Kategori *Downtime*)
Total data	550 baris
Data training	495 bari
Data testing	55 baris
Kategori *Downtime*	Singkat, Sedang, Lama

Berdasarkan tabel 4 data parameter, pembagian data dilakukan dengan *test_size* = 0,1 yang menghasilkan 10% data sebagai data uji dan 90% sebagai data latih. Penggunaan *random_state* = 42 bertujuan untuk memastikan konsistensi pembagian saat kode dijalankan kembali, sedangkan *stratify* = y menjaga proporsi setiap kategori durasi *downtime* pada data latih dan uji agar seimbang dengan data awal.

E. Algoritma *Support Vector Machine* (SVM)

Setelah tahap *split data* selesai dilakukan, tahap berikutnya adalah melakukan pemodelan menggunakan algoritma SVM. Model SVM dioptimalkan menggunakan *GridSearchCV* dengan 5-*fold cross-validation* untuk mencari *hyperparameter* terbaik. Gambar 3 menunjukkan pemodelan SVM.

```
base_svm = LinearSVC(dual=False, random_state=42)

grid_svm = GridSearchCV(
    base_svm,
    param_grid,
    cv=5,
    scoring='accuracy',
    n_jobs=-1,
    verbose=1
```

Gambar 3. *Source Code* Pemodelan SVM

Pada gambar 3 *source code* Pemodelan SVM menunjukkan kode *GridSearchCV* yang berfungsi untuk secara otomatis mencari kombinasi parameter terbaik dalam model SVM. Proses ini menguji berbagai nilai dalam *param_grid* melalui validasi silang 5-*fold* dan memilih konfigurasi yang memberikan akurasi tertinggi.

F. Algoritma *Naive Bayes*

Selanjutnya dilakukan pemodelan menggunakan algoritma *Naive Bayes*. Model *Naive Bayes* menggunakan parameter *default* dan hanya memanfaatkan fitur TF-IDF. Gambar 4 menunjukkan pemodelan *Naive Bayes*.

```
nb_model = MultinomialNB()
nb_model.fit(X_train_nb, y_train)
```

Gambar 4. *Source Code* Pemodelan *Naive Bayes*

Pada gambar 4 *source code* pemodelan *Naive Bayes* menunjukkan bahwa model tersebebut dibentuk dan dilatih dengan data latih yang telah diubah menjadi fitur numerik sehingga algoritma mempelajari pola keterkaitan antara teks deksripsi *downtime* dan kategori durasi *downtime* yang selanjutnya digunakan untuk memprediksi data uji.

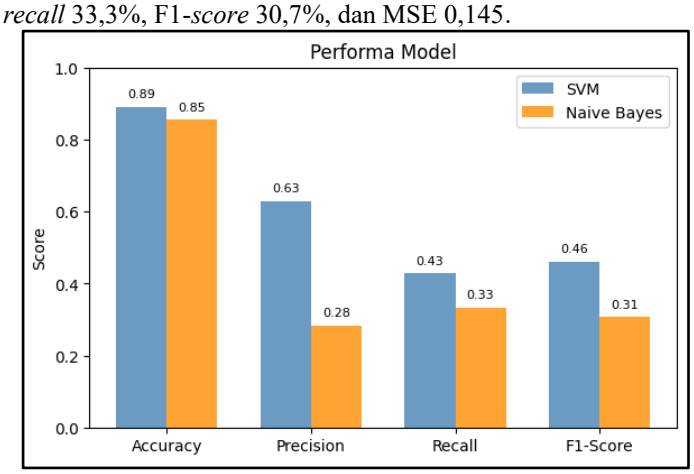
G. *Evaluation*

Evaluasi model dilakukan untuk mengukur kinerja klasifikasi algoritma SVM dan *Naive Bayes* pada data uji. Metrik yang digunakan dalam evaluasi meliputi *accuracy, precision, recall,* F1-*Score,* dan *Mean Square Error* (MSE). Hasil evaluasi kedua algoritma dapat dilihat pada tabel 5.

Tabel 5. Hasil Evaluasi SVM dan *Multinomial Naive Bayes*

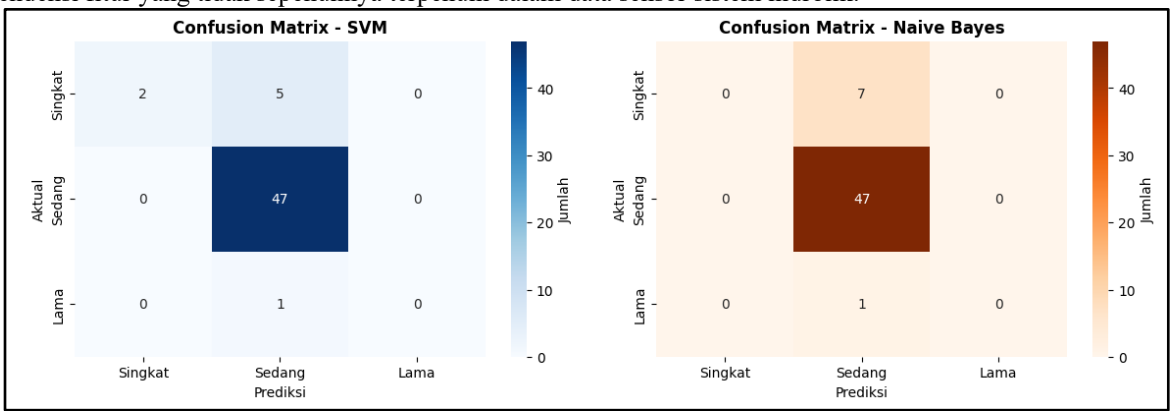
Model Algoritma	*Accuracy*	*Precision*	*Recall*	F1-*Score*	MSE
Support Vector Machine (SVM)	0,891	0,629	0,429	0,461	0,109
Naïve Bayes	0,855	0,285	0,333	0,307	0,145

Pada tabel 5 menunjukkan model SVM menghasilkan evaluasi dengan *accuracy* sebesar 89,1% yang berarti dapat memprediksi dengan benar 49 dari 55 data uji. *Precision* mencapai 62,9% mengindikasikan sekitar 63% dari semua prediksi model untuk mendeteksi 43% dari semua kelas, khususnya kelas minoritas 'Singkat' dan 'Lama'. F1-*score* yang mencapai 46,1% menunjukkan keseimbangan antara *precision* dan *recall. Mean Squared Error* (MSE) sebesar 0,109 menunjukkan rata-rata kesalahan prediksi yang rendah dengan label 'Singkat' diubah menjadi 0, 'Sedang' menjadi 1, dan 'Lama' menjadi 2. Sebaliknya, model *Naive Bayes* memiliki performa lebih rendah dengan akurasi 85,5%, *precision* 28,5%, *recall* 33,3%, F1-*score* 30,7%, dan MSE 0,145.



Gambar 5. Grafik Perbandingan Model SVM dan *Naive Bayes*

Pada gambar 5 grafik perbandingan performa menunjukkan bahwa model SVM yang ditunjukkan dengan diagram batang biru lebih unggul dibandingkan model *Naive Bayes* yang ditandai dengan diagram batang oranye pada semua metrik evaluasi. Keunggulan SVM terletak pada kemampuannya mengelola data dengan fitur kompleks dan memisahkan kelas-kelas yang mirip melalui *hyperplane* optimal. Sebaliknya, *Naive Bayes* terbatas oleh asumsi independensi fitur yang tidak sepenuhnya terpenuhi dalam data sensor sistem hidrolik.



Gambar 6. *Confusion Matrix* Model SVM dan *Naive Bayes*

Selanjutnya, pada gambar 6 menunjukkan *confusion matrix* untuk model SVM dan *Naive Bayes* dalam mengklasifikasikan durasi *downtime* singkat, sedang, dan lama. Model SVM berhasil memprediksi seluruh 47 data

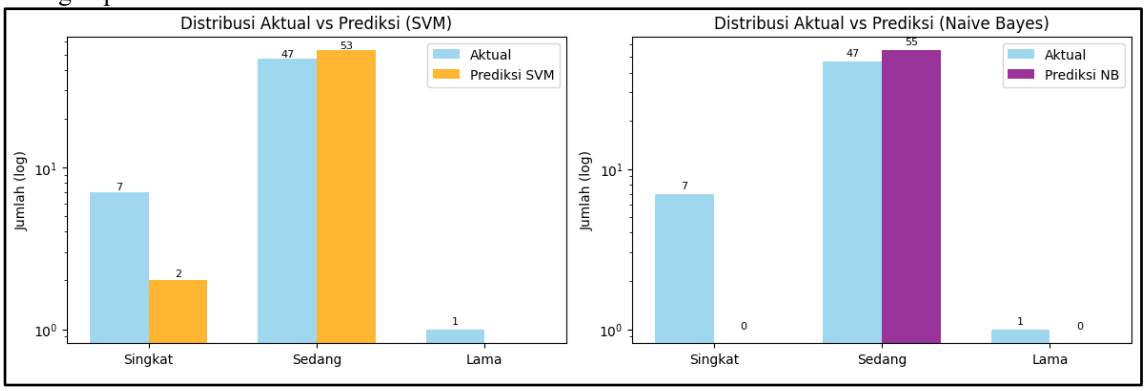
kategori ‘Sedang’ dengan tepat, tetapi hanya 2 dari 7 data kategori ‘Singkat’ yang terklasifikasikan dengan benar, sementara 5 lainnya salah diprediksi sebagai ‘Sedang’. Untuk kategori ‘Lama’, 1 data terklasifikasikan sebagai ‘Sedang’, sehingga tidak ada *downtime* ‘Lama’ yang diidentifikasi dengan baik. Model *Naive Bayes* juga memprediksi semua data kategori ‘Sedang’ dengan tepat, namun seluruh *downtime* ‘Singkat’ dan ‘Lama’ terklasifikasikan sebagai ‘Sedang’, sehingga pemisahan kategori di luar kelas ‘Sedang’ belum optimal. SVM dan *Naive Bayes* cenderung mengklasifikasikan sebagian besar durasi *downtime* ‘Sedang’. Akibatnya performa pada kelas minoritas ‘Singkat’ dan ‘Lama’ menjadi lebih lemah, meskipun akurasi keseluruhan terlihat tinggi. Hal ini mengindikasikan adanya masalah ketidakseimbangan data dan keterbatasan model dalam membedakan pola karakteristik *downtime* yang singkat atau lama.

H. *Model Comparison*

Hasil evaluasi menunjukkan bahwa algoritma SVM memiliki kinerja lebih baik daripada *Naive Bayes* dalam seluruh metrik yang dianalisis. Pada data uji, SVM mencapai akurasi 0,891 dengan *precision* 0,629, *recall* 0,429, dan F1-*score* 0,461. Sebaliknya, *Naive Bayes* memperoleh akurasi 0,855 dengan *precision* 0,285, *recall* 0,333 dan F1-*score* 0,307. Selain itu, MSE SVM sebesar 0,109 lebih rendah dari *Naive Bayes* yang mencapai 0,145, menunjukkan bahwa prediksi SVM lebih dekat dengan label sebenarnya. Secara keseluruhan, SVM menunjukkan performa yang lebih stabil dan akurat dalam klasifikasi teks berbasis TF-IDF [32].

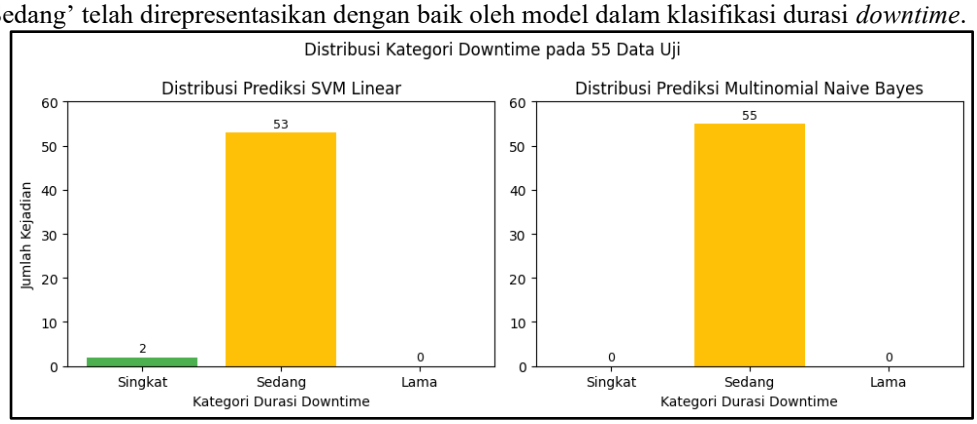
I. *Result Analysis*

Analisis hasil menunjukkan bahwa SVM memiliki akurasi dan MSE lebih baik daripada *Naive Bayes,* kedua model tersebut masih cenderung bias terhadap kelas mayoritas ‘Sedang’. Hal ini mengakibatkan kemampuan deteksi kelas minoritas ‘Singkat’ dan ‘Lama’ yang masih lemah. Berikut adalah gambar 12 yang menunjukkan distribusi aktual dengan prediksi.



Gambar 7. Distribusi Aktual dengan Prediksi

Pada gambar 7 terlihat grafik menunjukkan bahwa distribusi data aktual terdiri dari 7 data kategori ‘Singkat’, 47 data kategori ‘Sedang’, dan 1 data kategori ‘Lama’. Hasil prediksi menggunakan algoritma SVM menunjukkan perubahan menjadi 2 data kategori ‘Singkat’, 53 data kategori ‘Sedang’, dan tidak ada data kategori ‘Lama’. Sementara itu, prediksi dengan algoritma *Naive Bayes* mengelompokkan semua 55 data ke dalam kategori ‘Sedang’ tanpa prediksi untuk kategori ‘Singkat’ atau ‘Lama’. Pola ini menunjukkan bahwa kedua model cenderung memfokuskan prediksi pada kategori ‘Sedang’ dan belum mampu merepresentasikan kategori ‘Singkat’ dan ‘Lama’ dengan baik dalam pemeliharaan prediktif. Kedua model berhasil mengidentifikasi pola utama bahwa sebagian besar durasi *downtime* tergolong ‘Sedang’, baik dalam data aktual maupun hasil prediksi. Hal ini menunjukkan bahwa karakteristik *downtime* ‘Sedang’ telah direpresentasikan dengan baik oleh model dalam klasifikasi durasi *downtime*.



Gambar 8. Distribusi Kategori *Downtime* Pada 55 Data Uji

Pada gambar 8 terlihat grafik distribusi prediksi menunjukkan bahwa model SVM *linear* mengelompokkan 55 data uji dengan 53 kejadian dalam kategori durasi *downtime* 'Sedang' (1-3 jam) dan 2 kejadian dalam kategori 'Singkat' (<1 jam), sedangkan tidak ada prediksi untuk kategori 'Lama' (>3 jam). Di sisi lain, model *Multinomial Naive Bayes* mengklasifikasikan semua data dalam kategori 'Sedang' sebanyak 55 kejadian tanpa prediksi untuk kategori lainnya. Pola ini menunjukkan bahwa kedua model lebih fokus pada kejadian *downtime* berdurasi sedang sehingga hasil klasifikasi dapat dijadikan dasar awal dalam mengelopokkan kategori durasi *downtime* pada sistem hidrolik untuk penyusunan jadwal perawatan preventif.

J. Perhitungan Manual Metrik

Berikut adalah perhitungan manual untuk nilai *accuracy, precision,* dan F1-*score* yang didasarkan pada gambar 6 hasil *confusion matrix* model SVM dan *Naive Bayes* dengan pembagian data latih 90% dan data uji 10%. Langkah-langkahnya adalah sebagai berikut.

1. *Accuracy* SVM $= \frac{\text{TP} + \text{TNet} + \text{TNeg}}{\text{Total data yang diprediksi}} = \frac{\text{TP} + \text{TNet} + \text{TNeg}}{\text{Total data yang diprediksi}} = \frac{2+ 47 + 0}{55} = \frac{49}{55} = 0{,}891$

   Accuracy Naive Bayes $= \frac{\text{TP} + \text{TNet} + \text{TNeg}}{\text{Total data yang diprediksi}} = \frac{\text{TP} + \text{TNet} + \text{TNeg}}{\text{Total data yang diprediksi}} = \frac{0+ 47 + 0}{55} = \frac{49}{55} = 0{,}855$

Berdasarkan hasil perhitungan tersebut, diperoleh nilai akurasi untuk metode SVM sebesar 0,891 atau 89,1%. Sedangkan metode *Naive Bayes* menghasilkan nilai akurasi sebesar 0,855 atau 85,5%.

2. *Precision* $_{\text{Singkat}}$ SVM $= \frac{\text{TPP}}{\text{TPP} + \text{NtFP} + \text{NFP}} = \frac{2}{2 + 0 + 0} = \frac{2}{2} = 1{,}00$

   Precision $_{\text{Sedang}}$ SVM $= \frac{\text{TNtNt}}{\text{PFNt} + \text{TNtNt} + \text{NFNt}} = \frac{47}{5 + 47 + 1} = \frac{47}{53} = 0{,}887$

   Precision $_{\text{Lama}}$ SVM $= \frac{\text{TNN}}{\text{PFN} + \text{NtFN} + \text{TNN}} = \frac{0}{0 + 0 + 0} = \frac{0}{0} = 0$

   Macro Precision SVM $= \frac{\textit{Precision}_{\text{Singkat}} + \textit{Precision}_{\text{Sedang}} + \textit{Precision}_{\text{Lama}}}{\text{Jumlah Kelas}} = \frac{1{,}00 + 0{,}887 + 0}{3} = 0{,}629$

   Precision $_{\text{Singkat}}$ *Naive Bayes* $= \frac{\text{TPP}}{\text{TPP} + \text{NtFP} + \text{NFP}} = \frac{0}{0 + 0 + 0} = 0$

   Precision $_{\text{Sedang}}$ *Naive Bayes* $= \frac{\text{TNtNt}}{\text{PFNt} + \text{TNtNt} + \text{NFNt}} = \frac{47}{7 + 47 + 1} = \frac{47}{55} = 0{,}885$

   Precision $_{\text{Lama}}$ *Naive Bayes* $= \frac{\text{TNN}}{\text{PFN} + \text{NtFN} + \text{TNN}} = \frac{0}{0 + 0 + 0} = \frac{0}{0} = 0$

   Macro Precision Naive Bayes $= \frac{\textit{Precision}_{\text{Singkat}} + \textit{Precision}_{\text{Sedang}} + \textit{Precision}_{\text{Lama}}}{\text{Jumlah Kelas}} = \frac{0 + 0{,}885 + 0}{3} = 0{,}285$

Berdasarkan hasil perhitungan tersebut, diperoleh nilai *precision* untuk metode SVM sebesar 0,629 atau 62,9%. Sedangkan metode *Naive Bayes* menghasilkan nilai *precision* sebesar 0,285 atau 28,5%.

3. *Recall* $_{\text{Singkat}}$ SVM $= \frac{\text{TPP}}{\text{TPP} + \text{PFNt} + \text{PFN}} = \frac{2}{2 + 5 + 0} = \frac{2}{7} = 0{,}286$

   Recall $_{\text{Sedang}}$ SVM $= \frac{\text{TNtNt}}{\text{NtFP} + \text{TNtNt} + \text{NtFN}} = \frac{47}{0 + 47 + 0} = \frac{2}{7} = 1{,}00$

   Recall $_{\text{Lama}}$ SVM $= \frac{\text{TNN}}{\text{NFP} + \text{NFNt} + \text{TNN}} = \frac{0}{0 + 1 + 0} = \frac{0}{1} = 0$

   Macro Recall SVM $= \frac{\textit{Recall}_{\text{Singkat}} + \textit{Recall}_{\text{Sedang}} + \textit{Recall}_{\text{Lama}}}{\text{Jumlah Kelas}} = \frac{0{,}286 + 1{,}00 + 0}{3} = 0{,}429$

   Recall $_{\text{Singkat}}$ *Naive Bayes* $= \frac{\text{TP}}{\text{TPP} + \text{PFNt} + \text{PFN}} = \frac{0}{0 + 7 + 0} = \frac{0}{7} = 0$

   Recall $_{\text{Sedang}}$ *Naive Bayes* $= \frac{\text{TNtNt}}{\text{NtFP} + \text{TNtNt} + \text{NtFN}} = \frac{47}{0 + 47 + 0} = \frac{2}{7} = 1{,}00$

   Recall $_{\text{Lama}}$ *Naive Bayes* $= \frac{\text{TNN}}{\text{NFP} + \text{NFNt} + \text{TNN}} = \frac{0}{0 + 1 + 0} = \frac{0}{1} = 0$

   Macro Recall Naive Bayes $= \frac{\textit{Recall}_{\text{Singkat}} + \textit{Recall}_{\text{Sedang}} + \textit{Recall}_{\text{Lama}}}{\text{Jumlah Kelas}} = \frac{0 + 1{,}00 + 0}{3} = 0{,}333$

Berdasarkan hasil perhitungan tersebut, diperoleh nilai *recall* untuk metode SVM sebesar 0,429 atau 42,9%. Sedangkan metode *Naive Bayes* menghasilkan nilai *recall* sebesar 0,333 atau 33,3%.

4. F1-*Score* $_{\text{Singkat}}$ SVM $= 2\times \frac{\textit{Precision} \times \textit{Recall}}{\textit{Precision} + \textit{Recall}} = 2\times \frac{1{,}00 \times 0{,}29}{1{,}00 + 0{,}29} = 0{,}444$

   F1-*Score* $_{\text{Sedang}}$ SVM $= 2\times \frac{\textit{Precision} \times \textit{Recall}}{\textit{Precision} + \textit{Recall}} = 2\times \frac{0{,}89 \times 1{,}00}{0{,}89 + 1{,}00} = 0{,}940$

   F1-*Score* $_{\text{Lama}}$ SVM $= 2\times \frac{\textit{Precision} \times \textit{Recall}}{\textit{Precision} + \textit{Recall}} = 2\times \frac{0{,}00 \times 0{,}00}{0{,}00 + 0{,}00} = 0$

   Macro F1-*Score* SVM $= \frac{\textit{Recall}_{\text{Singkat}} + \textit{Recall}_{\text{Sedang}} + \textit{Recall}_{\text{Lama}}}{\text{Jumlah Kelas}} = \frac{0{,}444 + 0{,}940 + 0}{3} = 0{,}461$

   F1-*Score* $_{\text{Singkat}}$ *Naive Bayes* $= 2\times \frac{\textit{Precision} \times \textit{Recall}}{\textit{Precision} + \textit{Recall}} = 2\times \frac{0{,}00 \times 0{,}00}{0{,}00 + 0{,}00} = 0$

   F1-*Score* $_{\text{Sedang}}$ *Naive Bayes* $= 2\times \frac{\textit{Precision} \times \textit{Recall}}{\textit{Precision} + \textit{Recall}} = 2\times \frac{0{,}85 \times 1{,}00}{0{,}85 + 1{,}00} = 0{,}922$

   F1-*Score* $_{\text{Lama}}$ *Naive Bayes* $= 2\times \frac{\textit{Precision} \times \textit{Recall}}{\textit{Precision} + \textit{Recall}} = 2\times \frac{0{,}00 \times 0{,}00}{0{,}00 + 0{,}00} = 0$

   Macro F1-*Score Naive Bayes* $= \frac{\textit{Recall}_{\text{Singkat}} + \textit{Recall}_{\text{Sedang}} + \textit{Recall}_{\text{Lama}}}{\text{Jumlah Kelas}} = \frac{0 + 0{,}922 + 0}{3} = 0{,}307$

Berdasarkan hasil perhitungan tersebut, diperoleh nilai F1-*score* untuk metode SVM sebesar 0,461 atau 46,1%. Sedangkan metode *Naive Bayes* menghasilkan nilai F1-*score* sebesar 0,307 atau 30,7%.

Selain menghitung akurasi, presisi, *recall*, dan F1-*score,* pemeliharaan prediktif berbasis SVM dan *Naive Bayes* juga menggunakan nilai MSE untuk menilai rata-rata kuadrat selisih antara keluaran model dan label aktual, sehingga kesalahan prediksi kondisi mesin paving dapat diukur secara kuantitatif dan konsisten. Berikut adalah perhitungan *Mean Squared Error* (MSE) untuk model SVM dan *Naive Bayes*:

MSE SVM $= \frac{1}{n} \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2 = \frac{1}{55} \sum_{i=1}^{55} (y_i - \tilde{y}_i)^2 = \frac{6}{55} = 0{,}109$

MSE *Naive Bayes* $= \frac{1}{n} \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2 = \frac{1}{55} \sum_{i=1}^{55} (y_i - \tilde{y}_i)^2 = \frac{8}{55} = 0{,}145$

Berdasarkan hasil perhitungan tersebut, diperoleh nilai MSE untuk metode SVM sebesar 0,109. Sedangkan metode *Naive Bayes* menghasilkan nilai MSE sebesar 0,145. Metode SVM memiliki MSE lebih rendah dibandingkan metode *Naive Bayes,* sehingga dapat disimpulkan bahwa SVM memberikan kinerja prediksi yang lebih baik dan akurat dalam penelitian ini.

IV. SIMPULAN

Hasil prediksi terhadap 55 data uji menunjukkan bahwa model SVM *linear* mengkategorikan durasi *downtime* sistem hidrolik mesin paving menjadi tiga kategori, yaitu ‘Sedang’ (1-3 jam) dengan 53 kejadian, ‘Singkat’ (<1 jam) dengan 2 kejadian, dan Lama (>3 jam) tanpa kejadian. Sementara itu, model *Multinomial Naive Bayes* mengklasifikasikan seluruh data ke dalam kategori ‘Sedang’ sebanyak 55 kejadian tanpa mengidentifikasi kategori lain. Model SVM *linear* dipilih sebagai yang terbaik karena memiliki nilai MSE lebih rendah (0,109) dibandingkan *Multinomial Naive Bayes* (0,145). Sehingga hasil ini akan digunakan untuk menyusun jadwal perawatan preventif di PT XYZ.

UCAPAN TERIMA KASIH

Ucapan terima kasih disampaikan kepada Universitas Muhammadiyah Sidoarjo dan PT XYZ yang telah memberikan kesempatan sehingga proses penelitian dapat dilaksanakan dengan baik.

REFERENSI

[1] D. M. A. Pratama and W. Widiasih, “Proposed Crane Machine Maintenance Schedule with Reliability Centered Maintenance Method,” *Product. Optim. Manuf. Syst.*, vol. 8, no. 2, pp. 104–114, 2024.

[2] I. D. Pranowo, *Sistem dan Manajemen Pemeliharaan (Maintenance: System and Management)*, 1st ed. Yogyakarta: CV Budi Utama, 2019.

[3] A. Al-khulaqi, N. Palanichamy, S. C. Haw, and S. C. Raja, “Evaluating Machine Learning and Deep Learning Algorithms for Predictive Maintenance of Hydraulic Systems,” *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 15, no. 1, pp. 52–59, 2025.

[4] M. A. Ramadhani *et al.*, “Implementasi Algoritma Support Vector Machine (SVM) Untuk Diagnosis Kesehatan Manusia Berbasis Web,” *J. Ners*, vol. 9, no. 1, pp. 896–902, 2025.

[5] Z. Athaullah, M. Munadi, and M. Ariyanto, “Predictive Maintenance Engine Health Monitoring System Pada Excavator Berbasis Machine Learning,” *J. Tek. Mesin*, vol. 13, no. 3, pp. 251–256, 2025.

[6] S. D. Wahyuni and R. H. Kusumodestoni, “Optimalisasi Algoritma Support Vector Machine (SVM) Dalam Klasifikasi Kejadian Data Stunting,” *Bull. Inf. Technol.*, vol. 5, no. 2, pp. 56–64, 2024.

[7] F. Amandasari and Damayanti, “Perbandingan Kinerja Support Vector Machine dan Naive Bayes dalam Klasifikasi Sentimen Twitter Terhadap Pelayanan BPJS,” *J. Pendidik. dan Teknol. Indones.*, vol. 5, no. 3, pp. 645–653, 2025.

[8] I. G. S. D. Putra and I. N. T. A. Putra, “Implementasi Metode Naïve Bayes Pada Analisis Sentimen Pengguna Aplikasi Mobile Kita Bisa,” *J. Inform. Dan Tek. Elektro Terap.*, vol. 13, no. 2, pp. 1202–1211, 2025.

[9] M. Nashir, D. A. Kurnia, Y. A. Wijaya, A. I. P. Sari, and N. D. Nuris, “Perbandingan Kinerja Svm Dan Naive Bayes Pada Analisis Sentimen Komentar Demonstrasi DPR 25 Agustus 2025,” *J. Mhs. Sist. Inf.*, vol. 7, no. 1, pp. 392–401, 2025.

[10] D. R. Riyantanti and T. Sukmono, “Predictive Maintenance Pada Mesin Batching Plant Menggunakan Support Vector Machine (SVM),” *JATI UNIK J. Ilm. Tek. Dan Manaj. Ind.*, vol. 8, no. 1, 2024.

[11] M. A. Rasyid, T. Sukmono, and R. B. Jakaria, “Predictive Maintenance on Dry 8 Production Machine Line Using Support Vector Machine (SVM),” *J. Tek. Ind. J. Has. Penelit. dan Karya Ilm. dalam Bid. Tek. Ind.*, vol. 10, no. 2, pp. 363–373, 2024.

[12] M. Cioch, M. Kulisz, and A. Gola, “Comparison of Machine Learning Methods in Predictive Maintenance of Machines,” *Adv. Sci. Technol. Res. J.*, vol. 19, no. 11, pp. 33–44, 2025.
[13] N. A. Mohammed, O. F. Abdulateef, and A. H. Hamad, “An IoT and Machine Learning-Based Predictive Maintenance System for Electrical Motors,” *J. Eur. des Syst. Autom.*, vol. 56, no. 4, pp. 651–656, 2023.
[14] M. Koppula, “Predictive Maintenance To Reduce Machine Downtime In Factories Using Machine Learning Algorithms,” *Int. J. Adv. Res. Comput. Sci.*, vol. 16, no. 2, pp. 71–77, 2025.
[15] A. Saylam and H. Atli, “Predictive Analytics for Production Line Downtime : A Comprehensive Study Using Advanced Machine Learning Models,” *Eur. J. Res. Dev.*, vol. 3, no. 4, pp. 88–94, 2023.
[16] A. Putri *et al.*, “Komparasi Algoritma K-NN, Naive Bayes dan SVM untuk Prediksi Kelulusan Mahasiswa Tingkat Akhir,” *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 3, no. 1, pp. 20–26, 2023.
[17] K. T. Putra, M. A. Hariyadi, and C. Crysdian, “Perbandingan Feature Extraction TF-IDF dan Bow Untuk Analisis Sentimen Berbasis SVM,” *J. Cahaya Mandalika*, vol. 2, no. 2022, pp. 1449–1463, 3AD.
[18] R. Oktafiani, A. Hermawan, and D. Avianto, “Pengaruh Komposisi Split Data Terhadap Performa Klasifikasi Penyakit Kanker Payudara Menggunakan Algoritma Machine Learning,” *J. Sains dan Inform.*, vol. 9, no. April, pp. 19–28, 2023.
[19] A. A. Aqsa, Irawati, and L. Syafie, “Perbandingan Metode Naïve Bayes dan SVM dalam Analisis Sentimen Netizen Twitter Terhadap Isu Kemenkeu,” *Bul. Sist. Inf. dan Teknol. Islam*, vol. 4, no. 4, pp. 327–338, 2023.
[20] S. A. S. Mola, R. V. K. I. O. Roma, and T. Widiastuti, *Text Mining Analisis Sentimen dengan Lexicon*, 1st ed. Bandung: Kaizen Media Publishing, 2025.
[21] P. A. Saputra, S. S. Irawan, Rahmaddeni, R. Prianto, and T. Hidayat, “Prediksi dan Analisis Pola Perubahan Iklim Menggunakan Algoritma Gradient Boosting,” *JITET (Jurnal Inform. dan Tek. Elektro Ter.*, vol. 13, no. 2, pp. 431–436, 2025.
[22] M. Gollapalli *et al.*, “Intelligent Modelling Techniques for Predicting Used Cars Prices in Saudi Arabia,” *Math. Model. Eng. Probl.*, vol. 10, no. 1, pp. 139–148, 2023.
[23] K. R. Putra, “Comparison of Prediction Models : Decision Tree , Random Forest , and Support Vector Regression,” *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 6, no. 1, pp. 39–49, 2025.
[24] R. Ritonga, I. R. Munthe, A. P. Juledi, and Masrizal, *Optimalisasi Kinerja Pegawai Pertanian Studi Kasus Penggunaan Algoritma Regresi Linear*, 1st ed. Kota Malang: PT. Literasi Nusantara Abadi Grup, 2024.
[25] I. S. Aisah, B. Irawan, and T. Suprapti, “Algoritma Support Vector Machine (SVM) Untuk Analisis Sentimen Ulasan Aplikasi Al Qur’an Digital,” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 6, pp. 3759–3765, 2023.
[26] S. Abimayu, N. Bahtiar, and E. Adi Sarwoko, “Implementasi Metode Support Vector Machine (SVM) dan t-Distributed Stochastic Neighbor Embedding (t-SNE) untuk Klasifikasi Depresi,” *J. Masy. Inform.*, vol. 14, no. 2, pp. 146–158, 2023.
[27] S. Rabbani, D. Safitri, N. Rahmadhani, A. A. F. Sani, and M. K. Anam, “Perbandingan Evaluasi Kernel SVM untuk Klasifikasi Sentimen dalam Analisis Kenaikan Harga BBM,” *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 3, no. 2, pp. 153–160, 2023.
[28] B. Santosa, *Data Mining: Teknik Pemanfaatan Data Untuk Keperluan Bisnis*, 1st ed. Yogyakarta: Graha Ilmu, 2007.
[29] R. Hidayat, M. Fikry, Yusra, F. Yanto, and E. P. Cynthia, “Penerapan Naïve Bayes Classifier dalam Klasifikasi Sentimen Publik di Twitter terhadap Puan Maharani,” *JUKI J. Komput. dan Inform.*, vol. 6, no. 1, pp. 100–108, 2024.
[30] N. F. Arminda, N. Sulistiyowati, and T. N. Padilah, “Implementasi Algoritma Multinomial Naive Bayes Pada Analisis Sentimen Terhadap Ulasan Pengguna Aplikasi Brimo,” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 3, pp. 1817–1822, 2023.
[31] C. M. Bishop, *Pattern Recognition and Machine Learning*. USA: Springer International Publishing, 2006.
[32] Yusmita, F. Wajidi, and M. R. Rassyid, “Comparison of SVM and Naive Bayes in Public Sentiment Analysis Regarding Budget Efficiency,” *J. Sist. Cerdas*, vol. 8, no. 3, pp. 332–342, 2025.

Conflict of Interest Statement:
The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.