

## ANALISIS MODEL MACHINE LEARNING UNTUK JENIS ASPAL DI JAWA BARAT MENGGUNAKAN ALGORITMA DECISION TREE DAN RANDOM FOREST

Miftah Raka Sujono<sup>1</sup>, Agus Bahtiar<sup>2</sup>, Bambang Irawan<sup>3</sup>

<sup>1</sup>Teknik Informatika, STMIK IKMI Cirebon

<sup>2</sup>Sistem Informasi, STMIK IKMI Cirebon

<sup>3</sup>Komputerisasi Akuntansi, STMIK IKMI Cirebon

Jalan Perjuangan No. 10 B Cirebon, Indonesia

*rakasujono123@gmail.com*

### ABSTRAK

Dalam hal mengolah data dan melakukan perintah tertentu, Machine Learning (ML) sangat berguna. Meskipun demikian, hasil dari model yang digunakan sangat bergantung pada proses data yang dipelajari mesin. Tujuan dari penelitian ini adalah untuk mengevaluasi kinerja model-model pembelajaran mesin yang digunakan dalam klasifikasi berbagai jenis aspal di Jawa Barat. Dalam rangka mencapai tujuan ini, digunakan algoritma Pohon Keputusan (Decision Tree) dan Hutan Acak (Random Forest). Data set yang digunakan terdiri dari 1.191 data yang dikumpulkan dari [opendata.jabarprov.go.id](https://opendata.jabarprov.go.id) dari tahun 2019 hingga 2022. Dengan menggunakan aplikasi Google Colab, analisis dilakukan berdasarkan tahapan Knowledge Discovery in Database (KDD). Hasil penelitian menunjukkan bahwa model yang menggunakan algoritma Random Forest memiliki akurasi sebesar 90,79% pada dataset yang mengandung aspal di Jawa Barat. Random Forest menunjukkan hasil yang sempurna 100% pada data pelatihan dan 90,79% pada data pengujian dalam evaluasi model. Untuk setiap kelas, metrik evaluasi presisi, recall, dan s F1-score juga menunjukkan hasil yang baik. Model Random Forest mendapatkan nilai presisi 91,41%, recall 97,31%, dan F1-score 94,27% untuk kelas HOTMIX, sementara model Decision Tree mendapatkan nilai presisi 91,01%, recall 92,47%, dan F1-score 91,73% untuk kelas HOTMIX. Model Random Forest juga mendapatkan hasil yang cukup baik, nilai presisi 72,00%, recall 67,92%, dan F1-score 94,27% untuk kelas HOTMIX. Hasil penelitian ini memberikan kontribusi penting dalam pengembangan model pembelajaran mesin untuk klasifikasi jenis aspal.

**Kata kunci:** *DecisionTree, RandomForest, Mechine Learning, KDD.*

### 1. PENDAHULUAN

Data mining telah menjadi alat penting dalam era digital saat ini, memungkinkan penemuan pola dan informasi berharga dari kumpulan data yang besar. Dalam konteks ini, berbagai teknik dan metode data mining telah dikembangkan, dan pemilihan algoritma yang tepat sangat penting untuk mencapai tujuan analisis yang diinginkan. Algoritma yang digunakan dalam data mining dapat bervariasi, dan pemilihan algoritma yang tepat sangat bergantung pada tujuan dan proses knowledge discovery in database (KDD) secara keseluruhan [1].

Jawa Barat, yang secara luas diakui sebagai salah satu pusat utama kemajuan ekonomi di Indonesia, telah menyaksikan lonjakan luar biasa di ranah industri konstruksi dan pemeliharaan infrastruktur jalan raya. Di tengah lanskap pengelolaan jalan raya yang terus berkembang, pemilihan varian aspal yang tepat secara cermat mengasumsikan pentingnya untuk memastikan tingkat kualitas dan daya tahan yang diinginkan dari infrastruktur penting tersebut dalam menghadapi segudang kesulitan lingkungan, mencakup tetapi tidak terbatas pada suhu ekstrim, curah hujan lebat, dan beban lalu lintas kendaraan berat yang tak henti-hentinya.

Penelitian ini bertujuan untuk mengembangkan model prediktif yang akurat menggunakan algoritma Decision Tree dan Random Forest untuk memprediksi kualitas dan kinerja jenis aspal yang digunakan di Jawa Barat. Dengan memanfaatkan dataset yang relevan, diharapkan model yang dikembangkan dapat mengidentifikasi pola-pola kompleks yang berkaitan dengan karakteristik aspal dan memberikan dampak signifikan dalam sektor infrastruktur dan industri konstruksi.

Metode yang digunakan dalam penelitian ini melibatkan pengumpulan data jenis aspal dari berbagai lokasi di Jawa Barat, pembersihan data untuk menghilangkan nilai yang hilang atau anomali, serta proses normalisasi dan pengkodean variabel kategorikal. Model Decision Tree akan dibangun dengan membagi data berdasarkan aturan keputusan yang dihasilkan dari atribut yang ada, sementara Random Forest akan menggabungkan hasil dari berbagai pohon keputusan yang dibangun secara acak untuk menghasilkan prediksi yang lebih akurat dan dapat diandalkan.

Dengan fokus pada analisis model pembelajaran mesin untuk jenis aspal di Jawa Barat menggunakan algoritma Decision Tree dan Random Forest, penelitian ini dapat memberikan kontribusi penting

dalam pengembangan infrastruktur yang lebih baik dan lebih tahan lama di wilayah Jawa barat.

## 2. TINJAUAN PUSTAKA

### 2.1. Penelitian Terdahulu

Studi sebelumnya menunjukkan bahwa Random Forest cenderung memberikan hasil yang lebih baik dibandingkan dengan Decision Tree. Misalnya, dalam penelitian yang dilakukan untuk memprediksi harga mobil, kedua metode ini digunakan dan hasilnya menunjukkan bahwa Random Forest memberikan prediksi yang lebih baik [2].

Dalam penelitian yang dilakukan oleh Pamuji & Ramadhan (2021), algoritma Random Forest menunjukkan tingkat akurasi yang lebih tinggi dalam prediksi keberhasilan pengobatan imunoterapi untuk penyakit kutil dibandingkan dengan Decision Tree, dengan tingkat akurasi 85,5% berbanding 84,4% [3].

Penelitian Selanjutnya oleh Fiqri dkk, (2020), membahas mengenai cara mendeteksi serangan atau keamanan siber menggunakan klasifier Decision Tree, Random Forest, dan AdaBoost. Dataset yang digunakan adalah dataset KDDcup99. Hasil klasifikasi menggunakan tiga algoritma Decision Tree menghasilkan akurasi rata-rata sebesar 97,05%, precision sebesar 96,41%, dan recall sebesar 89,53%. Random Forest menghasilkan Precision 99.68%, Recall 97.4%, dan Accuracy 97.7%, dan algoritma adabost menghasilkan Precision 99.8%, Recall 96.99%, dan Accuracy 97.47%, dari hasil tersebut menjelaskan hasil algoritma random forest adalah yang terbaik [4].

Pada penelitian lain membahas mengenai perbandingan algoritma K-Nearest Neighbor (KNN), Decision Tree, dan Random Forest untuk klasifikasi promosi karyawan. Meskipun KNN dianggap memiliki performa terbaik dalam studi ini, Random Forest juga menunjukkan hasil yang baik. Dalam pengujian, Random Forest mencapai akurasi 87,55% dalam pelatihan dan 85,29% dalam pengujian, dengan precision 91,29% dan 88,81% serta recall 83,03% dan 80,74% masing-masing [5].

### 2.2. Machine Learning

Pembelajaran mesin (Machine Learning) adalah bidang kecerdasan buatan yang menggunakan teknik statistik untuk memberikan kemampuan sistem komputer untuk "belajar" dari data, tanpa diprogram secara eksplisit. Pembelajaran mesin diciptakan pada tahun 1959 oleh Arthur Samuel dan digunakan dalam berbagai tugas komputerisasi dimana merancang dan memprogram algoritma eksplisit dengan kinerja yang baik, sulit atau tidak layak [6].

Machine Learning, yang berada di bawah payung kecerdasan buatan, menyediakan sarana bagi sistem untuk memperoleh pengetahuan dan meningkatkan kinerja mereka melalui pengalaman, semua tanpa perlu pemrograman eksplisit. Dalam studi baru-baru ini yang dilakukan oleh Esther et al., (2022), berbagai algoritma Machine Learning,

seperti, Support Vector Machine, Decision Tree, dan Random Forest, digunakan untuk menganalisa penyakit berdasarkan gejala. Hasil dari penelitian ini menunjukkan bahwa algoritma Machine Learning memiliki kemampuan luar biasa secara efektif memprediksi penyakit hanya berdasarkan gejala [7].

### 2.3. Decision Tree

Pohon Keputusan atau Decision Tree adalah algoritma pembelajaran mesin yang digunakan untuk tugas-tugas klasifikasi dan regresi. Algoritma ini membangun model prediksi berdasarkan struktur pohon atau hirarki keputusan model prediksi dalam bentuk struktur pohon hierarkis. Struktur ini terdiri dari node keputusan, yang mewakili pertanyaan atau kondisi, dan node daun, yang mewakili keputusan atau hasil akhir. Node keputusan diatur secara berurutan dan bercabang menjadi dua atau lebih jalur berdasarkan jawaban dari pertanyaan atau kondisi tersebut. Node daun berfungsi sebagai representasi dari keputusan akhir atau nilai prediksi. Decision Tree sangat berguna dalam menangani data kategoris dan kontinu. Keuntungan utama dari algoritma ini adalah kemampuannya untuk dengan cepat beradaptasi dengan kumpulan data yang ada, yang memungkinkan proses pemodelan menjadi lebih efisien. Selain itu, model yang dihasilkan oleh Decision Tree dapat dengan mudah divisualisasikan dan dipahami melalui penggunaan diagram "pohon", yang memudahkan interpretasi dan analisis dari proses pengambilan keputusan yang mendasarinya. Hal ini menjadikan Decision Tree sebagai alat yang berharga dalam analisis data dan sistem pendukung keputusan [8].

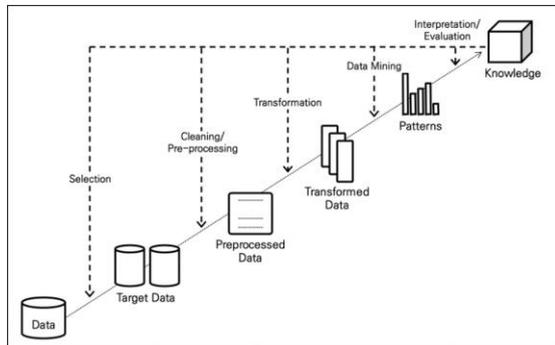
### 2.4. Random Forest

Random Forest (RF) adalah teknik dalam pembelajaran mesin yang melibatkan pembuatan sejumlah besar pohon keputusan. Setiap pohon dalam "hutan" ini dibuat berdasarkan sampel data yang dipilih secara acak. Ketika membuat prediksi, setiap pohon dalam hutan memberikan prediksi dan kelas dengan jumlah suara (vote) terbanyak dipilih sebagai hasil prediksi model [9].

Dalam konteks klasifikasi, Random Forest bekerja dengan cara membuat sejumlah besar pohon keputusan dan setiap pohon memberikan prediksi kelas. Kelas yang mendapatkan suara terbanyak dari semua pohon dalam hutan akan menjadi prediksi model. Salah satu keunggulan Random Forest adalah kemampuannya untuk menangani data yang memiliki banyak fitur dan dapat mengurangi risiko overfitting yang sering terjadi dalam model pohon keputusan tunggal. Random Forest telah digunakan dalam berbagai aplikasi, termasuk klasifikasi warna kulit berdasarkan warna piksel citra, klasifikasi data bank marketing, dan prediksi potensial gempa bumi. Dalam semua kasus ini, Random Forest menunjukkan performa yang baik dalam klasifikasi, dengan tingkat akurasi yang tinggi [10].

### 3. METODE PENELITIAN

Pada penelitian ini, menggunakan metode Knowledge Discovery In Database (KDD). Berikut merupakan tahapan – tahapan dan proses yang ada pada Knowledge Discovery In Database:



Gambar 1. Langkah-langkah KDD

#### 3.1. Selection Data

Tahap seleksi atau selection merupakan langkah awal dalam proses data mining, di mana data yang relevan dipilih dari database.

#### 3.2. Pre-processing

Tahap *preprocessing* adalah tahap penting dalam proses data mining yang melibatkan pembersihan dan persiapan data sebelum analisis lebih lanjut. Pada tahap ini, dilakukan proses pengecekan data lebih lanjut agar data bisa diolah pada tahap berikutnya.

#### 3.3. Transformation

Tahap *transformasi* dalam proses data mining melibatkan pengubahan data menjadi format yang sesuai untuk pengolahan data pada tahap selanjutnya. Salah satu teknik yang digunakan dalam tahap ini adalah *One-Hot Encoding* [11].

#### 3.4. Data Mining

Pada tahap *data mining*, analisis dan klasifikasi data dilakukan menggunakan algoritma Decision Tree dan Random Forest. Algoritma ini digunakan untuk mengolah dataset jenis aspal di Jawa Barat dan menghasilkan evaluasi dari kedua algoritma tersebut.

#### 3.5. Evaluation

Selama fase evaluasi penambangan data, kinerja algoritma Pohon Keputusan dan random forest dinilai dengan cara menggunakan metrik yang menghasilkan nilai akurasi, presisi, recall, dan F1-score yang memainkan peran penting dalam menentukan efektivitas dan efisiensi algoritma ini dalam memproses dan menganalisis kumpulan data yang diberikan.

### 4. HASIL DAN PEMBAHASAN

Pada tahap ini menampilkan Hasil dari penelitian berdasarkan penggunaan algoritma Decision Tree dan Random Forest.

#### 4.1. Selection Data

Dalam konteks ini, data yang dipilih adalah data jenis aspal di Jawa Barat pada tahun 2019 - 2022 yang berjumlah 1191 data. Proses seleksi ini penting karena tidak semua data yang ada dalam database dibutuhkan dalam proses data mining

rs	nama_ruas_jalan	jenis_permukaan	panjang_ruas_jalan	satuan	tahun
97	NAROGONG - CILEUNGI	HOTMIX	6,04	KILOMETER	2019
97	CILEUNGI - CIBINONG (CITEUREUP)	HOTMIX	14,13	KILOMETER	2019
97	JL. MAYOR OKING (CITEUREUP)	HOTMIX	3,61	KILOMETER	2019
97	JL. MAYOR OKING (CIBINONG)	HOTMIX	3,79	KILOMETER	2019
163	JL. RAYA CISEWU (PANGALENGAN)	HOTMIX	0,52	KILOMETER	2019
171	JL. NANIUNG - PATROL	BETON	5,61	KILOMETER	2019
349	NAGREG - BTS.BANDUNG/GARUT	HOTMIX	2,33	KILOMETER	2019
365	TERUSAN BUAHBATU (BTS.KOTA/KAB.BANDUNG) - BOI...	BETON	2	KILOMETER	2019
365	BOJONGSOANG - Sp.MUNIUL (JL. SILIWANGI)	BETON	5,08	KILOMETER	2019
365	Sp.MUNIUL - CIPARAY (JL. Sp. MUNIUL - JL. RAYA LASW...	BETON	6,97	KILOMETER	2019
365	JL. RAYA LASWI (CIPARAY)	BETON	3,4	KILOMETER	2019
365	JL. RAYA LASWI (s.d SP.3 JL. CKAREO/JL.TENGAH) MAJ...	BETON	5,13	KILOMETER	2019
367	MAJALAYA (SP.3 JL.CIKAREO/JL.TENGAH) - SAWAHBERA...	BETON	13,66	KILOMETER	2019
398	BTS. KAB. BANDUNG/CIANJUR - PONDOK DATAR	BETON	0,9	KILOMETER	2019
400	CIPAMOKOLAN (BTS. KOTA BANDUNG/JBT TOL) - SP. M...	BETON	15,12	KILOMETER	2019
401	SP.3 PANENJOAN - SAWAHBERA (SP.CIAPATI)	BETON	5,27	KILOMETER	2019
402	PARAKAN MUNCAENG - SP.3 PANENJOAN	HOTMIX	1,85	KILOMETER	2019
164	CUKUL (BTS.BANDUNG/GARUT) - SP.GENTENG	HOTMIX	2,95	KILOMETER	2019
164	SP.GENTENG - SP.TALEGONG (SUKAMULA)	HOTMIX	6,96	KILOMETER	2019
164	SP.TALEGONG (SUKAMULA)-CISEWU-SIKARAME-RANCA...	HOTMIX	47,2	KILOMETER	2019
284	MALANGBONG - BTS. GARUT/SUMEDANG	HOTMIX	8,5	KILOMETER	2019
345	JL. SUHERMAN (GARUT)	HOTMIX	1,37	KILOMETER	2019
345	JL. MERDEKA (GARUT)	HOTMIX	0,39	KILOMETER	2019

Gambar 2. Hasil selection Data

#### 4.2. Pre-processing

Tahap awal yaitu melakukan pemeriksaan data yang hilang atau *handling missing data* adalah tahap penting dalam proses preprocessing data. Tahap ini melibatkan identifikasi dan penanganan data yang hilang dalam dataset. Data yang hilang dapat terjadi karena berbagai alasan, seperti kesalahan entri, kegagalan sistem, atau data yang tidak dikumpulkan.

```
data.isnull().sum()
id 0
kode_provinsi 0
nama_provinsi 0
kode_kabupaten_kota 0
nama_kabupaten_kota 0
wilayah_uptd 0
no_ruas 0
nama_ruas_jalan 0
jenis_permukaan 0
panjang_ruas_jalan 0
satuan 0
tahun 0
dtype: int64
```

Gambar 3. Handling Missing Data

Dari Gambar 3 menampilkan hasil tidak ada data yang hilang. Selanjutnya Dilakukan pemeriksaan apakah ada data Yang duplikat menggunakan operator “data.duplicated().sum()” pada google collab dan menghasilkan output 0, hal ini mengindikasikan tidak ada duplikasi data.

```
[ ] data.duplicated().sum()
0
```

Gambar 4. Pengecekan Duplikasi Data

### 4.3. Transformation

Teknik yang digunakan pada tahap ini adalah *One-Hot Encoding* adalah teknik yang digunakan dalam pengolahan data untuk mengubah variabel kategori menjadi *vektor biner*. Teknik ini sangat berguna dalam *pembelajaran mesin*, di mana algoritma biasanya memerlukan input numerik. Dengan *One-Hot Encoding*, setiap kategori dalam variabel kategori diubah menjadi kolom baru dan diberi nilai 1 jika kategori tersebut ada dalam baris tersebut dan 0 jika tidak.

```
# DataFrame dengan kolom kategori
data = pd.DataFrame({'jenis_permukaan': ['BETON', 'HOTMIX']})

# One-Hot Encoding menggunakan pandas
data_encoded = pd.get_dummies(data, columns=['jenis_permukaan'], prefix='permukaan')
print(data_encoded)
```

	permukaan_BETON	permukaan_HOTMIX
0	1	0
1	0	1

Gambar 5. Proses Transformasi

Pada Gambar 5, variabelnya adalah 'jenis\_permukaan' yang mana variabel tersebut memiliki dua kategori: 'BETON' dan 'HOTMIX', variabel 'jenis\_permukaan' diubah menjadi dua kolom baru: 'permukaan\_BETON' dan 'permukaan\_HOTMIX'. Setiap baris dalam DataFrame sekarang memiliki nilai 1 pada kolom yang sesuai dengan kategori aslinya dan 0 pada kolom lainnya. Data yang dihasilkan dari proses ini memiliki format *vektor biner* dan siap untuk diolah pada tahap berikutnya.

### 4.4. Data Mining

Proses data mining yang melibatkan dua tahapan utama: pemilihan model dan pelatihan model. Berikut adalah penjelasan lebih rinci tentang kedua tahapan tersebut:

#### 4.4.1. Pemilihan Model

Pemilihan model adalah tahap awal dalam proses data mining, di mana algoritma yang akan digunakan untuk analisis data dipilih. Pada penelitian ini, memilih dua model, yaitu Decision Tree dan Random Forest pemodelan dalam data pelatihan dan data pengujian.

##### 1) Model Decision Tree

Model Decision Tree adalah model yang menggunakan struktur pohon atau hierarki keputusan. Setiap node dalam pohon mewakili suatu fitur atau atribut, dan setiap cabang mewakili aturan keputusan. Model ini sangat baik untuk interpretasi karena struktur pohon memungkinkan visualisasi yang jelas dari proses pengambilan keputusan.

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report

# Membuat model Decision Tree
dt_classifier = DecisionTreeClassifier(random_state=42)

# Melatih model Decision Tree menggunakan data training
dt_classifier.fit(X_train, y_train)
```

DecisionTreeClassifier  
DecisionTreeClassifier(random\_state=42)

Gambar 6. Pemilihan Model Decision Tree

##### 2) Model Random Forest

Model Random Forest adalah teknik dalam pembelajaran mesin yang melibatkan pembuatan sejumlah besar pohon keputusan. Setiap pohon dalam "hutan" ini dibuat berdasarkan sampel data yang dipilih secara acak. Ketika membuat prediksi, setiap pohon dalam hutan memberikan prediksi dan kelas dengan jumlah suara (vote) terbanyak dipilih sebagai hasil prediksi model. Salah satu keunggulan Random Forest adalah kemampuannya untuk menangani data yang memiliki banyak fitur dan dapat mengurangi risiko overfitting yang sering terjadi dalam model pohon keputusan tunggal.

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

# Membuat model Random Forest
rf_classifier = RandomForestClassifier(random_state=42)

# Melatih model Random Forest menggunakan data training
rf_classifier.fit(X_train, y_train)
```

RandomForestClassifier  
RandomForestClassifier(random\_state=42)

Gambar 7. Pemilihan Model Random Forest

#### 4.4.2. Pelatihan Model

Proses pelatihan model machine learning yang digunakan melibatkan dua jenis model, yaitu Decision Tree classifier dan Random Forest classifier. Berikut adalah penjelasan detail prosesnya:

##### a) Model Decision Tree

```
# using the model DecisionTreeClassifier
Tree_model = DecisionTreeClassifier(max_depth=10)

# fit model
Tree_model.fit(X_train, y_train)
y_pred_T = Tree_model.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred_T)

# Convert accuracy to a percentage string
accuracy_percentage = f"{accuracy * 100:.2f}%"

# Print the accuracy as a percentage
print("Score the X-train with Y-train is : ", Tree_model.score(X_train, y_train))
print("Score the X-test with Y-test is : ", Tree_model.score(X_test, y_test))
print("Model Evaluation Decision Tree : accuracy score ", accuracy_percentage)
```

Score the X-train with Y-train is : 0.9695378151260504  
Score the X-test with Y-test is : 0.8702928870292888  
Model Evaluation Decision Tree : accuracy score 87.03%

Gambar 8. Pelatihan Model Decision Tree

Dari Gambar 8, model Decision Tree dibuat dengan kedalaman pohon keputusan maksimum 10. Kedalaman ini merujuk pada jumlah maksimum tingkat dalam pohon keputusan. Model ini kemudian dilatih menggunakan data pelatihan ( $X_{train}$  dan  $y_{train}$ ). Proses pelatihan ini melibatkan

pembelajaran struktur pohon keputusan berdasarkan data input ( $X_{train}$ ) dan target yang diinginkan ( $y_{train}$ ). Setelah model dilatih, model ini digunakan untuk memprediksi target ( $y_{test}$ ) berdasarkan data pengujian ( $X_{test}$ ). Hasil prediksi ini kemudian dibandingkan dengan target sebenarnya untuk menghitung akurasi model. Akurasi ini dihitung sebagai persentase prediksi yang benar dari total prediksi. Akurasi dihitung dua kali: sekali untuk data pelatihan dan sekali lagi untuk data pengujian. Akurasi ini kemudian dicetak sebagai persentase. Akurasi yang dihasilkan pada data latih Decision Tree Classifier sebesar 96,9% dan akurasi pada data Ujinya sebesar 87,8%.

b) Model Random Forest

```
# using the model Random Forest Classifier
RF_model = RandomForestClassifier(n_estimators=300)
RF_model.fit(X_train, y_train)
y_pred_r = RF_model.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred_r)

# Convert accuracy to a percentage string
accuracy_percentage = f'{accuracy * 100:.2f}%'

# Print the accuracy as a percentage
print("Score the X-train with Y-train is : ", RF_model.score(X_train, y_train))
print("Score the X-test with Y-test is : ", RF_model.score(X_test, y_test))
print("Model Evaluation Random Forest Classifier: accuracy score ", accuracy_percentage)

Score the X-train with Y-train is : 1.0
Score the X-test with Y-test is : 0.9079457907945791
Model Evaluation Random Forest Classifier: accuracy score 90.79%
```

Gambar 9. Pelatihan Model Random Forest

Dari Gambar 9, model Random Forest dibuat dengan jumlah estimator sebanyak 300. Model ini kemudian dilatih menggunakan data latih ( $X_{train}$  dan  $y_{train}$ ). Setelah model dilatih, model ini digunakan untuk memprediksi data uji ( $X_{test}$ ) dan hasil prediksi ini disimpan dalam variabel  $y_{pred_r}$ . Akurasi dari model ini dihitung dengan membandingkan hasil prediksi ( $y_{pred_r}$ ) dengan nilai sebenarnya ( $y_{test}$ ) menggunakan fungsi `accuracy_score`. Akurasi ini kemudian dikonversi menjadi persentase. Akurasi yang dihasilkan pada data latih Random Forest Classifier sebesar 100% dan akurasi pada data Ujinya sebesar 90%.

4.5. Evaluation

Hasil evaluasi untuk dua model, yaitu Decision Tree dan Random Forest. Berikut interpretasi hasilnya:

a) Algoritma Decision Tree

```
from sklearn.metrics import accuracy_score

# calculate the classification report
report = classification_report(y_test, y_pred_t, target_names=["BETON", "HOTMIX"], output_dict=True)

# Convert values to percentage
for key in report.keys():
    if key not in ["accuracy", "macro avg", "weighted avg"]:
        report[key]["precision"] *= 100
        report[key]["recall"] *= 100
        report[key]["f1-score"] *= 100

# Print the modified classification report
print("Classification Report Model Decision Tree:")
print(" precision recall f1-score support")
for key in report.keys():
    if key not in ["accuracy", "macro avg", "weighted avg"]:
        print(f'{key} {report[key]["precision"]:.2f} {report[key]["recall"]:.2f} {report[key]["f1-score"]:.2f} {report[key]["support"]:.2f}')

Classification Report Model Decision Tree:
 precision recall f1-score support
BETON 72.00% 67.52% 69.96% 53
HOTMIX 91.41% 91.41% 91.41% 186
```

Gambar 10 Evaluasi Decision Tree

Hasil analisis Nilai akurasi 87,03% diperoleh untuk dataset tipe aspal di Jawa Barat melalui penerapan algoritma Pohon Keputusan, yang

menandakan efektivitas pendekatan ini dalam menganalisis data. Selanjutnya, ketika mengevaluasi model, diamati bahwa akurasi pada data pelatihan ( $X_{train}$  dengan kereta  $Y$ ) mencapai sekitar 96,95%. Tingkat akurasi yang tinggi ini menyoroti kinerja luar biasa dari model Pohon Keputusan ketika diterapkan pada data pelatihan. Akurasi pada data pengujian ( $X_{test}$  dengan  $Y_{test}$ ) adalah sekitar 87.03%, yang mengindikasikan bahwa model Decision Tree memiliki kinerja yang baik pada data pengujian. Ini menunjukkan bahwa model tidak terlalu overfitting. Classification Report menunjukkan metrik evaluasi presisi, recall, dan F1-score untuk setiap kelas (BETON dan HOTMIX). Model Decision Tree memiliki kinerja yang baik dalam memprediksi HOTMIX mendapatkan nilai presisi (91,01%), recall (92,47%), dan F1-score (91,73). Sedangkan dalam memprediksi BETON mendapatkan nilai presisi (72,00%), recall (67,92%), dan F1-score (69,90%).

b) Algoritma Random Forest

```
from sklearn.metrics import accuracy_score

# calculate the classification report
report = classification_report(y_test, y_pred_r, target_names=["BETON", "HOTMIX"], output_dict=True)

# Convert values to percentage
for key in report.keys():
    if key not in ["accuracy", "macro avg", "weighted avg"]:
        report[key]["precision"] *= 100
        report[key]["recall"] *= 100
        report[key]["f1-score"] *= 100

# Print the modified classification report
print("Classification Report Model Random Forest:")
print(" precision recall f1-score support")
for key in report.keys():
    if key not in ["accuracy", "macro avg", "weighted avg"]:
        print(f'{key} {report[key]["precision"]:.2f} {report[key]["recall"]:.2f} {report[key]["f1-score"]:.2f} {report[key]["support"]:.2f}')

Classification Report Model Random Forest:
 precision recall f1-score support
BETON 87.80% 67.92% 76.66% 53
HOTMIX 91.41% 97.31% 94.27% 186
```

Gambar 12 Evaluasi Random Forest

Hasil analisis menggunakan Random Forest mendapatkan nilai akurasi 90,79% pada dataset jenis aspal di Jawa Barat, Sedangkan untuk evaluasi model menunjukkan hasil Akurasi pada data pelatihan ( $X_{train}$  dengan  $Y_{train}$ ) adalah sempurna (100%), yang mengindikasikan bahwa model Random Forest Classifier mendapatkan hasil sempurna dengan data pelatihan. Akurasi pada data pengujian ( $X_{test}$  dengan  $Y_{test}$ ) adalah sekitar 90.79%, yang menunjukkan kinerja yang baik pada data pengujian. Ini menyiratkan bahwa, berdasarkan evaluasi akurasi, model Random Forest Classifier mengungguli model Pohon Keputusan. Classification Report menunjukkan metrik evaluasi presisi, recall, dan F1-score untuk setiap kelas HOTMIX mendapatkan nilai presisi (91.41%), recall (97.31%), dan F1-score (94.27%), Sedangkan dalam memprediksi BETON mendapatkan nilai presisi (87.80%), recall (67,92%), dan F1-score (76.60%). Model Random Forest Classifier memiliki kinerja yang baik dalam memprediksi kedua kelas (BETON dan HOTMIX) dengan akurasi yang tinggi.

## 5. KESIMPULAN DAN SARAN

Penelitian ini telah berhasil membangun dan mengevaluasi model Decision Tree dan Random Forest untuk klasifikasi jenis aspal di Jawa Barat. Meskipun Algoritma Decision Tree menunjukkan kinerja yang baik dengan akurasi sekitar 87,03% dan mampu memprediksi jenis aspal "HOTMIX" dengan presisi, recall, dan F1-score yang tinggi, kinerjanya dalam memprediksi jenis aspal "BETON" sedikit lebih rendah. Di sisi lain, Algoritma Random Forest menunjukkan kinerja yang lebih unggul dengan akurasi sekitar 90,79% dan mampu memprediksi kedua jenis aspal, "HOTMIX" dan "BETON," dengan presisi, recall, dan F1-score yang sangat baik. Secara keseluruhan, Random Forest mengungguli Decision Tree dalam hal akurasi dan metrik evaluasi lainnya, memberikan hasil yang lebih konsisten dan akurat dalam mengklasifikasikan jenis aspal.

Disarankan untuk penelitian selanjutnya menggunakan Random Forest karena kinerjanya yang superior dibandingkan dengan Decision Tree atau pohon keputusan, yang dapat meningkatkan akurasi dan keandalan model. Penting juga untuk memperhatikan keseimbangan data, karena data yang tidak seimbang dapat mempengaruhi hasil klasifikasi; metode seperti oversampling atau undersampling bisa digunakan untuk mengatasi masalah ini. Selain itu, evaluasi model tidak hanya harus bergantung pada akurasi, tetapi juga melibatkan metrik lain seperti presisi, recall, dan AUC-ROC untuk mendapatkan gambaran yang lebih komprehensif tentang kinerja model. Terakhir, hasil penelitian ini memiliki potensi untuk diaplikasikan dalam konteks lain yang membutuhkan klasifikasi jenis aspal, seperti dalam pemeliharaan jalan atau proyek konstruksi, dan lain-lain sehingga memperluas manfaat dari penelitian ini.

## DAFTAR PUSTAKA

- [1] Yuli Mardi, "Data Mining : Klasifikasi Menggunakan Algoritma C4 . 5 Data Mining Merupakan Bagian Dari Tahapan Proses Knowledge Discovery In Database ( Kdd ) . Jurnal Edik Informatika," *J. Edik Inform.*, Vol. 2, No. 2, Pp. 213–219, 2019.
- [2] P. H. Putra, A. Azanuddin, B. Purba, And Y. A. Dalimunthe, "Random Forest And Decision Tree Algorithms For Car Price Prediction," *J. Mat. Dan Ilmu Pengetah. Alam Lldikti Wil. 1*, Vol. 3, No. 2, Pp. 81–89, 2023, Doi: 10.54076/Jumpa.V3i2.305.
- [3] F. Y. Pamuji And V. P. Ramadhan, "Komparasi Algoritma Random Forest Dan Decision Tree Untuk Memprediksi Keberhasilan Immunotherapy," *J. Teknol. Dan Manaj. Inform.*, Vol. 7, No. 1, Pp. 46–50, 2021, Doi: 10.26905/Jtmi.V7i1.5982.
- [4] K. G. Fiqri, A. T. Hanuranto, And C. Setianingsih, "Analisa Perbandingan Klasifier Decision Tree, Random Forest, Dan Adaboost Dalam Mendeteksi Serangan Comparative Analysis Decision Tree, Random Forest, And Adaboost Classifier On Detecting Attack," *E-Proceeding Eng. Vol.7, No.1 April 2020*, Vol. 7, No. 1, Pp. 403–410, 2020, [Online]. Available: [https://Openlibrary.Telkomuniversity.Ac.Id/Pustaka/Files/156845/Jurnal\\_Eproc/Analisis-Perbandingan-Klasifier-Decision-Tree-Random-Forest-Dan-Adaboost-Dalam-Mendeteksi-Serangan-Siber.Pdf](https://Openlibrary.Telkomuniversity.Ac.Id/Pustaka/Files/156845/Jurnal_Eproc/Analisis-Perbandingan-Klasifier-Decision-Tree-Random-Forest-Dan-Adaboost-Dalam-Mendeteksi-Serangan-Siber.Pdf)
- [5] Louis Madaerdo Sotarjua And Dian Budhi Santoso, "Perbandingan Algoritma Knn, Decision Tree, Dan Random Forest Pada Data Imbalanced Class Untuk Klasifikasi Promosi Karyawan," *J. Instek (Informatika Sains Dan Teknol.*, Vol. 7, No. 2, Pp. 192–200, 2022, Doi: 10.24252/Instek.V7i2.31385.
- [6] A. S. Rajesh, M. S. Prabhushwamy, And S. Krishnasamy, "Smart Manufacturing Through Machine Learning: A Review , Perspective , And Future Directions To The Machining Industry," *Hindawi J. Eng.*, Vol. 2022, P. 6, 2022.
- [7] Ch. Esther, S. Nayana Sai, S. Sushma, B. V. R. Gupta, And Mr. G. Srinivasa Rao, "Disease Prediction Based On Symptoms By Using Decision Tree And Random Forest In Machine Learning," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, Vol. 3307, Pp. 419–427, 2022, Doi: 10.32628/Cseit2283105.
- [8] S. Kasus And B. Maal, "Sistem Pendukung Keputusan Penerimaan Beasiswa Menggunakan Algoritma C4 . 5," *J. Pendidik. Teknol. Dan Kejuru.*, Vol. 15, No. 1, Pp. 113–123, 2018.
- [9] H. Tantyoko, D. K. Sari, And A. R. Wijaya, "Prediksi Potensial Gempa Bumi Indonesia Menggunakan Metode Random Forest Dan Feature Selection," *Idealis Indones. J. Inf. Syst.*, Vol. 6, Pp. 83–89, 2023.
- [10] M. A. Ridla, F. Sains, U. Ibrahimy, And M. A. Ridla, "Perbandingan Algoritma Pembelajaran Mesin Untuk Klasifikasi Warna Kulit Berdasarkan Warna Piksel Citra," *Jusifor J. Sist. Inf. Dan Inform.*, Vol. 1, No. 1, Pp. 33–41, 2022.
- [11] D. A. C. N. Bayes, And D. A. N. Svm, "Perbandingan Metode Data Mining Untuk Prediksi Nilai Dan Waktu Kelulusan Mahasiswa Prodi Teknik Informatika," *J. Tekno Insentif*, Vol. 13, No. 1, Pp. 16–25, 2019.