

Seleksi Atribut Pada Diagnosis Penyakit Liver Menggunakan *Decision Tree* Dengan Algoritma Genetika

Aang Pangantyas Sampurna^{a1}, I Gede Santi Astawa^{a2}, Nqurah Aquus Sanjaya ER^{a3}, AAIN Eka Karyawati^{a4}, I Wayan Sanityasa^{a5}, I Made Widiartha^{a6}

^aProgram Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Bali, Indonesia

¹aangpangantyas@gmail.com

²santi.astawa@unud.ac.id

³agus_sanjaya@unud.ac.id

⁴eka.karyawati@unud.ac.id

⁵santiyasa@unud.ac.id

⁶madewidiartha@unud.ac.id

Abstract

Liver disease is caused by inflammation of the liver. WHO shows nearly 1.2 million people in Southeast Asia and Africa per year die from this disease. Therefore, a diagnosis is needed as soon as possible to get further treatment. To make a diagnosis, a classification algorithm is needed which in this study uses the C4.5 algorithm. However, the algorithm is not optimal for forming a decision tree because it requires loading all cases into memory. Therefore, it is necessary to optimize using genetic algorithms to form simpler rules by selecting attributes and trying various possible combinations of attributes until the most optimal combination is obtained. In the evaluation results, the rules generated by optimization are simpler, namely as many as 32 rules when compared to without optimization, which are more complex, which are 145 rules. Then in the evaluation of accuracy, the rules with optimization resulted in a better accuracy of 70,7% when compared to the accuracy of the rules without optimization of 68,9%.

Keywords: *Liver Disease, Classification, Decision Tree C4.5, Optimization, Genetic Algorithm*

1. Pendahuluan

Penyakit Liver merupakan penyakit krusial pada organ hati di mana penyakit tersebut diakibatkan oleh inflamasi pada hati yang disebabkan infeksi virus, bahan beracun, atau bakteri sehingga hati kesulitan untuk berfungsi sebagaimana mestinya [1]. Berdasarkan data yang dikeluarkan oleh WHO (World Health Organization) bahwa setidaknya hampir 1.2 juta orang per tahun di kawasan Asia Tenggara dan kawasan Afrika mengalami kematian akibat terserang penyakit ini [2]. Agar penyakit tersebut tidak semakin parah, maka dari itu diperlukan diagnosis sejak dini agar penderita mendapatkan penanganan sesegera mungkin.

Beberapa penelitian terkait pernah dilakukan sebelumnya. Penelitian Rahman [3] membahas tentang perbandingan antara algoritma *decision tree C4.5* dan algoritma *naïve bayes* dalam melakukan klasifikasi pada diagnosis penyakit liver menggunakan perangkat lunak *RapidMiner*. Berdasarkan hasil penelitian, didapatkan bahwa algoritma *naïve bayes* mendapatkan akurasi sebesar 67.05% sementara algoritma *decision tree C4.5* mendapatkan akurasi yang lebih tinggi yaitu sebesar 70.29%. Penelitian lainnya yaitu dilakukan oleh Utami [4] di mana dalam penelitian tersebut membahas tentang penggunaan teknik klasifikasi atribut menggunakan *naïve bayes*. Sebelum dilakukan klasifikasi atribut menggunakan *naïve bayes*, digunakan Algoritma Genetika dan Bagging untuk seleksi atribut yang berpengaruh agar lebih optimal untuk memprediksi penyakit liver dengan dataset yang didapatkan dari UCI Machine Learning Repository. Dari hasil analisis, didapatkan bahwa dengan hanya menggunakan algoritma klasifikasi *naïve bayes* saja, didapatkan akurasi sebesar 66,66%. Sedangkan jika algoritma *naïve bayes* tersebut dikombinasikan dengan algoritma genetika dan bagging, maka didapatkan akurasi yang lebih baik sebesar 72,02%.

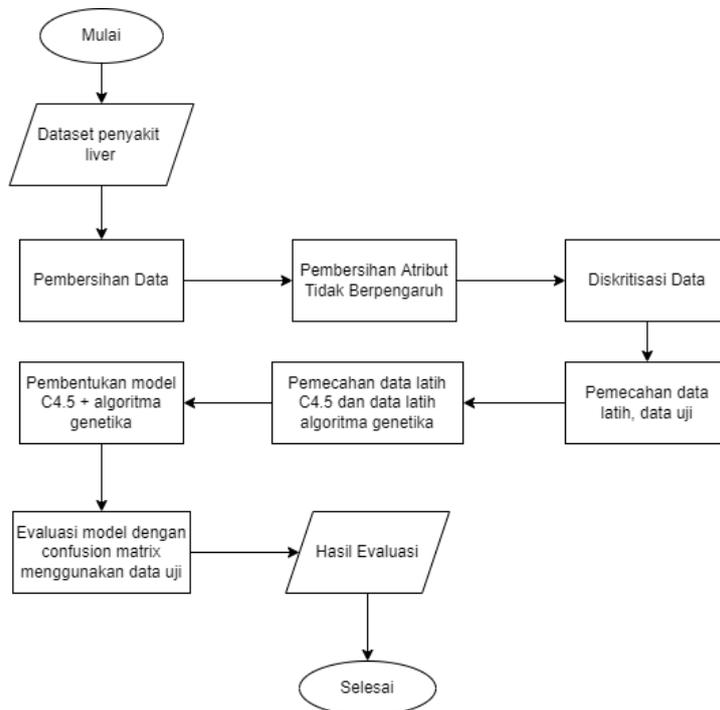
Pada penelitian ini penulis akan melakukan klasifikasi diagnosis penyakit liver menggunakan algoritma *decision tree C4.5*. Namun dikarenakan algoritma C4.5 kurang optimal dalam membentuk suatu pohon

keputusan/aturan yang mengharuskan memuat seluruh kasus ke dalam memori pohon keputusan, maka akan dilakukan kombinasi algoritma menggunakan Algoritma Genetika untuk pembentukan aturan yang lebih optimal dengan cara menyeleksi atribut. mencoba berbagai kemungkinan kombinasi atribut sampai didapatkan aturan yang paling optimal untuk dilakukan diagnosis dengan harapan akan mendapatkan akurasi yang lebih tinggi dan waktu diagnosis yang lebih cepat dibandingkan dengan diagnosis dengan Algoritma C4.5 tanpa kombinasi Algoritma Genetika untuk seleksi atribut.

2. Metode Penelitian

2.1. Alur Penelitian

Berdasarkan diagram alur penelitian pada gambar 1, dalam penelitian ini penulis menggunakan dataset yang diperoleh dari *UCI Machine Learning Repository*. Dari data penelitian yang telah ada kemudian dilakukan proses *preprocessing data* yang mencakup pembersihan data yang mengandung *missing value*, penghilangan atribut yang tidak relevan, dan diskritisasi data. Setelah tahap *preprocessing data* selesai, dilanjutkan dengan pembangunan model C4.5 tanpa seleksi atribut menggunakan algoritma genetika dan pembangunan model C4.5 dengan seleksi atribut menggunakan algoritma genetika. Pada tahap evaluasi dilakukan dengan membandingkan jumlah *rules*, dan *confusion matrix* menggunakan data uji antara model yang telah dilakukan seleksi atribut dan belum dilakukan seleksi atribut.



Gambar 1. Diagram Alur Penelitian

2.2. Data Penelitian

Penelitian ini menggunakan data sekunder yang didapat dari *Machine Learning Repository UCI (Universitas California Invene)* bernama *Indian Liver Patient Dataset (ILPD)* dengan alamat web: <http://archive.ics.uci.edu/ml/> dalam bentuk file *comma separated value (csv)*. Data ini terdiri dari 10 atribut berisi 583 record data, 416 record untuk kelas penderita penyakit liver dan 167 record untuk kelas tidak menderita penyakit liver. Deskripsi pada setiap atribut dapat di lihat pada Tabel 1.

Tabel 1. Deskripsi Data ILPD

Atribut	Deskripsi	Tipe Data
Age	Umur	Numerik
Gender	Jenis Kelamin	Kategorik
TB	Total Bilirubin	Numerik
DB	Direct Bilirubin	Numerik
Alkphos	Alkaline Phosphotase	Numerik
Sgpt	Alamine Aminotransferase	Numerik
Sgot	Aspartate Aminotransferase	Numerik
TP	Total Protiens	Numerik
ALB	Albumin	Numerik
A/G Ratio	Albumin and Globulin Ratio	Numerik
Class	Data Clases 1: Positive 2: Negative	Numerik

2.3. Pembersihan Atribut Tidak Berpengaruh

Pada tahap ini akan dilakukan proses seleksi atribut yang tidak berpengaruh terhadap pengambilan keputusan penyakit liver menggunakan perangkat lunak SPSS dengan metode *pearson correlation*.

2.4. Diskritisasi Data

Dataset yang digunakan pada penelitian ini memiliki beberapa atribut dengan data numerik yang sulit diterapkan pada algoritma *decision tree* C4.5 sehingga perlu dilakukan transformasi data dengan diskritisasi di mana pada penelitian ini menggunakan algoritma *entropy-based discretization* agar atribut dengan data numerik continyus bisa diubah menjadi data kategorikal dengan beberapa segmen atau interval. Berikut merupakan cara kerja dari algoritma ini:

- Mengurutkan data atribut yang akan dilakukan diskritisasi
- Hitung entropy dari keseluruhan dataset
- Untuk setiap split data didalam atribut yang telah diurutkan:
 - Hitung entropy di setiap bin data dengan rumus pada persamaan (1).
$$\text{Entropy}(S) = -\sum_{i=1}^n p_i \log_2(p_i) \quad (1)$$

Keterangan :
S = Himpunan kelas
n = jumlah kategori
 p_i = nilai probabilitas pada kategori i
 - Hitung *information gain* dengan menggunakan rumus pada persamaan (2).
$$\text{Info Gain}(S,A) = \text{Entropy}(s) - \sum_{i=1}^n \frac{|S_i|}{|S|} \text{Entropy}(S_i) \quad (2)$$

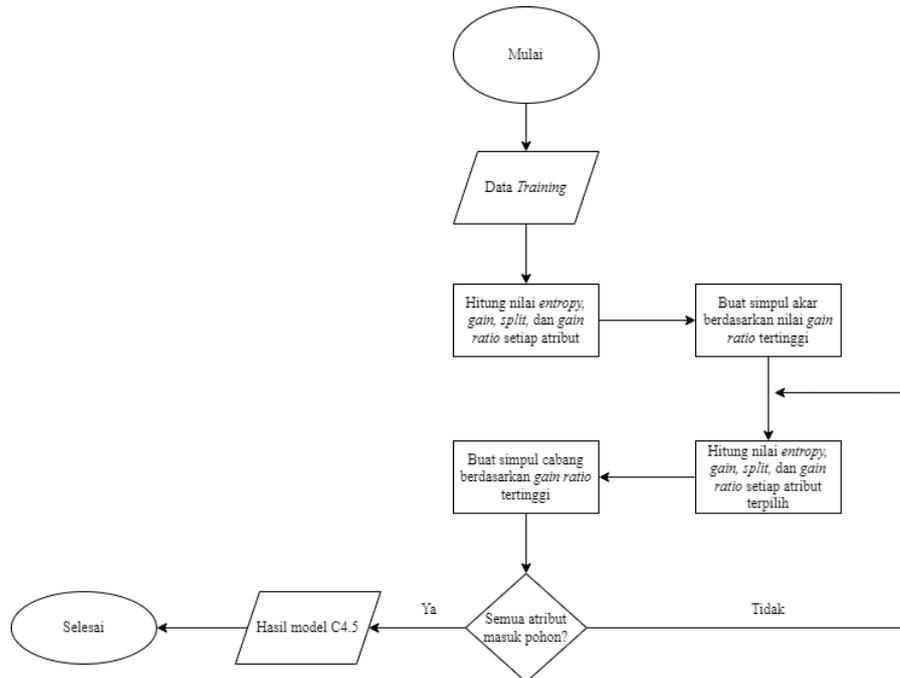
Keterangan:
S = Himpunan kelas
A = Atribut
n = Jumlah kategori
 $|S_i|$ = Jumlah kelas pada kategori ke i
 $|S|$ = Jumlah kelas dalam S
- Pilih posisi split (bin) dengan nilai gain paling tinggi
- Lakukan proses rekursif untuk setiap split yang telah ditentukan sampai dengan kondisi terminate. Kondisi terminate bisa dilakukan jika telah mencapai bin yang ditentukan atau jika gain dibawah dari treshold tertentu.

2.5. Pemecahan Data Latih dan Data Uji

Pada tahap ini akan dilakukan proses untuk pemecahan data latih dan data uji dengan perbandingan 80:20. Lalu dari data latih 80% tersebut akan dibagi menjadi data latih C4.5 dan data latih algoritma genetika dengan proporsi 80:20 di mana total data latih algoritma genetika akan sama dengan total data testing. Nantinya data latih C4.5 akan digunakan untuk membentuk model *decision tree* C4.5 lalu data latih algoritma genetika akan digunakan untuk mengukur tingkat akurasi dari model C4.5 yang nantinya juga dipergunakan untuk indikator nilai fitness. Kemudian data uji akan dilakukan untuk melakukan pengujian terhadap model *decision tree* yang belum dan telah dilakukan seleksi atribut.

2.6. Algoritma C4.5

Algoritma C4.5 adalah salah satu algoritma klasifikasi berbentuk decision tree yang merupakan pengembangan dari algoritma ID3 [5]. Algoritma ini akan membentuk decision tree menggunakan nilai *gain ratio*. Tahapan alur kerja proses dari algoritma ini dapat diilustrasikan melalui Gambar 2 [6].



Gambar 2. Diagram Alur Algoritma C4.5

Berdasarkan flowchart pada gambar 2, diawali dengan proses perhitungan nilai *entropy* di mana nilai tersebut menggambarkan distribusi probabilistik untuk mengukur tingkat homogenitas distribusi kelas pada suatu dataset. Semakin tinggi nilai *entropy* berarti bahwa nilai informasi pada dataset tersebut semakin *imputrity* (tidak pasti) [7]. Rumus perhitungan *entropy* terdapat pada persamaan (1). Setelah perhitungan nilai *entropy* selesai dilanjutkan dilakukan perhitungan nilai *gain* dan *split* untuk mengukur keefektifan suatu atribut dalam melakukan klasifikasi [8] dan mengukur informasi potensial suatu *entropy* [9]. Perhitungan *gain* dan *split info* dapat dilihat pada persamaan (2) dan (3). Terakhir yaitu proses penghitungan *gain ratio* untuk menentukan keputusan di mana *gain ratio* memiliki nilai besar jika data menyebar secara rata dan bernilai kecil jika semua data masuk ke dalam satu cabang [10]. Rumus untuk menghitung *gain ratio* dapat dilihat pada persamaan (4).

$$SplitInfo(S,A) = -\sum_{i=1}^n \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|} \quad (3)$$

$$GainRatio(S,A) = \frac{Gain(A)}{SplitEntropy(A)} \quad (4)$$

Keterangan :

A = Atribut

S = Himpunan kelas

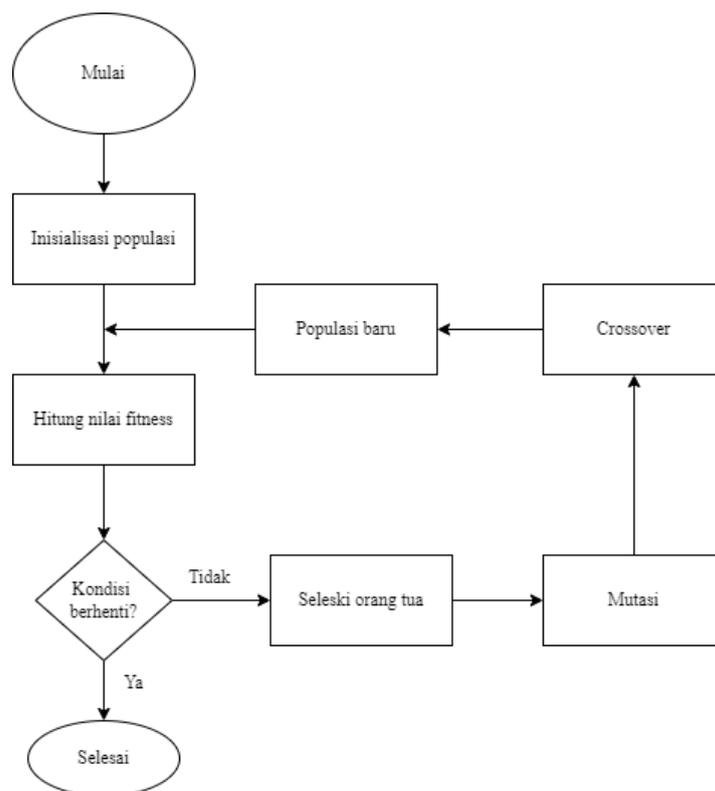
n = jumlah kategori

|S_i| = Jumlah kelas pada kategori ke i

|S| = Jumlah kelas dalam S

2.7. Pembentukan Model Algoritma C4.5 dengan Algoritma Genetika

Algoritma genetika merupakan algoritma yang diterapkan sebagai pendekatan untuk mengidentifikasi pencarian nilai dan solusi bagi berbagai permasalahan optimasi [11].



Gambar 3. Diagram Alur Algoritma Genetika

Berdasarkan diagram alur algoritma genetika pada gambar 3, algoritma genetika memiliki struktur sebagai berikut [12].

1. Inisialisasi populasi awal

Tahapan awal dalam algoritma genetika dimulai dengan inisialisasi populasi yang dibentuk dari sekumpulan N individu secara *random*. Setiap individu akan memiliki sebuah kromosom. Individu-individu ini mempresentasikan sekumpulan solusi yang diinginkan. Dalam penelitian ini, pembentukan kromosom menggunakan nilai biner yang merepresentasikan setiap atribut apakah digunakan atau tidak selama proses pelatihan. Lalu untuk penentuan jumlah individu pada populasi awal digunakan metode kombinasi faktorial dari atribut yang ada. Dari 10 atribut yang ada akan dilakukan kombinasi dengan menghilangkan 1 atribut pada setiap individu, maka jumlah kombinasi bisa didapatkan dengan rumus pada persamaan (5).

$$C_9^{10} = \frac{10!}{9!1!} = 10 \text{ Kombinasi} \quad (5)$$

2. Perhitungan nilai *fitness*

Tahap kedua dari proses algoritma genetika ada perhitungan nilai *fitness* untuk melakukan evaluasi pada suatu individu. Pada proses perhitungan nilai *fitness* setiap individu, akan memanfaatkan nilai akurasi data latih C4.5 dari model C4.5 yang terbentuk dari individu algoritma genetika menggunakan data latih algoritma genetika.

3. Seleksi orang tua

Tahap selanjutnya adalah proses seleksi orang tua yaitu memilih individu terbaik yang akan tetap berada pada populasi (*elitisme*). Pada penelitian ini, penulis menggunakan metode seleksi *Roulette-Whell* yang akan menghitung nilai probabilitas setiap individu dari nilai interval kumulatif nilai *fitness* masing masing individu dibagi dengan nilai total nilai *fitness* dari semua individu lalu memilih secara acak *parent* menggunakan bilangan random (0-1) dengan probabilitas setiap individu < bilangan random.

4. Mutasi

Mutasi merupakan proses dalam Algoritma Genetika yang bertujuan untuk mengubah gen-gen tertentu dalam sebuah kromosom. Pada penelitian ini, Semua gen yang ada didalam individu akan dilakukan proses mutasi. Proses mutasi pada penelitian ini dilakukan dengan cara membalikan bilangan biner. Mutasi setiap gen akan dilakukan jika pembangkitan bilangan random di setiap gen < Probabilitas mutasi.

5. Kondisi berhenti

Kriteria berhenti dari algoritma genetika yang dipakai yaitu menggunakan *steady state* atau telah konvergen dimana jika nilai *fitness* tidak mengalami perubahan selama 50 generasi dengan nilai *threshold* 0,001.

2.8. Confusion matrix

Confusion matrix merupakan salah satu metode yang digunakan untuk melakukan evaluasi terhadap hasil klasifikasi. Matriks tersebut dibangun berdasarkan nilai *true positive*, *true negative*, *false positive*, *false negative*. Contoh untuk *confusion matrix* dalam mengevaluasi hasil klasifikasi 2 kelas dapat dilihat pada tabel 2.

Tabel 2. Confusion Matrix

Kelas Sebenarnya	Prediksi Kelas Oleh Sistem	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

Keterangan:

TP = True Positive (total prediksi benar dari data positif)

FN = False Negative (total prediksi salah dari data positif)

TN = True Negative (total prediksi benar dari data negatif)

FP = False Positive (total prediksi salah dari data negatif)

Dari *confusion matrix* pada tabel 2, maka akan didapatkan nilai *accuracy*, *precision*, *recall*, *f-1 score* dengan rumus pada persamaan 6, 7, 8, dan 9.

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \tag{6}$$

$$Precision = \frac{(TP)}{(TP+FP)} \tag{7}$$

$$Recall = \frac{(TP)}{(TP+FN)} \tag{8}$$

$$F - 1 \text{ Score} = \frac{(2*Recall*Precision)}{(Recall+Precision)} \tag{9}$$

2.9. Desain Evaluasi

Evaluasi yang akan digunakan penulis pada penelitian ini yaitu dengan menggunakan *confusion matrix* dan membandingkan nilai *accuracy*, *precision*, *recall*, *f-1 score*, pada data testing dan jumlah *rules* yang terbentuk dari data *training* antara model *decision tree* yang telah dilakukan optimasi *rules* dan tanpa optimasi *rules* menggunakan algoritma genetika.

3. Hasil dan Pembahasan

Pada proses *preprocessing data* tahap pembersihan atribut tidak berpengaruh menggunakan perangkat lunak *SPSS* dengan metode *pearson correlation*, ditemukan bahwa atribut TP tidak memiliki korelasi yang signifikan terhadap atribut kelas sehingga atribut tersebut tidak akan digunakan selama proses pelatihan dan pengujian. Kemudian pada proses hasil kombinasi atribut sebelum dan setelah dilakukan seleksi dengan total waktu komputasi selama 3 menit 48 detik. Hasil kombinasi atribut tersebut dapat dilihat pada Tabel 3.

Tabel 3. Hasil perbandingan kombinasi atribut

	Kombinasi Atribut
Sebelum seleksi atribut	<i>Age, Gender, TB, DB, Alkphos, SGPT, SGOT, ALB, A/G Ratio</i>
Setelah seleksi atribut	<i>Age, SGOT, ALB, A/G Ratio</i>

Setelah proses seleksi selesai, dilanjutkan proses pengujian untuk *decision tree* C4.5 dengan optimasi menggunakan algoritma genetika dan tanpa optimasi menggunakan algoritma genetika. Terdapat 2 jenis pengujian, di mana pengujian yang pertama yaitu untuk melihat perbedaan jumlah (kesederhanaan) *rules decision tree* tanpa seleksi atribut dan dengan seleksi atribut pada data latih. Lalu pengujian yang kedua yaitu untuk pengukuran evaluasi *confusion matrix* dengan data uji pada *rules* yang telah dilakukan seleksi atribut dan belum dilakukan seleksi atribut. Dari hasil perbandingan seleksi atribut menggunakan algoritma genetika, maka didapatkan atribut yang terseleksi yaitu sebanyak 5 atribut, sehingga tersisa 4 atribut kombinasi terbaik.

3.1. Pengujian Jumlah (Kesederhanaan) Rules

Pengujian ini dilakukan proses perbandingan jumlah (kesederhanaan) *rules decision tree* C4.5 dengan dan tanpa optimasi menggunakan algoritma genetika.

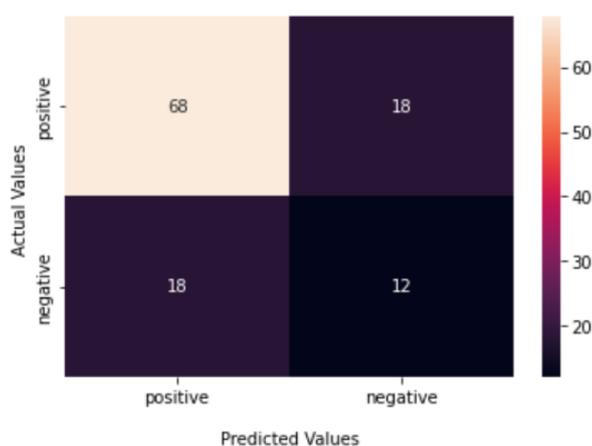
Tabel 4. Hasil perbandingan jumlah rules

	Jumlah Rules
Algoritma C4.5	145
Algoritma C4.5 + Algoritma Genetika	52

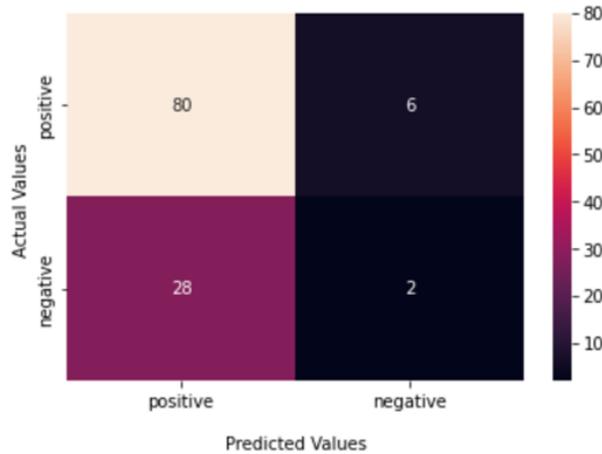
Pada Tabel 4 disajikan hasil perbandingan jumlah *rules* di mana *rules* yang dihasilkan jika menggunakan algoritma C45 tanpa optimasi yaitu sebanyak 145, sedangkan jika menggunakan algoritma C45 dengan optimasi yaitu sebanyak 52.

3.2. Pengujian Menggunakan Confusion Matrix

Pengujian ini dilakukan untuk melakukan evaluasi dengan membandingkan nilai akurasi, precision, recall, dan f1 score antara hasil klasifikasi model C4.5 dengan optimasi algoritma genetika dan model C4.5 tanpa optimasi algoritma genetika dengan data uji.



Gambar 4. Confusion matrix model C4.5



Gambar 5. *Confusion matrix* model C4.5 + Algoritma Genetika

Berdasarkan *Confusion matrix* hasil evaluasi data uji dengan model C4.5 tanpa optimasi dengan algoritma genetika pada Gambar 4 didapatkan nilai TP=68, FN=18, FP=18, TN=12 sedangkan model C4.5 tanpa optimasi algoritma genetika pada Gambar 5 didapatkan nilai TP=80, FN=6, FP=28, TN=2, sehingga didapatkan nilai *accuracy*, *precision*, *recall*, dan *f-1 score* pada Tabel 5.

Table 5. Hasil perbandingan evaluasi *confusion matrix*

	Accuracy	Precision	Recall	F-1 Score
Algoritma C4.5	68,9%	79%	79%	79%
Algoritma C4.5 + Algoritma Genetika	70,7%	74%	93%	82,4%

Berdasarkan hasil pengujian melalui *confusion matrix* pada Tabel 5, pembentukan model dengan algoritma genetika memiliki perbandingan cukup jauh di mana dengan dilakukannya seleksi atribut, mendapatkan akurasi sebesar 70,7%. Namun jika tanpa dilakukan seleksi atribut, didapatkan akurasi sebesar 68,9%.

4. Kesimpulan

Pada skenario pengujian pertama, didapatkan bahwa jumlah rules yang telah dilakukan seleksi atribut yaitu 52 rules, sedangkan jumlah rules yang tidak dilakukan seleksi atribut yaitu 145 rules. Hal ini menunjukkan bahwa dengan dilakukannya optimasi, berhasil menyederhanakan rules yang dibentuk untuk mengambil suatu keputusan terkait dengan penyakit liver.

Sedangkan pada skenario pengujian kedua, didapatkan bahwa dengan dilakukannya seleksi atribut, dapat meningkatkan akurasi secara signifikan. Pada hasil pengujian tanpa seleksi atribut, didapatkan akurasi sebesar 68,9% kemudian pada pengujian dengan seleksi atribut, didapatkan akurasi lebih tinggi, yaitu sebesar 70,7%.

References

- [1] N. Musyaffa and B. Rifai, "Model Support Vector Machine Berbasis Particle Swarm Optimization Untuk Prediksi Penyakit Liver," *JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER*, vol. 3, no. 2, pp. 189–194, 2018.
- [2] A. P. Ayudhitama and U. Pujiyanto, "Analisa 4 Algoritma dalam Klasifikasi Penyakit Liver Menggunakan Rapidminer," *JIP (Jurnal Informatika Polinema)*, vol. 6, no. 2, pp. 1–9, 2020.
- [3] N. T. Rahman, "Analisa Algoritma Decision Tree dan Naïve Bayes pada Pasien Penyakit Liver," *JURNAL FASILKOM*, vol. 10, no. 2, pp. 144–151, 2020.

- [4] D. Y. Utami, E. Nurlelah, and N. Hikmah, "Attribute Selection in Naive Bayes Algorithm Using Genetic Algorithms and Bagging for Prediction of Liver Disease," *JOURNAL OF INFORMATICS AND TELECOMMUNICATION ENGINEERING*, vol. 4, no. 1, pp. 76–85, Jul. 2020, doi: 10.31289/jite.v4i1.3793.
- [5] A. H. Nasrullah, "PENERAPAN METODE C4.5 UNTUK KLASIFIKASI MAHASISWA BERPOTENSI DROP OUT," *ILKOM Jurnal Ilmiah*, vol. 10, no. 2, pp. 244–250, 2018.
- [6] Mirqotussa'adah, M. A. Muslim, E. Sugiharti, B. Prasetyo, and S. Alimah, "Penerapan Dizcretization dan Teknik Bagging untuk Meningkatkan Akurasi Klasifikasi Berbasis Ensemble pada Algoritma C4.5 dalam Mendiagnosa Diabetes," *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, vol. 8, no. 2, pp. 135–143, Aug. 2017.
- [7] Y. S. Nugroho, "PENERAPAN ALGORITMA C4.5 UNTUK KLASIFIKASI PREDIKAT KELULUSAN MAHASISWA FAKULTAS KOMUNIKASI DAN INFORMATIKA UNIVERSITAS MUHAMMADIYAH SURAKARTA," 2014.
- [8] I. Handayani, T. Yogyakarta, J. Siliwangi, R. Utara,) Jombor, and Y. Sleman, "PENERAPAN ALGORITMA C4.5 UNTUK KLASIFIKASI PENYAKIT DISK HERNIA DAN SPONDYLOLISTHESIS DALAM KOLUMNA VERTEBRALIS," *JASIEK*, vol. 1, no. 2, 2019, doi: 10.12928/JASIEK.v13i2.xxxx.
- [9] H. Widayu, S. D. Nasution, N. Silalahi, and Mesran, "DATA MINING UNTUK MEMPREDIKSI JENIS TRANSAKSI NASABAH PADA KOPERASI SIMPAN PINJAM DENGAN ALGORITMA C4.5," *Media Informatika Budidarma*, vol. 1, no. 2, pp. 32–37, 2017.
- [10] A. Asroni, B. Masajeng Respati, and S. Riyadi, "Penerapan Algoritma C4.5 untuk Klasifikasi Jenis Pekerjaan Alumni di Universitas Muhammadiyah Yogyakarta," *Semesta Teknika*, vol. 21, no. 2, 2018, doi: 10.18196/st.212222.
- [11] H. Harafani, T. Informatika, S. Nusa, and M. Jakarta, "OPTIMASI ALGORITMA GENETIKA PADA K-NN UNTUK MEMPREDIKSI KECENDERUNGAN 'BLOG POSTING,'" *Jurnal Pendidikan Teknologi dan Kejuruan*, vol. 15, no. 1, p. 20, 2018, [Online]. Available: <https://ejournal.undiksha.ac.id/index.php/JPTK/issue/view/780>
- [12] N. Wahyudi, S. Wahyuningsih, F. Deny, and T. Amijaya, "Optimasi Klasifikasi Batubara Berdasarkan Jenis Kalori dengan menggunakan Genetic Modified K-Nearest Neighbor (GMK-NN) (Studi Kasus: PT Jasa Mutu Mineral Indonesia Samarinda, Kalimantan Timur) Optimization of Coal Classification Based on Calorie using Genetic Modified K-Nearest Neighbor (GMK-NN) (Case Study: PT Jasa Mutu Mineral Indonesia Samarinda, Kalimantan Timur)," *Jurnal EKSPONENSIAL*, vol. 10, no. 2, 2019.

This page is intentionally left blank.