

DAFTAR PUSTAKA

- Adwang, J. (2020). Kajian Sertifikasi Uji Laik Fungsi Jalan (ULFJ) Pada Ruas Jalan Nasional Pulau Sumatera (Poso) Nomor Ruas 32.12. K Km. 220+ 240 sd Km. 221+ 000 Provinsi *Tekno*, 18.
<https://ejournal.unsrat.ac.id/v3/index.php/tekno/article/view/30157>
- Bagian-Bagian Jalan Nasional, P., Kasus di Ruas Jalan MERR Surabaya, S., Kusuma Dewi, D., Asih Aryani Soemitro, R., Suprayitno, H., Budianto, H., Pejabat, M., Bina Marga, D. P., & Provinsi Jawa Timur, P. (2020). Utilization of National Road Segment, Case Study in MERR Street Section Surabaya. In *Jurnal Manajemen Aset Infrastruktur & Fasilitas* (Vol. 4, Issue 3).
- Bidang, P., Sains, K., Mardi, Y., Gajah, J., No, M., & Barat, S. (n.d.). *Jurnal Edik Informatika Data Mining : Klasifikasi Menggunakan Algoritma C4 . 5 Data mining merupakan bagian dari tahapan proses Knowledge Discovery in Database (KDD) . Jurnal Edik Informatika*.
- Frastian, N., Hendrian, S., & Valentino, V. H. (2018). Komparasi Algoritma Klasifikasi Menentukan Kelulusan Mata Kuliah Pada Universitas. *Faktor Exacta*, 11(1), 66.
<https://doi.org/10.30998/faktorexacta.v11i1.1826>
- Heryana, A. (2020). Jumlah kelompok Fungsi Syarat data. *Universitas Esa Unggul, May*, 1–20. <https://doi.org/10.13140/RG.2.2.23266.15047>
- Ii, B. A. B., & Pustaka, T. (n.d.). *No Title*. 6–9.
- Kiding, A. (2019). Analisis Sentimen Publik Terhadap Tes Cpnas Melalui Media Twiter Menggunakan Metode Naïve Bayes Classifier. *Http://E-Journal.Uajy.Ac.Id/7244/4/3TF03686.Pdf*, 2010, 15–48. <http://e-journal.uajy.ac.id/7244/4/3TF03686.pdf>
- Novianti, B., Rismawan, T., & Bahri, S. (2016). *IMPLEMENTASI DATA MINING DENGAN ALGORITMA C4 . 5 UNTUK PENJURUSAN SISWA (STUDI KASUS : SMA NEGERI 1 PONTIANAK)*. 04(3).
- Nuryadin, Ruslan, 2005. Panduan Menggunakan Mapserver, Informatika, Bandung.
- Pamuji, F. Y., & Ramadhan, V. P. (2021). Komparasi Algoritma Random Forest dan Tree untuk Memprediksi Keberhasilan Immunotherapy. *Jurnal Teknologi Informatika*, 7(1), 46–50. <https://doi.org/10.26905/jtmi.v7i1.5982>
- Pan, U., Rakyat, P., Jenderal, D., & Marga, B. (2021). *STANDAR JURNAL PROSEDUR TATA KERJA PENILIKAN JALAN SOP/UPM/DJBM-*
- Prahasta, Eddy, 2007. Membangun Aplikasi Web – based GIS dengan Mapsrever,



Informatika, Bandung

Saragi dkk. (2021). Analisis Lapisan Aspal Beton (AC-BC) Dengan Penambahan Limbah Kaleng Miniman Ditinjau Dari Karakteristik Marshall dan Uji Penetrasi. *Construct*, 1(1), 49–58.

Setiawan dan Rabassa, 1998, Aplikasi Open Source untuk Pemetaan Online, Bogor

Supriyadi, R., Gata, W., Maulidah, N., & Fauzi, A. (2020). Penerapan Algoritma Random Forest Untuk Menentukan Kualitas Anggur Merah. *E-Bisnis : Jurnal Ilmiah Ekonomi Dan Bisnis*, 13(2), 67–75. <https://doi.org/10.51903/e-bisnis.v13i2.247>

Teori, L. (n.d.). *lapisan tanah dasar { subgrade }, agar tanah mendapat tekanan tidak melampaui daya dukung tanahnya . Pada umumnya perkerasan terdiri dari beberapa lapis ,.* 11–22.

Wikipedia. (2023). Tana Luwu. https://id.wikipedia.org/wiki/Tana_Luwu



Lampiran 1 Tampilan Dataset Kerusakan Jalan

PPK	Provind	Satler	No Rusas Jalan	Nama Rusas Jalan	Tahun	No Identitas	Lokasi / KM	Sub Kategori	Rusak Rusak Kiri Jalat	Rusak Rusak Kanan Ji	Waktu Pengamatan
TP-01 Koridor Palop	SULAWESI SELATA	SKPD-TP Dinas Pele	33	Bts Kota Palopo - Bt	2023		4 399 + 500	120 (Terlelupad)	Tidak Rusak Kiri	Rusak Kanan	Oktober
TP-01 Koridor Palop	SULAWESI SELATA	SKPD-TP Dinas Pele	33	Bts Kota Palopo - Bt	2023		5 399 + 600	120 (Terlelupad)	Tidak Rusak Kiri	Rusak Kanan	Oktober
TP-01 Koridor Palop	SULAWESI SELATA	SKPD-TP Dinas Pele	33	Bts Kota Palopo - Bt	2023		1 400 + 000	511 (Patek KM Rusak)	Tidak Rusak Kiri	Rusak Kanan	Oktober
TP-01 Koridor Palop	SULAWESI SELATA	SKPD-TP Dinas Pele	33	Bts Kota Palopo - Bt	2023		6 400 + 175	120 (Terlelupad)	Tidak Rusak Kiri	Rusak Kanan	Oktober
TP-01 Koridor Palop	SULAWESI SELATA	SKPD-TP Dinas Pele	33	Bts Kota Palopo - Bt	2023		1 402 + 000	114 (Ambilasi)	Tidak Rusak Kiri	Rusak Kanan	Oktober
TP-01 Koridor Palop	SULAWESI SELATA	SKPD-TP Dinas Pele	33	Bts Kota Palopo - Bt	2023		8 406 + 360	120 (Terlelupad)	Rusak Kiri	Tidak Rusak Kanan	Oktober
TP-01 Koridor Palop	SULAWESI SELATA	SKPD-TP Dinas Pele	33	Bts Kota Palopo - Bt	2023		2 406 + 600	114 (Ambilasi)	Tidak Rusak Kiri	Rusak Kanan	Oktober
TP-01 Koridor Palop	SULAWESI SELATA	SKPD-TP Dinas Pele	33	Bts Kota Palopo - Bt	2023		1 409 + 450	252 (Kehilangan Per	Tidak Rusak Kiri	Rusak Kanan	Oktober
TP-01 Koridor Palop	SULAWESI SELATA	SKPD-TP Dinas Pele	34	Bts Kab Luwu - Masa	2023		1 411 + 650	840 (Kerusakan Sayt	Rusak Kiri	Tidak Rusak Kanan	Oktober
TP-01 Koridor Palop	SULAWESI SELATA	SKPD-TP Dinas Pele	34	Bts Kab Luwu - Masa	2023		1 412 + 100	117 (Retak Buaya)	Rusak Kiri	Tidak Rusak Kanan	Oktober
TP-01 Koridor Palop	SULAWESI SELATA	SKPD-TP Dinas Pele	34	Bts Kab Luwu - Masa	2023	001 / 001	412 + 700	117 (Retak Buaya)	Tidak Rusak Kiri	Rusak Kanan	Oktober
TP-01 Koridor Palop	SULAWESI SELATA	SKPD-TP Dinas Pele	34	Bts Kab Luwu - Masa	2023		2 412 + 800	840 (Kerusakan Sayt	Rusak Kiri	Tidak Rusak Kanan	Oktober
TP-01 Koridor Palop	SULAWESI SELATA	SKPD-TP Dinas Pele	34	Bts Kab Luwu - Masa	2023		1 413 + 300	114 (Ambilasi)	Rusak Kiri	Tidak Rusak Kanan	Oktober
TP-01 Koridor Palop	SULAWESI SELATA	SKPD-TP Dinas Pele	34	Bts Kab Luwu - Masa	2023		2 415 + 515	117 (Retak Buaya)	Tidak Rusak Kiri	Rusak Kanan	Oktober
TP-01 Koridor Palop	SULAWESI SELATA	SKPD-TP Dinas Pele	34	Bts Kab Luwu - Masa	2023		1 415 + 700	120 (Terlelupad)	Tidak Rusak Kiri	Rusak Kanan	Oktober
TP-01 Koridor Palop	SULAWESI SELATA	SKPD-TP Dinas Pele	34	Bts Kab Luwu - Masa	2023	003 / 001	417 + 950	111 (Lubang)	Rusak Kiri	Rusak Kanan	Oktober
TP-01 Koridor Palop	SULAWESI SELATA	SKPD-TP Dinas Pele	34	Bts Kab Luwu - Masa	2023		4 418 + 475	117 (Retak Buaya)	Rusak Kiri	Tidak Rusak Kanan	Oktober
TP-01 Koridor Palop	SULAWESI SELATA	SKPD-TP Dinas Pele	34	Bts Kab Luwu - Masa	2023		4 418 + 650	117 (Retak Buaya)	Tidak Rusak Kiri	Rusak Kanan	Oktober

Link Data Kerusakan Jalan : <https://bit.ly/3W9o8Ei>



Lampiran 2 Preprocessing Data

```

import pandas as pd
import numpy as np
from google.colab import drive
drive.mount('/content/drive')
# Mengganti path dengan direktori yang sesuai
path = '/content/drive/My Drive/Colab Notebooks/'
kumpulan_datajalan = pd.read_csv(path + 'Dataset Asli.csv',
encoding="ISO-8859-1")

features_datajalan = kumpulan_datajalan[["Sub Kategori", "Rusak Ruas
Kiri Jalan", "Rusak Ruas Kanan Jalan", "Nama Wilayah", "Panjang (M)",
"Lebar (M)", "Tebal/Tinggi (M)"]].copy()
# Menampilkan data dalam bentuk tabel
features_datajalan

# Print contoh data kosong dalam bentuk tabel
data_preprocessing = features_datajalan.dropna()
data_preprocessing

data_preprocessing.isnull().sum()

# Mengonversi kolom ke tipe data string
data_preprocessing['Panjang (M)'] = data_preprocessing['Panjang
(M)'].astype(str)
data_preprocessing['Lebar (M)'] = data_preprocessing['Lebar
(M)'].astype(str)
data_preprocessing['Tebal/Tinggi (M)'] =
data_preprocessing['Tebal/Tinggi (M)'].astype(str)

# Menghilangkan koma
data_preprocessing['Panjang (M)'] = data_preprocessing['Panjang
(M)'].str.replace(',', '').astype(float)
data_preprocessing['Lebar (M)'] = data_preprocessing['Lebar
(M)'].str.replace(',', '').astype(float)
data_preprocessing['Tebal/Tinggi (M)'] =
data_preprocessing['Tebal/Tinggi (M)'].str.replace(',',
').astype(float)

# Menampilkan data hasil akhir yang sudah diubah menjadi float
data_preprocessing[['Panjang (M)', 'Lebar (M)', 'Tebal/Tinggi (M)']]

# Multiply 'panjang (M)', 'lebar (M)', dan tebal/tinggi(M) and store the
new column 'Volume kerusakan'
data_preprocessing['Volume Kerusakan'] = data_preprocessing['Panjang
(M)'] * data_preprocessing['Lebar (M)'] * data_preprocessing['Tebal/Tinggi (M)']

# Menampilkan data hasil akhir yang sudah diubah menjadi float
data_preprocessing

# Print contoh data kosong dalam bentuk tabel

```



```

datas_preprocessing = data_preprocessing.dropna()
datas_preprocessing

def sub_kategori_kerusakan(Kategori):
    if Kategori in ['111 (Lubang)', '112 (Keriting)', '114 (Amblas)', '116 (Kerusakan Tepi)', '117 (Retak Buaya)', '118 (Retak Garis)', '119 (Kegemukan Aspal)', '120 (Terkelupas)', '160 (Permukaan Aspal Kasar)', '232 (Alur)', '252 (Kehilangan Permukaan)', '612 (Tergerus)', '710 (Longsor)']:
        return 'Ada Kerusakan'
    if Kategori in ['115 (Jembul)', '253 (Rumput Panjang)', '412 (Kerusakan Saluran)', '413 (Saluran Ditumbuhi Tanaman)', '431 (Pendangkalan)', '471 (Tersumbat)', '511 (Patok KM Rusak)', '531 (Marka Jalan Pudar)', '640 (Pas. Batu Amblas)', '641 (Kehilangan Batuan)', '840 (Kerusakan Sayap Gorong-gorong)', 'Kondisi Baik']:
        return 'Tidak Ada Kerusakan'
    return None # Atau dapat diisi dengan nilai default lainnya

# Terapkan fungsi pada kolom 'kategori kerusakan'
datas_preprocessing['Kategori Kerusakan'] = datas_preprocessing['Sub Kategori'].apply(sub_kategori_kerusakan)

# Tampilkan DataFrame hasil
datas_preprocessing

# Fungsi untuk menentukan kategori volume kerusakan berdasarkan rentang
def tentukan_volume(volume):
    if 0 <= volume <= 0.97825:
        return 'Volume Tidak Rusak'
    elif 1 <= volume <= 675:
        return 'Volume Rusak'
    else:
        return 'Volume Tidak Valid'

# Terapkan fungsi pada kolom 'Kategori Volume Kerusakan'
datas_preprocessing['Kategori Volume Kerusakan'] =
datas_preprocessing['Volume Kerusakan'].apply(tentukan_volume)

# Tampilkan DataFrame hasil
datas_preprocessing

# Menemukan baris yang memiliki nilai NaN
rows_with_nan =
datas_preprocessing[datas_preprocessing.isnull().any(axis=1)]

# Menampilkan baris yang memiliki nilai NaN
print(datas_preprocessing[rows_with_nan])

# Menampilkan jumlah baris yang memiliki nilai NaN
print(datas_preprocessing.isnull().sum())

# Menyimpan DataFrame ke file CSV
datas_preprocessing_csv = "data pengolahan.csv" # Nama file CSV yang ingin disimpan

```



```

path_direktori = "/content/drive/My Drive/Colab Notebooks" # Path
lengkap ke direktori penyimpanan (ubah sesuai dengan direktori Anda)
datas_preprocessing.to_csv(path_direktori + "/" +
datas_preprocessing_csv, index=False) # Menyimpan DataFrame ke file
CSV

# Mengganti path dengan direktori yang sesuai
path = '/content/drive/My Drive/Colab Notebooks/'

datas_preprocessings = pd.read_csv(path + 'data pengolahan.csv',
encoding="ISO-8859-1")

from sklearn.preprocessing import MultiLabelBinarizer
# Mengganti nilai NaN dengan nilai default atau nilai yang sesuai
datas_preprocessings['Rusak Ruas Kiri Jalan'].fillna('default',
inplace=True)
datas_preprocessings['Rusak Ruas Kanan Jalan'].fillna('default',
inplace=True)
datas_preprocessings['Kategori Kerusakan'].fillna('default',
inplace=True)

# Melakukan fit_transform menggunakan LabelBinarizer
mlb = MultiLabelBinarizer()
mlb.fit_transform(datas_preprocessings['Rusak Ruas Kiri Jalan'])
mlb.fit_transform(datas_preprocessings['Rusak Ruas Kanan Jalan'])
mlb.fit_transform(datas_preprocessings['Kategori Kerusakan'])

datas_preprocessings.info()

datas_preprocessings['Rusak Ruas Kiri
Jalan']=datas_preprocessings['Rusak Ruas Kiri Jalan'].str.split(',\s*')
datas_preprocessings['Rusak Ruas Kanan
Jalan']=datas_preprocessings['Rusak Ruas Kanan
Jalan'].str.split(',\s*')
datas_preprocessings['Kategori
Kerusakan']=datas_preprocessings['Kategori
Kerusakan'].str.split(',\s*')

mlb = MultiLabelBinarizer()

mlb.fit(datas_preprocessings['Rusak Ruas Kiri Jalan'])
mlb.fit(datas_preprocessings['Rusak Ruas Kanan Jalan'])
mlb.fit(datas_preprocessings['Kategori Kerusakan'])

from sklearn.preprocessing import LabelEncoder

datas_preprocessings['Rusak Ruas Kiri Jalan'].explode().unique()
datas_preprocessings['Rusak Ruas Kanan Jalan'].explode().unique()
datas_preprocessings['Kategori Kerusakan'].explode().unique()

mlb.transform(datas_preprocessings['Rusak Ruas Kiri Jalan'])

```



```

mlb.transform(datas_preprocessings['Rusak Ruas Kanan Jalan'])
mlb.transform(datas_preprocessings['Kategori Kerusakan'])

a1 = pd.DataFrame(mlb.fit_transform(datas_preprocessings['Rusak Ruas
Kiri Jalan']), columns=mlb.classes_)
a2 = pd.DataFrame(mlb.fit_transform(datas_preprocessings['Rusak Ruas
Kanan Jalan']), columns=mlb.classes_)
a3 = pd.DataFrame(mlb.fit_transform(datas_preprocessings['Kategori
Kerusakan']), columns=mlb.classes_)

binarizer=pd.concat([datas_preprocessings, a1, a2, a3], axis=1)

binarizer

binarizer.isnull().sum()

multilabel_binarizer = binarizer.drop(['Rusak Ruas Kiri Jalan', 'Rusak
Ruas Kanan Jalan', 'Kategori Kerusakan'], axis=1)

multilabel_binarizer

# Menyimpan DataFrame ke file CSV
multilabel_binarizer_csv = "binarizer_data.csv" # Nama file CSV yang
ingin disimpan
path_direktori = "/content/drive/My Drive/Colab Notebooks" # Path
lengkap ke direktori penyimpanan (ubah sesuai dengan direktori Anda)
multilabel_binarizer.to_csv(path_direktori + "/" +
multilabel_binarizer_csv, index=False) # Menyimpan DataFrame ke file
CSV

# Mengganti path dengan direktori yang sesuai
path = '/content/drive/My Drive/Colab Notebooks/'

multilabel_binarizer_csv = pd.read_csv(path + 'binarizer_data.csv',
encoding="ISO-8859-1")

#check missing value
multilabel_binarizer_csv.isnull().sum()

# Mengganti path dengan direktori yang sesuai
path = '/content/drive/My Drive/Colab Notebooks/'

labelling = pd.read_csv(path + 'labelling.csv', encoding="ISO-8859-1")
multilabel_binarizer = pd.read_csv(path + 'binarizer_data.csv',
encoding="ISO-8859-1")

#check missing value
multilabel_binarizer.isnull().sum()

multilabel_binarizer_labelling =pd.concat([multilabel_binarizer, labelling],
axis='columns')

```




```

data_jalan = binarizer_labelling.drop(['Sub Kategori', 'Nama
Wilayah', 'Kategori Volume Kerusakan'], axis=1)

data_jalan

# Menyimpan DataFrame ke file CSV
data_jalan_csv = "pupr_jalan.csv" # Name of the CSV file you want to
save
path_direktori = "/content/drive/My Drive/Colab Notebooks" # Full path
to the saving directory (modify this as per your directory)

# Save the DataFrame to a CSV file
data_jalan.to_csv(path_direktori + "/" + data_jalan_csv, index=False)

# Mengganti path dengan direktori yang sesuai
path = '/content/drive/My Drive/Colab Notebooks/'

data_jalan = pd.read_csv(path + 'pupr_jalan.csv', encoding="ISO-8859-
1")

# Cek jumlah baris pada data
jumlah_baris = data_jalan.shape[0]

print("Jumlah baris pada data:", jumlah_baris)

print(data_jalan['Label'].value_counts())
#print(df_dataTraining.value_counts())

import matplotlib.pyplot as plt

# Data label
counts = data_jalan['Label'].value_counts()

# Menentukan warna untuk setiap label
colors = ['red', 'green', 'purple'] # Merah untuk label 1, Hijau untuk
label 0, Ungu untuk label 2

# Plot
ax = counts.plot(kind='bar', color=colors)
plt.title('Distribusi Label')
plt.xlabel('Label')
plt.ylabel('Jumlah')
plt.xticks(rotation=0)

# Menambahkan keterangan di atas setiap batang
for label_count, count in enumerate(counts):
    ax.text(label_count, count + 5, str(label_count), ha='center')

# Menambahkan keterangan di bawah grafik
label_names = ['Rutin Biasa', 'Rutin Kondisi', 'Holding']
for i in range(len(label_names)):
    ax.text(i, 0, label_names[i], ha='center')
plt.xticks(range(len(label_names)), [f"{i} ({label_names[i]})" for i in
[1, 2, 0]]) # Urutan 1, 2, 0

```




```

plt.show()

X = data_jalan.drop("Label",axis=1).to_numpy()
X

y = data_jalan['Label'].to_numpy()
X = data_jalan.drop("Label",axis=1).to_numpy()

data_jalan['Label'] = pd.to_numeric(data_jalan['Label'])

y = data_jalan["Label"].to_numpy()
y[0:5]

from sklearn.model_selection import train_test_split

# Menggunakan train_test_split untuk membagi data dengan perbandingan
90:10
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, stratify=y, random_state=42)

print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)

from sklearn import feature_selection
# Baca file GAPED.csv dari Google Drive
path = '/content/drive/My Drive/Colab Notebooks/feature selection-Chi-
Square.csv'
feature_selection = pd.read_csv(path)

#fill null values
for col in feature_selection.columns:
    feature_selection[col] =
feature_selection[col].fillna(feature_selection[col].mode()[0])

# Menampilkan isi dataframe
feature_selection

# Label encoding
from sklearn.preprocessing import LabelEncoder
for col in feature_selection.columns:
    le = LabelEncoder()
    feature_selection[col] = le.fit_transform(feature_selection[col])

feature_selection[:5]

from sklearn.feature_selection import chi2
feature_selection.drop(columns=['Label'], axis=1)
feature_selection['Label']

chi2_scores = chi2(X, y)

```



```
chi_scores

chi_scores[0]

chi_scores[1]

# higher the chi value, higher the importance
chi_values = pd.Series(chi_scores[0], index=X.columns)
chi_values.sort_values(ascending=False, inplace=True)
chi_values.plot.bar()

# if p-value > 0.5, lower the importance
p_values = pd.Series(chi_scores[1], index=X.columns)
p_values.sort_values(ascending=False, inplace=True)
p_values.plot.bar()

# Menyimpan DataFrame ke file CSV
datas_preprocessings_csv = "feature selection.csv" # Nama file CSV
yang ingin disimpan
path_direktori = "/content/drive/My Drive/Colab Notebooks" # Path
lengkap ke direktori penyimpanan (ubah sesuai dengan direktori Anda)
datas_preprocessings.to_csv(path_direktori + "/" +
datas_preprocessings_csv, index=False) # Menyimpan DataFrame ke file
CSV
```



Lampiran 3 Class C4.5

```

import numpy as np
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import OneHotEncoder

class C45:
    def __init__(self, max_depth=13):
        self.max_depth = max_depth
        self.trees = []

    def fit(self, X, y):
        # One-hot encode the target labels
        enc = OneHotEncoder(sparse=False)
        y_onehot = enc.fit_transform(y.reshape(-1, 1))

        # Initialize the model
        self.f0 = np.mean(y_onehot, axis=0)
        pred = np.full((X.shape[0], y_onehot.shape[1]), self.f0)

        for i in range(self.max_depth):
            trees = []
            for j in range(y_onehot.shape[1]):
                # Fit a new tree to the current class
                tree = DecisionTreeClassifier(criterion='entropy',
max_depth=1)

                tree.fit(X, y_onehot[:, j])

                # Add the new tree to the model for each class
                trees.append(tree)

                # Update the predictions for the current class
                pred[:, j] = tree.predict(X)

            self.trees.append(trees)

    def predict_proba(self, X):
        # Initialize the predictions
        pred = np.zeros((X.shape[0], len(self.trees[-1])))

        # Compute the predictions by summing the outputs of each tree
        for each class
            for i, trees in enumerate(self.trees):
                for j, tree in enumerate(trees):
                    pred[:, j] += tree.predict(X)

        # Normalize the predictions to get class probabilities
        pred = pred / (i + 1)

    def predict(self, X):
        # Predict the class with the highest probability

```



```

        return np.argmax(self.predict_proba(X), axis=1)

#DATA TRAINING
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score
from sklearn.metrics import classification_report, confusion_matrix

# Initialize the C4.5 model using DecisionTreeClassifier
c45_model = DecisionTreeClassifier(max_depth=3)

# Fit the model on the training set
c45_model.fit(X_train, y_train)

# Predict the classes for the testing set
y_pred = c45_model.predict(X_train)

# Evaluate model performance
accuracy_c45 = accuracy_score(y_train, y_pred)
precision_c45 = precision_score(y_train, y_pred, average='weighted')
recall_c45 = recall_score(y_train, y_pred, average='macro')
f1_c45 = f1_score(y_train, y_pred, average='weighted')

print('C4.5 Accuracy:', accuracy_c45)
print('C4.5 Precision:', precision_c45)
print('C4.5 Recall:', recall_c45)
print('C4.5 F1-score:', f1_c45)

print(classification_report(y_train, y_pred))

print(y_pred.shape)

print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)

#DATA TESTING
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

# Initialize the C4.5 model using DecisionTreeClassifier
c45_model = DecisionTreeClassifier(max_depth=3)

# Fit the model on the training set
c45_model.fit(X_train, y_train)

# Predict the classes for the testing set
y_pred = c45_model.predict(X_test)

# Evaluate the accuracy of the model
accuracy_c45 = accuracy_score(y_test, y_pred)
print("Accuracy (C4.5):", accuracy_c45)

```



```

from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score
from sklearn.metrics import classification_report, confusion_matrix

# Initialize the C4.5 model using DecisionTreeClassifier
c45_model = DecisionTreeClassifier(max_depth=3)

# Fit the model on the training set
c45_model.fit(X_train, y_train)

# Predict the classes for the testing set
y_pred = c45_model.predict(X_test)

# Evaluate model performance
accuracy_c45 = accuracy_score(y_test, y_pred)
precision_c45 = precision_score(y_test, y_pred, average='weighted')
recall_c45 = recall_score(y_test, y_pred, average='macro')
f1_c45 = f1_score(y_test, y_pred, average='weighted')

print('C4.5 Accuracy:', accuracy_c45)
print('C4.5 Precision:', precision_c45)
print('C4.5 Recall:', recall_c45)
print('C4.5 F1-score:', f1_c45)

print(classification_report(y_test, y_pred))

import seaborn as sns
import matplotlib.pyplot as plt

# Cetak confusion matrix dalam bentuk gambar untuk model C4.5
cm_c45 = confusion_matrix(y_test, y_pred)
sns.heatmap(cm_c45, annot=True, cmap='Blues', fmt='g',
            annot_kws={"size": 16, "ha": 'center', "va": 'center'})
plt.xlabel('Predicted')
plt.ylabel('Actual')

# Atur label sumbu x dan y
plt.xticks([0.5, 1.5, 2.5], ['rutin biasa (0)', 'rutin kondisi (1)',
                              'holding (2)'])
plt.yticks([0.5, 1.5, 2.5], ['rutin biasa (0)', 'rutin kondisi (1)',
                              'holding (2)'])

plt.title('Confusion Matrix C4.5')

# Import necessary libraries
from joblib import dump

# Save the trained C4.5 model
model_filename = 'c45_model.joblib'
dump(c45_model, model_filename)

print(f"Model saved as {model_filename}")

```



```

# Load the saved model for predictions
loaded_model = joblib.load(model_filename)

# Now, let's make predictions on new data
# Replace the following values with your actual input data
new_data = {
    'Panjang (M)': 1.2,
    'Lebar (M)': 0.5 ,
    'Tebal/Tinggi (M)': 0.06,
    'Volume Kerusakan': 0.0360,
    'Rusak Kiri': 0,
    'Tidak Rusak Kiri': 1,
    'Rusak Kanan': 1,
    'Tidak Rusak Kanan': 0,
    'Ada Kerusakan': 0,
    'Tidak Ada Kerusakan': 1
}

# Convert the input data to a DataFrame
new_data_df = pd.DataFrame([new_data])

# Make predictions using the loaded model
new_predictions = loaded_model.predict(new_data_df)

# Display the predicted class
predicted_class = new_predictions[0]

print("Predicted Class:", predicted_class)

import pandas as pd
from IPython.display import display
import joblib

# Define a function to predict and add the label to the dataset
def predict_and_add_label(new_data, loaded_model):
    # Convert the input data to a DataFrame
    new_data_df = pd.DataFrame([new_data])

    # Make predictions using the loaded model
    new_predictions = loaded_model.predict(new_data_df)

    # Get the class labels from the model
    class_labels = loaded_model.classes_

    # Convert numerical prediction to the corresponding class label
    predicted_class = class_labels[new_predictions[0]]

    # Add new column "Label Prediksi Model" to the DataFrame
    new_data_df['Label Prediksi Model'] = predicted_class

    return new_data_df

```



```

# Assuming 'new_data' is available from the previous example
# Load the saved model for predictions
loaded_model = joblib.load(model_filename)

# Call the function to predict and add the label
result_dataset = predict_and_add_label(new_data, loaded_model)

# Display the result dataset
print("Result Dataset with Predicted Label:")
display(result_dataset)

# Load the saved model for predictions
loaded_model = joblib.load(model_filename)

# Now, let's make predictions on new data
# Replace the following values with your actual input data
new_data = {
    'Panjang (M)': 1.2,
    'Lebar (M)': 0.5 ,
    'Tebal/Tinggi (M)': 0.06,
    'Volume Kerusakan': 0.0360,
    'Rusak Kiri': 0,
    'Tidak Rusak Kiri': 1,
    'Rusak Kanan': 1,
    'Tidak Rusak Kanan': 0,
    'Ada Kerusakan': 0,
    'Tidak Ada Kerusakan': 1
}

# Convert the input data to a DataFrame
new_data_df = pd.DataFrame([new_data])

# Make predictions using the loaded model
new_predictions = loaded_model.predict(new_data_df)

# Display the predicted class
predicted_class = new_predictions[0]

if predicted_class == 1:
    print("Predicted Class: Ada Kerusakan")
else:
    print("Predicted Class: Tidak Ada Kerusakan")

import pandas as pd
display import display

# Load the saved model for predictions
loaded_model = joblib.load(model_filename)

# Now, let's make predictions on new data
# Replace the following values with your actual input data
new_data = {

```




```

    'Panjang (M)': 1.2,
    'Lebar (M)': 0.5 ,
    'Tebal/Tinggi (M)': 0.06,
    'Volume Kerusakan': 0.0360,
    'Rusak Kiri': 0,
    'Tidak Rusak Kiri': 1,
    'Rusak Kanan': 1,
    'Tidak Rusak Kanan': 0,
    'Ada Kerusakan': 0,
    'Tidak Ada Kerusakan': 1
}

# Convert the input data to a DataFrame
new_data_df = pd.DataFrame([new_data])

# Make predictions using the loaded model
predicted_class = loaded_model.predict(new_data_df)[0]

# Assuming 'lokasi_jalan.csv' is available in the same directory as
this script
lokasi_jalan_file = '/content/drive/My Drive/Colab
Notebooks/lokasi_jalan.csv'

# Load the 'lokasi_jalan.csv' dataset
lokasi_jalan_df = pd.read_csv(lokasi_jalan_file)

# Filter 'lokasi_jalan.csv' based on the input values
filtered_info = lokasi_jalan_df[
    (lokasi_jalan_df['Panjang (M)'] == new_data['Panjang (M)']) &
    (lokasi_jalan_df['Lebar (M)'] == new_data['Lebar (M)']) &
    (lokasi_jalan_df['Tebal/Tinggi (M)'] == new_data['Tebal/Tinggi
(M)']) &
    (lokasi_jalan_df['Volume Kerusakan'] == new_data['Volume
Kerusakan']) &
    (lokasi_jalan_df['Rusak Kiri'] == new_data['Rusak Kiri']) &
    (lokasi_jalan_df['Tidak Rusak Kiri'] == new_data['Tidak Rusak
Kiri']) &
    (lokasi_jalan_df['Rusak Kanan'] == new_data['Rusak Kanan']) &
    (lokasi_jalan_df['Tidak Rusak Kanan'] == new_data['Tidak Rusak
Kanan']) &
    (lokasi_jalan_df['Ada Kerusakan'] == new_data['Ada Kerusakan']) &
    (lokasi_jalan_df['Tidak Ada Kerusakan'] == new_data['Tidak Ada
Kerusakan'])
]

# Print the filtered information including nama wilayah and
# Print the filtered information based on Input Values:")
print(filtered_info[['Nama Wilayah', 'Sub Kategori']])

```



```

from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split

```

```
# Bagi data menjadi training dan testing
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Inisialisasi model C4.5 (DecisionTreeClassifier)
c45_model = DecisionTreeClassifier(max_depth=3)

# Mulai mengukur waktu
start_time = time.time()

# Latih model pada data training
c45_model.fit(X_train, y_train)

# Prediksi kelas untuk data testing
y_pred = c45_model.predict(X_test)

# Selesai mengukur waktu
end_time = time.time()

# Hitung waktu yang diperlukan
elapsed_time = end_time - start_time
print("Waktu yang diperlukan untuk klasifikasi:", elapsed_time,
"detik")
```



Lampiran 4 Class Random Forest

```

from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
from collections import Counter
from sklearn.tree import DecisionTreeClassifier

class RandomForestClassifier:
    def __init__(self, n_estimators=100, max_depth=None,
min_samples_split=2, random_state=None):
        self.n_estimators = n_estimators
        self.max_depth = max_depth
        self.min_samples_split = min_samples_split
        self.random_state = random_state
        self.trees = []

    def fit(self, X, y):
        np.random.seed(self.random_state)
        for i in range(self.n_estimators):
            # bootstrap sample
            idx = np.random.choice(X.shape[0], size=X.shape[0],
replace=True)
            X_boot, y_boot = X[idx], y[idx]

            # train decision tree
            dt = DecisionTreeClassifier(max_depth=self.max_depth,
min_samples_split=self.min_samples_split)
            dt.fit(X_boot, y_boot)

            # save tree
            self.trees.append(dt)

    def predict(self, X):
        # make predictions for each tree
        predictions = np.array([tree.predict(X) for tree in
self.trees])
        # calculate mode (most frequent class) for each sample
        return np.apply_along_axis(lambda x:
Counter(x).most_common(1)[0][0], axis=0, arr=predictions)

#DATA TRAINING
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, precision_score,
f1_score
from sklearn.metrics import classification_report, confusion_matrix

# Create the Random Forest model
rf = RandomForestClassifier(n_estimators=100, max_depth=3)

# Train the model on the training set
rf.fit(X_train, y_train)

```



```

# Predict the classes for the testing set
y_pred = rf_model.predict(X_test)

# Evaluate model performance
accuracy_rf = accuracy_score(y_train, y_pred)
precision_rf = precision_score(y_train, y_pred, average='weighted')
recall_rf = recall_score(y_train, y_pred, average='macro')
f1_rf = f1_score(y_train, y_pred, average='weighted')

print('Random Forest Accuracy:', accuracy_rf)
print('Random Forest Precision:', precision_rf)
print('Random Forest Recall:', recall_rf)
print('Random Forest F1-score:', f1_rf)

print(classification_report(y_train, y_pred))

#DATA TESTING
RandomForest = RandomForestClassifier(n_estimators=200, max_depth=13,
min_samples_split=2, random_state=42)
RandomForest.fit(X_train, y_train)

y_pred = RandomForest.predict(X_test)

acc = np.mean(y_pred == y_test)
print("Accuracy:", acc)

from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score
from sklearn.metrics import classification_report

print("Random Forest Accuracy: ", accuracy_score(y_test, y_pred))
print("Random Forest Precision: ", precision_score(y_test,
y_pred, average='weighted'))
print("Random Forest Recall: ", recall_score(y_test, y_pred,
average='weighted'))
print("Random Forest f1_score: ", f1_score(y_test, y_pred,
average='weighted'))
print(classification_report(y_test, y_pred))

import seaborn as sns
import matplotlib.pyplot as plt

# Mengganti model RandomForestClassifier dengan model yang sesuai
from sklearn.ensemble import RandomForestClassifier

# Model Random Forest
rf_clf = RandomForestClassifier(n_estimators=100, random_state=42)

# Melatih model Random Forest
rf_clf.fit(X_train, y_train)

# Melakukan prediksi menggunakan model Random Forest
y_pred_rf = rf_clf.predict(X_test)

```



```

# Membuat confusion matrix
cm_rf = confusion_matrix(y_test, y_pred_rf)

# Plot confusion matrix dalam bentuk gambar
sns.heatmap(cm_rf, annot=True, cmap='Blues', fmt='g',
             annot_kws={"size": 16, "ha": 'center', "va": 'center'})
plt.xlabel('Predicted')
plt.ylabel('Actual')

# Atur label sumbu x dan y
plt.xticks([0.5, 1.5, 2.5], ['rutin biasa (0)', 'rutin kondisi (1)',
                              'holding (2)'])
plt.yticks([0.5, 1.5, 2.5], ['rutin biasa (0)', 'rutin kondisi (1)',
                              'holding (2)'])

plt.title('Random Forest Confusion Matrix')
plt.show()

# Save the trained Random Forest model with its fitted state
model_filename = 'random_forest_model.joblib'
joblib.dump(loader_model, model_filename)
print(f"Model saved as {model_filename}")

import pickle

# Save the trained Random Forest model with its fitted state
model_filename = 'random_forest_model.pkl'
with open(model_filename, 'wb') as file:
    pickle.dump(loader_model, file)
print(f"Model saved as {model_filename}")

from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
import joblib

# Assuming you have your training data and labels
# X_train, y_train = ...

# Create and train the RandomForestClassifier
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Save the trained model
joblib.dump(model, model_filename)

# Load the saved model for predictions
loader_model = joblib.load(model_filename)

# Use the loaded model to make predictions on new data
# Replace the following values with your actual input data
new_data = {

```



```

    'Panjang (M)': 2.0,
    'Lebar (M)': 0.7,
    'Tebal/Tinggi (M)': 0.06,
    'Volume Kerusakan': 0.0840,
    'Rusak Kiri': 0,
    'Tidak Rusak Kiri': 1,
    'Rusak Kanan': 1,
    'Tidak Rusak Kanan': 0,
    'Ada Kerusakan': 1,
    'Tidak Ada Kerusakan': 0
}

# Convert the input data to a DataFrame
new_data_df = pd.DataFrame([new_data])

# Make predictions using the loaded model
new_predictions = loaded_model.predict(new_data_df)

# Display the predicted class
predicted_class = new_predictions[0]

print("Predicted Class:", predicted_class)

import pandas as pd
from IPython.display import display
import joblib

# Define a function to predict and add the label to the dataset
def predict_and_add_label(new_data, loaded_model):
    # Convert the input data to a DataFrame
    new_data_df = pd.DataFrame([new_data])

    # Make predictions using the loaded model
    new_predictions = loaded_model.predict(new_data_df)

    # Get the class labels from the model
    class_labels = loaded_model.classes_

    # Map the numerical prediction to the corresponding class label
    predicted_class = class_labels[new_predictions[0]]

    # Add a new column "Label Prediksi Model" to the DataFrame
    new_data_df['Label Prediksi Model'] = predicted_class

new_data_df

# 'new_data' is available from the previous example
# 'loaded_model' is the saved model for predictions
loaded_model = joblib.load(model_filename)

# Define a function to predict and add the label
def predict_and_add_label(new_data, loaded_model):
    result_dataset = predict_and_add_label(new_data, loaded_model)

```



```

# Display the result dataset
print("Result Dataset with Predicted Label:")
display(result_dataset)

# Load the saved model for predictions
loaded_model = joblib.load(model_filename)

# Now, let's make predictions on new data
# Replace the following values with your actual input data
new_data = {
    'Panjang (M)': 1.2,
    'Lebar (M)': 0.5,
    'Tebal/Tinggi (M)': 0.06,
    'Volume Kerusakan': 0.036,
    'Rusak Kiri': 1,
    'Tidak Rusak Kiri': 1,
    'Rusak Kanan': 1,
    'Tidak Rusak Kanan': 1,
    'Ada Kerusakan': 1,
    'Tidak Ada Kerusakan': 0
}

# Convert the input data to a DataFrame
new_data_df = pd.DataFrame([new_data])

# Make predictions using the loaded model
new_predictions = loaded_model.predict(new_data_df)

# Display the predicted class
predicted_class = new_predictions[0]

if predicted_class == 1:
    print("Predicted Class: Ada Kerusakan")
else:
    print("Predicted Class: Tidak Ada Kerusakan")

import time
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split

# Bagi data menjadi training dan testing
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.2, random_state=42)

```



Optimized using
trial version
www.balesio.com

```

# Si model Random Forest
random_forest_model = RandomForestClassifier(n_estimators=100,
                                             random_state=42)

# Ukur waktu
start_time = time.time()

# Latih model pada data training
random_forest_model.fit(X_train, y_train)

```

```
# Prediksi kelas untuk data testing
y_pred = random_forest_model.predict(X_test)

# Selesai mengukur waktu
end_time = time.time()

# Hitung waktu yang diperlukan
elapsed_time = end_time - start_time
print("Waktu yang diperlukan untuk klasifikasi dengan Random Forest:",
elapsed_time, "detik")
```



Lampiran 5 Klasifikasi

```

from sklearn.model_selection import train_test_split

# Pisahkan dataset menjadi fitur (X) dan label (y)
X = data_jalan.drop(columns=['Label']) # Sesuaikan dengan nama kolom
label sebenarnya
y = data_jalan['Label']

# Bagi data menjadi data pelatihan dan data pengujian
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

from sklearn.tree import DecisionTreeClassifier

# Inisialisasi model C4.5
c45_model = DecisionTreeClassifier()

# Latih model C4.5 dengan data pelatihan
c45_model.fit(X_train, y_train) # Pastikan Anda telah mendefinisikan
X_train dan y_train sebelumnya

# Buat prediksi menggunakan model C4.5 pada seluruh data
c45_predictions = c45_model.predict(X)

# Tambahkan kolom "Label Jalan raya" ke DataFrame
data_jalan['Label Jalan Raya'] = c45_predictions

# Simpan DataFrame dengan kolom baru ke file CSV
data_jalan.to_csv(path + 'pupr_jalan_c45.csv', index=False)

# Membaca dataset
path = '/content/drive/My Drive/Colab Notebooks/'
data_jalan = pd.read_csv(path + 'pupr_jalan_c45.csv', encoding="ISO-
8859-1")
data_jalan

print(data_jalan['Label Jalan Raya'].value_counts())
#print(df_dataTraining.value_counts())

# Tambahkan kolom "Label jalan raya" ke DataFrame
data_preprocessing['Label Jalan Raya'] = c45_predictions

# Simpan DataFrame dengan kolom baru ke file CSV
data_preprocessing.to_csv(path + 'data prediksi jalan.csv',
encoding='utf-8')

# Membaca dataset
path = '/content/drive/My Drive/Colab Notebooks/'
data_preprocessing = pd.read_csv(path + 'data prediksi jalan.csv',
encoding="ISO-8859-1")
data_preprocessing

```



```

# Fungsi untuk menentukan kualitas berdasarkan rentang
def keterangan_label(label):
    if label == 0:
        return "Rutin Biasa"
    elif label == 1:
        return "Rutin Kondisi"
    elif label == 2:
        return "Holding"
    else:
        return "Label tidak valid"

# Terapkan fungsi pada kolom 'Kualitas'
data_klasifikasi['Keterangan Label'] = data_klasifikasi['Label Jalan
Raya'].apply(keterangan_label)

# Tampilkan DataFrame hasil
data_klasifikasi

# Mengambil nama-nama wilayah unik
nama_wilayah = data_klasifikasi['Nama Wilayah'].unique()

# Menampilkan nama-nama wilayah
print("Nama-nama wilayah:")
for nama in nama_wilayah:
    print(nama)

data_klasifikasi.info()

import pandas as pd

# Definisi wilayah per kabupaten
wilayah_per_kabupaten = {
    'Kabupaten Luwu Utara': ['Baliase', 'Bone', 'Bone-Bone', 'Buntu
Terpedo', 'Kalotok', 'Lampuawa', 'Lero', 'Malimbu', 'Pongko', 'Radda',
'Sabbang', 'Baebunta', 'Kappuna', 'Kasimbong', 'Lodang', 'Mappedeceng',
'Mari-Mari', 'Marobo', 'Masamba', 'Minanga Tallu', 'Patila',
'Patoloan', 'Pompaniki', 'Saptamarga', 'Sidomukti', 'Sukamaju',
'Sumilin', 'Tamboke', 'Terpedo Jaya', 'Uraso', 'Buangin', 'Salassa'],
    'Kabupaten Luwu Timur': ['Benteng', 'Bonepute', 'Harapan',
'Kasintuwu', 'Kayulangi', 'Lambarese', 'Lampenai', 'Malili',
'Maliwowo', 'Margolembo', 'Pongkeru', 'Tampinna', 'Tarengge', 'Wotu',
'Angkona', 'Batuputih', 'Bawalipu', 'Jalajja', 'Kalaena', 'Karambua',
'Lamaeto', 'Lanosu', 'Laskap', 'Lewonu', 'Lumbewe', 'Mangkutana'],
    'Kabupaten Luwu': ['Bolong', 'Salulino', 'Walenrang', 'Bosso',
'amara', 'Marabuana', 'Salutubu', 'Buntu Awo', 'Barammase',
'ak'],
    'Palopo': ['Bara', 'Batu Walenrang', 'Palopo', 'Telluanua']
}

# Fungsi untuk menentukan keterangan berdasarkan nama wilayah
def keterangan_wilayah(nama_wilayah):
    for wilayah in wilayah_per_kabupaten['Kabupaten Luwu Utara']:
        if nama_wilayah == wilayah:
            return "Kabupaten Luwu Utara"
    elif nama_wilayah in wilayah_per_kabupaten['Kabupaten Luwu Timur']:

```



```

        return 'Kabupaten Luwu Timur'
    elif nama_wilayah in wilayah_per_kabupaten['Kabupaten Luwu']:
        return 'Kabupaten Luwu'
    elif nama_wilayah in wilayah_per_kabupaten['Kota Palopo']:
        return 'Kota Palopo'
    else:
        return 'Wilayah tidak valid'

# Menambahkan kolom keterangan wilayah ke dalam DataFrame
data_klasifikasi['Kabupaten'] = data_klasifikasi['Nama
Wilayah'].apply(nama_wilayah_per_kabupaten)

# Menampilkan DataFrame hasil
print(data_klasifikasi)

#check missing value
data_klasifikasi.isnull().sum()

# Menyimpan DataFrame ke file CSV
data_klasifikasi_fix_csv = "data_klasifikasi_fix.csv" # Nama file CSV
yang ingin disimpan
path_direktori = "/content/drive/My Drive/Colab Notebooks/" # Path
lengkap ke direktori penyimpanan (ubah sesuai dengan direktori Anda)
data_klasifikasi.to_csv(path_direktori + "/" +
data_klasifikasi_fix_csv, index=False) # Menyimpan DataFrame ke file
CSV

import numpy as np
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import LabelEncoder
import pandas as pd

# Membaca dataset
path = '/content/drive/My Drive/Colab Notebooks/'
data_klasifikasi = pd.read_csv(path + 'data prediksi jalan.csv',
encoding="ISO-8859-1")

# Inisialisasi model C4.5 (Decision Tree)
c45_clf = DecisionTreeClassifier()

encoder_kategori = LabelEncoder()

data_klasifikasi['Sub Kategori'] =
encoder_kategori.fit_transform(data_klasifikasi['Sub Kategori'])

# Mengganti nilai NaN dengan nilai NaN di 'Label C4.5'
data_klasifikasi.dropna(subset=['Label Jalan Raya'], inplace=True)

data_klasifikasi[['Sub Kategori']]
data_klasifikasi['Label Jalan Raya']

# Latih model C4.5 dengan data pelatihan

```



```

c45_clf.fit(X_train, y_train)

kategori_label_counts = data_klasifikasi.groupby(['Sub Kategori',
'Label Jalan Raya']).size().unstack(fill_value=0)

# Menghitung jumlah total dari setiap label
total_label0 = kategori_label_counts[0].sum()
total_label1 = kategori_label_counts[1].sum()
total_label2 = kategori_label_counts[2].sum()

# Membuat diagram lingkaran
fig, ax = plt.subplots(figsize=(8, 8))
colors = plt.cm.Set3.colors
wedges, texts, autotexts = ax.pie([total_label0, total_label1,
total_label2], labels=['Label 0', 'Label 1', 'Label 2'],
autopct='%1.1f%%', startangle=140, colors=colors)
ax.set_title('Proporsi Label Jalan Raya 0, 1, dan 2')
plt.setp(autotexts, size=8, weight="bold")
ax.axis('equal') # Membuat diagram lingkaran menjadi bulat sempurna
plt.show()

import matplotlib.pyplot as plt
import pandas as pd
from sklearn.preprocessing import LabelEncoder

# Membaca dataset
path = '/content/drive/My Drive/Colab Notebooks/'
data_klasifikasi = pd.read_csv(path + 'data prediksi jalan.csv',
encoding="ISO-8859-1")

# Konversi kategori ke numerik menggunakan LabelEncoder
encoder_kategori = LabelEncoder()

data_klasifikasi['Sub Kategori'] =
encoder_kategori.fit_transform(data_klasifikasi['Sub Kategori'])

# Membuat filter untuk data dengan label 0
data_label0 = data_klasifikasi[data_klasifikasi['Label Jalan Raya'] ==
2]

# Menghitung jumlah sub kategori dengan label 0
kategori_label0_counts = data_label0['Sub Kategori'].value_counts()

# Membuat diagram lingkaran
fig, ax = plt.subplots(figsize=(8, 8))
wedges, texts, autotexts = ax.pie(kategori_label0_counts.values, startangle=140)
ax.set_title('Proporsi Sub Kategori dengan Label Rutin Biasa')
plt.setp(autotexts, size=8, weight="bold")
ax.axis('equal') # Membuat diagram lingkaran menjadi bulat sempurna
plt.show()

# Membuat persentase dan membuat label untuk legend
percentages =
100.*kategori_label0_counts.values/kategori_label0_counts.values.sum()

```



```

# Modifikasi label legend untuk menambahkan jumlah data
legend_labels = ['{0} - {1:1.1f} % ({2} ruas)'.format(i,j,k) for i,j,k
in zip(encoder_kategori.classes_[kategori_label0_counts.index],
percentages, kategori_label0_counts.values)]

# Menambahkan keterangan/legend dengan sub kategori, persentase, dan
jumlah data
ax.legend(wedges, legend_labels,
          title="Sub Kategori",
          loc="center left",
          bbox_to_anchor=(1, 0, 0.5, 1))

plt.show()

import matplotlib.pyplot as plt
import pandas as pd
from sklearn.preprocessing import LabelEncoder

# Membaca dataset
path = '/content/drive/My Drive/Colab Notebooks/'
data_klasifikasi = pd.read_csv(path + 'data prediksi jalan.csv',
encoding="ISO-8859-1")

# Konversi kategori ke numerik menggunakan LabelEncoder
encoder_kategori = LabelEncoder()

data_klasifikasi['Sub Kategori'] =
encoder_kategori.fit_transform(data_klasifikasi['Sub Kategori'])

# Membuat filter untuk data dengan label 1
data_label0 = data_klasifikasi[data_klasifikasi['Label Jalan Raya'] ==
1]

# Menghitung jumlah sub kategori dengan label 1
kategori_label0_counts = data_label0['Sub Kategori'].value_counts()

# Membuat diagram lingkaran
fig, ax = plt.subplots(figsize=(8, 8))
wedges, texts = ax.pie(kategori_label0_counts.values, startangle=140)

# Menambahkan judul
ax.set_title('Proporsi Sub Kategori dengan Label Rutin Kondisi')

persentase dan membuat label untuk legend
=
_label0_counts.values/kategori_label0_counts.values.sum()

label legend untuk menambahkan jumlah data
s = ['{0} - {1:1.1f} % ({2} ruas)'.format(i,j,k) for i,j,k
er_kategori.classes_[kategori_label0_counts.index],
percentages, kategori_label0_counts.values)]

```



```

# Menambahkan keterangan/legend dengan sub kategori, persentase, dan
jumlah data
ax.legend(wedges, legend_labels,
          title="Sub Kategori",
          loc="center left",
          bbox_to_anchor=(1, 0, 0.5, 1))

plt.show()

import matplotlib.pyplot as plt
import pandas as pd
from sklearn.preprocessing import LabelEncoder

# Membaca dataset
path = '/content/drive/My Drive/Colab Notebooks/'
data_klasifikasi = pd.read_csv(path + 'data prediksi jalan.csv',
                                encoding="ISO-8859-1")

# Konversi kategori ke numerik menggunakan LabelEncoder
encoder_kategori = LabelEncoder()

data_klasifikasi['Sub Kategori'] =
encoder_kategori.fit_transform(data_klasifikasi['Sub Kategori'])

# Membuat filter untuk data dengan label 2
data_label0 = data_klasifikasi[data_klasifikasi['Label Jalan Raya'] ==
0]

# Menghitung jumlah sub kategori dengan label 2
kategori_label0_counts = data_label0['Sub Kategori'].value_counts()

# Membuat diagram lingkaran
fig, ax = plt.subplots(figsize=(8, 8))
wedges, texts = ax.pie(kategori_label0_counts.values, startangle=140)

# Menambahkan judul
ax.set_title('Proporsi Sub Kategori dengan Label Holding')

# Menghitung persentase dan membuat label untuk legend
percentages =
100.*kategori_label0_counts.values/kategori_label0_counts.values.sum()

# Modifikasi label legend untuk menambahkan jumlah data
s = ['{0} - {1:1.1f} % ({2} ruas)'.format(i,j,k) for i,j,k
    encoder_kategori.classes_[kategori_label0_counts.index],
    kategori_label0_counts.values)]

# Menambahkan keterangan/legend dengan sub kategori, persentase, dan
jumlah data
ax.legend(wedges, legend_labels,
          title="Sub Kategori",
          loc="center left",

```



```

        bbox_to_anchor=(1, 0, 0.5, 1))

plt.show()

import matplotlib.pyplot as plt

# Data
metrics = ['Presisi', 'Recall', 'F1-Score', 'Akurasi']
c45 = [87.9, 82.6, 87.8, 88.0]
randomforest = [86.6, 86.8, 86.6, 86.8]

# Plotting
plt.figure(figsize=(10, 6))
plt.plot(metrics, c45, marker='o', label='C4.5')
plt.plot(metrics, randomforest, marker='o', label='RandomForest')
plt.ylabel('Persentase (%)')
plt.title('Perbandingan Kinerja Model')
plt.legend()
plt.grid(True, which='both', linestyle='--', linewidth=0.5)
plt.show()

import matplotlib.pyplot as plt

# Data
metrics = ['Presisi', 'Recall', 'F1-Score', 'Akurasi']
c45 = [87.9, 82.6, 87.8, 88.0]
randomforest = [86.6, 86.8, 86.6, 86.8]

# Transposed data for the new layout
data = [
    c45,
    randomforest
]

# Algorithms for the x-axis
algorithms = ['C4.5', 'Random Forest']

# Setting the positions of the bars on the x-axis
bar_width = 0.2
indices = range(len(algorithms))

# Colors for the metrics
metric_colors = ['#ff0000', '#00ff00', '#0000ff', '#7f007f'] #
# to Presisi, Recall, F1-Score, Akurasi

plt.figure(figsize=(10, 6))

# Create bars with labels for each metric
for i, metric in enumerate(metrics):
    for algorithm, data in enumerate(algorithms):
        data = [algorithm[i] for algorithm in data]

```



```

    bar_positions = [x + (bar_width * i) - (bar_width * 1.5) for x in
indices]
    bars = plt.bar(bar_positions, metric_data, width=bar_width,
color=metric_colors[i], label=metric)

    # Adding value labels on top of each bar
    for bar, value in zip(bars, metric_data):
        plt.text(bar.get_x() + bar.get_width() / 2, value + 1,
str(value), ha='center', color='black', fontsize=8)

# Setting the labels for the x-axis
plt.xticks(ticks=[i - bar_width / 2 for i in indices],
labels=algorithms)

# Adding labels and title
plt.ylabel('Persentase (%)')
plt.title('Perbandingan Kinerja Model')

# Moving the legend to the right side of the plot
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))

plt.grid(True, which='both', linestyle='--', linewidth=0.5)
plt.tight_layout() # Adjust layout to make room for the legend
plt.show()

#menampilkan hasil
print("Jumlah data training :", jumlah_training)
print("Jumlah data testing :", jumlah_testing)

df = datas_preprocessing.copy()
location_df = df[["Nama Wilayah", "Kabupaten"]].copy()
location_df = location_df.drop_duplicates(subset=["Nama Wilayah",
"Kabupaten"])
location_df.reset_index(drop=True, inplace=True)
print(location_df.shape)
location_df.head()

import geopy
from geopy.geocoders import Nominatim
from geopy.exc import GeocoderTimeout
from datetime import datetime
from tqdm import tqdm_notebook

location_df['lat'] = ''
location_df['long'] = ''

time = datetime.now().strftime("%H:%M:%S")

geocoder = Nominatim(user_agent="my_app4")

for i in tqdm_notebook(range(len(location_df))):

```




```

        address = (location_df["Nama Wilayah"][i]+' ', '+
location_df["Kabupaten"][i]).strip()
        location = locator.geocode(address, timeout=8)

        if location != None:
            location_df['lat'][i] = location.latitude
            location_df['long'][i] = location.longitude

        else:
            print(f"[{count}]. {address}")
            location_df['lat'][i] = np.nan
            location_df['long'][i] = np.nan
            count += 1
        except GeocoderTimedOut as e:
            print("Error: geocode failed on input %s with message
%s"%(address, e.message))

    print("start:", start_time)
    print("finish: ", datetime.now().strftime("%H:%M:%S"))
    location_df.head()

    hasil = pd.concat([multilabel_binarizer_csv, data_jalan["Label"]],
axis=1)
    hasil = hasil.drop(["Rusak", "Tidak Rusak", "Ada Kerusakan", "Tidak
Ada Kerusakan"], axis=1)
    hasil = hasil.merge(location_df, how='left', left_on=['Nama
Wilayah', 'Kabupaten'], right_on=['Nama Wilayah', 'Kabupaten'])
    hasil = hasil.dropna()

    column_to_move = 'Label'
    hasil[column_to_move] = hasil.pop(column_to_move)

    ket= {0: 'Rutin Biasa', 1: 'Rutin Kondisi', 2: 'Holding'}

    hasil['ket'] = hasil['Label'].map(ket)
    column_to_move1 = 'ket'
    hasil[column_to_move1] = hasil.pop(column_to_move1)

    hasil[:4]

    hasil.to_json(r'./result/hasil.json')

    return {}

```



Lampiran 6 Surat Penugasan Bimbingan



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

Nomor : 621/UN4.7.7/ TD.06/2023
Lamp : -
Hal : Penugasan Bimbingan Tugas Akhir

Kepada Yth : Bapak Wakil Dekan Bidang Akademik
Fakultas Teknik UNHAS
Di-
Makassar

Dengan hormat,
Kiranya dosen pembimbing tugas akhir (skripsi) dari mahasiswa :

Nama : Putri Ayu Nengsih
Stambuk : D121 19 1070
Departemen : Teknik Informatika

Dengan judul Tugas Akhir :

**“ Perbandingan Algoritma C4.5 dan Random Forest Pada Pemetaan Kondisi
Kerusakan Jalan Berbasis WebGIS (Studi Kasus : Luwu Raya) ”**

Dosen Pembimbing :

- | | |
|-----------------------------------|----------|
| 1. Dr. Ir. Ingrid Nurtanio, M.T. | Pemb. I |
| 2. Dr. Ir. Zahir Zainuddin, M.Sc. | Pemb. II |

Dapat dibuatkan Surat Penugasan Bimbingan Tugas Akhir.
Demikian penyampaian kami, atas bantuan dan perhatiannya diucapkan terima kasih.

Gowa, 25 Juli 2023
Ketua Departemen
Teknik Informatika FT-UH



Prof. Dr. Ir. Indrabayu.,ST, MT, M.Bus.Sys., IPM, ASEAN.Eng
Nip.197507162002121004

Tembusan :
1. Arsip



Lampiran 7 Surat Penugasan



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

Poros Malino Km.6Bontomarannu(92172) Gowa, Sulawesi Selatan 92172, Sulawesi Selatan
Telp. (0411) 586015, 586262 Fax (0411) 586015
<http://eng.unhas.ac.id>, Email : teknik@unhas.ac.id

SURAT PENUGASAN

No. 15929/UN4.7.1/TD.06/2023

Dari : Dekan Fakultas Teknik Universitas Hasanuddin

Kepada : 1. Dr. Ir. Ingrid Nurtanio, M.T Pemb. I
2. Dr. Ir. Zahir Zainuddin, M.Sc Pemb. II

Isi : 1. Berdasarkan Surat Ketua Departemen Teknik Informatika Fakultas Teknik Nomor. 621/UN4.7.7/TD.06/2023 tanggal 25 Juli 2023 tentang usul DOSEN PEMBIMBING MAHASISWA, maka dengan ini kami menugaskan Saudara untuk membimbing penulisan Skripsi/Tugas Akhir mahasiswa Teknik Informatika Fakultas Teknik Universitas Hasanuddin di bawah ini :

Nama : No. Stambuk :
Putri Ayu Nengsih D121 19 1070

Judul Skripsi/Tugas Akhir :

“ Perbandingan Algoritma C4.5 dan Random Forest Pada Pemetaan Kondisi Kerusakan Jalan Berbasis WebGIS (Studi Kasus : Luwu Raya) ”

2. Surat penugasan pembimbing ini mulai berlaku sejak tanggal ditetapkannya dan berakhir sampai selesainya penulisan Skripsi/Tugas Akhir mahasiswa tersebut.
3. Agar penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.

Ditetapkan di Gowa

Pada tanggal 25 Juli 2023

a.n. Dekan,

Wakil Dekan Bidang Akademik dan Kemahasiswaan
Fakultas Teknik Unhas



Dr. Amil Ahmad Ilham, ST., M.IT
NIP. 197310101998021001

Tembusan :

1. Dekan FT-UH,
2. Ketua Departemen Teknik Informatika FT-UH,
3. Mahasiswa yang bersangkutan



• Dokumen ini telah ditandatangani secara elektronik menggunakan sertifikat elektronik yang diterbitkan BSrE
• UU ITE No 11 Tahun 2008 Pasal 5 Ayat 1
"Informasi Elektronik dan/atau Dokumen Elektronik dan/atau hasil cetaknya merupakan alat bukti hukum yang sah"

Lampiran 8 Daftar Hadir Seminar Hasil



KEMENTERIAN PENDIDIKAN , KEBUDAYAAN
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

DAFTAR HADIR SEMINAR HASIL

Nama/Stambuk : 1. Putri Ayu Nengsih D121191070

Judul Skripsi/T.A : "Perbandingan Algoritma C4.5 dan Random Forest pada Pemetaan Kondisi Kerusakan Jalan Berbasis WebGIS (Studi Kasus : Luwu Raya)"

Hari/Tanggal : Jumat, 17 Mei 2024

Jam : 13.00 Wita – Selesai

Tempat : Ruang Lab. CBS Departemen Teknik Informatika Gowa

No.	Jabatan	Nama Dosen	Tanda Tangan
L.	Pembimbing I	1. Dr. Ir. Ingrid Nurtanio, M.T	1.
	Pembimbing II	2. Dr. Ir. Zahir Zainuddin, M.Sc	2.
II.	Anggota Penguji	3. Prof. Dr. Ir. Indrabayu, ST., M.T.M. Bus. Sys. IPM. ASEAN. Eng	3.
		4. Dr. Eng. Zulkifli Tahir, ST., M.Sc	4.

PANITIA UJIAN

Ketua,

 Dr. Ir. Ingrid Nurtanio, M.T

Sekretaris,

 Dr. Ir. Zahir Zainuddin, M.Sc


Lampiran 9 Berita Acara Seminar Hasil



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

BERITA ACARA SEMINAR HASIL

Pada hari ini **Jumat**, tanggal 17 Mei 2024 Pukul 13.00 WITA - Selesai bertempat di Ruang Lab.CBS Departemen Teknik Informatika Gowa, telah dilaksanakan Seminar Hasil bagi Saudara :

Nama : Putri Ayu Nengsih
No. Stambuk : D121191070
Fakultas/Departemen : Teknik/Teknik Informatika
Judul Skripsi : "Perbandingan Algoritma C4.5 dan Random Forest pada Pemetaan Kondisi Kerusakan Jalan Berbasis WebGIS (Studi Kasus : Luwu Raya)"

Yang dihadiri oleh Tim Penguji Seminar Hasil sebagai berikut :

No.	Nama	Jabatan	Tanda tangan
1.	Dr. Ir. Ingrid Nurtanio, M.T	Pemb I/Ketua	1.
2.	Dr. Ir. Zahir Zainuddin, M.Sc	Pemb II/Sekretaris	2.
3.	Prof. Dr. Ir. Indrabayu, ST., M.T.M.Bus.Sys.IPM.ASEAN.Eng	Anggota	3.
4.	Dr. Eng. Zulkifli Tahir, ST., M.Sc	Anggota	4.

Hasil keputusan Tim Penguji Seminar Hasil : **Lulus** / ~~Tidak Lulus~~ dengan nilai angka **A** dan huruf **86,75**

Gowa, 17 Mei 2024

Ketua/Sekretaris Panitia Ujian,

Dr. Ir. Ingrid Nurtanio, M.T



Lampiran 10 Surat Penerbitan Panitia/Penguji Seminar Hasil



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA**

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

Nomor : 585/UN4.7.7/TD.06/2024
Lamp : -
Hal : Penerbitan Surat Penugasan Panitia/Penguji
Seminar Hasil Strata Satu (S1)

Kepada Yth :

Wakil Dekan Bidang Akademik dan Kemahasiswaan
Fakultas Teknik Universitas Hasanuddin

Di-

Gowa

Dengan hormat,

Berdasarkan Persetujuan Pembimbing Mahasiswa, Bersama ini diusulkan susunan Panitia/Penguji Seminar Hasil Strata Satu (S1) bagi mahasiswa Departemen Teknik Informatika Fakultas Teknik tersebut di bawah ini :

Nama / Stambuk	: Putri Ayu Nengsih	D121191070
Judul TA	: Perbandingan Algoritma C4.5 dan Random Forest Pada Pemetaan Kondisi Kerusakan Jalan Berbasis WebGIS (Studi Kasus : Luwu Raya)	

Dengan ini kami sampaikan Susunan Panitia Seminar Hasil Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :

Pembimbing I/ Ketua	: 1. Dr. Ir. Ingrid Nurtanio, M.T
Pembimbing II / Sekretaris	: 2. Dr. Ir. Zahir Zainuddin, M.Sc.
Anggota	: 3. Prof. Dr. Ir. Indrabayu, ST, MT., M.Bus.Sys
	: 4. Dr.Eng. Zulkifli Tahir, ST., M.Sc.

Untuk dapat diterbitkan surat penugasannya

Demikian penyampaian kami, atas perhatian dan kerjasamanya diucapkan terima kasih.

Gowa, 14 Mei 2024

Ketua Departemen Tek.Informatika,



Prof. Dr. Ir. Indrabayu, ST, MT., M.Bus.Sys., IPM, ASEAN.Eng
Nip.19750716 200212 1 004



san :
ip

Lampiran 11 Surat Penugasan



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK**

Poros Malino Km.6Bontomarannu(92172) Gowa, Sulawesi Selatan 92172, Sulawesi Selatan
Telp. (0411) 586015, 586262 Fax (0411) 586015
<http://eng.unhas.ac.id>, Email : teknik@unhas.ac.id

SURAT PENUGASAN
No. 10394/UN4.7.1/TD.06/2024

Dari : Dekan Fakultas Teknik Universitas Hasanuddin

Kepada : Mereka yang tercantum namanya dibawah ini

Isi : 1. Bahwa merujuk kepada Peraturan Rektor Universitas Hasanuddin **Nomor : 29/UN4.1/2023 tentang Penyelenggaraan Program Sarjana Universitas Hasanuddin**, dengan ini menugaskan Saudara sebagai PENGUJI/PANITIA SEMINAR HASIL Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :

Pembimbing I/ Ketua : 1. Dr. Ir. Ingrid Nurtanio, M.T
Pembimbing II / Sekretaris : 2. Dr. Ir. Zahir Zainuddin, M.Sc
Anggota : 3. Prof. Dr. Ir. Indrabayu, ST, MT., M.Bus.Sys.
4. Dr.Eng. Zulkifli Tahir, ST., M.Sc.

Untuk menguji bagi mahasiswa tersebut dibawah ini :

Nama/NIM : Putri Ayu Nengsih D121191070
Program Studi : Teknik Informatika
Judul thesis/Skripsi : Perbandingan Algoritma C4.5 dan Random Forest Pada Pemetaan Kondisi Kerusakan Jalan Berbasis WebGIS (Studi Kasus : Luwu Raya)

2. Waktu seminar ditetapkan oleh Panitia Seminar Hasil Program Strata Satu (S1)

3. Agar Surat Penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.

4. Surat penugasa ini berlaku sejak tanggal ditetapkan sampai dengan berakhirnya seminar tersebut dengan ketentuan bahwa segala sesuatunya akan ditinjau dan diperbaiki sebagaimana mestinya apabila dikemudia hari terdapat kekeliruan dalam keputusan ini.

Ditetapkan di Gowa
Pada tanggal 14 Mei 2024
a.n. Dekan,
Wakil Dekan Bidang Akademik dan Kemahasiswaan
Fakultas Teknik Unhas



Dr. Amil Ahmad Ilham, ST., M.IT
NIP. 197310101998021001

Tembusan :



teknik Unhas
men Teknik Informatika FT-UH
ng bersangkutan



• Dokumen ini telah ditandatangani secara elektronik menggunakan sertifikat elektronik yang diterbitkan BSrE
• UU ITE No 11 Tahun 2008 Pasal 5 Ayat 1
"Informasi Elektronik dan/atau Dokumen Elektronik dan/atau hasil cetaknya merupakan alat bukti hukum yang sah"

Lampiran 12 Daftar Hadir Ujian Skripsi



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN

FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://cg.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

DAFTAR HADIR UJIAN SKRIPSI MAHASISWA
FAKULTAS TEKNIK UNHAS

Nama/Stambuk : 1. Putri Ayu Nengsih D121191070

Judul Skripsi/T.A : "Perbandingan Algoritma C4.5 dan Random Forest pada Pemetaan Kondisi Kerusakan Jalan Berbasis WebGIS (Studi Kasus : Luwu Raya)"

Hari/Tanggal : Rabu, 12 Juni 2024

Jam : 14- 30 Wita – Selesai

Tempat : Ruang Lab. CBS Departemen Teknik Informatika Gowa

No.	Jabatan	Nama Dosen	Tanda Tangan
L.	Pembimbing I	1. Dr. Ir. Ingrid Nurtanio, M.T	1.
	Pembimbing II	2. Dr. Ir. Zahir Zainuddin, M.Sc	2.
II.	Anggota Penguji	3. Prof. Dr. Ir. Indrabayu, ST., M.T.M.Bus.Sys.IPM.ASEAN.Eng 3	3.
		4. Muhammad Alief Fahdal Imran Oemar, ST., M.Sc	4.

PANITIA UJIAN

Ketua,

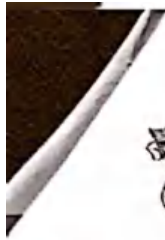
Dr. Ir. Ingrid Nurtanio, M.T

Sekretaris,

Dr. Ir. Zahir Zainuddin, M.Sc



Lampiran 13 Berita Acara Ujian Skripsi



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

BERITA ACARA UJIAN SKRIPSI

Pada hari ini Rabu, tanggal 12 Juni 2024 Pukul 14.30 WITA - Selesai bertempat di Lab. CBS Departemen Teknik Informatika Gowa, telah dilaksanakan Ujian Skripsi bagi Saudara :

Nama : Putri Ayu Nengsih
No. Stambuk : D121191070
Fakultas/Departemen : Teknik /Teknik Informatika
Judul Skripsi : "Perbandingan Algoritma C4.5 dan Random Forest pada Pemetaan Kondisi Kerusakan Jalan Berbasis WebGIS (Studi Kasus : Luwu Raya)"

Yang dihadiri oleh Tim Penguji Ujian Skripsi sebagai berikut :

No.	N a m a	Jabatan	Tanda tangan
1.	Dr. Ir. Ingrid Nurtanio, M.T	Pemb I/Ketua	1.
2.	Dr .Ir. Zahir Zainuddin, M.Sc	Pemb II/Sekretaris	2.
3.	Prof.Dr. Ir.Indrabayu, ST., M.T.M.Bus.Sys.IPM.ASEAN.Eng	Anggota	3.
4.	Muhammad Alief Fahdal Imran Omar, ST., M.Sc	Anggota	4.

Hasil keputusan Tim Penguji Ujian Skripsi/Tugas Akhir : **Lulus** / ~~Tidak lulus~~ dengan nilai angka dan huruf

Gowa, 12 Juni 2024

Ketua/Sekretaris Panitia Ujian,

Dr. Ir. Ingrid Nurtanio, M.T



Lampiran 14 Usulan Susunan Panitia/Penguji Ujian Sarjana



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

Jalan Poros Malino Km. 6 Bontomarannu, Gowa, 92171, Sulawesi Selatan
☎ +62811 4420 909, E-mail: teknik@unhas.ac.id , <https://eng.unhas.ac.id>

Gowa, 10 Juni 2024

Nomor : 753/UN4.7.7.1/TD.06/2024
Lamp : -
Hal : Usulan Susunan Panitia/Penguji Ujian Sarjana

Yth. : Bapak Wakil Dekan Bidang Akademik dan Kemahasiswaan
Fakultas Teknik Unhas
Di
Gowa

Dalam rangka penyelesaian studi pada Departemen Teknik Informatika Fakultas Teknik Unhas, bersama ini kami usulkan susunan Panitia/Penguji Ujian Sarjana Program Strata Satu (S1) bagi mahasiswa Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas nama :

Pembimbing I / Ketua : 1. Dr. Ir. Ingrid Nurtanio, M.T.
Pembimbing II / Sekretaris : 2. Dr. Ir. Zahir Zainuddin, M.Sc.
Anggota : 3. Prof. Dr. Ir. Indrabayu, ST, MT, M.Bus.Sys.
4. Dr.Eng. Ir. Zulkifli Tahir, ST., M.Sc

Untuk Bertugas sebagai Penguji/ Penanggap Ujian Sarjana bagi Mahasiswa :

Nama : Putri Ayu Nengsih
Stambuk : D121 19 1070

Dengan Judul Skripsi

“ Perbandingan Algoritma C4.5 dan Random Forest Pada Pemetaan Kondisi
Kerusakan Jalan Berbasis WebGIS (Studi Kasus : Luwu Raya) “

Pada :
Hari/Tanggal : Rabu, 12 Juni 2024
Jam : 14.30 Wita - Selesai
Tempat : Ruang Sidang Lab. CBS

Demikian penyampaian kami, atas perhatiannya diucapkan terimah kasih.

Ketua Departemen Tek.Informatika,



Prof. Dr. Ir. Indrabayu, ST, MT, M.Bus.Sys., IPM, ASEAN.Eng
Nip.197507016 200212 1 004

Tembusan :
1. Arsip



Optimized using
trial version
www.balesio.com



Lampiran 15 Surat Penugasan Panitia/Penguji Ujian Sarjana



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

Jalan Poros Malino Km. 6 Bontomarannu, Gowa, 92171, Sulawesi Selatan
☎ +62811 4420 909, E-mail: teknik@unhas.ac.id , <https://eng.unhas.ac.id>

SURAT PENUGASAN

No. 13231/UN4.7.1/TD.06/2024

Dari : Dekan Fakultas Teknik Universitas Hasanuddin.

Kepada : Mereka yang tercantum namanya di bawah ini.

Isi : 1. Bahwa merujuk kepada Peraturan Rektor Universitas Hasanuddin **Nomor : 29/UN4.1/2023 tentang Penyelenggaraan Program Sarjana Universitas Hasanuddin**, dengan ini menugaskan Saudara sebagai **PENGUJI/PANITIA UJIAN SARJANA** Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :

Pembimbing I / Ketua : 1. Dr. Ir. Ingrid Nurtanio, M.T.
Pembimbing II / Sekretaris : 2. Dr. Ir. Zahir Zainuddin, M.Sc.
Anggota : 3. Prof. Dr. Ir. Indrabayu, ST, MT, M.Bus.Sys.
4. Dr.Eng. Ir. Zulkifli Tahir, ST., M.Sc

untuk menguji bagi mahasiswa tersebut di bawah ini :

Nama/NIM : Putri Ayu Nengsih D121 19 1070

Program Studi : Teknik Informatika

Judul Thesis/Skripsi : "Perbandingan Algoritma C4.5 dan Random Forest Pada Pemetaan Kondisi Kerusakan Jalan Berbasis WebGIS (Studi Kasus : Luwu Raya) "

2. Waktu Ujian ditetapkan oleh Panitia Ujian Sarjana Program Strata Satu (S1).

3. Agar Surat penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.

4. Surat penugasan ini berlaku sejak tanggal ditetapkan sampai dengan berakhirnya Ujian Sarjana tersebut, dengan ketentuan bahwa segala sesuatunya akan ditinjau dan diperbaiki sebagaimana mestinya apabila dikemudian hari ternyata terdapat kekeliruan dalam keputusan ini.

Ditetapkan di Gowa,
Pada tanggal 10 Juni 2024

a.n. Dekan

Wakil Dekan Bidang Akademik dan Kemahasiswaan
Fakultas Teknik Unhas



Dr. Amil Ahmad Ilham, ST., M.IT
NIP.197310101998021001

Tembusan :

1. Dekan Fak. Teknik Unhas
2. Ketua Departemen Teknik Informatika FT-UH
3. Kasubag. Umum dan Perlengkapan FT-UH



- Dokumen ini telah ditandatangani secara elektronik menggunakan sertifikat elektronik yang diterbitkan BSrE
- UU ITE No 11 Tahun 2008 Pasal 5 Ayat 1
- "Informasi Elektronik dan/atau Dokumen Elektronik dan/atau hasil cetaknya merupakan alat bukti hukum yang sah"



Lampiran 16 Lembar Perbaikan Skripsi

LEMBAR PERBAIKAN SKRIPSI

**“PERBANDINGAN ALGORITMA C4.5 DAN RANDOM
FOREST PADA PEMETAAN KONDISI KERUSAKAN JALAN
BERBASIS WEBGIS
(STUDI KASUS : LUWU RAYA)”**

OLEH:

**PUTRI AYU NENGSIH
D121191070**

Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 12 Juni 2024.

Telah dilakukan perbaikan penulisan dan isi skripsi berdasarkan usulan dari penguji dan pembimbing skripsi.

Persetujuan perbaikan oleh tim penguji:

	Nama	Tanda Tangan
Ketua	Dr. Ir. Ingrid Nurtanio, M.T.	
Sekretaris	Dr. Ir. Zahir Zainuddin, M.Sc.	
Anggota	Prof. Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys., IPM. ASEAN. Eng.	
	Muhammad Alief Fahdal Imran Oemar, S.T., M.Sc.	

Persetujuan Perbaikan oleh pembimbing:

Pembimbing	Nama	Tanda Tangan
I	Dr. Ir. Ingrid Nurtanio, M.T.	
II	Dr. Ir. Zahir Zainuddin, M.Sc.	

