

# Optimalisasi Prediksi Tingkat Obesitas di Negara Mexico Menggunakan Perbandingan Support Vector Machine dan Naïve Bayes

**Diterima:** 10 Juni 2024  
**Revisi:** 10 Juli 2024  
**Terbit:** 1 Agustus 2024

<sup>1\*</sup>Eggy Cahya Ardianta, <sup>2</sup>Prasetyo Ari Wibowo, <sup>3</sup>Nazzel Maulana Mustofa, <sup>4</sup>Aidina Ristyawan, <sup>5</sup>Erna Daniati  
<sup>1-5</sup>Universitas Nusantara PGRI Kediri  
<sup>1</sup>[eggycahyac@gmail.com](mailto:eggycahyac@gmail.com), <sup>2</sup>[pras.ari69@gmail.com](mailto:pras.ari69@gmail.com),  
<sup>3</sup>[nazzelmaulana@gmail.com](mailto:nazzelmaulana@gmail.com), <sup>4</sup>[aidinaristi@unpkediri.ac.id](mailto:aidinaristi@unpkediri.ac.id),  
<sup>5</sup>[ernadaniati@unpkediri.ac.id](mailto:ernadaniati@unpkediri.ac.id)

**Abstrak**—Obesitas, yang sering dikaitkan dengan penyakit jantung, diabetes, dan stroke, merupakan masalah besar bagi kesehatan global. Jumlah orang obesitas di Indonesia telah meningkat secara signifikan, mencapai 21,80 persen dari populasi dewasa pada tahun 2018. Studi ini menggunakan dataset yang tersedia untuk umum untuk memperkirakan tingkat obesitas di Meksiko. Teknik data mining yang digunakan termasuk algoritma Naive Bayes dan Support Vector Machine (SVM). Tujuannya adalah untuk membuat model prediksi yang dapat menemukan orang yang lebih berisiko terkena obesitas dan membantu mengambil langkah pencegahan yang tepat.

**Kata Kunci**—Obesitas;Naive Bayes;Support Vector Machine

**Abstract**— Obesity, which is often linked to heart disease, diabetes and stroke, is a major problem for global health. The number of obese people in Indonesia has increased significantly, reaching 21.80 percent of the adult population in 2018. This study uses publicly available datasets to estimate obesity rates in Mexico. The data mining techniques used include the Naive Bayes algorithm and Support Vector Machine (SVM). The goal is to create a prediction model that can find people who are more at risk of obesity and help take appropriate preventive steps..

**Keywords**—Obesity;Naive Bayes;Support Vector Machine

This is an open access article under the CC BY-SA License.



---

## Penulis Korespondensi:

Aidina Ristyawan,  
Sistem Informasi,  
Universitas Nusantara PGRI Kediri,  
Email: [aidinaristi@unpkediri.ac.id](mailto:aidinaristi@unpkediri.ac.id)  
ID Orcid: [<https://orcid.org/0009-0003-2712-1507>]  
Handphone: 081232624460

---

## I. PENDAHULUAN

Dampak negatif akibat modernisasi ialah perubahan gaya hidup dari traditional life style menjadi sedentary life style seperti kurangnya aktivitas fisik serta penyimpangan pola makan yang cenderung tinggi asupan energi (lemak, protein, karbohidrat) dan rendah serat.[1] Puspitasari menjelaskan bahwa obesitas di pengaruhi oleh jenis kelamin, tingkat oengetahuan, tingkat pendidikan, jenis pekerjaan, status pernikahan, Riwayat keturunan, aktvitas fisik, status merokok dan intake kalori.[2] Di Meksiko, angka obesitas mencapai 74,5% pada wanita dewasa dan 69,1% pada pria dewasa. Obesitas tidak hanya terjadi pada usia dewasa,[3]dan remaja.[4] Mellihat besarnya dampak negative yang di sebabkan oleh obesitas, ini menarik peneliti untuk menyusun algoritma yang dapat memprediksi level obesitas.[5]

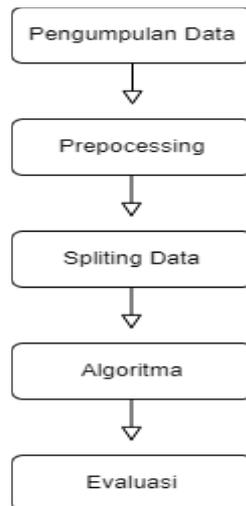
Untuk membuat prediksi, beberapa peneliti menggunakan machine learning dan deep learning. Machine learning mengacu secara luas pada proses penyesuaian model prediktif dengan data atau mengidentifikasi pengelompokan informatif dalam data.[6]Sedangkan deep learnig merupakan konsep dari machine learning yang didasarkan pada jaringan saraf tiruan, menurut Janiesch, untuk banyak aplikasi model deep learning mengungguli model machine learning.[7]Mesin learning juga telah membantu mengotomatisasi penelitian dengan menggunakan model yang dipelajari mesin untuk menyarankan eksperimen atau simulasi baru, machine learning ini menghasilkan loop umpan balik data dari pembuatan data, pelatihan ulang model dan prediksi yang ditingkatkan mewakili perubahan paradigma dalam bahan penelitian [8]

Metode SVM bekerja dengan mencari hyperplane atau garis pemisah terbaik yang memiliki margin atau jarak antara kelas terbesar. SVM menggunakan beberapa kernel dalam klasifikasi, termasuk Linear, Radial Basis Function (RBF), dan Polynomial, untuk memperoleh klasifikasi yang optimal. Algoritma Naive Bayes adalah algoritma pembelajaran mesin untuk masalah klasifikasi, yang terutama digunakan untuk klasifikasi teks yang berisi kumpulan data pelatihan berdimensi tinggi [9]

Tujuan penelitian ini adalah untuk memanfaatkan teknik data mining, terutama algoritma nave bayes dan support vector machine , untuk melakukan prediksi tingkat obesitas di Meksiko dengan menggunakan Phyton. Data yang digunakan dalam penelitian berasal dari dataset <https://www.kaggle.com/datasets/fatemehmehrpavar/obesity-levels>. Dataset ini mencakup informasi tentang individu dari Meksiko, Peru, dan Kolombia, termasuk faktor risiko obesitas serta tingkat obesitas yang mereka alami, serta memberikan kontribusi signifikan dalam penanggulangan masalah obesitas di Meksiko melalui pengembangan model prediksi yang akurat

## II. METODE

Penelitian ini bertujuan untuk membangun model klasifikasi tingkat obesitas dengan membandingkan dua metode, yaitu Support Vector Machine dan Naïve Bayes. Alur penelitian ini digambarkan pada Gambar 1. Penelitian ini juga akan mengevaluasi kinerja kedua metode berdasarkan akurasi, presisi, dan recall untuk menentukan metode mana yang lebih efektif dalam mengklasifikasikan tingkat obesitas. Dengan demikian, diharapkan hasil penelitian ini dapat memberikan kontribusi signifikan dalam pengembangan alat prediksi yang lebih akurat dan andal untuk mendeteksi obesitas pada populasi..



Gambar 2.1. Alur Penelitian

### 2.1. Pengumpulan Data

Data utama yang digunakan dalam penelitian ini adalah dataset publik tentang tingkat obesitas di negara Meksiko yang tersedia di platform Kaggle. Dataset ini terdiri dari 2.111 entri data, masing-masing dengan 16 fitur yang menjelaskan berbagai aspek yang terkait dengan tingkat obesitas. Terakhir kali dataset ini diperbarui adalah 2 bulan yang lalu, menunjukkan bahwa data tersebut lengkap dan akurat. Secara keseluruhan, kumpulan data ini memberikan landasan yang kokoh untuk melakukan analisis menyeluruh tentang tingkat obesitas yang terkait dengan penelitian ini.

### 2.2. Preprocessing

Preprocessing adalah sebuah langkah penting dalam proses penambangan data. Data yang akan digunakan dalam proses penambangan data tidak selalu dalam kondisi terbaik untuk diproses. Ada kalanya dalam data tersebut terdapat beberapa masalah yang nantinya dapat mempengaruhi hasil yang diberikan dari proses penambangan itu sendiri seperti terdapat nilai yang hilang, data yang berlebihan, outlier, atau format data yang tidak sesuai dengan sistem. Oleh karena itu untuk mengatasi masalah tersebut perlu dilakukan tahap preprocessing. Preprocessing adalah salah satu langkah dalam menghilangkan masalah yang dapat mengganggu hasil dari pada proses klasifikasi data. [11]

#### 1. Data Cleaning

Data Cleaning adalah menghapus atau mengisi nilai kosong untuk seluruh kumpulan data dengan nilai rata-rata dari tiap kolom pada nilai yang kosong [12]

#### 2. Data Transform

Data Transformation adalah mengubah data untuk mendapatkan data yang lebih baik [13] dilakukan untuk mengubah data yang akan dianalisis sehingga dapat menghasilkan nilai yang lebih relevan. [14]

#### 3. Normalization

Normalisasi Data merupakan proses transformasi skala atribut numerik ke dalam bentuk lebih sederhana seperti 0 hingga 1. [15]

### 2.3. Splitting Data

Splitting data berfungsi untuk membagi dataset menjadi dua bagian, yaitu data pelatihan dan data uji dengan proporsi tertentu. [16] Penelitian ini membagi data dengan rasio 80% data latih dan 20% sebagai data uji. Metode ini memungkinkan eksplorasi kinerja model dalam berbagai skenario pembagian data latih dan uji. Data yang telah diikuti dua tahap ini kemudian diuji menggunakan algoritma support vector machine dan Naïve Bayes untuk mengevaluasi kinerja model pada setiap skenario pembagian data

### 2.4. Algoritma

Dua algoritma yang digunakan dalam penelitian ini adalah Naïve Bayes dan metode Support Vector Machine (SVM). Naïve Bayes menggunakan prinsip probabilitas untuk menentukan kelas instance, sedangkan SVM menggunakan hyperplane atau garis pemisah untuk memperoleh klasifikasi.

#### 1. Naïve bayes

Klasifikasi Bayes adalah pengelompokan statistik yang memperkirakan kemungkinan keanggotaan kelas misalnya, kemungkinan bahwa sebuah tuple masuk ke dalam kelas tertentu. [17] Teorema Bayes adalah algoritma Naïve Bayes yang menggunakan pengalaman masa lalu untuk memprediksi apa yang akan terjadi di masa depan. [18] Tahapan proses Naïve Bayes sebagai berikut :

$$P(A/B) = \frac{P\left(\frac{B}{A}\right) * P(A)}{P(B)}$$

Keterangan dari persamaan diatas yaitu :

$P(A|B)$  = Probabilitas A dan B terjadi bersama-sama

$P(B|A)$  = Probabilitas B dan A terjadi bersama-sama

$P(A)$  = Probabilitas kejadian

$P(B)$  = Probabilitas kejadian B

#### 2. Support Vector Machine

SVM (Support Vector Machine) adalah algoritma Machine Learning yang menerapkan fungsi hyperplane pada data sehingga terbentuk daerah-daerah untuk tiap kelas. Hyperplane sendiri merupakan sebuah fungsi yang digunakan sebagai pemisah antar kelas yang ada. Dalam memprediksi suatu kelas dari data, SVM akan melabelinya berdasarkan daerah kelas mana yang merupakan tempat dari data tersebut. SVM biasanya digunakan pada dataset besar yang diambil dari situs online dan menjadi populer karena penerapannya dalam klasifikasi teks. [19] Pada penelitian ini akan difokuskan menggunakan fungsi linear dan membagi menjadi 2 kelas yaitu positif dan negatif. Fungsi mencari hyperlane dapat didefinisikan pada persamaan 2 dan garis tepi pada hyperplane untuk membagi kelas positif dan negatif didefinisikan pada persamaan 3.

$$(W.X) + b = 0$$

$$(W.X) + b = +1 \text{ (untuk kelas positif)}$$

$$(W.X) + b = -1 \text{ (untuk kelas negative)}$$

Sama seperti proses klasifikasi Naïve Bayes Classifier, Untuk proses klasifikasi dengan metode Support Vector Machine juga menggunakan bantuan library Scikit-Learn. Adapun library Scikit-Learn yang dipakai di antaranya adalah CountVectorizer, TfidfVectorizer, svm, accuracy\_score, precision\_score, recall\_score, f1\_score, classification\_report, dan confusion\_matrix. Langkah pertama untuk melakukan

klasifikasi adalah menginstall library yang diperlukan. Selanjutnya setelah semua library yang diperlukan dipastikan sudah terinstal maka dilanjutkan dengan proses prediksi menggunakan data test untuk mendapatkan nilai accuracy, precision, recall, dan f1\_score.

## 2.5. Evaluasi

Pada tahap ini, evaluasi metode prediksi dilakukan untuk mengukur kinerja algoritma Support Vector Machine (SVM) dan Naïve Bayes (NB). Evaluasi ini akan membandingkan nilai akurasi yang dihasilkan oleh kedua algoritma tersebut. Selain itu, akan dilakukan analisis mendalam terhadap presisi, recall, dan F1-score untuk memberikan gambaran yang lebih komprehensif mengenai efektivitas masing-masing algoritma dalam mengklasifikasikan tingkat obesitas. Hasil evaluasi ini diharapkan dapat memberikan panduan yang jelas dalam memilih algoritma yang paling sesuai untuk prediksi obesitas.

## III. HASIL DAN PEMBAHASAN

### 3.1. Pengumpulan Data

Data diperoleh dari situs keggale, data yang diperoleh sebesar 2111 data. Dengan 16 atribut dan 1 label yaitu “nobeyesdad”. Berikut detail datanya:

Tabel 1

Atribut	Keterangan
FAVC	kebiasaan orang mengkonsumsi makanan ber kalori tinggi,.
FCVC	frekuensi orang mengkonsumsi sayuran.
NCP	jumlah makan utama yang dimakan.
CAEC	kebiasaan orang mengkonsumsi makanan diantara waktu makan.
CH20	kebiasaan orang mengkonsumsi air setiap harinya.
CALC	Kebiasaan orang mengkonsumsi minuman beralkohol.

Tabel 2

Atribut	Keterangan
SCC	kebiasaan orang dalam melakukan pemantauan mengenai perihalkonsumsi
FAF	frekuensi aktivitas fisik yang dilakukan
TUE	lama waktu menggunakan perangkat teknologi
MTRANS	jenis transportasi yang digunakan

Tabel 3

Atribut	Keterangan
Gender	Jenis Kelamin
Age	Umur
Height	Tinggi Badan
Weight	Berat Badan
Family History With Overweight	Riwayat Obesitas
Smoke	Kebiasaaan Merokok

### 3.2. Preprocessing

#### 1. Data Cleansing

Tidak dilakukan pembersihan data karena data yang didapat merupakan data yang bersih tetapi masih terdapat outlier.

## 2. Data Transform

Tidak mengubah nama atribut dikarenakan untuk menjaga keaslian data, apabila ingin mengubah nama atribut juga diperbolehkan.

## 3. Normalization

Untuk mempermudah proses Analisa data, diharuskan untuk mengubah data menjadi numerik. Detail contohnya sebagai berikut:

```
[4]: # Konversi variabel kategoris menjadi numerik
le = LabelEncoder()
df['Gender'] = le.fit_transform(df['Gender'])
df['CALC'] = le.fit_transform(df['CALC'])
df['FAVC'] = le.fit_transform(df['FAVC'])
df['SCC'] = le.fit_transform(df['SCC'])
df['SMOKE'] = le.fit_transform(df['SMOKE'])
df['family_history_with_overweight'] = le.fit_transform(df['family_history_with_overweight'])
df['MTRANS'] = le.fit_transform(df['MTRANS'])
df['NObeyesdad'] = le.fit_transform(df['NObeyesdad'])

df

[4]:
```

	Age	Gender	Height	Weight	CALC	FAVC	FCVC	NCP	SCC	SMOKE	CH2O	family_history_with_overweight	FAF	TUE	CAEC	MTRANS	NObeyesdad
0	21.000000	0	1.620000	64.000000	3	0	2.0	3.0	0	0	2.000000	1	0.000000	1.000000	2	3	1
1	21.000000	0	1.520000	56.000000	2	0	3.0	3.0	1	1	3.000000	1	3.000000	0.000000	2	3	1
2	23.000000	1	1.800000	77.000000	1	0	2.0	3.0	0	0	2.000000	1	2.000000	1.000000	2	3	1
3	27.000000	1	1.800000	97.000000	1	0	3.0	3.0	0	0	2.000000	0	2.000000	0.000000	2	4	5
4	22.000000	1	1.780000	89.800000	2	0	2.0	1.0	0	0	2.000000	0	0.000000	0.000000	2	3	6
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2106	20.978842	0	1.710730	131.408528	2	1	3.0	3.0	0	0	1.728139	1	1.676269	0.906547	2	3	4
2107	21.992242	0	1.748584	133.742943	2	1	3.0	3.0	0	0	2.005130	1	1.341390	0.599270	2	3	4
2108	22.524036	0	1.752206	133.689352	2	1	3.0	3.0	0	0	2.054193	1	1.414209	0.646288	2	3	4
2109	24.361936	0	1.739450	133.346641	2	1	3.0	3.0	0	0	2.852339	1	1.139107	0.588035	2	3	4
2110	23.654709	0	1.738836	133.472641	2	1	3.0	3.0	0	0	2.863513	1	1.026452	0.714137	2	3	4

2111 rows x 17 columns

Gambar 1

## 3.3. Splitting Data

Dalam proses ini kita membagi data pelatihan dan data pengujian. Sedangkan yang saya pakai yaitu rasio 80% data latih dan 20% data uji. Detail gambarnya sebagai berikut:

```
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, accuracy_score

# Pisahkan fitur dan target
X = df_clipped.drop('NObeyesdad', axis=1)
y = df_clipped['NObeyesdad']

# Pisahkan data menjadi data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=14)

# Inisialisasi model Naive Bayes (misalnya, Gaussian Naive Bayes)
model = GaussianNB()

# Latih model dengan data latih
model.fit(X_train, y_train)

# Prediksi dengan data uji
y_pred = model.predict(X_test)

# Evaluasi model
accuracy = accuracy_score(y_test, y_pred)
print(f"Akurasi model Naive Bayes: {accuracy:.2f}")

print("Classification Report:")
print(classification_report(y_test, y_pred))
```

Gambar 2

## 3.4. Algoritma

### 1. Naïve Bayes

Klasifikasi naïve bayes adalah pengelompokka statistic yang memperkirakan kemungkinan keanggotaan kelas misalnya, kemungkinan bahwa tuple masuk kedalam kelas tertentu. Diperoleh bahwa akurasiya yaitu 65%. Detail gambarnya sebagai berikut:

```

from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, accuracy_score

# Pisahkan fitur dan target
X = df_clipped.drop('NObayesdad', axis=1)
y = df_clipped['NObayesdad']

# Pisahkan data menjadi data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=14)

# Inisialisasi model Naive Bayes (misalnya, Gaussian Naive Bayes)
model = GaussianNB()

# Latih model dengan data latih
model.fit(X_train, y_train)

# Prediksi dengan data uji
y_pred = model.predict(X_test)

# Evaluasi model
accuracy = accuracy_score(y_test, y_pred)
print(f"Akurasi model Naive Bayes: {accuracy:.2f}")

print("Classification Report:")
print(classification_report(y_test, y_pred))

Akurasi model Naive Bayes: 0.65
Classification Report:

```

	precision	recall	f1-score	support
0	0.71	0.88	0.78	89
1	0.56	0.39	0.46	89
2	0.41	0.63	0.49	101
3	0.63	0.87	0.73	77
4	1.00	0.98	0.99	119
5	0.64	0.37	0.47	86
6	0.58	0.25	0.35	73
accuracy			0.65	634
macro avg	0.65	0.63	0.61	634
weighted avg	0.66	0.65	0.63	634

Gambar 3

## 2. Support Vector Machine

Metode SVM bekerja dengan cara mencari hyperplane atau garis pemisah terbaik yang memiliki margin atau jarak antar kelas terbesar menggunakan beberapa kernel dalam klasifikasi SVM antara lain Linier, Radial Basic Function (RBF), dan Polynomial untuk memperoleh klasifikasi. Untuk akurasi yang diperoleh melalui metode ini yaitu 89%. Untuk detail gambarnya sebagai berikut:

```

[23]: from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report, accuracy_score

# Pisahkan fitur dan target
X = df_clipped.drop('NObayesdad', axis=1)
y = df_clipped['NObayesdad']

# Pisahkan data menjadi data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=14)

# Inisialisasi model SVM
model = SVC(kernel='linear', random_state=14) # Gunakan kernel linear sebagai contoh

# Latih model dengan data latih
model.fit(X_train, y_train)

# Prediksi dengan data uji
y_pred = model.predict(X_test)

# Evaluasi model
accuracy = accuracy_score(y_test, y_pred)
print(f"Akurasi model SVM: {accuracy:.2f}")

print("Classification Report:")
print(classification_report(y_test, y_pred))

Akurasi model SVM: 0.89
Classification Report:

```

	precision	recall	f1-score	support
0	0.87	1.00	0.93	89
1	0.95	0.78	0.85	89
2	0.88	0.91	0.90	101
3	0.94	0.97	0.96	77
4	0.99	0.99	0.99	119
5	0.81	0.86	0.84	86
6	0.77	0.68	0.72	73
accuracy			0.89	634
macro avg	0.89	0.89	0.88	634
weighted avg	0.89	0.89	0.89	634

Gambar 4

### 3.5. Evaluasi

Bisa dilihat untuk perbandingan dari kedua metode tersebut bahwa *Support Vector Machine(SVM)* diperoleh sebesar 89%, sedangkan untuk *naïve bayes* sebesar 65%. Jadi akurasi lebih besar didapatkan menggunakan metode *SVM*

## IV. KESIMPULAN

Penelitian ini bertujuan untuk membangun model klasifikasi tingkat obesitas dengan membandingkan dua metode, yaitu Support Vector Machine (SVM) dan Naïve Bayes (NB). Berdasarkan data yang diperoleh dari situs Kaggle dengan 2111 entri dan 16 atribut, dilakukan beberapa langkah pemrosesan data termasuk normalisasi dan pembagian data menjadi set pelatihan dan pengujian dengan rasio 80% untuk pelatihan dan 20% untuk pengujian. Hasil evaluasi menunjukkan bahwa metode SVM memberikan akurasi sebesar 89%, sementara

metode Naïve Bayes hanya mencapai akurasi 65%. Selain itu, analisis lebih lanjut terhadap presisi, recall, dan F1-score memperkuat temuan ini, menunjukkan bahwa SVM lebih unggul dalam mengklasifikasikan tingkat obesitas dibandingkan dengan Naïve Bayes. Dengan demikian, dapat disimpulkan bahwa Support Vector Machine merupakan metode yang lebih efektif dan akurat untuk prediksi tingkat obesitas dibandingkan dengan Naïve Bayes. Hasil penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam pengembangan alat prediksi yang lebih akurat dan andal untuk mendeteksi obesitas pada populasi.

#### DAFTAR PUSTAKA

- [1] U. L. Wijaya, B. Widjanarko, and R. Indraswari, "Faktor-Faktor Yang Berhubungan Dengan Perilaku Makan Berisiko Gizi Lebih Pada Remaja Sma Di Kota Semarang," *Jurnal Kesehatan Masyarakat*, vol. 8, no. 3, pp. 426–431, 2020.
- [2] N. Puspitasari, "Faktor kejadian obesitas sentral paa usia dewasa," *Higeia J. puplic Heal. Res. Dev.*, vol. 2, no. 2, pp. 249–259, 2018.
- [3] R. R. Suryadinata and D. A. Sukarno, "Pengaruh Aktivitas Fisik Terhadap Risiko Obesitas Pada Usia Dewasa= The Effect Of Physical Activity On The Risk Of Obesity In Adulthood," *The Indonesian Journal of Public Health*, vol. 14, no. 1, pp. 106–116, 2019.
- [4] K. Kusdalina, A. Mutia, and J. Jumiyati, "Pola makan dan aktivitas fisik terhadap kejadian obesitas remaja pada masa pandemi Covid-19," *Journal of Nutrition College*, vol. 11, no. 1, pp. 26–34, 2022.
- [5] L. Setiyani, A. N. Indahsari, and R. Roestam, "Analisis Prediksi Level Obesitas Menggunakan Perbandingan Algoritma Machine Learning dan Deep Learning," *JTERA J. Teknol. Rekayasa*, vol. 8, no. 1, pp. 2022–2139, 2023.
- [6] J. G. Greener, S. M. Kandathil, L. Moffat, and D. T. Jones, "A guide to machine learning for biologists," *Nat Rev Mol Cell Biol*, vol. 23, no. 1, pp. 40–55, 2022.
- [7] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electronic Markets*, vol. 31, no. 3, pp. 685–695, 2021.
- [8] G. L. W. Hart, T. Mueller, C. Toher, and S. Curtarolo, "Machine learning for alloys," *Nat Rev Mater*, vol. 6, no. 8, pp. 730–755, 2021.
- [9] A. Ridwan, "Penerapan Algoritma Naïve Bayes Untuk Klasifikasi Penyakit Diabetes Mellitus," *J. SISKOM-KB (Sistem Komput. dan Kecerdasan Buatan)*, vol. 4, no. 1, pp. 15–21, 2020.
- [10] R. W. Abdullah, D. Hartanti, H. Permatasari, A. W. Septyanto, and Y. Abi Bagaskara, "Penerapan Data Mining untuk Memprediksi Jumlah Produk Terlaris Menggunakan Algoritma Naive Bayes Studi Kasus (Toko Prapti)," *Jurnal Ilmiah Informatika Global*, vol. 13, no. 1, 2022.
- [11] M. D. Purbolaksono, M. I. Tantowi, A. I. Hidayat, and A. Adiwijaya, "Perbandingan support vector machine dan modified balanced random forest dalam deteksi pasien penyakit diabetes," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 2, pp. 393–399, 2021.
- [12] G. A. B. Suryanegara and M. D. Purbolaksono, "Peningkatan Hasil Klasifikasi pada Algoritma Random Forest untuk Deteksi Pasien Penderita Diabetes Menggunakan Metode Normalisasi," *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, vol. 5, no. 1, pp. 114–122, 2021.
- [13] B. Prasojo and E. Haryatmi, "Analisa Prediksi Kelayakan Pemberian Kredit Pinjaman dengan Metode Random Forest," *J. Nas. Teknol. dan Sist. Inf*, vol. 7, no. 2, pp. 79–89, 2021.
- [14] A. E. Satriatama et al., "Analisis Klaster Data Pasien Diabetes untuk Identifikasi Pola dan Karakteristik Pasien," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 5, no. 3, pp. 172–182, 2023.

- [15] M. D. Purbolaksono, M. I. Tantowi, A. I. Hidayat, and A. Adiwijaya, “Perbandingan support vector machine dan modified balanced random forest dalam deteksi pasien penyakit diabetes,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 2, pp. 393–399, 2021.
- [16] V. A. P. Putri, A. B. Prasetyo, and D. Eridani, “Perbandingan Kinerja Algoritme Naïve Bayes Dan K-Nearest Neighbor (Knn) Untuk Prediksi Harga Rumah,” *Transmisi: Jurnal Ilmiah Teknik Elektro*, vol. 24, no. 4, pp. 162–171, 2022.
- [17] D. Zamri, “Perbandingan Metode Data Mining untuk Prediksi Banjir Dengan Algoritma Naïve Bayes dan KNN: Comparison of Data Mining Methods for Prediction of Floods with Naïve Bayes and KNN Algorithm,” in *SENTIMAS: Seminar Nasional Penelitian dan Pengabdian Masyarakat*, 2022, pp. 40–48.
- [18] D. D. Putri, G. F. Nama, and W. E. Sulistiono, “Analisis Sentimen Kinerja Dewan Perwakilan Rakyat (DPR) Pada Twitter Menggunakan Metode Naive Bayes Classifier,” *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 10, no. 1, 2022.
- [19] S. Rana and A. Singh, “Comparative analysis of sentiment orientation using SVM and Naive Bayes techniques,” in *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, IEEE, 2016, pp. 106–111.