# IMPLEMENTASI TEKNIK REGULARISASI MENGGUNAKAN PRUNING PADA ALGORITMA RANDOM FOREST DALAM MEMPREDIKSI PENYAKIT GINJAL KRONIS

**Yuna Adi Purnama; Devi Afriyantari Puspa Putri**
**Teknik Informatika, Fakultas Komunikasi dan Informatika,**
**Universitas Muhammadiyah Surakarta**

**Abstrak**

Penyakit ginjal kronis (PGK) adalah masalah kesehatan masyarakat utama diseluruh dunia, terutama di negara berpenghasilan rendah dan menengah. Penyakit ginjal kronis (PGK) adalah gangguan di mana ginjal berhenti bekerja dengan benar dan tidak dapat menyaring darah. Tujuan dari penelitian ini yaitu mendapatkan hasil terbaik prediksi untuk Penyakit Ginjal Kronis menggunakan Algoritma *Random Forest* dengan Metode *Pruning*. Perbandingan *Naïve Bayes Classifier* dengan Algoritma *Random Forest* dengan Metode *Pruning* untuk mendapatkan regularisasi yang terbaik. *Pruning* mengacu pada inovasi ini yang berfokus pada identifikasi subset *Random Forest* terbaik. Peneliti membandingkan Algoritma penelitian terdahulu yaitu menggunakan *Naïve Bayes Classifier* dengan Algoritma yang akan diteliti yaitu Algoritma *Random Forest* dengan Metode *Pruning* dalam prediksi untuk Penyakit Ginjal Kronis. Hasil penelitian ini menunjukan bahwa penggunaan *pruning* pada algoritma *Random Forest* menunjukkan performa yang lebih baik dengan accuracy 99.58% dari penelitian terdahulu menggunakan algoritma *Naïve Bayes Classifier* dengan accuracy 96.43%.

**Kata Kunci:** Algoritma Random Forest, Machine Learning, Penyakit Ginjal Kronis, Prediksi, Teknik Pruning.

***Abstract***

*Chronic kidney disease (CKD) is a major public health problem worldwide, especially in low- and middle-income countries. Chronic kidney disease (CKD) is a disorder in which the kidneys stop working properly and cannot filter the blood. The aim of this research is to get the best prediction results for Chronic Kidney Disease using the Random Forest Algorithm with the Pruning Method. Comparison of the Naïve Bayes Classifier with the Random Forest Algorithm with the Pruning Method to get the best regularization. Pruning refers to this innovation that focuses on identifying the best Random Forest subset. Researchers compared previous research algorithms using the Naïve Bayes Classifier with the algorithm to be studied, namely the Random Forest Algorithm with the Pruning Method in predictions for Chronic Kidney Disease. The results of this study indicate that the use of pruning in the Random Forest algorithm yields better performance with an accuracy of 99.58% compared to the previous study that utilized the Naïve Bayes Classifier algorithm, which achieved an accuracy of 96.43%.*

***Keywords****: Chronic Kidney Disease, Machine Learning, Prediction, Pruning Technique, Random Forest Algorithm.*

## 1. PENDAHULUAN

Penyakit ginjal kronis (PGK) merupakan masalah kesehatan masyarakat global terutama di negara berpenghasilan rendah dan menengah (Gliselda, 2021). PGK merupakan penyebab kematian

peringkat ke-27 di dunia tahun 1990, meningkat menjadi urutan ke-18 pada tahun 2010, dan meningkat lagi menjadi urutan ke-12 pada tahun 2017 (Gliselda, 2021). Penyakit ginjal kronis (CKD) memerlukan skrining dini yang andal, namun pengumpulan data klinis seringkali agak tidak andal, yang dapat memengaruhi model pembelajaran mesin dan menurunkan kualitas deteksi (Imaduddin et al., 2026). Negara yang termasuk dalam negara berpenghasilan rendah, misalnya Nigeria, Afrika Tengah, Afrika Selatan, Afrika Barat, Madagaskar dan lain sebagainya (Ainun et al., 2017; Anggryni et al., 2021). Oleh karena itu perlu adanya metode untuk mengatasi permasalahan PGK di dunia ini. Salah satu pendekatan adalah *Machine Learning*.

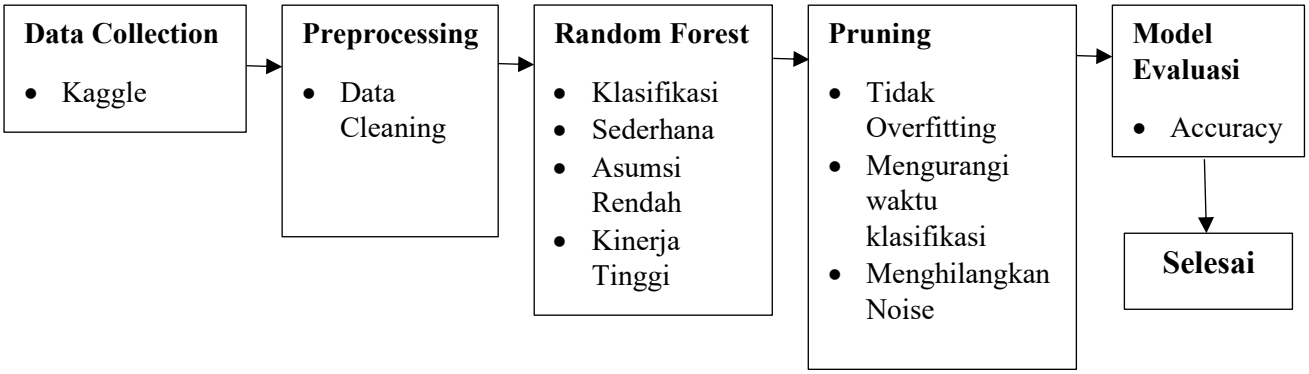
*Machine learning* digunakan untuk mendeteksi penyakit. *Dataset* yang mendukung penelitian ini adalah *dataset* yang memprediksi apakah seseorang memiliki penyakit ginjal kronis berdasarkan ciri-cirinya menggunakan *Machine Learning* menggunakan Algoritma *Random Forest* dengan Metode *Pruning*. *Dataset* tersebut bernama *Pre-processed Chronic Kidney Disease Dataset* oleh Mahmoud Limam (*Pre-Processed Chronic Kidney Disease Dataset. | Kaggle*, n.d.).

Pendeteksian penyakit ginjal kronis harus lebih ditingkatkan dari penelitian terdahulu. Penelitian yang dilakukan oleh Qurotul A'yuniyah, Ena Tasia, Nanda Nazira, dkk. Penelitian terdahulu menggunakan *Naïve Bayes Classifier*. Hasil klasifikasi dengan tools RapidMiner yang dilakukan dengan penerapan algoritma NBC maka diperoleh nilai akurasi 96.43%, dan hasil rata-rata *recall* 93.18%, rata-rata *precision* 93.02% dan AUC 93.2% (A'yuniyah et al., 2022). Penelitian kali ini akan menggunakan *Pruning* pada algoritma *Random Forest* sebagai salah satu metode regularisasi dalam memprediksi untuk Penyakit Ginjal Kronis. Algoritma *Random Forest* mempunyai kelebihan termasuk kemampuan untuk menangani noise dan nilai yang hilang serta mengatasi data dalam jumlah besar (Harahap et al., 2021). Kemudian, terdapat sebuah penelitian yang dilakukan oleh Dimas Ariyoga bahwa teknik regularisasi dapat meningkatkan performa dari algoritma Random Forest (Ariyoga, 2022). Selain itu, penggunaan pruning juga dapat meningkatkan performa dari algoritma Random Forest. Kelebihan pruning dapat mengurangi kompleksitas dan meningkatkan akurasi prediktif dengan pengurangan *overfitting* (Wibowo et al., 2020).

Tujuan dari penelitian ini yaitu mendapatkan performa terbaik prediksi untuk Penyakit Ginjal Kronis menggunakan Algoritma *Random Forest* dengan Metode *Pruning* yang terbaik. Luaran penelitian ini berupa kode program dengan algoritma *random forest* dan *pruning* untuk memprediksi PGK. Manfaat pencegah seseorang terkena PGK yaitu memperlambat laju penurunan fungsi ginjal dan mencegah kondisi penyakit ginjal tahap akhir yang memerlukan dialisis, mengingat tingginya pembiayaan yang dibutuhkan untuk terapi pengganti ginjal (Wayan & Wardani, 2022).

## 2. METODE

Alur metode penelitian dapat dilihat pada Gambar 1.




Gambar 1. Alur Metode Penelitian

### 2.1 Data Collection

*Dataset* yang digunakan dari Mahmoud Limam pada Kaggle (*Pre-Processed Chronic Kidney Disease Dataset. | Kaggle*, n.d.). Atribut dataset disajikan pada Tabel 1.

Tabel 1. Atribut Dataset

| Atribut Dataset | | |
|---|---|---|
| No | Atribut | Keterangan |
| 1 | Age (yrs) | Umur seseorang |
| 2 | Blood Pressure (mm/Hg) | Tekanan darah seseorang |
| 3 | Specific Gravity | masa jenis urine yang dalam kadar normal bernilai lebih kurang 1,003-1,030 |
| 4 | Albumin | kadar albumin dalam darah seseorang |
| 5 | Sugar | Kadar gula darah seseorang |
| 6 | Blood Glucose Random (mgs/dL) | Kadar glukosa atau gula yang beredar dalam darah seseorang |
| 7 | Blood Urea (mgs/dL) | Kadar urea dalam darah seseorang |
| 8 | Serum Creatinine (mgs/dL) | Kadar sampah hasil metabolisme otot yang mengalir pada sirkulasi darah |
| 9 | Sodium (mEq/L) | Kadar Sodium dalam darah seseorang |
| 10 | Potassium (mEq/L) | Kadar Potassium dalam darah seseorang |
| 11 | Hemoglobin (gms) | Kadar protein yang berfungsi membawa oksigen menuju ke semua jaringan tubuh seseorang |
| 12 | Packed Cell Volume | Volume yang ditempati oleh sel darah merah ketika sampel darah antikoagulan disentrifugas |

Tabel 1. Atribut Dataset

| Atribut Dataset | | |
|---|---|---|
| No | Atribut | Keterangan |
| 13 | White Blood Cells (cells/cmm) | Jumlah sel darah putih seseorang |
| 14 | Red Blood Cells (millions/cmm) | Jumlah Sel darah merah seseorang |
| 15 | Red Blood Cells (normal) | Normal tidaknya jumlah sel darah merah seseorang |
| 16 | Pus Cells (normal) | Kumpulan cairan, sel darah putih, mikroorganisme, dan bahan selular yang menunjukkan adanya luka yang terinfeksi atau abses |
| 17 | Pus Cell Clumps (present) | Sel darah putih yang bergerombol dalam pemeriksaan urin, biasanya salah satu pertanda ada infeksi |
| 18 | Bacteria (present) | Adanya bakteri atau tidak dalam tubuh seseorang |
| 19 | Hypertension (yes) | Memiliki tidaknya tekanan darah tinggi |
| 20 | Diabetes Mellitus (yes) | Penyakit akibat terlalu banyak kadar gula dalam darah |
| 21 | Coronary Artery Disease (yes) | suatu kondisi di mana suplai darah ke otot jantung tersumbat sebagian atau seluruhnya. |
| 22 | Appetite (poor) | Memiliki nafsu makan yang baik atau tidak |
| 23 | Pedal Edema (yes) | Memiliki atau tidak penumpukan cairan pada bagian betis serta kaki bagian bawah |
| 24 | Anemia (yes) | Memiliki anemia atau tidak |
| 25 | Chronic Kidney Disease (yes) | Penyakit ginjal kronis positif atau tidak |

*Chronic Kidney Disease* dan mempunyai jumlah data sebanyak 400 data. *Dataset* ini terbagi dalam 2 kelas/label (*Pre-Processed Chronic Kidney Disease Dataset. | Kaggle*, n.d.).

### 2.2 Data Preprocessing

*Preprocessing* data adalah proses mengubah data berkualitas rendah menjadi data berkualitas tinggi. Proses ini mempersiapkan data untuk digunakan dalam analisis atau model pembelajaran *machine learning*. Pemrosesan data dengan baik, model dapat belajar secara lebih efektif, menghindari bias, dan menghasilkan prediksi yang lebih akurat.

#### 2.2.1 Data Cleaning

Pembersihan data paling sering digunakan untuk mengidentifikasi nilai yang hilang, mengidentifikasi *outlier*, mengurangi *noise* data, meningkatkan konsistensi data, dan mengurangi efek integrasi data yang mungkin menyebabkan redudansi (Prasojo & Haryatmi, 2021).

## 2.3 Data Processing

Pemrosesan data adalah proses penyiapan data sebelum digunakan dalam pembelajaran mesin *machine learning*. Proses ini mencakup berbagai tahapan seperti pengumpulan, modifikasi, dan pengurangan dimensi data guna memastikan kualitas serta representasi data yang sesuai dengan karakteristik model pembelajaran mesin. Pemrosesan data sangatlah penting karena hal ini memengaruhi kinerja model pembelajaran mesin tersebut.

### 2.3.1 Decision Tree

Decision tree adalah alat yang sangat berguna untuk pemodelan prediktif dalam pembelajaran mesin dan pembelajaran mesin. Algoritma Decision tree digunakan untuk memprediksi nilai suatu variabel berdasarkan range data variabel yang disediakan. Kemudahan penggunaan Decision tree dalam menginterpretasikan hasil, sering digunakan untuk klasifikasi dan prediksi. Pengembangan Decision tree dipandu oleh penyertaan satu set fitur dalam hierarki, serta ekspresi dari banyak fitur dan variabel (Kurniawan et al., 2023). Penggunaan rumus entropy memungkinkan kita untuk menghitung nilai *entropy* dan *information gain*, yang pada gilirannya dapat membantu menentukan seluruh struktur dari pohon Keputusan. Persamaan 1 merupakan rumus untuk menghasilkan *entropy* dan persamaan 2 merupakan rumus untuk menghasilkan *gain*.

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i \quad (1)$$

Keterangan Rumus:

S = Himpunan kasus

A = Atribut

N = Jumlah dari partisi S

Pi = Proporsi dari Si terhadap S

Rumus information gain:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (2)$$

Keterangan Rumus:

S = himpunan permasalahan

A = Atribut

|Sv| = jumlah masalah pada partisi ke-v

|S| = jumlah masalah dalam S (Firmansyach et al., 2023).

### 2.3.2 Random Forest

*Random Forest* pertama kali ditunjuk oleh Breimen et al., adalah pendekatan ansambel untuk membangun model prediktif untuk tugas klasifikasi dan regresi. Random Forest adalah cara menggabungkan model dasar yang kurang prediktif dengan model prediktif yang lebih baik. Karena sifatnya yang sederhana, asumsi rendah, dan kinerja tinggi, model *Random Forest* telah banyak digunakan dalam pembelajaran *Machine Learning*. Model *Random Forest* mengklasifikasikan variabel berdasarkan kepentingannya untuk mencapai model *Random Forest* terbaik (Nhu et al., 2020). *Random Forest* banyak digunakan karena kelebihan nya pada akurasi yang lebih baik, dan ketahanan dari berbagai gangguan, kecepatan dan kemudahan dalam implementasi model (Afif et al., 2024). *Random Forest* merupakan algoritma berbasis esemble yang dikembangkan berdasarkan algoritma *Decision Tree* dan dikenal memiliki kinerja yang baik. Algoritma esemble adalah kombinasi dari beberapa teknik pembelajaran mesin yang menghasilkan model prediksi tunggal. Algoritma dibuat untuk mengurangi bias dan meningkatkan akurasi prediksi (Sari et al., 2023). Berikut merupakan rumus dari *Random Forest*:

Menghitung nilai *entropy* berdasarkan persamaan 1

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i \quad (1)$$

Keterangan Rumus:

S = Himpunan kasus

A = Atribut

N = Jumlah dari patisi S

Pi = Proporsi dari Si terhadap S

Menghitung nilai *Information Gain* berdasarkan persamaan 2

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (2)$$

Keterangan Rumus:

S = himpunan permasalahan

A = Atribut

|Sv| = jumlah masalah pada partisi ke-v

|S| = jumlah masalah dalam S (G. Sandag, 2020)

## 2.4 Regularisasi

Teknik Regularisasi adalah teknik yang digunakan untuk mengurangi kemungkinan model *overfitting.* Regularisasi membantu pengembangan model yang melakukan generalisasi lebih baik terhadap data baru dengan meningkatkan penalti pada *loss function*, menghasilkan parameter model yang lebih kecil dan lebih kompleks (Anindyo et al., 2023). Regularisasi digunakan untuk mengurangi *overfitting*, dan biasanya menghasilkan kinerja prediksi yang tinggi dengan analisis yang efisien (Imaduddin et al., 2026).

### 2.4.1 Pruning

*Pre-pruning* dan *Post-pruning* adalah dua teknik standar untuk menangani noise dalam pembelajaran pohon keputusan. *Pre-pruning* menangani noise selama pembelajaran, sedangkan *post-pruning* mengatasi masalah ini setelah teori *overfitting* telah dipelajari (Urnkranz, 1997). *Pruning* adalah proses memotong atau membuang dahan (cabang) yang tidak perlu. *Pruning* digunakan untuk meningkatkan akurasi generalisasi dan prediksi dengan menghapus *node* dari *Decision Tree* yang tidak lagi diperlukan. Konsep utamanya adalah untuk memperkenalkan bias untuk teori yang lebih sederhana untuk menghilangkan *noise* (kesalahan hasil prediksi) (Wardhani et al., 2022). Pruning adalah cara untuk mengurangi ukurannya guna mengurangi waktu klasifikasi dan meningkatkan (atau setidaknya mempertahankan) akurasi klasifikasi (Gadomer & Sosnowski, 2020). Masalah overfitting dapat diselesaikan dengan menggunakan teknik pruning. Regularisasi pruning dengan smoothing menggunakan Laplace correction method dapat mengatasi masalah overfitting (Rahayu et al., 2015). Metode pemangkasan ini secara intuitif menarik karena memperhitungkan baik kesalahan klasifikasi maupun kompleksitas pohon dengan ukuran pengurangan kesalahan per daun. Metode ini juga menghasilkan pilihan pohon untuk dipelajari oleh ahli. Namun, metode ini membutuhkan set data uji terpisah untuk tujuan seleksi (Bsrcd, 1989).

## 2.5 Evaluation

Setelah pembuatan model selanjutnya melakukan *evaluation*. *Evaluation* berguna untuk meningkatkan kinerja pengklasifikasi (Sandag et al., 2018). Setelah model akhir dibuat, evaluasi terhadap model tersebut akan dilakukan dengan membandingkan kedua jenis algoritma untuk menentukan algoritma mana yang akan digunakan. Setelah model yang akan digunakan ditentukan, langkah selanjutnya adalah membuat model yang dapat digunakan oleh pengguna.(Afif et al., 2024)

### 2.5.1 Accuracy

Akurasi adalah metrik yang paling maju dan banyak digunakan untuk mengurangi kinerja *classifier* dan ketepatan klasifikasi data (Juba & Le, 2019). Berikut adalah rumus *Accuracy*:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

Keterangan:

| | | | |
|---|---|---|---|
| P | : Jumlah data *positive* | TP | : Nilai *true positive* |
| N | : Jumlah data *negative* | TN | : Nilai *true negative* |
| FN | : Nilai *false negative* | FP | : Nilai *false positive* (G. A. Sandag et al., 2018) |

## 3. HASIL DAN PEMBAHASAN

Pada tahap ini, penulis akan membahas tentang hasil proses Regularisasi menggunakan *Pruning* pada Algoritma *Random Forest.* Tahap ini meliputi data *Preprocessing*, data *processing*, dan evaluasi model. Hasilnya dicantumkan pada evaluasi model berupa *accuration.*

### 3.1 Data *Preprocessing*

Langkah pertama yang dilakukan adalah menginstall library yang dibutuhkan seperti sklearn, *import* numpy, matplotlib, seaborn, dan pandas. Kemudian membuka google collab untuk memasukkan *dataset* agar bisa terproses didalamnya. *Dataset* dibersihkan menggunakan data cleaning sebelum melakukan proses *training*. Proses data *cleaning* meliputi mengisi data yang hilang atau kosong dengan nilai *mean*, kemudian menghapus atau menduplikat bila diperlukan, Kemudian mengubah fitur kategorikal menjadi *numerical* agar bisa terbaca oleh model, memisahkan fitur (X) dan target (Y), dan membagi data *training* dan *testing* yang disajikan pada Gambar 2, 3, 4, 5, 6, dan 7.

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
```

Gambar 2. Mengimport *Library*

```
from sklearn.datasets import make_blobs, make_classification
from sklearn.ensemble import RandomForestClassifier, ExtraTreesClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.ensemble._forest import _generate_unsampled_indices as get_unsampled_indices
from sklearn.ensemble._forest import _generate_sample_indices as get_insample_indicess
```

Gambar 3. Mengimport fitur

```
import pandas as pd

# Read the CSV file
data = pd.read_csv('CKD_Preprocessed.csv')
```

Gambar 4. Mengimport *Dataset*

```
print(data)
```

```
396                      1.2          141.0              3.5  ...
397                      0.6          137.0              4.4  ...
398                      1.0          135.0              4.9  ...
399                      1.1          141.0              3.5  ...

     Pus Cells: normal  Pus Cell Clumps: present  Bacteria: present  \
0                  1.0                       0.0                0.0
1                  1.0                       0.0                0.0
2                  1.0                       0.0                0.0
3                  0.0                       1.0                0.0
4                  1.0                       0.0                0.0
..                 ...                       ...                ...
395                1.0                       0.0                0.0
396                1.0                       0.0                0.0
397                1.0                       0.0                0.0
398                1.0                       0.0                0.0
399                1.0                       0.0                0.0

     Hypertension: yes  Diabetes Mellitus: yes  Coronary Artery Disease: yes  \
0                  1.0                     1.0                           0.0
1                  0.0                     0.0                           0.0
2                  0.0                     1.0                           0.0
3                  1.0                     0.0                           0.0
4                  0.0                     0.0                           0.0
..                 ...                     ...                           ...
395                0.0                     0.0                           0.0
396                0.0                     0.0                           0.0
397                0.0                     0.0                           0.0
398                0.0                     0.0                           0.0
399                0.0                     0.0                           0.0
```

Gambar 5. Menampilkan *dataset*

```
     Appetite: poor  Pedal Edema: yes  Anemia: yes  \
0               0.0               0.0          0.0
1               0.0               0.0          0.0
2               1.0               0.0          1.0
3               1.0               1.0          1.0
4               0.0               0.0          0.0
..              ...               ...          ...
395             0.0               0.0          0.0
396             0.0               0.0          0.0
397             0.0               0.0          0.0
398             0.0               0.0          0.0
399             0.0               0.0          0.0

     Chronic Kidney Disease: yes
0                            1.0
1                            1.0
2                            1.0
3                            1.0
4                            1.0
..                           ...
395                          0.0
396                          0.0
397                          0.0
398                          0.0
399                          0.0

[400 rows x 25 columns]
```

Gambar 6. Menampilkan *dataset*

```
Y = data['Chronic Kidney Disease: yes']
X = data.drop('Chronic Kidney Disease: yes', axis =1)

print(X.shape)

(400, 24)

print(Y)
print(Y.shape)

0      1.0
1      1.0
2      1.0
3      1.0
4      1.0
      ...
395    0.0
396    0.0
397    0.0
398    0.0
399    0.0
Name: Chronic Kidney Disease: yes, Length: 400, dtype: float64
(400,)
```

Gambar 7. Menginput data pada fitur X dan Y

### 3.2 Data *Processing*

Alur pada *Processing* disajikan dalam bentuk kode pada Gambar 7.

```
from sklearn.metrics import accuracy_score

# Split dataset into training and testing sets
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=43)

# Create a Random Forest classifier with pruning
rf_classifier = RandomForestClassifier(n_estimators=200, max_depth=2, random_state=7)

# Train the classifier
rf_classifier.fit(X_train, Y_train)

# Make predictions on the test set
Y_pred = rf_classifier.predict(X_test)
```

Gambar 8. *Training* dan *Testing* Data

Membuat model *Random Forest* kemudian dilatih menggunakan data testing untuk dijadikan model. *Random Forest* menggunakan cara menggabungkan model dasar yang kurang prediktif dengan model prediktif yang lebih baik. Karena sifatnya yang sederhana, asumsi rendah, dan kinerja tinggi, model Random Forest telah banyak digunakan dalam pembelajaran *Machine Learning*. Model *Random Forest* mengklasifikasikan variabel berdasarkan kepentingannya untuk mencapai model *Random Forest* terbaik (Nhu et al., 2020). *Random Forest* merupakan algoritma berbasis esemble yang dikembangkan berdasarkan algoritma *Decision Tree* dan dikenal memiliki kinerja yang baik. Algoritma esemble kombinasi dari beberapa pembelajaran mesin yang menghasilkan model prediksi. Algoritma dibuat untuk mengurangi bias dan meningkatkan akurasi prediksi (Sari et al.,

2023). Langkah Pertama melakukan split dataset dengan data training dan data testing. Langkah kedua membuat *random forest classifier* dengan menggunakan pruning. Langkah ketiga *training* data *classifier* yang sudah dibuat. Langkah keempat membuat prediksi pada *random forest classifier*.

### 3.3 Evaluasi Model

Evaluasi model yang dilakukan menggunakan *accuracy* pada proses *pruning* pada *node* tertentu untuk mengetahui hasilnya apakah sudah bagus, kurang bagus, ataupun *overfitting*. Berikut hasil evaluasi model yang disajikan pada Gambar 8, 9, 10, 11, dan 12.

```
# Calculate accuracy
accuracy = accuracy_score(Y_test, Y_pred)
print(list(Y_test))
print(Y_pred)
print("Accuracy:", accuracy)
```

```
[1.0, 0.0, 0.0, 1.0, 1.0, 0.0, 1.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 1
[1. 0. 0. 0. 1. 0. 1. 0. 0. 1. 1. 1. 1. 1. 0. 0. 1. 0. 0. 0. 0. 1. 1. 1.
 0. 1. 1. 1. 1. 1. 0. 0. 1. 1. 0. 1. 1. 1. 1. 1. 0. 0. 1. 1. 1. 0. 0. 0.
 1. 0. 1. 1. 1. 0. 1. 1. 1. 1. 0. 0. 0. 1. 0. 1. 0. 0. 1. 0. 0. 1. 0. 1.
 0. 1. 0. 1. 1. 0. 0. 1.]
Accuracy: 0.975
```

Gambar 9. Evaluasi model pada data *training* 80% dan *pruning* pada node ke 2

```
# Calculate accuracy
accuracy = accuracy_score(Y_test, Y_pred)
print(list(Y_test))
print(Y_pred)
print("Accuracy:", accuracy)
```

```
[1.0, 0.0, 0.0, 1.0, 1.0, 0.0, 1.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 1.0,
[1. 0. 0. 1. 1. 0. 1. 0. 0. 1. 1. 1. 1. 1. 0. 0. 1. 0. 0. 0. 0. 1. 1. 1.
 0. 1. 1. 1. 1. 1. 0. 0. 1. 1. 0. 1. 1. 1. 1. 1. 0. 0. 1. 1. 1. 0. 0. 0.
 1. 0. 1. 1. 1. 0. 1. 1. 1. 1. 0. 0. 0. 1. 0. 1. 0. 0. 1. 0. 0. 1. 0. 1.
 0. 1. 0. 1. 1. 0. 0. 1.]
Accuracy: 0.9875
```

Gambar 10. Evaluasi model pada data *training* 80% dan *pruning* pada node ke 3

```
# Calculate accuracy
accuracy = accuracy_score(Y_test, Y_pred)
print(list(Y_test))
print(Y_pred)
print("Accuracy:", accuracy)
```

```
[1.0, 0.0, 0.0, 1.0, 1.0, 0.0, 1.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 1.0, 0
[1. 0. 0. 0. 1. 0. 1. 0. 0. 1. 1. 1. 1. 1. 0. 0. 1. 0. 0. 0. 0. 1. 1. 1.
 0. 1. 1. 1. 1. 1. 0. 0. 1. 1. 0. 1. 1. 1. 1. 1. 0. 0. 1. 1. 1. 0. 0. 0.
 1. 0. 1. 1. 1. 0. 1. 1. 1. 1. 0. 0. 0. 1. 0. 1. 0. 0. 1. 0. 0. 1. 0. 0.
 0. 1. 0. 1. 1. 0. 0. 1. 0. 1. 1. 1. 1. 1. 0. 1. 1. 0. 1. 1. 0. 0. 0. 1.
 0. 1. 1. 1. 1. 1. 1. 0. 0. 1. 0. 1. 0. 0. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
Accuracy: 0.9666666666666667
```

Gambar 11. Evaluasi model pada data *training* 70% dan *pruning* pada node ke 2

```
# Calculate accuracy
accuracy = accuracy_score(Y_test, Y_pred)
print(list(Y_test))
print(Y_pred)
print("Accuracy:", accuracy)
```

```
[1.0, 0.0, 0.0, 1.0, 1.0, 0.0, 1.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 1.0
[1. 0. 0. 0. 1. 0. 1. 0. 0. 1. 1. 1. 1. 1. 0. 0. 1. 0. 0. 0. 0. 1. 1. 1.
 0. 1. 1. 1. 1. 1. 0. 0. 1. 1. 0. 1. 1. 1. 1. 1. 0. 0. 1. 1. 1. 0. 0. 0.
 1. 0. 1. 1. 1. 0. 1. 1. 1. 1. 0. 0. 0. 1. 0. 1. 0. 0. 1. 0. 0. 1. 0. 0.
 0. 1. 0. 1. 1. 0. 0. 1. 0. 1. 1. 1. 1. 1. 0. 1. 1. 0. 1. 1. 0. 0. 0. 1.
 0. 1. 1. 1. 1. 1. 1. 0. 0. 1. 0. 1. 0. 0. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 0. 1. 1. 0. 1. 0. 0. 1. 0. 0. 1. 0. 0. 1. 1. 1. 1. 1. 1. 1. 0. 0. 0. 1.
 1. 1. 1. 1. 0. 0. 1. 1. 0. 0. 1. 1. 1. 1. 1. 0.]
Accuracy: 0.9625
```

Gambar 12. Evaluasi model pada data *training* 60% dan *pruning* pada node ke 2

```
# Calculate accuracy
accuracy = accuracy_score(Y_test, Y_pred)
print(list(Y_test))
print(Y_pred)
print("Accuracy:", accuracy)
```

```
[1.0, 0.0, 0.0, 1.0, 1.0, 0.0, 1.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 1.0
[1. 0. 0. 1. 1. 0. 1. 0. 0. 1. 1. 1. 1. 1. 0. 0. 1. 0. 0. 0. 0. 1. 1. 1.
 0. 1. 1. 1. 1. 1. 0. 0. 1. 1. 0. 1. 1. 1. 1. 1. 0. 0. 1. 1. 1. 0. 0. 0.
 1. 0. 1. 1. 1. 0. 1. 1. 1. 1. 0. 0. 0. 1. 0. 1. 0. 0. 1. 0. 0. 1. 0. 1.
 0. 1. 0. 1. 1. 0. 0. 1. 0. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 0. 0. 0. 1.
 0. 1. 1. 1. 1. 1. 1. 0. 0. 1. 0. 1. 0. 0. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 0. 1. 1. 0. 1. 0. 0. 1. 0. 0. 1. 0. 0. 1. 1. 1. 1. 1. 1. 1. 0. 1. 0. 1.
 1. 1. 1. 1. 0. 1. 1. 1. 0. 0. 1. 1. 1. 1. 1. 0. 0. 1. 1. 1. 1. 1. 1. 1.
 0. 1. 1. 1. 1. 0. 1. 0. 0. 0. 0. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 0. 0. 0. 0. 1. 0. 0. 0. 1. 1. 1. 0. 1. 0. 1. 1. 0. 1. 0. 1.
 0. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 1. 0. 0. 1. 1. 1. 0. 1. 1. 1. 1.]
Accuracy: 0.9958333333333333
```

Gambar 13. Evaluasi model pada data *training* 80% dan *pruning* pada node ke 5

## 4. PENUTUP

Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa penggunaan pruning pada algoritma Random Forest menunjukkan performa yang lebih baik dari penelitian terdahulu menggunakan algoritma *Naïve Bayes Classifier*. Hasil accuracy penelitian ini yaitu 99.58% lebih dengan proses pruning pada beberapa tahapan node dan perbandingan antara data training dan data testing. Pemilihan pruning pada node yang diinginkan dalam penelitian ini dengan mempertimbangkan hasil training yang terlalu bagus agar tidak *overfitting*.

## DAFTAR PUSTAKA


Afif, W., Rahman, N., Endang, ;, & Pamungkas, W. (2024). *Implementasi Machine Learning Untuk Memprediksi Tingkat Stres Berdasaerkan Gaya Hidup*. https://eprints.ums.ac.id/125907/

Ainun, N., Yunihastuti, E., & Roosheroe, A. G. (2017). HIV pada Geriatri HIV in Geriatrics. *Jurnal Penyakit Dalam Indonesia*, *3*(2), 106. https://doi.org/10.7454/jpdi.v3i2.17

Anggryni, M., Mardiah, W., Hermayanti, Y., Rakhmawati, W., Ramdhanie, G. G., & Mediani, H. S. (2021). Faktor Pemberian Nutrisi Masa Golden Age dengan Kejadian Stunting pada Balita di Negara Berkembang. *Jurnal Obsesi : Jurnal Pendidikan Anak Usia Dini*, *5*(2), 1764–1776. https://doi.org/10.31004/obsesi.v5i2.967

Anindyo, A., Isnanto, R. R., & Eridani, D. (2023). Pemilihan Model Terbaik Algoritma Convolutional Neural Network untuk Klasifikasi Jenis Bencana Alam. *Jurnal Teknik Komputer*, *2*(1), 58–67. https://doi.org/10.14710/jtk.v2i1.37656

Ariyoga, D. (2022). *Perbandingan Metode Seleksi Fitur Filter, Wrapper, Dan Embedded Pada Klasifikasi Data Nirs Mangga Menggunakan Random Forest Dan Support Vector Machine (Svm)*. https://dspace.uii.ac.id/handle/123456789/38955

A'yuniyah, Q., Tasia, E., Nazira, N., Pratama, P. F., Anugrah, M. R., Adhiva, J., & Mustakim, M. (2022). Implementasi Algoritma Naïve Bayes Classifier (NBC) untuk Klasifikasi Penyakit Ginjal Kronik. *Jurnal Sistem Komputer Dan Informatika (JSON)*, *4*(1), 72. https://doi.org/10.30865/json.v4i1.4781

Bsrcd, J. M. (1989). *An Empirical Comparison of Pruning Methods for Decision Tree Induction* (Vol. 4). *Machine Learning.* https://doi.org/10.1023/A:1022604100933

Firmansyach, W. A., Hayati, U., & Wijaya, Y. A. (2023). Analisa Terjadinya Overfitting Dan Underfiiting Pada Algoritma Naive Bayes Dan Descision Tree Dengan Teknik Cross Validation. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 7, Number 1).

Gadomer, Ł., & Sosnowski, Z. A. (2020). Pruning trees in C-fuzzy random forest. *Soft Computing*, *25*(3), 1995–2013. https://doi.org/10.1007/s00500-020-05270-3

Gliselda, V. K. (2021). *Diagnosis dan Manajemen Penyakit Ginjal Kronis (PGK)*. http://jurnalmedikahutama.com

Harahap, A. H., Muttaqin, I., Malik, B. A., Irfan, M., Imam, N., Thariq, M., Bilhaq, S., Nur, A. A., Lutfia, S., & Agustini, D. (2021). Klasifikasi Diagnosa Penyakit Jantung menggunakan Algoritma Random Forest. *Gunung Djati Conference Series*, *3*.

Imaduddin, H., Yusuf, S. A. A., & Adhantoro, M. S. (2026). A robust model for early detection of chronic kidney disease leveraging machine learning and data balancing techniques. *Bulletin of Electrical Engineering and Informatics*, *15*(2), 1442–1451. https://doi.org/10.11591/eei.v15i2.11247

Juba, B., & Le, H. S. (2019). *Precision-Recall versus Accuracy and the Role of Large Data Sets*. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(01), 4039–4048. https://doi.org/10.1609/aaai.v33i01.33014039

Kurniawan, I., Buani, D. C. P., Apriliah, W., & Fitriani, E. (2023). Penerapan Teknik Random Undersampling untuk Mengatasi Imbalance Class dalam Prediksi Kebakaran Hutan Menggunakan Algoritma Decision Tree. In *AJCSR [Academic Journal of Computer Science Research* (Vol. 5, Number 1).

Nhu, V. H., Shahabi, H., Nohani, E., Shirzadi, A., Al-Ansari, N., Bahrami, S., Miraki, S., Geertsema, M., & Nguyen, H. (2020). Daily water level prediction of zrebar lake (Iran): A comparison between m5p, random forest, random tree and reduced error pruning trees algorithms. *ISPRS International Journal of Geo-Information*, *9*(8). https://doi.org/10.3390/ijgi9080479

Prasojo, B., & Haryatmi, E. (2021). Analisa Prediksi Kelayakan Pemberian Kredit Pinjaman dengan Metode Random Forest. *Jurnal Nasional Teknologi Dan Sistem Informasi*, *7*(2), 79–89. https://doi.org/10.25077/teknosi.v7i2.2021.79-89

*Pre-processed Chronic Kidney Disease Dataset.* | *Kaggle*. (n.d.). Retrieved April 14, 2023, from https://www.kaggle.com/datasets/mahmoudlimam/preprocessed-chronic-kidney-disease-dataset

Rahayu, E. S., Satria, R., Dan, W., & Supriyanto, C. (2015). Penerapan Metode Average Gain, Threshold Pruning dan Cost Complexity Pruning untuk Split Atribut pada Algoritma C4.5. *Journal of Intelligent Systems*, *1*(2). http://journal.ilmukomputer.org

Sandag, G. (2020). Prediksi Rating Aplikasi App Store Menggunakan Algoritma Random Forest Application Rating Prediction on App Store using Random Forest Algorithm. *Cogito Smart Journal*, *6*(2).

Sandag, G. A., Leopold, J., & Ong, V. F. (2018). Klasifikasi Malicious Websites Menggunakan Algoritma K-NN Berdasarkan Application Layers dan Network Characteristics Malicious Websites Classification Using K-NN Algorithm Based on Application Layers and Network Characteristics. *Cogito Smart Journal*, *4*(1).

Sari, L., Romadloni, A., Listyaningrum, R., Studi, P., Informatika, T., Cilacap, P. N., & Soetomo, J. (2023). Penerapan Data Mining dalam Analisis Prediksi Kanker Paru Menggunakan Algoritma Random Forest. *Infotekmesin*, *14*(01). https://doi.org/10.35970/infotekmesin.v14i1.1751

Urnkranz, J. F. ¨. (1997). *Pruning Algorithms for Rule Learning* (Vol. 27). Kluwer Academic Publishers.

Wardhani, A. K., Nugraha, E., Ulfiana, Q., Medis, R., Kesehatan, I., Rukun, P., & Luhur, A. (2022). Optimization of the Decision Tree Method using Pruning on Liver Disease Classification. In *Journal of Applied Informatics and Computing (JAIC)* (Vol. 6, Number 2). http://jurnal.polibatam.ac.id/index.php/JAIC

Wayan, N., & Wardani, S. (2022). Pemberdayaan Masyarakat Dalam Pencegahan Penyakit Ginjal Pasien Diabetes Dan Hipertensi Di RSUD Sanjiwani Gianyar. *Jurnal Lingkungan & Pembangunan*, *6*(1). https://ejournal.warmadewa.ac.id/index.php/wicaksana

Wibowo, A. D., Himawan, I., Suryana, A., Artiyasa, M., & Pradiftha, A. (2020). Gambaran Umum Metode Klasifikasi Data Mining. *FIDELITY Jurnal Teknik Elektro 2(2)*, *2*(2), 34–38. https://doi.org/10.52005/fidelity.v2i2.111