Implementasi Random Forest pada Klasifikasi Penyakit Kardiovaskular dengan Hyperparameter Tuning Grid Search

I Ketut Adian Jayaditya[a1], I Gusti Agung Gede Arya Kadyanan[a2]

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Udayana, Bali
Jln. Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, 08261, Bali, Indonesia
[1]adianjay11@gmail.com
[2]gungde@unud.ac.id

Abstract

Cardiovascular disease has the potential to cause death if not treated right, because it interferes with the function of the heart. Machine Learning algorithm can be used to do early diagnosis of cardiovascular disease to lower the risk of death. In this study, the classification of cardiovascular disease uses the Random Forest algorithm to determine whether a person has cardiovascular disease or not. Grid Search is also used to do hyperparameter tuning to find the optimal hyperparameter for the Random Forest algorithm. The performance results of the classification model using Random Forest with Grid Search are 73.06% in accuracy, 75.15% in precision, 68.72% in recall, and 71.79% in f1-score.

Keywords: *Cardiovascular Disease, Random Forest, Hyperparameter Tuning, Grid Search*

1. Pendahuluan

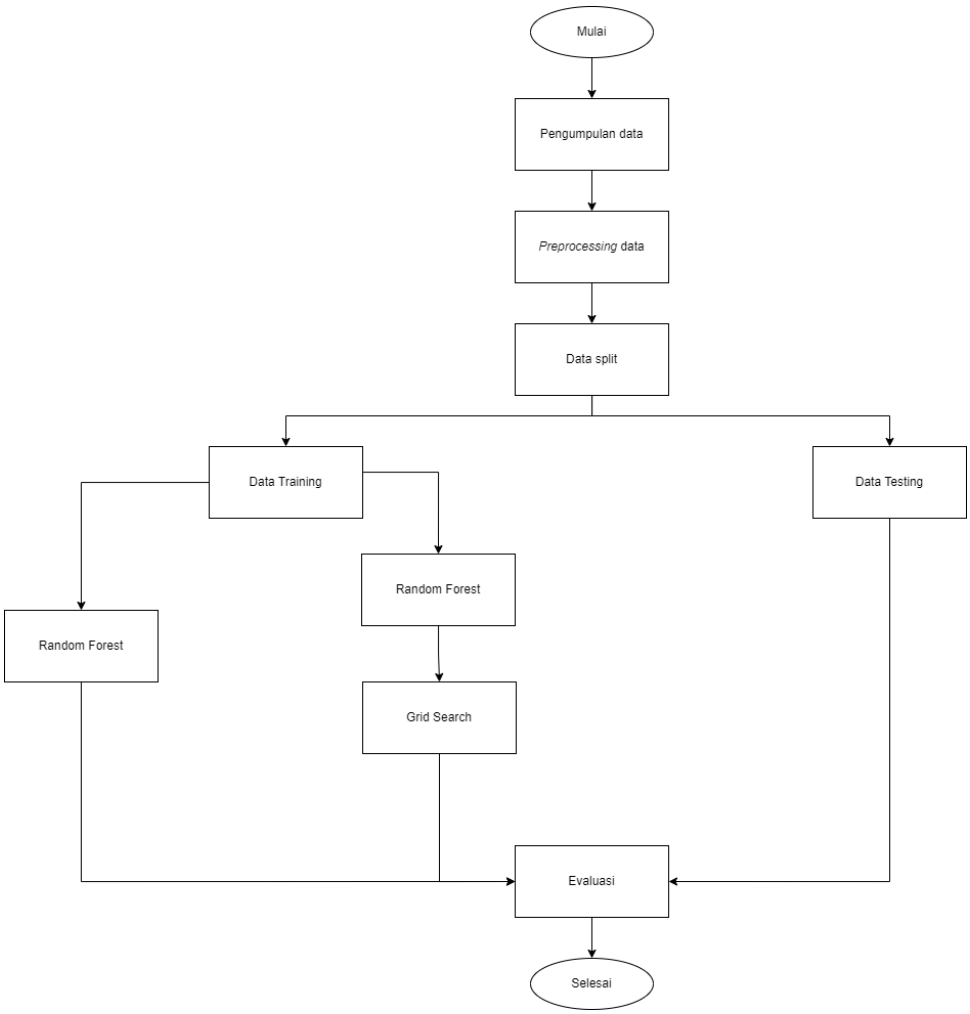
Penyakit kardiovaskular merupakan penyakit yang dapat mengakibatkan jumlah kematian nomor satu di dunia. Penyakit ini tergolong tidak menular dan penyakit ini biasanya terjadi gangguan pada jantung dan pembuluh darah seperti penyakit jantung koroner, gagal jantung, hipertensi, dan stroke [1]. Data dari World Health Organization mengatakan bahwa lebih dari 17 juta orang di dunia mengalami kematian yang diakibatkan oleh penyakit jantung dan pembuluh darah [2]. Dengan meningkatnya angka kematian setiap tahunnya, maka diperlukan suatu sistem klasifikasi yang dapat mendiagnosis sejak dini adanya penyakit kardiovaskular pada seseorang. *Machine Learning* dapat menjadi salah satu alat yang dapat digunakan untuk mengklasifikasikan penyakit kardiovaskular pada seseorang.

Terdapat beberapa algoritma *Machine Learning* yang dapat digunakan untuk permasalahan klasifikasi, diantaranya *Support Vector Machine, Logistic Regression, Random Forest, Decision Tree,* dan *Naïve Bayes*. Algoritma *Random Forest* menunjukkan performa yang cukup baik ketika mengklasifikasi pada data medis. Penelitian yang dilakukan oleh Sabrina, dkk pada tahun 2023, peneliti membandingkan algoritma *Decision Tree* dengan *Random Forest* untuk melakukan klasifikasi pada penyakit jantung. Algoritma *Decision Tree* meraih akurasi sebesar 77.44% dan *Random Forest* meraih akurasi sebesar 81.82% [3]. Selain itu, penelitian yang dilakukan oleh Wahyu Nugraha dan Agung Sasongko pada tahun 2023 melakukan *hyperparameter tuning* pada tujuh algoritma *Machine Learning* untuk mendapatkan performa yang optimal [4]. Penelitian tersebut menunjukkan hasil Algoritma *XGBoost* memperoleh nilai terbaik sebesar 0,772 sedangkan algoritma *Decision Tree* memperoleh nilai terendah sebesar 0,701.

Pada penelitian ini dilakukan klasifikasi terhadap penyakit kardiovaskular menggunakan algoritma *Random Forest* dengan *hyperparameter tuning* menggunakan *Grid Search*. Pada penelitian ini juga akan dilakukan perbandingan performa dari algoritma *Random Forest* sebelum dan sesudah melakukan *hyperparameter tuning* menggunakan *Grid Search*.

2. Metode Penelitian

Berikut merupakan tahapan – tahapan dari penelitian yang dilakukan.



Gambar 1. Alur Penelitian

2. 1. Pengumpulan data

Data yang digunakan dalam penelitian ini merupakan data sekunder yang diperoleh dari website kaggle.com dengan nama *cardiovascular disease dataset* dalam bentuk *comma-separated value* (csv). Data ini memiliki 12 atribut dengan total 70.000 *instance*, dimana sejumlah 34.979 *instance* untuk penderita penyakit kardiovaskular, sedangkan sejumlah 35.021 *instance* untuk kelas tidak menderita penyakit kardiovaskular.

Tabel 1. Deskripsi Dataset

Atribut	Deskripsi
Age	Umur
Height	Tinggi badan
Weight	Berat badan
Gender	Jenis kelamin
ap_hi	*Systolic blood pressure* atau tekanan darah sistolik

Atribut	Deskripsi
ap_lo	*Diastolic blood pressure* atau tekanan darah diastolik
cholesterol	Kadar kolesterol (1 = normal, 2 = diatas normal, 3 = jauh diatas normal)
gluc	Kadar gula darah atau glukosa (1 = normal, 2 = diatas normal, 3 = jauh diatas normal)
smoke	Perokok (1 = ya, 0 = tidak)
alco	Meminum alkohol (1 = ya, 0 = tidak)
active	Aktif berolahraga (1 = ya, 0 = tidak)
cardio	Label penyakit kardiovaskular (1 = menderita penyakit kardiovaskular, 0 = tidak menderita penyakit kardiovaskular)

2. 2. *Preprocessing* data

Sebelum data digunakan untuk melatih model *Random Forest*, diperlukan adanya *preprocessing* data agar tidak berdampak buruk pada performa dari model tersebut. Pada penelitian ini, *preprocessing* data mencakup penghapusan terhadap data duplikat, menghapus adanya data *outlier*, dan melakukan *label encoder* untuk data yang bersifat kategorikal. Setelah dilakukan *preprocessing* data, kemudian data tersebut akan dipecah menjadi data *training* dan data *testing* dengan rasio 70: 30.

2. 3. Random Forest

Random Forest adalah suatu model klasifikasi yang terdiri dari kumpulan beberapa pohon klasifikasi, dimana setiap pengklasifikasi menghasilkan suatu suara atau *voting* terhadap kelas tertentu berdasarkan dari *input vector* yang diberikan [5]. Pohon keputusan dimulai dengan menghitung *entropy* sebagai penentu ketidakmurnian atribut dan nilai *information gain*. Rumus persamaan 1 digunakan untuk menghitung *entropy*, sedangkan persamaan 2 digunakan untuk menghitung *information gain* [6].

$$Entropy\ (Y) = \sum_{i=1}^{n} -p\ (c_i) \log_2 (p(c_i) \tag{1}$$

Dimana Y merupakan himpunan kasus dan $p(c_i)$ merupakan probabilitas atau persentase dari kelas c_i pada suatu *node*.

$$Information\ Gain\ (Y, A) = Entropy(Y)\ - \sum_{v\ \epsilon\ Values(A)} \frac{|Y_v|}{|Y_a|} Entropy(Y_v) \tag{2}$$

Dimana *Values(A)* merupakan semua nilai yang mungkin pada himpunan kelas A. Y_v ialah *subclass* dari Y dengan kelas v yang berhubungan dengan kelas a. Y_a merupakan semua nilai yang sesuai dengan a. *Information Gain* tertinggi dari atribut-atribut yang ada menjadi dasar untuk pemilihat atribut pada simpul [6].

2. 4. Grid Search

Grid search merupakan suatu metode yang dapat digunakan untuk mencari *hyperparameter* yang optimal untuk meningkatkan performa dari model klasifikasi. Grid Search ini bekerja dengan cara mencoba semua kombinasi yang mungkin dari *hyperparameter* yang sudah didefinisikan sebelumnya dan menentukan kombinasi *hyperparameter* optimal yang menghasilkan kinerja model klasifikasi terbaik [7]. Grid Search biasanya digabungkan dengan dengan *k-fold cross-validation* untuk menentukan *hyperparameter* terbaik dan biasanya disebut dengan Grid Search Cross-Validation atau GridSearchCV [8].

2. 5. Confusion Matrix

Confusion matrix digunakan sebagai alat untuk mengukur jumlah ketepatan klasifikasi terhadap kelas dengan model *Machine Learning* yang dipakai [2].

Tabel 2. *Confusion Matrix*

	Nilai Prediksi	
Nilai Sebenarnya	**Positif (1)**	**Negatif (0)**
Positif (1)	True Positive (TP)	False Negative (FN)
Negatif (0)	False Positive (FP)	True Negative (TN)

Melalui *confusion matrix*, dapat dilakukan perhitungan untuk mengidentifikasikan performa dari model *Machine Learning* yang digunakan. Nilai yang dapat dihitung untuk mengidentifikasi performa, yaitu akurasi, *recall*, *precision*, dan *f1-score*.

a. Akurasi, dihitung dengan cara membagi jumlah data yang diklasifikasikan benar oleh model dengan total data.

$$akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

b. *Precision*, dihitung dengan cara membagi jumlah data *True Positive* dengan jumlah data *True Positive* ditambah data *False Positive*.

$$precision = \frac{TP}{TP + FP} \quad (4)$$

c. *Recall*, dihitung dengan cara membagi jumlah data *True Positive* dengan jumlah data *True Positive* ditambah data *False Negative*.

$$recall = \frac{TP}{TP + FN} \quad (5)$$

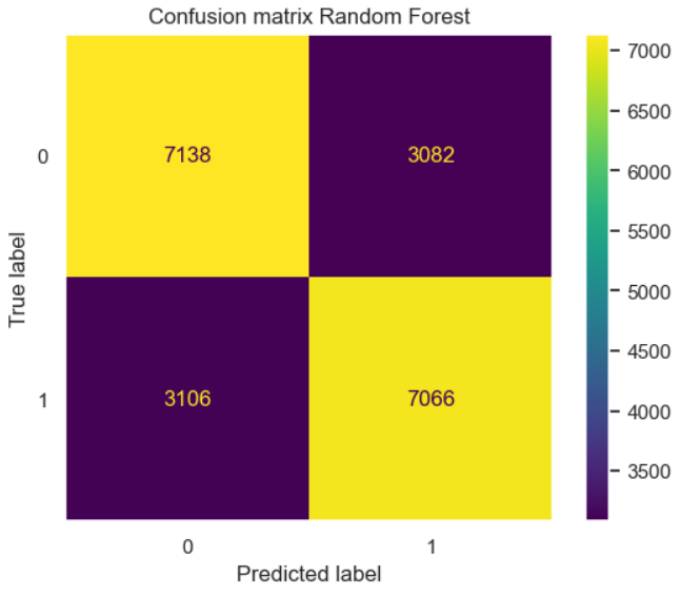
d. *F1-score,* didapatkan dengan cara pembagian hasil perkalian *precision* dan *recall* dengan hasil penjumlahannya lalu dikalikan dua.

$$F1\ score = \frac{2 \cdot precision \cdot recall}{precision + recall} \quad (6)$$

3. Hasil dan Pembahasan

3.1. Performa Random Forest sebelum Hyperparameter Tuning

Perfoma *Random Forest* sebelum dilakukan *hyperparameter tuning* diukur menggunakan data *testing* yang sebelumnya sudah dipisah. *Hyperparameter* dari *Random Forest* menggunakan nilai *default*, diantaranya yaitu *max_depth = None*, *max_features = sqrt, min_samples_leaf = 1, min_samples_split = 2, n_estimators = 100*. Gambar 2 menunjukkan *confusion matrix* dari model pada data *testing*.



Gambar 2. *Confusion matrix Random Forest*

Performa dari model klasifikasi *Random Forest* tanpa adanya *hyperparameter tuning* dapat dilihat pada Tabel 3.

Tabel 3. Performa *Random Forest*

	Accuracy	Precision	Recall	F1-Score
Random Forest	69.65%	69.62%	69.46%	69.54%

3.2. Performa Random Forest setelah Hyperparameter Tuning

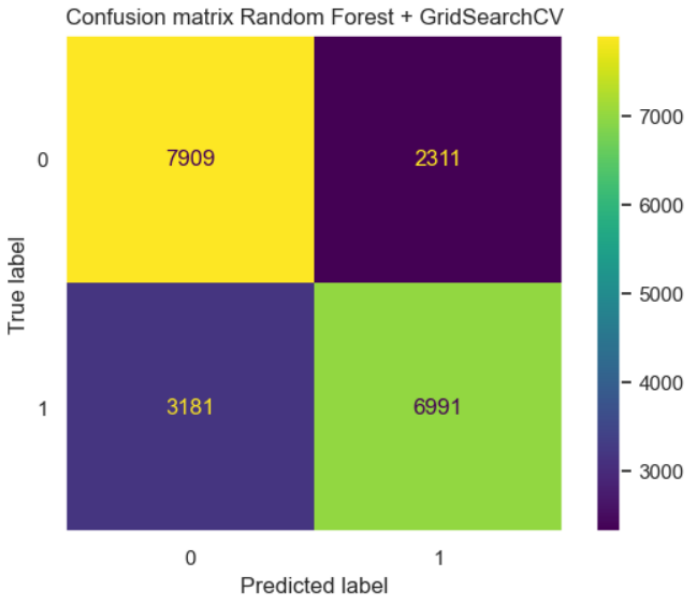
Hyperparameter tuning dilakukan untuk mencari *hyperparameter max_depth*, *max_features*, *min_samples_leaf*, *min_samples_split*, dan *n_estimators* yang optimal pada model *Random Forest*. *Hyperparameter tuning* dilakukan menggunakan *Grid Search Cross-Validation* dengan jumlah *k-fold* bernilai 5. Gambar 3 menunjukkan *hyperparameter* optimal yang didapatkan untuk model klasiifkasi.

```
Fitting 5 folds for each of 32 candidates, totalling 160 fits

{'max_depth': 80,
 'max_features': 2,
 'min_samples_leaf': 4,
 'min_samples_split': 10,
 'n_estimators': 200}
```

Gambar 3. *Hyperparameter* Optimal

Selanjutnya, dibangun model klasifikasi *Random Forest* sesuai dengan *hyperparameter* yang sudah didapatkan. Gambar berikut menunjukkan *confusion matrix* dari model klasifikasi *Random Forest* dengan *hyperparameter tuning GridSearchCV*.



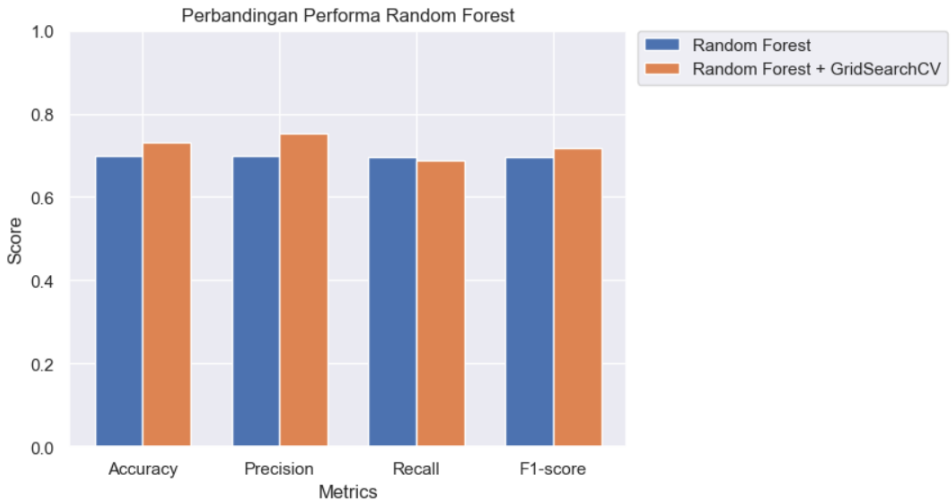
Gambar 4. *Confusion Matrix Random Forest + GridSearchCV*

Performa dari model klasifikasi *Random Forest* dengan *hyperparameter tuning* GridSearchCV dapat dilihat pada Tabel 4.

Tabel 4. Performa *Random Forest* + GridSearchCV

	Accuracy	Precision	Recall	F1-Score
Random Forest	73.06%	75.15%	68.72%	71.79%

Setelah dilakukan *hyperparameter tuning*, dapat dilihat bahwa performa dari model klasifikasi *Random Forest* memiliki kenaikan. Akurasi yang awalnya 69.65% naik menjadi 73.06%, *precision* yang awalnya 69.62% naik menjadi 75.15%, *recall* yang awalnya 69.46% turun menjadi 68.72%, dan *f1-score* yang awalnya 69.54% naik menjadi 71.79%. Gambar 5 menunjukkan perbandingan dari model *Random Forest* sesudah dan sebelum dilakukan *hyperparameter tuning* menggunakan GridSearchCV.



Gambar 5. Perbandingan Performa *Random Forest*

4. Kesimpulan

Berdasarkan hasil penelitian, ditemukan bahwa algoritma *Random Forest* dapat digunakan untuk mengklasifikasikan penyakit kardiovaskular. Performa dari algoritma ini sebelum dilakukan *hyperparameter tuning* ialah akurasi sebesar 69.65%, *precision* sebesar 69.62%, *recall* sebesar 69.46%, dan *f1-score* sebesar 69.54%. *Hyperparameter tuning* GridSearchCV dengan jumlah *k-fold* bernilai 5 dilakukan dengan cara mencoba beberapa kombinasi dari *hyperparameter* yang mendukung model *Random Forest*. *Hyperparameter* yang optimal kemudian diterapkan kembali pada model *Random Forest*. Hasilnya adalah akurasi mengalami kenaikan menjadi 73.06%, *precision* mengalami kenaikan menjadi 75.15%, *recall* mengamali penurunan menjadi 68.72%, dan *f1-score* mengalami kenaikan menjadi 71.79%.

Untuk penelitian selanjutnya, dapat diketahui bahwa GridSearchCV memiliki kelemahan berupa proses mencari *hyperparameter* optimal yang lama dikarenakan banyaknya *hyperparameter* yang harus dioptimalkan dan jumlah *k-fold* pada saat melakukan *cross validation*. Beberapa alternatif yang dapat digunakan untuk melakukan *hyperparameter tuning* ialah menggunakan algoritma koloni atau algoritma evolusi, seperti algoritma genetika.

Daftar Pustaka

[1] A. Desiani, M. Akbar, I. Irmeilyana, and A. Amran, “Implementasi Algoritma Naïve Bayes dan Support Vector Machine (SVM) Pada Klasifikasi Penyakit Kardiovaskular,” *Jurnal Teknik Elektro dan Komputasi (ELKOM)*, vol. 4, no. 2, pp. 207–214, Aug. 2022, doi: 10.32528/elkom. v4i2.7691.

[2] D. Andri, A. Mutoi, and R. Rahmat, “Penerapan Algoritma K-Nearest Neighbord untuk Prediksi Kematian Akibat Penyakit Gagal Jantung,” *Scientific Student Journal for Information, Technology and Science*, vol. 3, pp. 105–112, 2022.

[3] Sabrina Adnin Kamila, R. R. S. Sulistijowati, and I. Susanto, “Classification of Heart Disease Using Decision Tree and Random Forest,” *STAINS (Seminar Nasional Teknologi & SAINS)*, vol. 2, no. 1, pp. 7–12, Jan. 2023, [Online]. Available: https://proceeding.unpkediri.ac.id/index.php/stains/article/view/2816

[4] W. Nugraha and A. Sasongko, “Hyperparameter Tuning on Classification Algorithm with Grid Search,” *SISTEMASI*, vol. 11, no. 2, p. 391, May 2022, doi: 10.32520/stmsi. v11i2.1750.

[5] A. Hidayatullah, I. Muttaqin, M. Irfan, M. Thariq, A. Amini, and S. Lufia, “Classification of Heart Disease Diagnosis using the Random Forest Algorithm,” *Mini Seminar Kelas Data Mining*, vol. 3, pp. 42–51, 2021.

[6] V. Wanika and I. Elvina, “Prediksi Harga Ponsel Menggunakan Metode Random Forest,” *Annual Research Seminar (ARS)*, vol. 4, pp. 144–147, 2018.

[7] M. Fajri and A. Primajaya, “Komparasi Teknik Hyperparameter Optimization pada SVM untuk Permasalahan Klasifikasi dengan Menggunakan Grid Search dan Random Search,” *Journal of Applied Informatics and Computing*, vol. 7, no. 1, Jan. 2023, doi: 10.30871/jaic. v7i1.5004.

[8] A. Toha, P. Purwono, and W. Gata, “Model Prediksi Kualitas Udara dengan Support Vector Machines dengan Optimasi Hyperparameter GridSearch CV,” *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 4, no. 1, pp. 12–21, May 2022, doi: 10.12928/biste. v4i1.6079.

Halaman ini sengaja dibiarkan kosong