



Pendekatan Metode *Ensemble Learning* untuk Prakiraan Cuaca menggunakan *Soft Voting Classifier*

Steven Joses¹, Donata Yulvida², Siti Rochimah³

^{1,2,3}Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

¹stevenjos@widyadharma.ac.id, ²donata_yulvida@widyadharma.ac.id, ³siti@if.its.ac.id

Abstract

Weather conditions are one of the crucial factors that need attention. Changes in weather conditions significantly impact various activities. Weather condition changes are determined by numerous factors, often occurring within a relatively short period in the atmosphere, such as pressure, wind speed, rainfall, temperature, and other atmospheric phenomena. Issues in weather forecasting arise due to several factors, namely the fluctuating atmospheric conditions. This research proposes the development of a weather forecasting model using the ensemble learning method approach. The weather data used consist of 33746 records with attributes used after preprocessing, namely Temperature, Dew Point, Humidity, Wind Speed, Wind Gust, Pressure, Precipitation, and Condition. Testing in this research employs several single-machine learning methods such as K-Nearest Neighbor (KNN), Logistic Regression, Random Forest, Naive Bayes, and Multi-Layer Perceptron. The Naive Bayes method using default parameters achieves a high accuracy of 99.00%. In the ensemble method, combinations of three methods exhibit excellent accuracy for all combinations. The best combination methods are found in the Soft Voting Classifier method (Random Forest, MLP, Naive Bayes), Soft Voting Classifier (Logistic Regression, MLP, Naive Bayes), and Soft Voting Classifier (Random Forest, KNN, Naive Bayes) with an accuracy of 99.03%.

Keywords: weather prediction, machine learning, ensemble method, soft voting classifier

Abstrak

Kondisi cuaca merupakan salah satu faktor penting yang perlu diperhatikan. Perubahan kondisi cuaca berdampak signifikan pada berbagai aktivitas. Perubahan kondisi cuaca ditentukan oleh banyak faktor, seringkali dalam jangka waktu yang relatif singkat dalam kondisi atmosfer seperti tekanan, kecepatan angin, curah hujan, suhu dan fenomena atmosfer lainnya. Permasalahan pada prakiraan cuaca disebabkan oleh beberapa faktor yaitu kondisi atmosfer yang berubah-ubah. Penelitian ini mengusulkan pengembangan model prakiraan cuaca dengan pendekatan metode *ensemble learning*. Data cuaca yang digunakan sebanyak 33746 record dengan atribut yang digunakan setelah proses *preprocessing* yaitu Temperature, Dew Point, Humidity, Wind Speed, Wind Gust, Pressure, Precipitation, Condition. Pengujian dalam penelitian ini menggunakan beberapa metode *machine learning* tunggal seperti K-Nearest Neighbour (KNN), Logistic Regression, Random Forest, Naive Bayes dan Multi Layer Perceptron. Metode Naive Bayes menggunakan parameter default memperoleh akurasi tinggi sebesar 99,00%. Pada *ensemble method* dengan kombinasi 3 metode memiliki akurasi yang sangat baik untuk semua kombinasinya. Kombinasi metode terbaik adalah pada metode Soft Voting Classifier (Random Forest, MLP, Naive Bayes), Soft Voting Classifier (Logistic Regression, MLP, Naive Bayes), dan Soft Voting Classifier (Random Forest, KNN, Naive Bayes) dengan akurasi sebesar 99,03%.

Kata kunci: prakiraan cuaca, *machine learning*, *ensemble method*, *soft voting classifier*

1. Pendahuluan

Cuaca merupakan aspek penting dalam kehidupan sehari-hari masyarakat. Kondisi cuaca merupakan salah satu faktor penting yang perlu diperhatikan, karena cuaca pada suatu wilayah menentukan berbagai macam aktivitas manusia. Perubahan kondisi cuaca berdampak signifikan pada berbagai aktivitas, mulai dari perencanaan pariwisata, pertanian, hingga keselamatan masyarakat.

Pontianak, ibu kota Provinsi Kalimantan Barat, memang terkenal merupakan kota khatulistiwa karena

bersinggungan langsung dengan garis khatulistiwa [1]. Cuaca di kota Pontianak cenderung panas sepanjang tahun dengan suhu rata-rata harian sekitar 27 hingga 32 derajat celsius. Kota Pontianak memiliki iklim tropis basah dan kering yang dipengaruhi oleh kedekatannya dengan khatulistiwa. Kota Pontianak juga berada dalam negara yang terletak pada zona iklim tropis dan memiliki dua musim yaitu musim kemarau dan musim hujan. Musim kemarau dan musim hujan bergantian setiap enam bulan sekali. Indonesia merupakan negara dengan iklim hutan hujan tropis yang ditandai dengan suhu udara hampir seragam dan curah hujan tinggi yang



Lisensi

Lisensi Internasional Creative Commons Attribution-ShareAlike 4.0.

tersebar merata sepanjang tahun [2]. Namun, menurut WWF (*World Wildlife Fund for Nature*) Indonesia, dalam beberapa tahun terakhir, perubahan iklim global ditandai dengan pergantian musim kemarau dan musim hujan yang tidak menentu [1].

Salah satu lembaga yang mencatat perubahan iklim di Indonesia adalah Badan Meteorologi, Klimatologi, dan Geofisika (BMKG). BMKG merupakan lembaga nasional yang memberikan informasi penting, termasuk prakiraan kondisi cuaca [3]. BMKG menyediakan prakiraan cuaca dengan memperhatikan kondisi cuaca terkini yang terjadi saat ini, misalnya kondisi suhu, kelembaban dan kecepatan angin. Setelah mengamati kondisi cuaca, BMKG melakukan analisis kondisi cuaca yang diperoleh untuk memperoleh hasil prakiraan cuaca.

Kondisi cuaca ditentukan oleh banyak faktor, seringkali dalam jangka waktu yang relatif singkat dalam kondisi atmosfer, yang dapat dirumuskan menjadi beberapa atribut seperti tekanan, kecepatan angin, curah hujan, suhu dan fenomena atmosfer lainnya [4]. Besarnya pengaruh yang ditimbulkan oleh cuaca mendorong dikembangkannya sistem penentuan cuaca untuk mengetahui kondisi cuaca [5]. Untuk mengetahui perubahan cuaca tersebut, maka dirancanglah prakiraan cuaca untuk melihat dan menganalisa kondisi cuaca yang akan terbentuk kedepannya [1]. Pemilihan metode yang tepat untuk menentukan kondisi cuaca akhir-akhir ini menjadi prioritas utama bagi banyak ahli atmosfer atau cuaca.

Prakiraan cuaca merupakan kajian yang menarik untuk dilakukan, apalagi dengan pesatnya pertumbuhan kecerdasan buatan dalam penggunaannya di masyarakat global [6]. Permasalahan pada prakiraan cuaca disebabkan oleh beberapa faktor yaitu kondisi atmosfer yang berubah-ubah, kesalahan pengukuran, data yang terlalu banyak, dan pemahaman yang tidak lengkap mengenai faktor yang mempengaruhi prakiraan cuaca yang dihasilkan. Meskipun terdapat kemajuan pesat dalam prakiraan cuaca, cuaca merupakan sistem yang sangat kompleks dengan banyak parameter yang saling bergantung. Perubahan kecil pada suatu parameter atau kesalahan dalam mengelompokkan kondisi cuaca dapat menyebabkan perbedaan signifikan dalam prakiraan jangka panjang. Keakuratan dan ketepatan waktu prakiraan cuaca yang terbatas merupakan sebuah kesenjangan yang terus menjadi fokus penelitian. Teknik klasifikasi cuaca yang baik dapat memprediksi kemungkinan perubahan cuaca yang terjadi dengan lebih akurat [7].

Salah satu kegunaan klasifikasi adalah untuk menentukan jenis cuaca. Beberapa penelitian telah menggunakan algoritma pembelajaran mesin untuk klasifikasi dan prakiraan cuaca [7]. Melalui klasifikasi, dapat diidentifikasi jenis cuaca yang sedang terjadi atau yang akan datang dengan akurasi tinggi berdasarkan data yang tersedia, seperti suhu, kelembaban, tekanan udara,

dan data observasi lainnya. Studi terbaru menunjukkan bahwa berbagai algoritma *Machine Learning* telah digunakan untuk identifikasi dan prediksi. Metode *Machine Learning* telah menghasilkan efisiensi dan peningkatan yang sangat besar dibandingkan dengan metode konvensional.

Penelitian mengenai prakiraan cuaca dengan nilai akurasi *Backpropagation* pada musim hujan lebih rendah sebesar 68,33% dibandingkan pada musim kemarau (87,57%), sedangkan regularisasi Bayesian pada musim hujan mempunyai nilai akurasi tertinggi sebesar 99,70% dan menurun sebesar 99,06% pada musim kemarau. Perbedaan fluktuasi kenaikan dan penurunan ini disebabkan karena distribusi data yang tidak seimbang. Adapun masukan yang digunakan pada penelitian yaitu temperatur, tekanan udara, kelembaban udara, tudung awan, kecepatan angin, dan presipitasi. Hasil keluaran prakiraan cuaca berupa cuaca cerah, cuaca berawan, hujan ringan, dan hujan lebat. Untuk mengembangkan lebih jauh dapat dilakukan dengan membandingkannya dengan algoritma *deep learning*. Selain itu perlu dilakukan rasionalisasi kumpulan data secara seimbang agar akurasi dan error lebih stabil untuk mencapai performa terbaik, serta algoritma ini layak untuk direkomendasikan [6].

Penelitian berikutnya mengenai prediksi cuaca menghasilkan akurasi sebesar 70%. Penelitian tersebut menggunakan *hybrid method* yaitu *Locality Sensitive (LSH)* dan algoritma *K-Nearest Neighbour (KNN)* dengan 9 parameter cuaca yaitu suhu minimum, suhu maksimum, suhu rata-rata, kelembaban rata-rata, curah hujan, lama penyinaran, kecepatan angin rata-rata, kecepatan angin terbesar, dan arah angin. Penelitian tersebut juga menyarankan untuk menggunakan data cuaca yang lebih banyak agar memperoleh hasil akurasi lebih baik atau menggunakan metode klasifikasi lainnya [8].

Penelitian berikutnya mengenai prakiraan cuaca dengan menggunakan optimasi metode *Ensemble Learning* pada *Naive Bayes* dan C4.5, menghasilkan akurasi sebesar 49,76% pada metode *Naive Bayes* dan akurasi sebesar 45,76% pada metode algoritma C4.5. Penelitian tersebut menyatakan bahwa atribut pada dataset yang digunakan memiliki nilai yang hampir sama untuk setiap kategori cuaca. Oleh karena itu sulit untuk membuat model bahkan setelah optimasi menggunakan metode *Ensemble Learning*. Penelitian tersebut menyarankan untuk penelitian selanjutnya dapat menambah basis pengetahuan para ahli untuk memperoleh hasil akurasi yang lebih baik [9].

Penelitian berikutnya mengenai prediksi cuaca dengan metode *Ensemble Learning* memperoleh hasil akurasi adalah 81.21%. *Ensemble Learning* menggunakan pembelajaran yang terdiri dari *Naive bayes*, *Generalized Linear Model*, *Deep Learning*, *Decision Tree* dan *Random Forest*. Adapun atribut-atribut penelitian

berupa *Date*, Tanggal observasi, *Location*, *MinTemp*, *MaxTemp*, *Rainfall*, *Evaporation*, *Sunshine*, *WindGustDir*, *WindGustSpeed*, *WindDir9am*, *WindDir3pm*, *WindSpeed9am*, *WindSpeed3pm*, *Humidity9am*, *Humidity3pm*, *Pressure9am*, *Pressure3pm*, *Cloud9am*, *Cloud3pm*, *Temp9am*, *Temp3pm*, *RainToday*, *RISK_MM*, *RainTomorrow*. Penelitian tersebut membahas metode *Ensemble Learning* dengan banyak karakteristik metode yang berbeda-beda menyebabkan hasil akurasi metode *Ensemble Learning* lebih rendah dibandingkan beberapa metode lainnya. Penelitian tersebut menyarankan penelitian berikut bisa menggunakan algoritma klasifikasi lainnya [4].

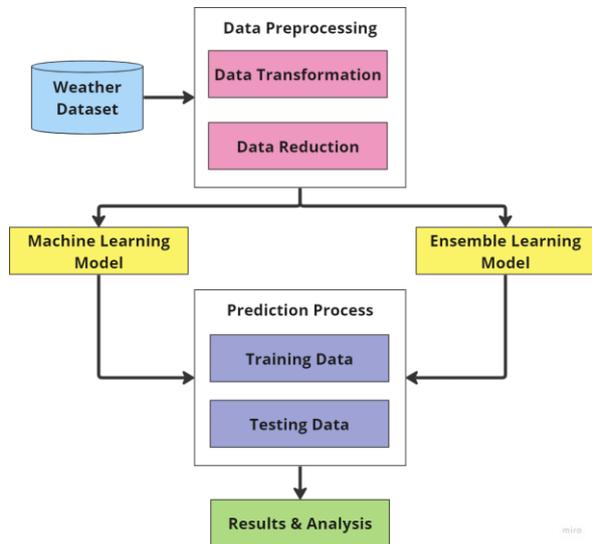
Penelitian berikutnya mengenai klasifikasi dan prediksi pada diabetes mellitus yang menggunakan pendekatan metode *Ensemble Learning* dengan *Soft Voting Classifier*. Peneliti telah mengusulkan model pengklasifikasi *Soft voting Classifier* dengan tiga algoritma pembelajaran mesin seperti *Random Forest*, *Logistic Regression*, dan *Naive Bayes*. Akurasi yang diperoleh yaitu 79,08% pada data diabetes Indian Pima dan 97,02% pada data kanker payudara [10].

Penelitian mengenai efek dari *Data Scaling Methods* pada algoritma *Machine Learning* menggunakan 11 algoritma *Machine Learning* yaitu *Logistic Regression (LR)*, *Linear Discriminant Analysis (LDA)*, *K-Nearest Neighbors (KNN)*, *Classification and Regression Trees (CART)*, *Naive Bayes (NB)*, *Support Vector Machine (SVM)*, *XGBoost (XGB)*, *Random Forest Classifier (RF)*, *Gradient Boost (GB)*, *AdaBoost (AB)*, *Extra Tree Classifier (ET)*. Penelitian tersebut menerapkan 6 *Data Scaling Methods* yaitu *Normalization (NR)*, *Standscale (SS)*, *MinMax (MM)*, *MaxAbs (MA)*, *Robust Scaler (RS)*, dan *Quantile Transformer (QT)*. Hasil akurasi terbaik diperoleh menggunakan algoritma *CART* dengan metode *Robust Scaler* dan *Quantum Transformer* dengan akurasi 100%, presisi 100%, *recall* 99% dan *F1 score* 100%. Hasil penelitian menunjukkan bahwa performa model bervariasi bergantung pada metode penskalaan data yang digunakan [11].

Berdasarkan temuan penelitian sebelumnya yang menerapkan pendekatan *Ensemble Learning*, penelitian ini bertujuan untuk mengevaluasi efektivitas teknik *Soft Voting Classifier* dalam meningkatkan akurasi prakiraan cuaca. Fokus penelitian ini adalah pada penggunaan *ensemble learning* yang menggabungkan beberapa algoritma, dengan harapan bahwa pengintegrasian algoritma-algoritma ini akan menghasilkan tingkat akurasi yang lebih tinggi dibandingkan dengan penggunaan algoritma tunggal dalam prediksi cuaca. Penelitian ini mengusulkan pengembangan model prakiraan cuaca dengan pendekatan metode *Ensemble Learning* dengan menggunakan beberapa metode *machine learning* tunggal, seperti *K-Nearest Neighbour (KNN)*, *Logistic Regression*, *Random Forest*, *Naive Bayes* dan *Multi Layer Perceptron* berdasarkan fitur-

fitur yang mendukung seperti suhu udara, titik embun, kelembaban udara, kecepatan angin, hembusan angin dan tekanan udara.

2. Metode Penelitian



Gambar 1. Metode Penelitian

Metode penelitian merupakan proses atau cara ilmiah untuk mendapatkan data yang akan digunakan untuk keperluan penelitian [12]. Pada penelitian ini menggunakan metode *Machine Learning* dan *Ensemble Learning* untuk prakiraan cuaca, dimana penelitian ini meliputi beberapa tahapan dimulai dari pengambilan data, data *preprocessing*, model *machine learning* atau model *Ensemble Learning*, proses prediksi data, dan analisis hasil, lihat Gambar 1.

2.1 Sumber Data

Penelitian ini menggunakan data cuaca yang diperoleh dari website <https://www.wunderground.com>. Data cuaca yang diambil adalah di mulai dari tanggal 1 September 2021 hingga 31 Agustus 2023. Untuk data cuaca diambil setiap 30 menit sekali. Jumlah data cuaca yang digunakan pada penelitian ini sebanyak 33.746 *record*. Data terdiri dari 11 atribut yaitu *Date*, *Time*, *Temperature*, *Dew Point*, *Humidity*, *Wind*, *Wind Speed*, *Wind Gust*, *Pressure*, *Precipitation*, dan *Condition*.

2.2 Pra-pemrosesan Data

Pra-pemrosesan data merupakan proses awal terhadap data masukan untuk menghasilkan data dalam format yang sesuai dan siap untuk diproses pada langkah berikutnya [13]. Dalam tahap ini, data disiapkan untuk pengolahan lebih lanjut dengan tujuan mencapai kinerja sistem yang optimal [14]. Setelah data diperoleh, data awal diperiksa melalui tahap pra-pemrosesan data untuk melakukan validasi dan membersihkan sumber data yang tidak lengkap, seperti data yang hilang atau tidak valid.

2.2.1 Transformasi Data

Penelitian ini melakukan tahap transformasi data yang bertujuan untuk mengubah sejumlah atribut yang awalnya tidak bersifat numerik menjadi format yang lebih sesuai untuk analisis. Transformasi ini melibatkan konversi satuan pada atribut *Temperature*, *Dew Point*, *Humidity*, *Wind Speed*, *Wind Gust*, *Pressure*, dan *Condition*. Pertama, atribut *Temperature* dan *Dew Point* dikonversi ke dalam satuan derajat *Fahrenheit*, sedangkan atribut *Humidity* dikonversi ke dalam bilangan desimal untuk mempermudah representasi nilai kelembaban.

Selanjutnya, atribut *Wind Speed* dan *Wind Gust* diubah ke dalam satuan kilometer per jam (km/h) agar memiliki kesesuaian satuan. Atribut *Pressure* dikonversi ke dalam satuan inchi (in) untuk representasi yang seragam. Atribut kelas *Condition* mengalami transformasi nilai nominal menjadi kategorikal untuk memudahkan identifikasi dan analisis dalam kaitannya dengan kondisi cuaca yang mendasar. Langkah transformasi ini bertujuan untuk menghadirkan konsistensi dalam representasi data dan mempersiapkan *dataset* agar lebih mudah dipahami serta digunakan dalam proses analisis lebih lanjut. Tabel 1 menjelaskan kelas *Condition*.

Tabel 1. Transformasi Kelas Cuaca

Kondisi Cuaca	Pelabelan Kategorikal
Smoke	0
Mist	1
Haze	2
Fog	3
Fair	4
Cloudy	5
Partly Cloudy	6
Mostly Cloudy	7
Thunder in the Vicinity	8
Thunder	9
Light Rain	10
Light Rain with Thunder	11
Rain	12
Heavy Rain	13
T-Storm	14
Heavy T-Storm	15

2.2.2 Reduksi Data

Dalam upaya untuk menyederhanakan dan mengoptimalkan dataset, penelitian ini melakukan tahap reduksi data yang bertujuan untuk menghilangkan atribut yang dianggap tidak relevan atau kurang berpengaruh. Salah satu langkah yang diambil dalam tahap ini adalah menghapus atribut *Wind* dan *Precipitation*. Alasan penghilangan kedua atribut tersebut adalah nilai-nilai yang tercatat dalam atribut *Wind* dan *Precipitation* adalah 0 untuk semua entri data dalam dataset.

Dengan melakukan langkah ini, penelitian bertujuan untuk mengeliminasi atribut-atribut yang tidak menyumbang secara signifikan terhadap informasi yang relevan dan tidak memberikan kontribusi yang signifikan terhadap variabel target atau atribut kelas

yang menjadi fokus analisis. Tahap reduksi data ini diharapkan dapat membantu dalam mengoptimalkan dan mempersiapkan dataset untuk proses analisis yang lebih akurat dan relevan.

2.3 Machine Learning Method

Prakiraan cuaca memiliki peran penting dalam mendukung berbagai aspek kehidupan sehari-hari, dan penggunaan teknologi machine learning diharapkan dapat meningkatkan akurasi prediksi serta membantu mengatasi kompleksitas dari data cuaca yang terus berkembang.

Pada bagian ini akan dilakukan eksplorasi menyeluruh terkait penelitian prakiraan cuaca dengan menerapkan metode *machine learning*. Penelitian ini menggunakan empat algoritma *machine learning*, yakni *K-Nearest Neighbors* (KNN), *Logistic Regression*, *Random Forest*, dan *Naive Bayes*.

2.3.1. K-Nearest Neighbour (KNN)

K-Nearest Neighbour merupakan algoritma klasifikasi paling sederhana dan paling umum digunakan dalam pembelajaran mesin. *K-Nearest Neighbor* adalah jenis klasifikasi yang dikenal juga sebagai *lazy learner classifier* yang berdasarkan pada pembelajaran melalui kesamaan, yaitu dengan membandingkan karakteristik uji yang diberikan dengan atribut-atribut pelatihan yang mirip dengan itu [15]. Dalam algoritma KNN, "K" mengacu pada jumlah tetangga terdekat yang dipertimbangkan oleh algoritma untuk membuat prediksi. KNN akan memberikan tingkat keakuratan yang tinggi jika nilai K yang dipilih sesuai. Ini disebabkan oleh fakta bahwa KNN mengukur jarak terpendek antara sampel uji dan sampel pelatihan tanpa mempertimbangkan distribusi probabilitas dari masing-masing kelas [16]. Berikut adalah algoritma KNN yang digunakan:

Algoritma KNN

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
```

```
modelKNN = KNeighborsClassifier() //parameter default
# Latih model
modelKNN.fit(xLatih, yLatih)
# Lakukan prediksi pada data uji
yPred = modelKNN.predict(xUji)
hasilAkurasi = accuracy_score(yUji, yPred)
```

Pada implementasi dengan menggunakan perpustakaan *scikit-learn* di pemrograman Python, digunakan pernyataan `from sklearn.neighbors import KNeighborsClassifier` untuk mengimpor kelas *KNeighborsClassifier* dari perpustakaan *scikit-learn*. Kelas tersebut berperan dalam menjalankan proses klasifikasi dengan menggunakan metode *K-Nearest Neighbors* (KNN). Pembentukan objek model KNN dilakukan dengan menggunakan parameter default, diwakili oleh deklarasi `modelKNN = KNeighborsClassifier()`. Proses pelatihan model KNN

dilakukan dengan memanfaatkan data latih, yang terdiri dari fitur (disebut xLatih) dan label (disebut yLatih), melalui pemanggilan metode `.fit(xLatih, yLatih)`. Prediksi pada data uji (xUji) dilakukan dengan menggunakan model yang telah dilatih, melibatkan penggunaan metode `.predict(xUji)`. Evaluasi akurasi model dilakukan dengan membandingkan hasil prediksi (yPred) dengan label yang sebenarnya dari data uji (yUji). Untuk mengukur akurasi tersebut, digunakan fungsi `accuracy_score(yUji, yPred)` dari perpustakaan *scikit-learn*.

2.3.2. Logistic Regression

Logistic Regression adalah bagian dari metode penambangan data yang digunakan untuk menganalisis data yang menggambarkan suatu variabel respon (terikat) atau beberapa variabel prediktor [13]. *Logistic Regression* menggunakan fungsi matematika yang disebut fungsi sigmoid untuk memperkirakan probabilitas bahwa variabel terikat mengambil nilai tertentu [17]. Tujuan utama dari *Logistic Regression* adalah untuk memprediksi kemungkinan terjadinya suatu peristiwa dengan menghubungkannya dengan satu atau lebih variabel prediktor. Berikut algoritma *Logistic Regression*:

Algoritma Logistic Regression

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

```
modelLR = LogisticRegression(random_state=0) //parameter default
# Latih model
modelLR.fit(xLatih, yLatih)
# Lakukan prediksi pada data uji
yPred = modelLR.predict(xUji)
```

```
hasilAkurasi = accuracy_score(yUji, yPred)
```

Dengan mengimport kelas *LogisticRegression* dan `accuracy_score` dari perpustakaan *scikit-learn*, langkah pertama melibatkan deklarasi objek model regresi logistik menggunakan pernyataan `modelLR = LogisticRegression(random_state=0)`. Proses latihan model dilakukan dengan memanfaatkan data latih, yang terdiri dari fitur (xLatih) dan label (yLatih), melalui pemanggilan metode `.fit(xLatih, yLatih)`. Setelah model terlatih, langkah berikutnya adalah melakukan prediksi pada data uji (xUji) menggunakan model tersebut, diwakili oleh pernyataan `yPred = modelLR.predict(xUji)`. Untuk mengukur akurasi model, digunakan fungsi `accuracy_score(yUji, yPred)` dari *scikit-learn*.

2.3.3. Random Forest

Random Forest adalah sebuah algoritma machine learning yang digunakan untuk melakukan klasifikasi dan prediksi. Algoritma *Random Forest* dapat digunakan untuk mengklasifikasikan dataset yang besar. Kelebihan dari *Random Forest* terletak pada kemampuannya untuk menggabungkan banyak pohon, berbeda dengan pendekatan satu pohon dalam

melakukan klasifikasi dan meramalkan kelas [18]. *Random Forest* dapat mengurangi variansi sehingga membantu mencegah overfitting [19]. Dalam langkah pertama, dilakukan impor kelas *RandomForestClassifier* dan fungsi `accuracy_score` dari *scikit-learn* menggunakan pernyataan `from sklearn.ensemble import RandomForestClassifier` dan `from sklearn.metrics import accuracy_score`. Pembentukan objek model *Random Forest*, ditandai dengan deklarasi `modelRF = RandomForestClassifier(random_state=0)`, dilakukan dengan menggunakan parameter default. Proses pelatihan model dilaksanakan dengan memanfaatkan data latih (xLatih sebagai fitur dan yLatih sebagai label) melalui pemanggilan metode `.fit(xLatih, yLatih)`. Setelah pelatihan selesai, model *Random Forest* dapat digunakan untuk melakukan prediksi pada data uji (xUji) melalui metode `.predict(xUji)`. Evaluasi akurasi model dilakukan dengan membandingkan hasil prediksi (yPred) dengan label sebenarnya dari data uji (yUji) menggunakan fungsi `accuracy_score(yUji, yPred)`. Berikut adalah algoritma *Random Forest* yang digunakan:

Algoritma Random Forest

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
```

```
modelRF = RandomForestClassifier(random_state=0) //parameter default
```

```
# Latih model
modelRF.fit(xLatih, yLatih)
# Lakukan prediksi pada data uji
yPred = modelRF.predict(xUji)
```

```
hasilAkurasi = accuracy_score(yUji, yPred)
```

2.3.4. Naive Bayes

Algoritma *Naive Bayes* adalah suatu algoritma *Machine Learning* yang digunakan untuk tugas klasifikasi pada kumpulan data berdimensi tinggi. Algoritma klasifikasi *Naive Bayes* ini merupakan suatu metode pengklasifikasi kumpulan data statistika yang bertujuan untuk memprediksi probabilitas dari setiap anggota suatu kelas dalam kumpulan data [20]. Algoritma *Naive Bayes* menunjukkan keunggulan sebagai algoritma yang dapat mengelola kerumitan dataset yang melibatkan berbagai fitur melalui pelaksanaan analisis probabilitas yang teliti. Algoritma ini terbukti sangat bermanfaat dalam berbagai situasi, seperti klasifikasi teks, pengenalan pola, dan penggunaan pada aplikasi lain yang melibatkan beragam jenis data. Dibandingkan dengan banyak teknik klasifikasi lainnya, algoritma ini sederhana, namun memiliki kinerja yang lebih baik [21]. Berikut algoritma *Naive Bayes*:

Algoritma Naive Bayes

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

```
modelNB = GaussianNB() //parameter default
# Latih model
modelNB.fit(xLatih, yLatih)
```

```
# Lakukan prediksi pada data uji
yPred= modelNB.predict(xUji)

hasilAkurasi = accuracy_score(yUji, yPred)
```

Dalam implementasi menggunakan perpustakaan scikit-learn di Python, langkah pertama adalah mengimpor kelas GaussianNB dari modul naive_bayes. Pembentukan objek model Naive Bayes dilakukan dengan mendeklarasikan modelNB = GaussianNB() menggunakan parameter default. Proses pelatihan model dilakukan dengan menggunakan data latih (xLatih sebagai fitur dan yLatih sebagai label) melalui pemanggilan metode .fit(xLatih, yLatih). Setelah model dilatih, langkah selanjutnya adalah melakukan prediksi pada data uji (xUji) menggunakan metode .predict(xUji). Akurasi model dievaluasi dengan membandingkan hasil prediksi (yPred) dengan label yang sebenarnya dari data uji (yUji), menggunakan fungsi accuracy_score(yUji, yPred) dari modul metrics di scikit-learn.

2.3.5. Multilayer Perceptron

Multilayer Perceptron (MLP) adalah salah satu metode dalam jaringan syaraf tiruan yang luas digunakan. MLP merupakan salah satu jenis dari *feed-forward neural network* yang memiliki satu atau lebih lapisan tersembunyi. Pada umumnya, MLP terdiri dari lapisan input yang terdiri dari neuron-neuron yang digunakan untuk memasukkan data [22]. MLP merupakan bentuk algoritma jaringan saraf yang memiliki kemampuan fleksibilitas dan dapat diterapkan dalam berbagai konteks aplikasi, seperti pengenalan gambar, pemrosesan teks, pemodelan data berdasarkan waktu, dan sejumlah tugas lainnya. Berikut adalah algoritma MLP yang digunakan:

Algoritma Multi Layer Perceptron

```
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
```

```
modelMLP= MLPClassifier(random_state=0)
# Melatih model
modelMLP.fit(xLatih, yLatih)
# Membuat prediksi
yPred= modelMLP.predict(xUji)

hasilAkurasi = accuracy_score(yUji, yPred)
```

Dalam implementasinya dengan menggunakan perpustakaan scikit-learn di Python, kelas MLPClassifier dari modul sklearn.neural_network diimpor menggunakan pernyataan from sklearn.neural_network import MLPClassifier. Pada tahap awal, objek model MLP dibentuk dengan menerapkan konstruktor MLPClassifier(random_state=0), dengan penggunaan parameter random_state untuk menjaga reproduktibilitas hasil. Pelatihan model dilakukan dengan memanfaatkan data latih, yang terdiri dari fitur (disebut xLatih) dan label (disebut yLatih), melalui metode .fit(xLatih, yLatih). Setelah pelatihan selesai, model dapat digunakan untuk membuat prediksi pada data uji (xUji) dengan menggunakan metode

.predict(xUji). Untuk mengukur kinerja model, digunakan metode akurasi yang dihitung dengan membandingkan hasil prediksi (yPred) dengan label sebenarnya dari data uji (yUji) menggunakan fungsi accuracy_score(yUji, yPred) dari modul sklearn.metrics.

2.4. Ensemble Method using Soft Voting Classifier

Ensemble classification merupakan metode pembelajaran mesin yang mengumpulkan hasil prediksi dari beberapa metode klasifikasi tunggal untuk membentuk prediksi yang lebih akurat dan stabil [23]. Teknik klasifikasi dengan *Ensemble Method using Soft Voting Classifier* mengacu pada pendekatan teknik *Ensemble Learning* di mana model yang terlibat membuat prediksi dalam bentuk probabilitas atau skor keanggotaan pada kelas tertentu, bukan hanya label kelas biner. Tujuan utama dari penerapan *Soft Voting Classifier* adalah untuk mengurangi bias dan varians sehingga menghasilkan prediksi yang lebih baik [24]. Berikut algoritma *Ensemble Method using Soft Voting Classifier*:

Algoritma Ensemble Method using Soft Voting Classifier

```
from sklearn.ensemble import VotingClassifier
from sklearn.metrics import accuracy_score
```

```
model1 = algoritma1()
model2 = algoritma2()
model3 = algoritma3()
```

```
modelEnsemble= VotingClassifier (estimators = [('algoritma1',
model1), ('algoritma2', model2), ('algoritma3',model3)],
voting='soft')
```

```
# Melatih model
modelEnsemble.fit(xLatih, yLatih)
# Membuat prediksi
yPred= modelEnsemble.predict(xUji)
```

```
hasilAkurasi = accuracy_score(yUji, yPred)
```

Algoritma *Ensemble Method* menggabungkan tiga model *machine learning* yang berbeda (model1, model2, dan model3) ke dalam satu entitas *ensemble* dengan metode *voting 'soft'*. Langkah pertama adalah membuat tiga model *machine learning* yang berbeda untuk kemudian digunakan dalam *ensemble*. Selanjutnya, model-model ini digabungkan menggunakan *Voting Classifier*, di mana setiap model diidentifikasi dengan nama 'algoritma1', 'algoritma2', dan 'algoritma3' dan disertakan dalam estimators bersama dengan *instance* model masing-masing. *Ensemble* ini dilatih menggunakan data latih (xLatih, yLatih) dan kemudian digunakan untuk membuat prediksi pada data uji (xUji). Akurasi prediksi dari model *ensemble* diukur menggunakan metrik akurasi dengan membandingkan hasil prediksi (yPred) dengan nilai sebenarnya dari data uji (yUji) menggunakan fungsi accuracy_score() dari scikit-learn.

3. Hasil dan Pembahasan

3.1 Hasil

Penelitian yang dilakukan dimulai dengan melibatkan tahap pra-pemrosesan data untuk beberapa atribut. Proses transformasi data yang dilakukan antara lain berupa konversi nilai nominal menjadi nilai kategorikal, konversi nilai menjadi satuan lain, dan konversi penyamaan satuan yang berbeda pada dataset. Setelah proses transformasi data, maka tahap selanjutnya adalah tahap reduksi data. Pada tahap ini dilakukan penghilangan atribut yang memiliki nilai 0 untuk semua data di dalamnya. Pada awalnya, *dataset* terdiri dari 11 atribut yaitu *date*, *time*, *temperature*, *dew point*, *humidity*, *wind*, *wind speed*, *wind gust*, *pressure*, *precipitation*, *condition*. Pada proses pelatihan dan pengujian, ada atribut yang dihilangkan dikarenakan semua data bernilai 0 yaitu atribut *wind* dan atribut *precipitation*. Atribut lain yang tidak digunakan adalah atribut *date* dan atribut *time*. Dari tahap reduksi data, *dataset* yang digunakan menjadi 7 atribut dengan 1 atribut sebagai kelas.

Tahap selanjutnya dalam penelitian ini adalah tahap pemodelan pada metode *Machine Learning* dan *Ensemble Method*. Untuk metode *Machine Learning* tunggal (KNN, *Logistic Regression*, *RandomForest*, *Naive Bayes* dan *Multi Layer Perceptron*) menggunakan parameter default yang ada pada *library Scikit Learn*. Pada *Ensemble Method* disusun menjadi 3 metode setiap kombinasi. Dikarenakan metode *Machine Learning* tunggal yang digunakan pada penelitian ini ada 5 metode sehingga untuk kombinasi per 3 metode dalam *Ensemble Method* sebanyak 10 kombinasi diantaranya seperti terlihat pada Tabel 2.

Tabel 2. Kombinasi metode pada *Ensemble Learning*

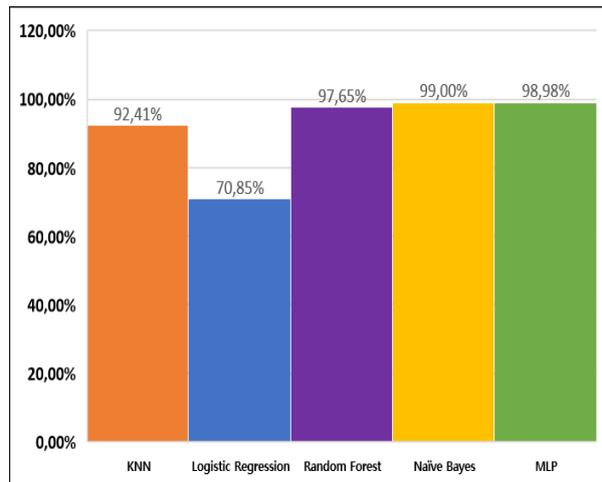
Ensemble Method dengan 3 Metode
<i>Logistic Regression</i> , <i>Random Forest</i> , <i>Naive Bayes</i> (LRN)
<i>Random Forest</i> , MLP, <i>Naive Bayes</i> (RMN)
<i>Random Forest</i> , MLP, <i>Logistic Regression</i> (RML)
<i>Random Forest</i> , <i>Logistic Regression</i> , KNN (RLK)
<i>Logistic Regression</i> , MLP, <i>Naive Bayes</i> (LMN)
<i>Logistic Regression</i> , MLP, KNN (LMK)
<i>Random Forest</i> , KNN, <i>Naive Bayes</i> (RKN)
<i>Random Forest</i> , MLP, KNN (RMK)
KNN, MLP, <i>Naive Bayes</i> (KMN)
KNN, <i>Logistic Regression</i> , <i>Naive Bayes</i> (KLN)

Langkah selanjutnya dalam penelitian ini adalah memisahkan data menjadi data latih dan data uji dengan proporsi 70:30, dengan tujuan untuk mencegah *overfitting* dan menguji kemampuan model dalam memperkirakan cuaca yang akan datang. Langkah selanjutnya melakukan proses pelatihan data dengan parameter setiap metode yang telah diatur secara default. Setelah proses pelatihan selesai, maka dilakukan proses pengujian untuk memperoleh akurasi. Pengukuran akurasi yang dilakukan menggunakan *library accuracy_score* yang terdapat pada *Scikit Learn*. Tabel 3 adalah hasil yang diperoleh dalam proses pengujian.

Tabel 3. Hasil Pengujian pada *Machine Learning Method*

Machine Learning Method	Accuracy
KNN	92,41 %
<i>Logistic Regression</i>	70,85 %
<i>Random Forest</i>	97,65 %
<i>Naive Bayes</i>	99,00 %
MLP	98,98 %

Berdasarkan hasil dari beberapa metode *Machine Learning*, *Naive Bayes* memiliki tingkat akurasi tertinggi yaitu sebesar 99,00%. Sedangkan akurasi terendah ada pada metode *Logistic Regression* yaitu sebesar 70,85%. Pada metode KNN, *Random Forest* dan MLP memiliki akurasi yang sangat baik. Lihat Gambar 2.

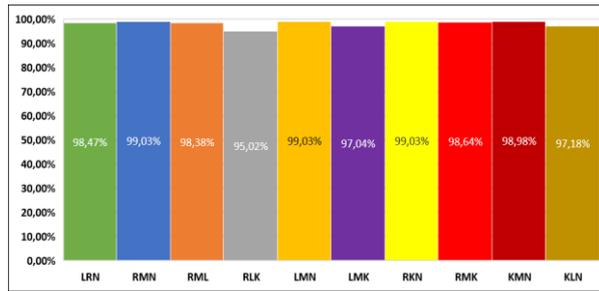


Gambar 2. Grafik Hasil Akurasi pada *Machine Learning Method*

Setelah pengujian dilakukan pada metode *Machine Learning* tunggal, langkah selanjutnya melakukan pengujian pada beberapa kombinasi metode pada *Ensemble Method*. Berdasarkan tabel hasil pengujian pada *Ensemble Method*, terdapat 3 kombinasi *Ensemble Method* yang memiliki akurasi tertinggi yaitu pada metode *Soft Voting Classifier (Random Forest, MLP, Naive Bayes)*, *Soft Voting Classifier (Logistic Regression, MLP, Naive Bayes)*, dan *Soft Voting Classifier (RandomForest, KNN, Naive Bayes)* dengan akurasi sebesar 99,03%. Akurasi terendah terdapat pada metode *Soft Voting Classifier (Random Forest, Logistic Regression, KNN)* dengan akurasi sebesar 95,02%. Tabel 4 adalah hasil yang diperoleh dalam proses pengujian. Lihat Gambar 3.

Tabel 4. Hasil Pengujian pada *Ensemble Learning Method*

Ensemble Method	Accuracy
<i>Soft Voting Classifier (LRN)</i>	98,47 %
<i>Soft Voting Classifier (RMN)</i>	99,03 %
<i>Soft Voting Classifier (RML)</i>	98,38 %
<i>Soft Voting Classifier (RLK)</i>	95,02 %
<i>Soft Voting Classifier (LMN)</i>	99,03 %
<i>Soft Voting Classifier (LMK)</i>	97,04 %
<i>Soft Voting Classifier (RKN)</i>	99,03 %
<i>Soft Voting Classifier (RMK)</i>	98,64 %
<i>Soft Voting Classifier (KMN)</i>	98,98 %
<i>Soft Voting Classifier (KLN)</i>	97,18 %



Gambar 3. Grafik Hasil Akurasi pada *Ensemble Learning Method*

3.2 Pembahasan

Berdasarkan hasil penelitian yang telah dipaparkan, terdapat 5 metode *Machine Learning*. KNN memperoleh akurasi tinggi dengan parameter default. Berdasarkan karakteristik model KNN, diperolehnya akurasi tinggi dikarenakan kesesuaian *dataset* terhadap *dataset* yang digunakan. Pada metode *Logistic Regression* memiliki akurasi lebih rendah dibandingkan metode lain. Lebih rendahnya akurasi pada metode ini dibandingkan metode lain disebabkan oleh hubungan antara variabel dependen dan independen tidak linier, maka model *Logistic Regression* mungkin tidak mampu menangkap pola yang kompleks. Akurasi tinggi pada metode *Random Forest* dikarenakan agregasi dari banyak pohon keputusan (*decision trees*) sehingga cenderung mengurangi varians dan kecenderungan *overfitting*. Metode *Naive Bayes* memperoleh akurasi terbaik karena pada model *Naive Bayes* yang relatif sederhana dan efisien secara komputasi cocok untuk digunakan pada *dataset* yang besar dan jumlah fitur yang besar juga. Metode MLP mempelajari representasi yang sangat kompleks dari data karena kemampuannya untuk memiliki kumpulan *hidden layer*. Melalui *hidden layer* tersebut, MLP dapat menangkap pola yang lebih kompleks dan abstrak pada data, sehingga dapat meningkatkan kemampuan model dalam melakukan klasifikasi dengan akurasi tinggi.

Berdasarkan hasil penelitian pada *Ensemble Method*, terdapat 10 kombinasi metode. Dengan penggabungan beberapa model *Machine Learning* yang berbeda, *ensemble method using Soft Voting Classifier* dapat mengurangi varian dan mencegah *overfitting* pada model *Machine Learning* tunggal. Pemilihan model *Machine Learning* dasar yang kuat menjadikan salah satu peran penting dalam meningkatkan kinerja *ensemble method using Soft Voting Classifier* dan mencapai akurasi prediksi yang tinggi. Hasil pengujian yang diperoleh menunjukkan bahwa meskipun *Ensemble Method* memberikan tingkat akurasi yang baik, namun peningkatannya tidak signifikan dibandingkan dengan pendekatan *Machine Learning* tunggal untuk metode *Naive Bayes*.

Dari 10 kombinasi pada *Ensemble Method*, terdapat 3 kombinasi terbaik yaitu pada metode *Soft Voting Classifier (Random Forest, MLP, Naive Bayes)*, *Soft Voting Classifier (Logistic Regression, MLP, Naive*

Bayes), dan *Soft Voting Classifier (Random Forest, KNN, Naive Bayes)*. Tiga kombinasi dalam metode *Ensemble Learning* menampilkan satu model *Machine Learning* tunggal, yaitu *Naive Bayes*, yang secara konsisten terdapat dalam setiap kombinasi tersebut. Dalam metode *Machine Learning* tunggal, *Naive Bayes* juga menghasilkan akurasi tinggi, sehingga dapat disimpulkan bahwa *Naive Bayes* menunjukkan korelasi yang signifikan antara model *Machine Learning* tunggal dan model *Ensemble Learning*.

4. Kesimpulan

Berdasarkan hasil penelitian, dapat disimpulkan bahwa beberapa metode *Machine Learning* tunggal cukup baik dalam prakiraan cuaca menggunakan parameter default dengan akurasi tinggi sebesar 99,00% pada metode *Naive Bayes* dan akurasi terendah sebesar 70,85% pada metode *Logistic Regression*. Pada *Ensemble Method* dengan kombinasi 3 metode juga memiliki akurasi yang sangat baik untuk semua kombinasinya. Kombinasi metode terbaik adalah pada metode *Soft Voting Classifier (Random Forest, MLP, Naive Bayes)*, *Soft Voting Classifier (Logistic Regression, MLP, Naive Bayes)*, dan *Soft Voting Classifier (Random Forest, KNN, Naive Bayes)* dengan akurasi sebesar 99,03%. Akurasi terendah terdapat pada metode *Soft Voting Classifier (Random Forest, Logistic Regression, KNN)* dengan akurasi sebesar 95,02%. Walaupun demikian, semua kombinasi metode pada *Ensemble Method* memiliki akurasi yang baik. Dari tingginya angka akurasi membuktikan bahwa *Ensemble Method* sangat baik dalam penelitian prakiraan cuaca.

Daftar Rujukan

- [1] D. Desmonda, T. Tursina, and M. A. Irwansyah, "Prediksi Besaran Curah Hujan Menggunakan Metode Fuzzy Time Series," *Jurnal Sistem dan Teknologi Informasi (JUSTIN)*, vol. 6, no. 4, 2018, doi: 10.26418/justin.v6i4.27036.
- [2] A. R. I. Pratama, S. A. Latipah, and B. N. Sari, "Optimasi Klasifikasi Curah Hujan Menggunakan Support Vector Machine (SVM) dan Recursive Feature Elimination (RFE)," *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 7, no. 2, pp. 314–324, May 2022, doi: 10.29100/jupi.v7i2.2675.
- [3] G. I. Merdekawati and Ismail, "Prediksi Curah Hujan di Jakarta Berbasis Algoritma Levenberg Marquardt," *Jurnal Ilmiah Informatika Komputer*, vol. 24, no. 2, pp. 116–128, 2019, doi: 10.35760/ik.2019.v24i2.2366.
- [4] A. M. Siregar, "Klasifikasi Untuk Prediksi Cuaca Menggunakan Ensemble Learning," *PETIR*, vol. 13, no. 2, 2020, doi: 10.33322/petir.v13i2.998.
- [5] A. Roy and A. MS. Hendriyawan, "Implementasi Metode Naive Bayes Classifier Untuk Perkiraan Cuaca," 2019.
- [6] I. Intan, S. Aminah Dinayati Ghani, A. T. Koswara, U. Dipa Makassar, K. Arsip Nasional Republik Indonesia, and J. P. Kemerdekaan, "Performance Analysis of Weather Forecasting using Machine Learning Algorithms (Analisis Performansi Prakiraan Cuaca Menggunakan Algoritma Machine Learning)," *Jurnal Pekommas*, vol. 6, no. 2, 2021, Accessed: Oct. 19, 2023. [Online]. Available: <http://dx.doi.org/10.30818/jpkm.2021.206022>
- [7] F. Hamami and I. A. Dahlan, "Klasifikasi Cuaca Provinsi DKI Jakarta Menggunakan Algoritma Random Forest dengan Teknik

- Oversampling,” *Jurnal Teknoinfo*, vol. 16, no. 1, 2022, doi: 10.33365/jti.v16i1.1533.
- [8] Utami, D. P. Rini, and E. & Lestari, “Prediksi Cuaca di Kota Palembang Berbasis Supervised Learning Menggunakan Algoritma K-Nearest Neighbour,” *JUPITER: Jurnal Penelitian Ilmu Dan Teknologi Komputer*, vol. 13, no. 1, 2021.
- [9] V. I. Yani, A. Aradea, and H. Mubarak, “Optimasi Prakiraan Cuaca Menggunakan Metode Ensemble pada Naïve Bayes dan C4.5,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 8, no. 3, Dec. 2022, doi: 10.28932/jutisi.v8i3.5455.
- [10] S. Kumari, D. Kumar, and M. Mittal, “An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier,” *International Journal of Cognitive Computing in Engineering*, vol. 2, 2021, doi: 10.1016/j.ijcce.2021.01.001.
- [11] M. M. Ahsan, M. A. P. Mahmud, P. K. Saha, K. D. Gupta, and Z. Siddique, “Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance,” *Technologies (Basel)*, vol. 9, no. 3, 2021, doi: 10.3390/technologies9030052.
- [12] Alvina Felicia Watratan, Arwini Puspita. B, and Dikwan Moeis, “Implementasi Algoritma Naive Bayes Untuk Memprediksi Tingkat Penyebaran Covid-19 Di Indonesia,” *Journal of Applied Computer Science and Technology*, vol. 1, no. 1, pp. 7–14, Jul. 2020, doi: 10.52158/jacost.v1i1.9.
- [13] F. R. Suprihati, “Analisis Klasifikasi SMS Spam Menggunakan Logistic Regression,” *Jurnal Sistem Cerdas*, vol. 4, no. 3, 2021, doi: 10.37396/jsc.v4i3.166.
- [14] S. Sudianto, A. D. Sripamuji, I. Ramadhanti, R. R. Amalia, J. Saputra, and B. Prihatnowo, “Penerapan Algoritma Support Vector Machine dan Multi-Layer Perceptron pada Klasifikasi Topik Berita,” *Jurnal Nasional Pendidikan Teknik Informatika : JANAPATI*, vol. 11, no. 2, pp. 84–91, 2022, [Online]. Available: <https://ejournal.undiksha.ac.id/index.php/janapati/article/view/44151>
- [15] K. Sumwiza, C. Twizere, G. Rushingabigwi, P. Bakunzibake, and P. Bamurigire, “Enhanced cardiovascular disease prediction model using random forest algorithm,” *Inform Med Unlocked*, vol. 41, 2023, doi: 10.1016/j.imu.2023.101316.
- [16] O. Saeful Bachri and R. M. Herdian Bhakti, “Penentuan Status Stunting pada Anak dengan Menggunakan Algoritma KNN,” *Jurnal Ilmiah Intech: Information Technology Journal of UMUS*, vol. 3, no. 02, pp. 130–137, Nov. 2021, doi: 10.46772/intech.v3i02.533.
- [17] O. Iparraguirre-Villanueva, K. Espinola-Linares, R. O. Flores Castañeda, and M. Cabanillas-Carbonell, “Application of Machine Learning Models for Early Detection and Accurate Classification of Type 2 Diabetes,” *Diagnostics*, vol. 13, no. 14, 2023, doi: 10.3390/diagnostics13142383.
- [18] B. Prasajo and E. Haryatmi, “Analisa Prediksi Kelayakan Pemberian Kredit Pinjaman dengan Metode Random Forest,” *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 7, no. 2, pp. 79–89, Sep. 2021, doi: 10.25077/teknosi.v7i2.2021.79-89.
- [19] N. M. Lutimath, N. Sharma, and B. K. Byregowda, “Prediction of Heart Disease using Biomedical Data through Machine Learning Techniques,” *EAI Endorsed Trans Pervasive Health Technol.*, vol. 7, no. 29, 2021, doi: 10.4108/eai.30-8-2021.170881.
- [20] A. Ridwan, “Penerapan Algoritma Naïve Bayes Untuk Klasifikasi Penyakit Diabetes Mellitus,” *Jurnal SISKOM-KB (Sistem Komputer dan Kecerdasan Buatan)*, vol. 4, no. 1, 2020, doi: 10.47970/siskom-kb.v4i1.169.
- [21] U. Nagavelli, D. Samanta, and P. Chakraborty, “Machine Learning Technology-Based Heart Disease Detection Models,” *J Healthc Eng*, vol. 2022, 2022, doi: 10.1155/2022/7351061.
- [22] M. S. Wibawa and I. M. D. Maysanjaya, “Multi Layer Perceptron dan Principal Component Analysis untuk Diagnosis Kanker Payudara,” *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 7, no. 1, p. 90, May 2018, doi: 10.23887/janapati.v7i1.12909.
- [23] G. Akilandasowmya, G. Nirmaladevi, S. U. Suganthi, and A. Aishwariya, “Skin cancer diagnosis: Leveraging deep hidden features and ensemble classifiers for early detection and classification,” *Biomed Signal Process Control*, 2023, doi: 10.1016/j.bspc.2023.105306.
- [24] N. Agustina and C. N. Ihsan, “Pendekatan Ensemble untuk Analisis Sentimen Covid19 Menggunakan Pengklasifikasi Soft Voting,” *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 10, no. 2, p. 263, Apr. 2023, doi: 10.25126/jtiik.20231026215.