

**ANALISIS PERBANDINGAN ALGORITMA *DECISION TREE*
DAN *k-NN* (*k-NEAREST NEIGHBOR*) DENGAN OPTIMASI
BAGGING UNTUK KLASIFIKASI PENYAKIT JANTUNG**

**COMPARATIVE ANALYSIS OF *DECISION TREE* AND *k-NN*
(*k-NEAREST NEIGHBOR*) ALGORITHMS WITH *BAGGING*
OPTIMIZATION FOR HEART DISEASE CLASSIFICATION**

Fitri Yanti¹, Zainal Abidin², Bijanto³

Prodi Informatika Sekolah Tinggi Teknik Pati, Indonesia^{1,2,3}

e-mail: yanti061202@gmail.com¹, zainal.fsr@yahoo.id², biyantokakoi@gmail.com³

Abstract : Heart disease is one of the leading causes of death in the world, so an accurate method is needed to detect and classify this disease early. This study aims to compare the performance of the Decision tree Algorithm and k-Nearest Neighbor (k-NN) in the classification of heart disease with the application of Bagging optimization. The evaluation was carried out by measuring accuracy, error, Precision, and recall to determine the most effective algorithm. The results showed that Decision tree with Bagging had the highest accuracy of 97.82%, while k-NN only reached 93.78%. In terms of error rate, Decision tree had a lower error rate (2.18%) compared to k-NN (6.22%). In addition, the Precision and recall in Decision tree were 96.94% and 99.15%, respectively, higher than k-NN which reached 95.60% and 92.88%. With these results, it can be concluded that Decision tree with Bagging optimization is superior to k-NN in the classification of heart disease. Bagging optimization has proven effective in increasing model accuracy and reducing the level of prediction errors. The results of this study are expected to be a reference in the development of an artificial intelligence-based medical decision support system for more accurate and efficient heart disease diagnosis.

Keywords: Heart disease, Algorithm Decision tree, Algorithm k-Nearest Neighbor, Bagging, classification.

Abstrak : Penyakit jantung merupakan salah satu penyebab utama kematian di dunia, sehingga diperlukan metode yang akurat untuk mendeteksi dan mengklasifikasikan penyakit ini secara dini. Penelitian ini bertujuan untuk membandingkan performa Algoritma *Decision tree* dan *k-Nearest Neighbor* (*k-NN*) dalam klasifikasi penyakit jantung dengan penerapan optimasi *Bagging*. Evaluasi dilakukan dengan mengukur *accuracy*, *error*, *Precision*, dan *recall* untuk menentukan algoritma yang paling efektif. Hasil penelitian menunjukkan bahwa *Decision tree* dengan *Bagging* memiliki akurasi tertinggi sebesar 97,82%, sementara *k-NN* hanya mencapai 93,78%. Dari segi *error rate*, *Decision tree* memiliki tingkat kesalahan lebih rendah (2,18%) dibandingkan *k-NN* (6,22%). Selain itu, *Precision* dan *recall* pada *Decision tree* masing-masing sebesar 96,94% dan 99,15%, lebih tinggi dibandingkan *k-NN* yang mencapai 95,60% dan 92,88%. Dengan hasil tersebut, dapat disimpulkan bahwa *Decision tree* dengan optimasi *Bagging* lebih unggul dibandingkan *k-NN* dalam klasifikasi penyakit jantung. Optimasi *Bagging* terbukti efektif dalam meningkatkan akurasi model serta mengurangi tingkat kesalahan prediksi. Hasil penelitian ini diharapkan dapat menjadi referensi dalam pengembangan sistem pendukung keputusan medis berbasis kecerdasan buatan untuk diagnosis penyakit jantung yang lebih akurat dan efisien.

Kata Kunci: Penyakit jantung, Algoritma *Decision tree*, Algoritma *k-Nearest Neighbor*, *Bagging*, klasifikasi.

PENDAHULUAN

Menjaga kesehatan tubuh hal yang sangat penting untuk memastikan fisik tetap bugar dan produktif [1]. Kesehatan sangat penting untuk kelangsungan hidup manusia. Tanpa tubuh yang sehat, seseorang akan menghadapi kesulitan dalam menjalankan berbagai aktivitasnya [2]. Kesehatan mengacu pada kondisi di mana fungsi fisiologis tubuh berjalan normal tanpa adanya gangguan yang menyebabkan rasa sakit. Karena itu, jika keadaan ini diabaikan, akan memberikan dampak negatif pada tubuh. Tubuh akan mengalami penurunan kekuatan karena kurangnya aktivitas fisik yang dilakukan, sehingga daya tahan tubuh pun menurun [3]. Namun demikian, setiap orang memiliki harapan untuk tetap sehat [4]. Karena itu, menjaga kesehatan sangatlah penting untuk mencegah berbagai penyakit, seperti stroke, gangguan jantung, hepatitis, kanker, dan lainnya.

Jantung merupakan organ yang terdiri dari otot berongga yang berbentuk kerucut, terbagi menjadi empat ruang, yaitu ventrikel kanan, ventrikel kiri, atrium kanan, dan atrium kiri [5]. Fungsi utama jantung adalah memompa darah ke seluruh tubuh. Apabila pembuluh darah mengalami penyempitan, hal ini akan mengganggu fungsi jantung dan dapat menyebabkan timbulnya penyakit jantung [6]. Penyakit ini dapat disebabkan oleh berbagai faktor, seperti pola makan yang tidak sehat, kurangnya aktivitas fisik, stres, faktor keturunan, usia, serta kondisi medis lain seperti hipertensi dan diabetes [7][8]. Penyakit jantung adalah gangguan yang mempengaruhi sistem pembuluh darah, sehingga menyebabkan jantung dan sistem peredaran darah tidak berfungsi dengan semestinya [9]. Maka, jika kurangnya pasokan darah ke tubuh atau organ seperti otak dapat mengakibatkan kematian.

Penyakit jantung merupakan salah satu penyakit dengan tingkat kematian yang tinggi [10]. Menurut statistik dunia, setiap tahunnya terjadi sekitar 9,4 juta kematian akibat penyakit kardiovaskular. Penyakit Kardiovaskular (PKV) adalah kondisi yang terjadi akibat gangguan pada fungsi jantung dan pembuluh darah [11]. Dari jumlah tersebut, sebanyak 45% disebabkan oleh Penyakit Jantung. Organisasi Kesehatan Dunia (WHO) memperkirakan bahwa angka tersebut akan meningkat menjadi sekitar 23,3 juta pada tahun 2030 mendatang [12]. Karena kurangnya mereka dalam memperhatikan kesehatan jantung mereka. Di berbagai negara, penyakit jantung koroner juga menjadi salah satu penyebab utama kematian akibat penyakit kardiovaskular. Contohnya, di Amerika Serikat, sekitar 56% kematian disebabkan oleh penyakit kardiovaskular, yang sebagian besar disebabkan oleh Penyakit Jantung Koroner (PJK). Hal serupa terjadi di Inggris, di mana penyakit kardiovaskular juga menjadi penyebab kematian utama, dengan Penyakit Jantung Koroner (PJK) sebagai penyebabnya [13]. Di Indonesia sendiri, angka kejadian penyakit jantung mencapai 1,5% pada seluruh penduduk, artinya dari setiap 100 orang, 1,5% mengalami penyakit jantung. Sebanyak 15 provinsi juga memiliki tingkat kejadian di atas rata-rata nasional. Kelompok usia 65-74 tahun memiliki tingkat kematian tertinggi akibat Penyakit Jantung Koroner (PJK), meskipun terdapat kasus PJK pada usia muda, yaitu 15-24 tahun [16]. Provinsi Jawa Timur menjadi yang teratas dengan jumlah kasus Penyakit Jantung Koroner (PJK) tertinggi, mencapai 144.279 orang. Kota Surabaya merupakan kota di Jawa Timur yang menempati posisi kedua tertinggi setelah kota Madiun [14]. Di Sumatera Utara, jumlah kasus akibat Penyakit Jantung Koroner (PJK) yang terdiagnosis oleh dokter adalah 0,5%, sementara yang belum pernah didiagnosis tetapi mengalami gejala serupa penyakit jantung koroner adalah 1,1% [15]. Diperlukannya penanganan pendeteksi penyakit jantung secara dini

dengan cara menggunakan teknologi yang dapat membantu dalam penanganan penyakit.

Perkembangan teknologi merupakan hal yang tidak dapat diabaikan dalam kehidupan, dalam hal ini pemanfaatan teknologi digunakan dalam upaya pengolahan data yang akurat dan dapat membantu dokter dalam menganalisa gejala-gejala penyakit jantung. Teknologi juga dapat memberikan informasi dalam bidang medis termasuk kesehatan jantung [16]. Pemantauan terhadap perkembangan kesehatan jantung akan memudahkan pihak para medis untuk mendeteksi gejala penyakit jantung [17]. Untuk deteksi kesehatan jantung memerlukan penerapan data mining.

Data mining tidak hanya diterapkan di bidang teknologi, tetapi juga digunakan dalam bidang kesehatan. Data mining dapat digunakan sebagai referensi untuk memprediksi dan mendiagnosis berbagai jenis penyakit dengan memanfaatkan metode-metode yang dapat diterapkan [18]. Salah satu teknik data mining yaitu *Decision tree* dan *k-Nearest Neighbor*. Algoritma *Decision tree* digunakan untuk melakukan klasifikasi terhadap data dengan membuat pohon keputusan [19]. Sedangkan *k-Nearest Neighbor* digunakan untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut [20]. Analisis penyakit jantung menggunakan pengolahan data dan akan dilakukan dengan mengklasifikasikan atribut menjadi 0 (Normal) dan 1 (Penderita Penyakit Jantung), sehingga teknik klasifikasi dapat digunakan dalam penelitian ini.

Beberapa penelitian mengenai klasifikasi penyakit jantung telah banyak dilakukan antara lain, penelitian yang dilakukan oleh Yovi Pratama, Anton Prayitno, Defri Nazrian, Nur Aini, Yoga Rizki R, Errissya Rasywir tahun 2022 yang berjudul *Klasifikasi Penyakit Gagal Jantung Menggunakan Algoritma K-Nearest Neighbor*. Penulis menerapkan algoritma *KNN* terhadap data penyakit gagal jantung dilakukan dengan pengujian akurasi yang dilakukan dengan menggunakan aplikasi *Rapidminer* [18]. Penelitian lain yang dilakukan oleh Arni Sepharni, Iwansyah Edo Hendrawan, Chaerur Rozikin pada tahun 2022 dengan judul *Klasifikasi Penyakit Jantung Dengan Menggunakan Algoritma C4.5* [21]. Penelitian ini bertujuan untuk membuat sistem prediksi dengan menggunakan algoritma *C4.5* dimana Algoritma ini membuat prediksi berdasarkan data historis pasien yang akan diperiksa. Hasil penelitian yang dilakukan menggunakan Algoritma *C4.5* yang diterapkan dengan eksperimen menggunakan *k-fold cross validation* dengan jumlah $k=10$. Pembagian dataset menghasilkan nilai *accuracy*, *Precision*, *recall* yang berbeda-beda tiap pembagian datanya dari hasil Algoritma *C4.5*. Penelitian lain yang dilakukan oleh Ardea Bagas Wibisono dan Achmad Fahrurrozi pada tahun 2019 dengan judul *Perbandingan Algoritma Klasifikasi Dalam Pengklasifikasian Data Penyakit Jantung Koroner*. Metode yang diterapkan meliputi *Naïve Bayes*, *K-Nearest Neighbor*, *Decision tree*, dan *Random Forest*. Metode *Naïve Bayes* menggunakan probabilitas disetiap data, metode *K-Nearest Neighbor* menggunakan perhitungan jarak, metode *Decision tree* menggunakan pohon keputusan, sedangkan metode *Random Forest* menggunakan beberapa pohon keputusan yang disatukan. Penelitian ini bertujuan untuk membandingkan keempat algoritma tersebut dalam mengklasifikasikan data penyakit jantung koroner. Perbandingan algoritma akan dilihat berdasarkan *performance measure* yang terdiri dari tingkatan *accurasi*, *recall* disetiap kelas, dan presisi disetiap kelas [19]. Dari beberapa penelitian yang sudah dilakukan, belum ada penerapan optimasi *bagging* yang digunakan sehingga penelitian ini perlu

dilakukan dengan menggunakan dua algoritma yaitu algoritma *Decision tree*, dan *k-Nearest Neighbor* sebagai pembandingan.

Berdasarkan latar belakang di atas, penelitian ini dilakukan untuk membandingkan kedua algoritma dengan optimasi *Bagging* guna menentukan algoritma mana yang paling akurat dalam melakukan klasifikasi terhadap data penyakit jantung, dengan mengukur performanya berdasarkan *accuracy*, *error*, *presisi* dan *recall*

METODE PENELITIAN

Dataset yang digunakan dalam penelitian ini adalah dataset penyakit jantung berjudul *Heart Failure Prediction*, yang diperoleh dari platform *Kaggle Learning Repository*. Situs ini bersifat publik dan telah dimanfaatkan oleh banyak peneliti di berbagai belahan dunia. *Kaggle* sendiri merupakan platform yang menyediakan kumpulan data global untuk keperluan penelitian. Dataset yang digunakan dalam studi ini mencakup 11 atribut serta 1 label dengan total 918 data. Penelitian ini akan membandingkan kinerja Algoritma *Decision tree* dan *k-Nearest Neighbor*, yang selanjutnya dioptimalkan dengan metode *Bootstrap Aggregating (Bagging)* guna meningkatkan efektivitas dalam proses klasifikasi.

A. Optimasi *Bootstrap Aggregating (Bagging)*

Bagging (*Bootstrap Aggregating*) adalah teknik ensemble learning yang pertama kali diperkenalkan oleh Leo Breiman pada tahun 1996 [22]. *Bagging* adalah metode yang digunakan untuk meningkatkan stabilitas dan akurasi model machine learning dengan mengurangi ketidakstabilan dari model dasar yang digunakan [23]. Metode ini menerapkan teknik bootstrap sampling, yaitu pengambilan *sampel* acak dengan penggantian, untuk membuat beberapa *sampel* data dari dataset asli [24] [25]. *Bagging* kemudian melatih beberapa model dasar pada masing-masing bagian data secara independent (terpisah), dan hasil dari model-model ini digabungkan untuk menghasilkan prediksi akhir.

Alasan mengapa *Bagging* sering digunakan dibandingkan dengan metode optimasi lainnya adalah sebagai berikut [26]:

- 1) *Bagging* membantu mengurangi variasi model sehingga menghasilkan model ensemble yang lebih stabil dan konsisten.
- 2) *Bagging* dapat mengatasi masalah ketidakseimbangan kelas dengan memberikan bobot yang seimbang pada setiap kelas
- 3) *Bagging* relatif mudah diimplementasikan dan dapat diterapkan pada berbagai jenis model pembelajaran mesin tanpa perlu penyesuaian yang signifikan.

Tujuan *bagging* adalah untuk meningkatkan akurasi klasifikasi dengan menggabungkan hasil dari beberapa pengklasifikasi, dengan menghasilkan kinerja yang lebih baik dari pada pengambilan *sampel* acak [27]. *Bagging* bekerja dengan cara membagi data training menjadi beberapa bagian, kemudian dibuat model klasifikasi dari masing masing bagian untuk mendapatkan hasil prediksi dari model tersebut [58]. Hasil akhir didapatkan dengan cara voting. Voting yang dilakukan dengan cara mengambil suara terbanyak.

Teknik *Bootstrap Aggregating (Bagging)* dapat diterapkan dengan mengimplementasikan algoritma lain yang mendukung proses kinerja *bagging*.

Beberapa algoritma yang sering digunakan untuk mendukung *Bagging* adalah Algoritma *Decision tree* dan Algoritma *k-NN*.

a. Kelebihan *Bagging*. *Bagging* memiliki beberapa kelebihan diantaranya [28]:

1) Mengurangi varians

Dalam konteks statistik dan machine learning, varians merujuk pada ukuran seberapa jauh nilai-nilai dari suatu dataset atau model menyimpang dari rata-ratanya. Varians yang tinggi menunjukkan bahwa data atau hasil prediksi model bervariasi secara luas, sedangkan varians yang rendah menunjukkan bahwa data atau hasil prediksi lebih konsisten dan berdekatan dengan rata-rata. *Bagging* dapat mengurangi varians yang muncul dari setiap model dengan menggabungkan hasil dari beberapa model, sehingga menjadikannya lebih stabil.

2) Peningkatan akurasi

Secara umum, metode ini dapat meningkatkan akurasi prediksi dibandingkan dengan model tunggal, terutama pada dataset dengan noise tinggi.

3) Tahan terhadap overfitting

Bagging membantu mengurangi risiko overfitting yang sering terjadi pada model yang kompleks, karena prediksi akhir adalah hasil gabungan dari beberapa model.

b. Kekurangan *Bagging*. Meskipun *bagging* memiliki kelebihan, terdapat beberapa kekurangan pada *bagging* antara lain [29]:

1) Biaya komputasi tinggi

Pelatihan banyak model memerlukan waktu dan sumber daya komputasi yang lebih besar, yang dapat menjadi kendala dalam skala besar.

2) Model dasar yang terlalu mirip

Jika model dasar yang digunakan dalam *Bagging* terlalu mirip, efek agregasi mungkin tidak akan terlihat secara signifikan. Sebab itu, pemilihan model yang beragam menjadi hal yang krusial untuk mengoptimalkan hasil.

3) Klasifikasi Menggunakan Algoritma *Decision tree*

Algoritma *Decision tree* merupakan salah satu algoritma yang dapat digunakan untuk membangun pohon keputusan (*Decision tree*). Algoritma ini sangat populer dan sering digunakan oleh peneliti di seluruh dunia [30]. Algoritma *Decision tree* adalah pengembangan dari algoritma ID3 yang diciptakan oleh J. Rose Quinlan. Algoritma *Decision tree* secara berulang mengunjungi setiap simpul keputusan dan memilih percabangan yang terbaik hingga tidak ada lagi cabang yang dapat ditambahkan [31]. Tahapan dalam penerapan Algoritma *Decision tree* meliputi menyiapkan data pelatihan, menentukan akar pohon dengan menghitung *entropy*, menghitung nilai *gain*, dan kemudian menentukan tupel yang akan dipartisi. Langkah-langkah dalam perhitungan Algoritma *Decision tree* adalah sebagai berikut [32]:

1. Melakukan pengelompokan setiap atribut ke dalam setiap kelas tertentu untuk dijadikan akar atau *node*.
2. Menentukan akar yang memiliki nilai *gain* tertinggi atau nilai terendah dari *entropy*. Untuk menentukan Nilai *Entropy* dapat menggunakan persamaan 1, berikut:

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

Keterangan :

S : Himpunan Kasus

N : Jumlah partisi S

Pi : Proporsi dari Si terhadap S

Sedangkan untuk menentukan Nilai *Gain* menggunakan persamaan 2 di bawah ini:

$$Gain(S, A) = entropy(S) - \sum_{i=1}^n \frac{|S_i|}{S} \times Entropy(S) \quad (2)$$

Keterangan :

S : himpunan kasus

A : atribut

N : jumlah partisi atribut A

|Si| : jumlah kasus pada partisi ke-i

|S| : jumlah kasus dalam S

3. Mengulangi proses ke-2 hingga semua node terisi secara merata.
4. Proses partisi pohon keputusan akan berhenti disaat:
 - a. Semua tupel dalam *record* dalam simpul N mendapat kelas yang sama.
 - b. Tidak ada atribut dalam *record* yang dipartisi lagi.
 - c. Tidak ada *record* di dalam cabang yang kosong.

Algoritma *Decision tree* adalah metode klasifikasi berbasis pohon keputusan yang populer karena keunggulannya dibandingkan algoritma lainnya [66]. Keunggulan algoritma *Decision tree* yaitu dapat menghasilkan pohon keputusan yang mudah dipahami, tingkat akurasi yang baik, efisiensi dalam menangani atribut diskret, serta kemampuannya untuk mengelola atribut baik diskret maupun numerik [33]. Dengan demikian, Algoritma *Decision tree* sangat cocok digunakan untuk mengambil keputusan berdasarkan kondisi tertentu sesuai dengan data yang ada.

4) Klasifikasi Menggunakan Algoritma *k-Nearest Neighbor*

Algoritma *k-Nearest Neighbor* (*k-NN*) merupakan salah satu metode dalam machine learning yang digunakan untuk mengklasifikasikan objek berdasarkan data latih yang tersedia. Algoritma ini bekerja dengan mengklasifikasikan objek berdasarkan jarak tetangga terdekat dari data yang ada [34]. Sebagai algoritma supervised learning, *k-NN* mengklasifikasikan instance baru dengan melihat mayoritas kategori dari tetangga terdekatnya, di mana kelas yang paling sering muncul akan menjadi hasil dari klasifikasi tersebut.

Algoritma *k-NN* dimana merupakan algoritma yang digunakan untuk melakukan klasifikasi terhadap suatu data, berdasarkan *k* data latih yang jaraknya paling dekat dengan data tersebut [35]. Penentuan nilai *k* dalam algoritma klasifikasi *k-Nearest Neighbor* dapat dicari berdasarkan nilai *sampel k* terdekat k_1, k_2, \dots, k_s . Semakin banyak data yang ada semakin kecil jumlah *k* yang dipilih, sedangkan jika ukuran dimensi data yang ada lebih besar, jumlah *k* yang dipilih harus lebih tinggi. Dalam menentukan nilai *k*, lebih baik menggunakan angka ganjil seperti $k = 1, 3, 5, \dots$, dst. Salah satu alasan mengapa menggunakan nilai *k* ganjil untuk melatih dataset adalah berdasarkan penelitian yang dilakukan oleh Indriani, yang menunjukkan bahwa nilai *k* ganjil menghasilkan tingkat akurasi yang lebih baik. Selain itu, penggunaan *k* ganjil dapat mengurangi kesalahan dalam proses klasifikasi oleh *k-NN* karena jika nilai *K* genap, akan ada jumlah tetangga yang sama [36]. Nilai *k* harus memenuhi syarat yaitu $k < N$ dimana *N* merupakan jumlah dari dataset latih, karena nilai *k* digunakan untuk mencari jumlah mayoritas dari kelas/label pada data latih maka nilai *k* tidak boleh lebih dari jumlah dataset latih. Berikut adalah tahapan langkah-langkah dalam Algoritma *k-NN* [37]:

1. Menentukan nilai *k* (jumlah tetangga paling dekat).
2. Melakukan perhitungan nilai jarak (*Euclidean distance*) terhadap masing masing objek data yang diberikan. Rumus menghitung *Euclidean distance*.

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^n (x_{1i} - x_{2i})^2} \quad (3)$$

Keterangan:

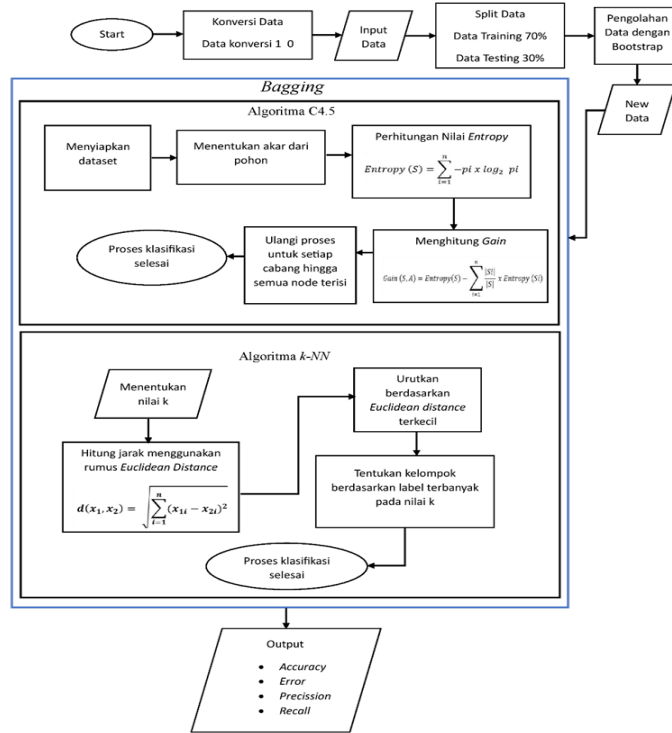
X_{1i} = Data Training

X_{2i} = Data Testing

3. Melakukan pengelompokan data sesuai dengan perhitungan jarak (*Euclidean distance*).
4. Melakukan pengelompokan data sesuai dengan nilai tetangga terdekat (*nearest neighbor*) atau berdasarkan data yang mempunyai jarak *Euclidean* terkecil.
5. Memilih nilai mayoritas dari tetangga terdekat sebagai hasil klasifikasi.

5) Tahap Penelitian

Tahapan-tahapan penelitian yang dilakukan dalam penelitian ini disajikan dalam bentuk diagram alur atau *Flowchart*. Tahapan tersebut dapat dilihat pada gambar 1, sebagai berikut:



Gambar 1 Flowchart Penelitian

(Sumber: Pustaka Peneliti)

HASIL

1. Persiapan Data

Dalam tahap persiapan data, penelitian ini menggunakan dataset yang diperoleh dari Kaggle dengan judul "Heart Failure Prediction Dataset", yang diunggah oleh fedesoriano pada September 2021. Dataset tersebut terdiri dari 918 record, dengan 11 atribut dan 1 label. Label target pada dataset ini adalah "HeartDisease", yang dikategorikan ke dalam dua variabel: 1 untuk menunjukkan individu yang menderita penyakit jantung, dan 0 untuk individu yang tidak memiliki penyakit jantung (normal). Data yang digunakan dalam penelitian ini disajikan pada Tabel 1 berikut.

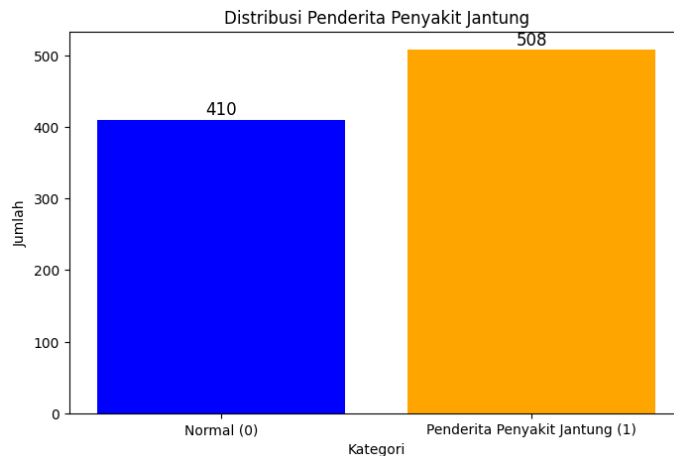
Tabel 1 Data sampel Penelitian

No	Age	Sex	chest Pain Type	Resting BP	Cholesterol	Fasting BS	Resting ECG	Max HR	Exercise Angina	Oldpeak	ST_Slope	Heart Disease
1	40	M	ATA	140	289	0	Normal	172	N	0	Up	0
2	49	F	NAP	160	180	0	Normal	156	N	1	Flat	1
3	37	M	ATA	130	283	0	ST	98	N	0	Up	0
4	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
5	54	M	NAP	150	195	0	Normal	122	N	0	Up	0

...
91	45	M	TA	110	264	0	Normal	132	N	1.2	Flat	1
4												
91	68	M	ASY	144	193	1	Normal	141	N	3.4	Flat	1
5												
91	57	M	ASY	130	131	0	Normal	115	Y	1.2	Flat	1
6												
91	57	F	ATA	130	236	0	LVH	174	N	0	Flat	1
7												
91	38	M	NAP	138	175	0	Normal	173	N	0	Up	0
8												

Sumber: (Kaggle 2021)

Berdasarkan tabel 4.1 dapat dijelaskan bahwa atribut “HeartDiseases” merupakan variabel target dimana terdapat 2 variabel target yaitu 0 (normal) dan 1 (penderita penyakit jantung).



Gambar 2 Grafik pembagian value pada atribut HeartDisease

Pada gambar 2 menunjukkan bahwa terdapat sebanyak 410 data masuk ke dalam kategori pasien 0 (normal) dan 508 data masuk ke dalam kategori pasien 1 (Penderita Penyakit jantung). Berdasarkan pembagian dari keseluruhan kategori tersebut, maka terdapat total keseluruhan data sebanyak 918 data. Untuk atribut yang lain dapat dijelaskan pada tabel 4.2 sebagai berikut:

Tabel 2 Penjelasan atribut data

No	Atribut	Keterangan	Keterangan
1	Age	Umur pasien	Atribut
2	Sex	Jenis kelamin pasien, memiliki 2 kategori yaitu M (Male) dan F (Female)	Atribut
3	ChestPainType	Jenis nyeri dada yang diderita pasien. Atribut ini memiliki 4 kategori, yaitu: TA = Typical Angina ATA = Atypical Angina NAP = Non-Anginal Pain	Atribut

No	Atribut	Keterangan	Keterangan
		<i>ASY = Asymptomatic</i>	
4	<i>RestingBP</i>	Tekanan Darah	Atribut
5	<i>Cholesterol</i>	<i>Cholesterol</i> yaitu kadar kolesterol dalam darah pasien, dengan satuan mm/dl	Atribut
6	<i>FastingBS</i>	<i>Fasting blood sugar</i> yaitu kadar gula darah pasien. Atribut fbs memiliki 2 kategori yaitu 1 jika kadar gula darah pasien lebih dari 120 mg/dl, dan 0 jika kadar gula darah pasien kurang dari sama dengan 120 mg/dl.	Atribut
7	<i>RestingECG</i>	<i>Resting electrocardiographic</i> yaitu kondisi ECG pasien ketika dalam keadaan istirahat. Atribut ini memiliki 3 kategori yaitu: Normal = untuk keadaan normal ST = untuk keadaan yang memiliki kelainan gelombang ST-T (Inversi gelombang T dan elevasi atau depresi ST meningkat maupun menurun lebih dari 0,05 Mv LVH = Menunjukkan hipertrofi ventrikel kiri yang mungkin atau pasti berdasarkan kriteria Estes	Atribut
8	<i>MaxHR</i>	Detak jantung maksimum yang dicapai	Atribut
9	<i>ExerciseAngina</i>	<i>Angina</i> yang disebabkan oleh Latihan fisik. Atribut ini memiliki 2 kategori yaitu Y (Ya) dan N (Tidak)	Atribut
10	<i>Oldpeak</i>	Penurunan ST akibat olahraga	Atribut
11	<i>ST_Slope</i>	<i>Slope</i> dari puncak ST setelah berolahraga. Atribut ini memiliki 3 kategori yaitu <i>Up</i> untuk menaik, <i>Flat</i> untuk datar, dan <i>Down</i> untuk menurun.	Atribut
12	<i>HeartDiasis</i>	Detak jantung pasien. Atribut ini memiliki 2 kategori yaitu 1 untuk penderita penyakit jantung, dan 0 untuk normal.	Label

2. Konversi Data

Pada tahap ini, dilakukan dikonversi terlebih dahulu agar memudahkan proses perhitungan klasifikasi menggunakan Algoritma *Decision tree* dan Algoritma *k-NN*. Berikut penjelasan cara konversi data dapat dilihat pada tabel 3.

Tabel 3 Cara Konversi Data

Atribut	Record	Kategori
Age	29 - 45	0

Atribut	Record	Kategori
Sex	46 - 61	1
	62 - 77	2
ChestPainType	M (Male)	0
	F (Female)	1
RestingECG	TA	0
	ATA	1
	NAP	2
	SAY	3
ExerciseAngina	Normal	0
	ST	1
	LVH	2
ST_Slope	Y	0
	N	1
	Flat	0
	Up	1
	Down	2

Tabel 4 Data Hasil Konversi Data

No	Age	Sex	Chest Pain Type	Resting BP	Choles terol	Fasting BS	Resting ECG	Max HR	Exercise Angina	Old peak	ST_Slope	Heart Disease
1	0	0	1	140	289	0	0	172	1	0	1	0
2	1	1	2	160	180	0	0	156	1	1	0	1
3	0	0	1	130	283	0	1	98	1	0	1	0
4	1	1	3	138	214	0	0	108	0	1.5	0	1
5	1	0	2	150	195	0	0	122	1	0	1	0
...
91	0	0	0	110	264	0	0	132	1	1.2	0	1
4												
91	2	0	3	144	193	1	0	141	1	3.4	0	1
5												
91	1	0	3	130	131	0	0	115	0	1.2	0	1
6												
91	1	1	1	130	236	0	2	174	1	0	0	1
7												
91	0	0	2	138	175	0	0	173	1	0	1	0
8												

3. Split Data

Split data merupakan tahapan pembagian data menjadi dua bagian, yaitu data *training* dan data *testing*. Perbandingan rasio antara data *testing* dan *training* dalam penelitian ini sebesar 70% : 30% dari total keseluruhan data. Jadi, data yang akan digunakan untuk *testing* sebesar 275 *record*, sedangkan data *training* sebesar 643 *record*. Tujuan dari penerapan split data untuk melatih dan menguji model secara efektif serta menghindari *overfitting* (kondisi di mana model yang dikembangkan terlalu "menghafal" pola-pola yang ada dalam data latih, sehingga tidak dapat secara efektif memprediksi data yang belum pernah dilihat sebelumnya (data uji atau data baru)) dan memastikan generalisasi yang baik dari model yang dikembangkan.

4. Pengolahan menggunakan *Bootstrap Aggregating (Bagging)*

Teknik *bootstrap* menggunakan metode yang berbeda untuk memperkirakan distribusi *sampel*. Metode ini mengambil data *sampel* yang diperoleh, dan kemudian melakukan *resample* data tersebut secara berulang untuk membuat banyak simulasi *sampel*. Proses *resampling* dilakukan secara acak dan dapat membentuk berbagai

variasi dataset simulasi. Dengan demikian, setiap kali dilakukan *bootstrap*, akan dihasilkan parameter sampel yang baru setelah proses tersebut selesai. Hasil dari proses *bootstrapping* dapat disajikan pada tabel 5 berikut:

Tabel 5 Data sampel hasil *bootstrapping* data

No	Heart Disease	Age	Sex	Chest Pain Type	Resting BP	Cholesterol	Fasting BS	Resting ECG	Max HR	Exercise Angina	Oldpeak	ST_Slope
1	1	1	1	2	160	180	0	0	156	1	1	0
2	1	1	1	3	138	214	0	0	108	0	1.5	0
3	0	1	0	2	150	195	0	0	122	1	0	1
...
301	1	2	0	3	140	260	0	1	112	0	3	0
302	1	2	0	3	140	260	0	1	112	0	3	0
303	0	0	0	3	130	209	0	1	127	1	0	1
...
640	1	1	1	1	130	236	0	2	174	1	0	0
641	0	0	0	2	138	175	0	0	173	1	0	1
642	0	0	0	2	138	175	0	0	173	1	0	1
643	0	0	0	2	138	175	0	0	173	1	0	1

Dari 918 data kemudian membuat data baru menjadi 643 data karena data tersebut di ambil dari data latih. Berdasarkan tabel 4.7 terdapat beberapa data yang memiliki duplikasi sama dengan data sebelumnya. Sebagai contoh pada data nomor 301 dan 302 yang memiliki duplikasi. Duplikasi tersebut diperoleh dengan menerapkan teknik *sample bootstrapping*, setiap observasi dalam data dapat dipilih kembali dalam setiap *sample bootstrap* maka data yang sama dapat muncul beberapa kali dalam satu *sample bootstrap*. Data baru tersebut kemudian digunakan untuk melatih model *ensemble learning* dalam proses *Bagging*. Tujuan utama dari metode ini adalah meningkatkan akurasi klasifikasi dengan mengombinasikan hasil dari beberapa pengklasifikasi, sehingga menghasilkan kinerja yang lebih baik dibandingkan dengan pengambilan *sample* secara acak.

5. Klasifikasi menggunakan Algoritma C4.5

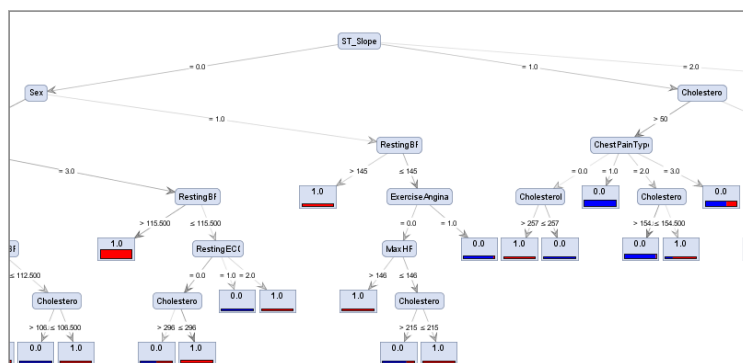
Algoritma *Decision tree* diterapkan untuk membangun pohon keputusan yang digunakan dalam menghitung nilai *accuracy*, *error*, *Precision*, dan *recall*. Pohon keputusan terdiri dari beberapa akar percabangan yang dapat ditentukan dengan menggunakan perhitungan *Entropy* dan *Gain* [114]. Nilai dari setiap atribut yang akan dijadikan cabang diperoleh dengan menentukan nilai *gain* tertinggi. Nilai *gain* dapat dihitung Setelah memperoleh nilai *entropy* dari setiap atribut yang ada. Peneliti hanya mengambil beberapa *sample* untuk mempermudah perhitungan, sehingga tidak perlu menghitung semua atribut. *Sample* yang digunakan mencakup atribut *chest pain type*, *fasting BS*, *exercise angina*, *ST-slope*, dan *heart disease*.

Hasil dari perhitungan nilai *entropy* dan *gain* dari masing-masing atribut dapat dilihat pada tabel 6, berikut:

Tabel 6 Hasil perhitungan nilai *entropy* dan *gain*

Atribut	Jumlah	Label 0	Label 1	Entropy	Gain
TOTAL :	643	292	351	0,9939	
ST_SLOPE :					0,3387
• 0	308	54	254	0,6697	
• 1	279	231	48	0,6623	
• 2	56	7	49	0,5435	
EXERCISE ANGINA :					0,1747
• 0	254	41	213	0,6376	
• 1	389	251	138	0,9382	
CHEST PAIN TYPE :					0,2211
• 0	24	12	12	1	
• 1	110	98	12	0,4971	
• 2	143	96	47	0,9135	
• 3	366	86	280	0,7865	
FASTING BS :					0,0372
• 0	488	252	236	0,9992	
• 1	155	40	115	0,8238	

Nilai *entropy* yang telah diperoleh dengan menggunakan persamaan 1, kemudian akan digunakan untuk mencari nilai *gain*. Nilai *gain* tertinggi digunakan sebagai dasar dalam menentukan cabang utama pada pohon keputusan. *Gain* digunakan untuk mengukur seberapa besar atribut ini dapat mengurangi ketidakpastian dalam klasifikasi penyakit jantung. Berdasarkan dari nilai *gain* dari masing-masing atribut yang telah dihitung dengan menggunakan persamaan 2 di atas, kemudian akan dicari nilai *gain* yang memiliki nilai tertinggi. Nilai *gain* tertinggi terdapat pada atribut *ST_Slope*. Sehingga, atribut *ST_Slope* akan dijadikan sebagai cabang utama dalam pohon keputusan (*root*). Penentuan cabang selanjutnya, dilakukan dengan nilai *gain* dan *entropy* menggunakan cara yang sama, tapi dengan mengacu pada nilai cabang utama yang memiliki nilai *entropy* tertinggi. Hasil pohon keputusan yang ditunjukkan pada gambar 3 berikut:



Gambar 3 Pohon Keputusan Algoritma *Decision Tree*

Penelitian ini menunjukkan hasil penerapan algoritma C4.5 yang di optimasi *Bootstrap Aggregating (Bagging)* untuk klasifikasi. Hasil nilai *entropy* dan *gain* berpengaruh dalam menentukan hasil pohon keputusan untuk proses klasifikasi. Nilai *entropy* dan *gain* tertinggi terdapat pada atribut *ST_Slope* yang memiliki nilai *gain* = 0,3387. Tahapan klasifikasi menggunakan Algoritma C4.5 dilakukan dengan menentukan nilai *Accuracy*, *Error*, *Precision*, dan *Recall*. Hasil klasifikasi menggunakan Algoritma C4.5 yang telah dioptimasi menggunakan *Bagging* disajikan pada tabel 6, sebagai berikut:

Tabel 6 Nilai *accuracy*, *error*, *Precision*, dan *recall*

Parameter	Hasil
<i>Accuracy</i>	97,82%
<i>Error</i>	2,18%
<i>Precision</i>	96,94%
<i>Recall</i>	99,15%

6. Klasifikasi Algoritma *k-Nearest Neighbor*

Tahap Klasifikasi menggunakan Algoritma *k-NN* dalam penelitian ini dalam mencari nilai *accuracy*, *error*, *precision* dan *recall*. Dalam proses perhitungan Algoritma *k-NN* perlu penghitungan jarak terdekat dari sebuah data. Penelitian ini menggunakan rumus *Euclidean distance* untuk menentukan jarak terdekat. Rumus *Euclidean distance* dalam penelitian ini dijabarkan sebagai berikut:

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^n (x_{1i} - x_{2i})^2} \tag{4}$$

Peneliti akan membandingkan menggunakan nilai *k* yaitu 3,5,7,9 untuk mencari hasil klasifikasi terbaik. Hasil terbaik akan dipilih dan digunakan sebagai output dari penelitian ini. Penelitian ini menggunakan nilai *k* ganjil guna menghindari jumlah jarak yang sama dalam proses klasifikasi sehingga hasil klasifikasi menjadi samar dalam penentuan prediksi tiap data.

Tabel 7 Hasil perbandingan pengujian klasifikasi nilai *k*

K=?	<i>Accuracy</i>	<i>Error</i>	<i>Precision</i>	<i>Recall</i>
3	93,78%	6,22%	91,72%	94,86%
5	86,94%	13,06%	85,37%	85,96%
7	81,34%	18,66%	78,67%	80,82%
9	80,40%	19,60%	76,77%	81,51%

Berdasarkan pada tabel 4.21 data perbandingan percobaan nilai k , dengan nilai $k=3$ dengan *accuracy* tertinggi sebesar 93,78%. Sehingga nilai $k=3$ dipilih untuk pengambilan model di Algoritma k -NN.

PEMBAHASAN

Setelah proses pengujian data dari hasil klasifikasi menggunakan Algoritma *Decision tree* dan Algoritma k -Nearest Neighbor yang optimasi *Bagging* selesai, selanjutnya adalah membandingkan hasil penelitian terbaik dari kedua Algoritma tersebut. Berdasarkan perhitungan yang telah dilakukan maka didapatkan hasil perbandingan *accuracy*, *error*, *Precision* dan *recall* dari kedua Algoritma yang dapat dilihat pada tabel 4.22 berikut ini:

Tabel 8 Hasil Perbandingan Algoritma *Decision tree* dan Algoritma k -NN

Metode	Akurasi	Error	Precision	Recall
<i>Decision tree</i>	97,82%	2,18%	96,94%	99,15%
k -NN	93,78%	6,22%	95,60%	92,88%

KESIMPULAN

Berdasarkan penelitian yang telah dilakukan mengenai analisis perbandingan Algoritma *Decision tree* dan Algoritma k -NN dengan optimasi *Bagging* untuk klasifikasi Penyakit jantung, dapat disimpulkan bahwa:

1. Hasil analisis menunjukkan bahwa *Decision tree* dengan optimasi *Bagging* memiliki akurasi lebih tinggi, yaitu 97,82%, dibandingkan k -NN yang hanya mencapai 93,78%. Selain itu, tingkat kesalahan (*error rate*) pada *Decision tree* lebih rendah, yaitu 2,18%, dibandingkan k -NN yang memiliki 6,22%. Dari segi evaluasi lainnya, *Decision tree* memiliki *Precision* sebesar 96,94% dan *recall* 99,15%, lebih baik dibandingkan k -NN yang memperoleh *Precision* 95,60% dan *recall* 92,88%. Hal ini membuktikan bahwa *Decision tree* dengan optimasi *Bagging* lebih efektif dalam mengklasifikasikan penyakit jantung dibandingkan k -NN.
2. Penelitian ini menunjukkan peningkatan akurasi dibandingkan dengan penelitian sebelumnya, di mana Algoritma *Decision tree* yang digunakan oleh I Made Agus Oka Gunawan dkk. (2023) menghasilkan akurasi 80,43% dengan *error* 19,57% dan AUC 0,798, serta Algoritma k -Nearest Neighbor yang digunakan oleh Ni Ketut Setiawati dan I Gede Arta Wibawa (2023) mencapai akurasi 91% pada $k=7$, sedangkan dalam penelitian ini, optimasi *Bootstrap Aggregating (Bagging)* meningkatkan akurasi menjadi 97,82% dengan *error* 2,18%, *Precision* 96,94%, *recall* 99,15% untuk *Decision tree*, serta akurasi 93,78%, *error* 6,22%, *Precision* 95,60%, dan *recall* 92,88% untuk k -NN pada $k=3$.

SARAN

Berdasarkan penelitian yang telah dilakukan, adapun saran yang akan diajukan peneliti untuk penelitian selanjutnya, sebagai berikut:

1. Pada penelitian selanjutnya tidak hanya mengeksplorasi lebih banyak metode, tetapi juga memanfaatkan *Python* atau *Google Colab* sebagai platform analisis. Penggunaan *Python* dengan pustaka seperti Scikit-learn, TensorFlow, atau *PyTorch* dapat mempermudah implementasi berbagai algoritma, meningkatkan efisiensi pemrosesan, serta memungkinkan eksperimen dengan teknik optimasi yang lebih luas.
2. Penelitian selanjutnya diharapkan bisa menggunakan teknik optimasi lain yang dapat diimplementasikan dengan algoritma berbeda untuk memperoleh hasil yang lebih maksimal.

DAFTAR PUSTAKA

- A. Fauzi, R. S. (2020). Deteksi Penyakit Kanker Payudara dengan Seleksi Fitur berbasis Principal Component Analysis dan Random Forest. *J. Infortech*, 2, 96-101. doi:10.31294/infortech.v2i1.8079
- A. Isnadiya, F. R. (2019). Pengaruh Emotional Freedom Technique (EFT) terhadap Tingkat Kecemasan Pasien Pre Percutaneous Coronary Intervention (PCI) di SMC RS Telogorejo. *J. Ilmu Keperawatan Med. Bedah*, 1, 12. doi:10.32584/jikmb.v1i2.187
- A. Sepharni, I. E. (2022). Klasifikasi Penyakit Jantung dengan Menggunakan Algoritma C4.5. *STRING (Satuan Tulisan Ris. dan Inov. Teknol.)*, 7, 117. doi:10.30998/string.v7i2.12012
- Agustine, Q. M. (2021). Study of Bagging Application in the Safe-Level Smote Method in Handling Unbalanced Classification. *Indones. J. Stat. Its Appl*, 5, 105-116. doi:10.29244/ijsa.v5i1p105-116.
- B. Rina, I. A. (2020). Kemitraan Program Posyandu dalam Upaya Pemberdayaan Masyarakat di Bidang Kesehatan. *Diklus J. Pendidik. Luar Sekol.*, 4, 112-123. doi:10.21831/diklus.v4i2.31620
- Bertalina, B. (2017). Hubungan Asupan Natrium, Gaya Hidup, Dan Faktor Genetik Dengan Tekanan Darah Pada Penderita Penyakit Jantung Koroner. *J. Kesehat*, 8, 240. doi:10.26630/jk.v8i2.467
- Buani, D. C. (2021). Penerapan Algoritma Naïve Bayes dengan Seleksi Fitur Algoritma Genetika Untuk Prediksi Gagal Jantung. *EVOLUSI J. Sains dan Manaj*, 9, 43-48. doi:10.31294/evolusi.v9i2.11141
- D. Pradana, M. L. (2022). Klasifikasi Penyakit Jantung Menggunakan Metode Artificial Neural Network. *Indones. J. Data Sci*, 2, 55-60. doi:10.56705/ijodas.v3i2.35
- E. Alfaro, M. G. (2013). Adabag: An R package for classification with boosting and bagging. *J. Stat. Softw*, 54. doi:10.18637/jss.v054.i02
- Ernawati, Y. W. (2021). Klasifikasi Penyakit Batu Ginjal Menggunakan Algoritma Decision Tree C4 . 5 Dengan Membandingkan Hasil Uji Akurasi. *J. IKRA-ITH Inform*, 5, 128.

- H. A. Dwi Fasnuari, H. Y. (2022). Penerapan Algoritma K-Nearest Neighbor Untuk Klasifikasi Penyakit Diabetes Melitus. *Antivirus J. Ilm. Tek. Inform*, 16, 133-142. doi:10.35457/antivirus.v16i2.2445
- H. M. Nawawi, J. J. (2019). Komparasi Algoritma Neural Network Dan Naïve Bayes Untuk Memprediksi Penyakit Jantung. *J. Pilar Nusa Mandiri*, 15, 189-194. doi:10.33480/pilar.v15i2.669.
- Haris, M. M. (n.d.). Risiko Penyakit Kardiovaskuler Pada Peserta Program Pengelolaan Penyakit Kronis (Prolanis) Di Puskesmas Kota Bima: Korelasinya Dengan Ankle Brachial Index Dan Obesitas. *J. Keperawatan Indones*, 22, 200-208.
- I. Yulianti, R. A.—4. (2020). Optimasi Akurasi Algoritma C4.5 Berbasis Particle Swarm Optimization dengan Teknik Bagging pada Prediksi Penyakit Ginjal Kronis. *Techno.Com*, 19, 411-421. doi:10.33633/tc.v19i4.3579
- Ilham, A. (2020). Hybrid Metode Bootstrap Dan Teknik Imputasi Pada Metode C4-5 Untuk Prediksi Penyakit Ginjal Kronis. *Statistika*, 8, 43-51.
- J. Pseudocode, V. N. (2014). ALGORITMA KLASIFIKASI DATA MINING NAÏVE BAYES BERBASIS PARTICLE SWARM. 11-14.
- M. A. Adrian, M. R. (2021). Health Monitoring System dengan Indikator Suhu Tubuh, Detak Jantung dan Saturasi Oksigen Berbasis Internet of Things (IoT). *J. Petik*, 7, 108-118. doi:10.31980/jpetik.v7i2.1230
- M. D. Khairani. (2020). Prilaku Hidup Bersih dan Sehat: Perspektif Al-Qur'an dan Sunnah Rasul. *J. Darussalam Islam. Stud.*, 1, 31-44. doi: 10.47747/jdis.v1i1.89.
- M. Ginting, G. S. (2020). Pengetahuan Keluarga tentang Penyakit Jantung Koroner di Rumah Sakit Umum Bina Kasih Medan Sunggal. *J. Darma Agung Husada*, 7, 13-21.
- M. M. Baharuddin, H. A. (2019). Analisis Performa Metode K-Nearest Neighbor Untuk Identifikasi Jenis Kaca. *Ilk. J. Ilm*, 11, 269–274. doi:10.33096/ilkom.v11i3.489.269-274
- M. R. Sandi, S. M. (2019). the Description of Modifiable Risk Factors in Coronary Heart Disease At Dr. Soetomo Regional Public Hospital. *J. Berk. Epidomiol*, 7, 85. doi:10.20473/jeb.v7i22019.85-93.
- Mesran, D. P. (2020). Analisis Komparasi Metode Klasifikasi Data Mining dan Reduksi Atribut Pada Data Set Penyakit Jantung. *J. Media Inform. Budidarma*, 4, 437. doi:10.30865/mib.v4i2.2080.
- N. D. Saputri, K. K. (2022). Komparasi penerapan metode Bagging dan Adaboost pada Algoritma C4. 5 untuk prediksi Penyakit Stroke. *Sist. J. Sist. Inf*, 11, 567-577. Retrieved from <http://sistemasi.ftik.unisi.ac.id>
- N. Rahmad, A. D. (2022). Pengaruh Latihan Aktifitas Rehabilitasi Jantung Fase II Terhadap Tingkat Kebugaran dan Daya Tahan Pasien Penyakit Jantung Koroner. *J. Penelit. Kesehat. Suara Forikes*, 13, 1057-1058. Retrieved from <http://forikesejournal.com/index.php/SF>
- NIZIRWN ANWAR, J. M. (2022). Rancang Bangun Aplikasi Mobile Untuk Layanan Fisioterapi Online (Let'S Fisis). *J. INSTEK (Informatika Sains dan Teknol*, 7, 232-237. doi:10.24252/instek.v7i2.31655
- Octavianovol, A. M. (2023). Penerapan Data Mining Untuk Prediksi Hasil Keuntungan Lelang Mesin X-Ray Tahun 2020 Dengan Metode K-Nearest Neighbor (Studi

Jurnal EDU ELEKTROMATIKA (JEE)

ISSN: 2747-0784 (p); xxxxxx (e)

Vol 6, No. 1, Juni 2025

- Kasus : PT.Ramadika Mandiri). *J. Inform. MULTI*, 1, 2985-8860. Retrieved from <https://jurnal.publikasitecno.id/index.php/jim126>
- P. D. Nugraha, S. a. (2018). Klasifikasi Dokumen Menggunakan Metode Knn Dengan Information Gain. *eProceedings Eng*, 3, 1541–1550.
- P. M. Y. R. Atmaja, I. K. (2021). Aktivitas Fisik Serta Pola Hidup Sehat Masyarakat Sebagai Upaya Menjaga Kesehatan pada Masa Pandemi Covid-19. *Ilmu Keolahragaan Undiksha*, 9, 128. doi:10.23887/jiku.v9i231409
- Pratiwi, R. T. (2015). Penerapan Teknik Bagging Pada Algoritma Klasifikasi Untuk Mengatasi Ketidakseimbangan Kelas Dataset Medis. *J. Inform*, II, 395-403. Retrieved from <https://ejournal.bsi.ac.id/ejurnal/index.php/ji/article/view/118>
- R. P. Putra, E. R. (2022). Struktur Histologi dan Histomorfometri Jantung Kalkun (*Meleagris gallopavo*) Pada Tingkat Umur yang Berbeda. *J. Ilm. Mhs. Vet*, 6, 143–152.
- Rakasiwi, L. S. (2021). Pengaruh Faktor Demografi dan Sosial Ekonomi terhadap Status Kesehatan Individu di Indonesia. *Kaji. Ekon. dan Keuang*, 5, 146-157. doi:10.31685/kek.v5i2.1008.
- Rochmawati, M. I. (2023). Teknik Bagging Pada Algoritma Klasifikasi Decision Tree dan SVM Untuk Klasifikasi SMS Berbahasa Indonesia. *J. Informatics Comput. Sci*, 5, 265-271. doi:10.26740/jinacs.v5n02.p265-271
- Rusli, S. I. (2017). Ketepatan Klasifikasi Bagging CART Pada Klasifikasi Ketidaktepatan Waktu Kelulusan Mahasiswa STIKOM Bali. *Konf. Nas. Sist. dan Inform*, 237-240.
- Supriyatna, S. a. (2023). Prediksi Penyakit Diabetes Menggunakan Algoritma Random Forest. *Ijccs*, 17, 1-5.
- Wahono, T. S. (2015). Penerapan Metode Bagging Untuk Mengurangi Data Noise Pada Neural Network Untuk Estimasi Kuat Tekan Beton. *J. Intell. Syst.*, 1, 36-41.
- Y. Kusumawati, A. S. (2020). A Classification of Batik Kudus Based On Patterns Using K-NN AND GLCM. *no. February*, 1-6.
- Zein, Y. P. (2023). Model Balanced Bagging Berbasis Decision Tree Pada Dataset Imbalanced Class. *J. Sisfokom (Sistem Inf. dan Komputer)*, 12, 9-15. doi:10.32736/sisfokom.v12i1.1399