

PEMBENTUKAN POHON KEPUTUSAN UNTUK PENERIMA BANTUAN BERAS MISKIN MENGGUNAKAN ALGORITMA DECISION TREE C4.5

DECISION TREE FORMATION FOR SUBSIDIZED RICE AID RECIPIENTS USING THE C4.5 ALGORITHM

Donny Avianto¹⁾, Adityo Permana Wibowo²⁾

¹⁾Program Studi Teknik Informatika, Fakultas Sains & Teknologi, Universitas Teknologi Yogyakarta

²⁾Program Studi Sistem Informasi, Fakultas Sains & Teknologi, Universitas Teknologi Yogyakarta

Jl Siliwangi (Ring Road Utara), Jombor, Mlati, Sleman, DIY

Email : donny@uty.ac.id¹⁾, adityopw@uty.ac.id²⁾

Abstrak

Ketahanan pangan menjadi salah satu hal yang paling penting dalam keberlangsungan suatu negara. Untuk mendukung ketahanan pangan, pemerintah mengeluarkan satu program bertujuan untuk memberikan bantuan pangan pokok yaitu beras kepada masyarakat kurang mampu dengan nama beras miskin (Raskin). Namun, tantangan besar dalam implementasi program ini adalah ketidaktepatan sasaran, di mana terdapat kasus di mana warga yang seharusnya menerima bantuan malah tidak mendapatkannya, sementara sebagian yang tidak memenuhi syarat justru menerima bantuan. Penelitian ini bertujuan menghasilkan model pohon keputusan yang dapat membantu proses klasifikasi penerima bantuan beras miskin secara lebih mudah dan akurat, sehingga penyaluran program Raskin menjadi lebih tepat sasaran. Pembuatan model dilakukan menggunakan aplikasi RapidMiner Studio versi 10.3 dan algoritma C4.5 sebagai algoritma pembentukan pohon keputusannya. Dalam menentukan kelayakan penerima, aplikasi menggunakan tujuh kriteria utama: tingkat kesejahteraan, jumlah tanggungan, jenis pekerjaan, sarana sanitasi, sumber air, jenis atap, dan jenis lantai. Algoritma C4.5 pada penelitian ini menggunakan pembagian data sebesar 60% data latih dan 40% data uji dari total 120 data yang ada. Hasil eksperimen menunjukkan, model yang dibuat memiliki akurasi mencapai 79,17%. Adapun faktor yang paling menentukan dalam prediksi penerima Raskin adalah jenis lantai. Penelitian ini juga memvisualisasikan pohon keputusan yang terbentuk secara utuh untuk memudahkan interpretasi hasil prediksi dan peluang peningkatan di masa depan.

Kata kunci: Decision Tree C4.5, Klasifikasi Penerima Bantuan Beras Miskin, Raskin

Abstract

Food security is one of the most important things in the sustainability of a country. To support food security, the government issued a program aimed at providing staple food assistance, namely rice, to the underprivileged under the name Rice for the Poor (Raskin). However, a major challenge in the implementation of this program is the inaccuracy of targeting, where there are cases where people who should receive assistance do not get it, while some who are not eligible actually receive assistance. This research aims to produce a decision tree model that can help the classification of rice for the poor beneficiaries more easily and accurately, so that the distribution of the Raskin program becomes more targeted. Modeling is done using the RapidMiner Studio version 10.3 application and the C4.5 algorithm as the decision tree formation algorithm. In determining recipient eligibility, the application uses seven main criteria: welfare level, number of dependents, type of employment, sanitation facilities, water source, roof type, and floor type. The C4.5 algorithm in this research uses a data division of 60% training data and 40% test data from a total of 120 data. The experimental results show that the model created has an accuracy of 79.17%. The most determining factor in predicting Raskin recipients is the type of floor. This research also visualizes the decision tree formed as a whole to facilitate the interpretation of prediction results and opportunities for improvement in the future.

Keywords: Decision Tree C4.5, Classification of Rice for The Poor Beneficiaries, Raskin.

1. PENDAHULUAN

Program Raskin adalah program pemberian subsidi terhadap bahan pangan utama yang dicanangkan oleh pemerintah. Tujuan pelaksanaan program ini adalah untuk membangun ketahanan pangan sekaligus memberi perlindungan kepada keluarga kurang mampu dengan membagikan beras kepada mereka [1]. Subsidi ini terus dilaksanakan sesuai dengan anggaran

yang dialokasikan oleh pemerintah setiap tahunnya. Keberhasilan program Raskin ini dapat diukur dengan tercapainya 6T yaitu tepat sasaran penerima manfaat, tepat waktu, tepat jumlah, tepat harga, tepat administrasi dan tepat kualitas [2, 3].

Namun, dalam pelaksanaannya, program ini sering kali menemui kendala yang menghambat penyaluran bantuan secara optimal. Pihak kelurahan memiliki tanggung jawab besar dalam memastikan bahwa bantuan Raskin disalurkan kepada penerima manfaat yang sesuai. Salah satu tantangan utama adalah ketidaktepatan sasaran penerima bantuan, di mana warga yang memenuhi kriteria tidak mendapatkan bantuan, sementara yang tidak memenuhi syarat justru mendapatkannya. Tantangan ini sering kali timbul dari kurangnya dukungan sistem berbasis terutama dalam hal pengelolaan stok Raskin yang akan dibagikan dan ketidaksesuaian identitas tempat tinggal penerima dengan lokasi mendapatkannya [4].

Kriteria penerima bantuan seharusnya diputuskan berdasarkan data objektif, seperti tingkat kesejahteraan, jumlah tanggungan, jenis pekerjaan, dan akses terhadap fasilitas dasar seperti sanitasi dan sumber air. Akan tetapi, data ini sering kali tidak diolah secara efektif sehingga menyebabkan ketidaktepatan sasaran. Selain itu, kesadaran masyarakat mengenai hak atas bantuan ini juga rendah. Banyak penerima yang tidak benar-benar membutuhkan, tetapi tetap menerima bantuan, bahkan ada yang menjual beras bantuan tersebut karena alasan kualitas atau untuk memenuhi kebutuhan lainnya [5, 6].

Berdasarkan latar belakang ini, dikembangkan sebuah sistem yang menggunakan metode Decision Tree sebagai alat bantu dalam mengklasifikasikan penerima yang layak mendapatkan bantuan Raskin. Beberapa penelitian sebelumnya telah menunjukkan performa yang luar biasa dari metode *decision tree*, khususnya kasus-kasus yang berkaitan dengan klasifikasi kelayakan, seperti penelitian tentang klasifikasi kelayakan penerima kredit [7], kelayakan penerima beasiswa [8], hingga kelayakan penerima bantuan jaminan kesehatan [9]. Beberapa penelitian yang sudah dilakukan menunjukkan kelebihan dari metode *decision tree* terutama dalam hal interpretasi yang mudah dan fleksibilitasnya dalam pemodelan data. Algoritma *decision tree* juga efektif sebagai metode awal maupun model akhir dalam pemodelan data karena mampu memberikan hasil yang jelas dan terstruktur. Penelitian ini berkontribusi dalam menghasilkan model *decision tree* berbasis algoritma C4.5 untuk menentukan kelayakan penerima beras miskin dengan menguji pengaruh berbagai parameter, seperti proporsi *split data*, kedalaman maksimal pohon, dan *minimum gain*. Evaluasi model pada penelitian ini dilakukan menggunakan metrik akurasi, *precision*, dan *recall*, sehingga mampu memberikan wawasan terkait performa model berdasarkan konfigurasi parameter yang berbeda-beda.

2. DASAR TEORI

2.1 Decision Tree

Decision tree adalah algoritma *machine learning* yang menggunakan seperangkat aturan untuk membuat keputusan dengan struktur seperti pohon yang memodelkan kemungkinan hasil, biaya sumber daya, utilitas dan kemungkinan konsekuensi atau risiko [10]. Menurut [11], terdapat 3 jenis *node* pada *Decision tree* yaitu:

- a) *Root Node* (Simpul Akar): *Node* awal yang mewakili keseluruhan dataset dan memecah data sesuai dengan kondisi yang paling informatif [12].
- b) *Internal Nodes* (Simpul Internal): Titik percabangan yang menguji atribut tertentu dalam dataset, di mana setiap cabang mencerminkan hasil pengujian tersebut.
- c) *Leaf Nodes* (Simpul Daun): *Node* akhir yang menunjukkan keputusan atau hasil akhir dari jalur yang dilalui. Setiap jalur dari akar hingga daun merupakan aturan klasifikasi atau prediksi.

Node teratas dari pohon keputusan biasa disebut dengan istilah *node* akar. *Node* akar sendiri biasanya merupakan atribut yang mempunyai pengaruh paling besar pada penentuan kelas [13, 14]. Umumnya, pohon keputusan melakukan strategi pencarian solusi *top-down* [15], yang berarti pada saat data baru akan diklasifikasikan, proses penentuan kelasnya akan dilakukan

dengan cara menelusuri jalur mulai dari *node* akar ke *node* daun (*leaf*) dan mencari nilai entropi dan *gain* untuk masing-masing atribut yang dikunjungi [16].

2.2 Algoritma C4.5

Algoritma C4.5 dikembangkan oleh Ross Quinlan yang berfokus untuk membuat pohon keputusan [17]. Algoritma ini mencakup masukan berupa sampel pelatihan dan sampel. Sampel pelatihan adalah sampel data yang digunakan untuk membangun pohon yang diuji keakuratannya. Sebaliknya, sampel adalah bagian dari data yang kemudian digunakan sebagai parameter saat mengklasifikasikan data [18]. Pemilihan suatu atribut sebagai *node*, baik *node* akar maupun *node* internal, didasarkan pada nilai *gain* tertinggi dari atribut-atribut yang ada. Nilai *gain* dihitung secara bertahap berdasarkan nilai entropi dari masing-masing atribut. Adapun rumus untuk menentukan nilai entropi ditunjukkan pada Persamaan (1).

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

Dimana S adalah ruang (data) sampel yang digunakan untuk pelatihan, p_i adalah proporsi dari S_i terhadap S , dan n adalah jumlah partisi S . Entropi dihitung digunakan untuk mencari *information gain*, pada persamaan dua digunakan untuk mencari *information gain*.

Setelah mendapatkan nilai entropi, perhitungan dilanjutkan dengan mencari nilai *information gain*. Adapun rumus untuk mencari nilai *information gain* dapat dilihat pada Persamaan (2) berikut.

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

Dimana A adalah atribut, $|S_i|$ adalah jumlah sampel untuk nilai i , $|S|$ adalah jumlah seluruh sampel, dan $Entropy(S_i)$ adalah entropi untuk sampel-sampel yang memiliki nilai i . Setelah *information gain* dihitung menggunakan persamaan kedua, maka langkah selanjutnya adalah menghitung *split info*.

Split info adalah tahap ketiga yang digunakan pada algoritma C4.5, dan dihitung menggunakan formula yang dapat dilihat pada Persamaan (3).

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

Atribut dengan nilai *gain ratio* tertinggi dipilih sebagai atribut tes untuk simpul. Pendekatan ini merupakan langkah normalisasi pada *information gain* dengan menggunakan apa yang disebut dengan *split information*. Terdapat alasan mengapa C4.5 menggunakan nilai *gain ratio* dan bukan menggunakan nilai *gain* saja untuk pemilihan atributnya. Setelah diteliti nilai *gain* ternyata bersifat bias terhadap atribut yang memiliki nilai unik. Adapun persamaan untuk menghitung nilai *information gain* dapat dilihat pada Persamaan (4).

$$Gain Ratio (A) = \frac{Gain Information (A)}{Split Info (A)} \quad \dots(4)$$

Decision tree tidak menggunakan vektor jarak dalam mengklasifikasikan data. Observasi data sering kali memiliki atribut dengan nilai nominal. Sebagai contoh, jika objeknya adalah sekumpulan buah-buahan, buah tersebut dapat dibedakan berdasarkan atribut seperti bentuk, warna, ukuran, dan rasa. Atribut-atribut tersebut merupakan nilai nominal atau kategoris, di mana setiap nilainya tidak dapat dijumlahkan atau dikurangkan. Misalnya, atribut warna dapat memiliki nilai hijau, kuning, dan merah, sedangkan atribut ukuran mungkin terdiri dari nilai besar, sedang, dan kecil. Berdasarkan nilai-nilai ini, *decision tree* dapat dibentuk untuk menentukan jenis buah dari suatu objek dengan mempertimbangkan kriteria-kriteria tersebut.

2.3 Metode Pengukuran Performa Sistem

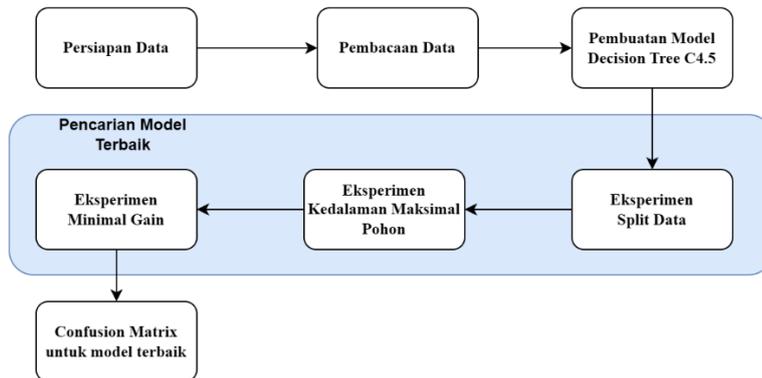
Penelitian ini menggunakan 2 metrik pengukuran performa sistem yaitu skor akurasi dan *confusion matrix*. Skor akurasi pada penelitian ini digunakan untuk memberikan gambaran seberapa besar jumlah output yang berhasil diklasifikasikan secara benar dari total seluruh data penelitian yang ada [19]. Adapun persamaan untuk mencari nilai akurasi pada penelitian ini seperti yang tersaji pada Persamaan (5).

$$Akurasi = \frac{\text{Jumlah Data yang Diklasifikasikan dengan Benar}}{\text{Jumlah Seluruh Data}} \dots\dots(5)$$

Setelah mendapatkan skor akurasi untuk setiap model *decision tree* yang diujikan, model terbaik dipilih untuk kemudian dilakukan pengukuran performa secara lebih mendalam. Pengukuran performa yang kedua ini dilakukan menggunakan *confusion matrix* dengan tujuan melihat kualitas hasil klasifikasi yang dimiliki oleh model terbaik [20]. Selain mengetahui skor akurasi, penggunaan *confusion matrix* juga dapat memberikan informasi tambahan berupa skor *precision* dan *recall* yang sangat penting dalam pengukuran performa suatu model [21].

3. METODOLOGI PENELITIAN

Pembuatan pohon keputusan untuk memprediksi penerima program Raskin pada penelitian ini dilakukan dalam beberapa tahapan seperti yang tampak pada Gambar 1 berikut.



Gambar 1. Langkah-langkah Penelitian Sistem Prediksi Penerima Raskin

Langkah pertama yang dilakukan adalah mempersiapkan data. Data yang digunakan pada penelitian ini diambil dari salah satu kabupaten di provinsi Jawa Tengah. Data yang digunakan memiliki total 120 *record* yang mewakili 120 kondisi keluarga di wilayah tersebut, dimana masing-masing data memiliki 7 fitur yang sekaligus menjadi faktor-faktor yang menentukan kelayakan penerima Raskin. Adapun ketujuh fitur tersebut dapat dilihat pada Tabel 1 berikut.

Tabel 1. Faktor Penentu Kelayakan Penerima Raskin

No	Nama Faktor	Nilai
1	Pekerjaan	Wiraswasta; Petani; Transportasi; Pedagang; ART; Jasa; Buruh
2	Tingkat Kesejahteraan	Miskin; Sedang; Kaya
3	Sarana Sanitasi	WC leher angsa; Kakus/cemplung; Numpang
4	Sumber Air	Sumur pompa; Sumur timba; Keran umum; PDAM; Sumber galian pribadi
5	Jenis Atap	Genteng; Genteng lama
6	Jenis Dinding	Batu; Bambu
7	Jenis Lantai	Keramik; Semen; Tanah

Masing-masing data juga sudah dilengkapi dengan target atau nilai kebenaran referensi yang menunjukkan apakah data tersebut menjadi penerima program Raskin atau tidak di dalam kenyataannya. Setelah data dipersiapkan, data disimpan dalam format CSV untuk kemudian masuk ke tahapan berikutnya yaitu pembacaan data. Pada tahap pembacaan data ini terdapat beberapa proses yang dilakukan antara lain membaca data dari *file* CSV, mendeteksi struktur data untuk mengetahui *header* data dan pemisah kolom yang digunakan. Setelah struktur data terbaca, proses penyesuaian tipe data dilakukan untuk memastikan proses klasifikasi dapat berjalan secara optimal. Terakhir proses penentuan kolom target dan kolom fitur dilakukan dan dataset sudah siap digunakan untuk tahapan selanjutnya.

Setelah itu, algoritma *decision tree* C4.5 mulai diimplementasikan menggunakan *tool* RapidMiner Studio versi 10.3. Hasil pembacaan data kemudian disambungkan sebagai input untuk algoritma C4.5 dan memastikan tidak ada kesalahan selama proses pembentukan pohon keputusan berlangsung. Proses ini akan menghasilkan pohon keputusan yang menjadi model *baseline* yang pertama. Model *baseline* yang pertama ini kemudian dioptimalkan secara bertahap melalui beberapa eksperimen untuk mendapatkan model terbaik pada penelitian ini.

Adapun eksperimen pertama yang dilakukan dalam rangka menemukan model terbaik pada penelitian ini adalah mencari nilai *split* data terbaik. *Split* data akan menentukan berapa jumlah data latih dan data uji yang akan digunakan untuk membuat pohon keputusan. Kemudian eksperimen yang kedua berfokus pada pencarian nilai kedalaman pohon maksimal. Adapun eksperimen terakhir adalah pencarian nilai *minimal gain* untuk menghasilkan model terbaik dalam memprediksi kelayakan penerima program Raskin. Setelah model terbaik dihasilkan dari tiga eksperimen tersebut, kualitas hasil prediksi dari model terbaik akan dilihat secara lebih mendalam dengan mempelajari *confusion matrix*-nya.

4. PENGUJIAN DAN PEMBAHASAN

Pengujian pada penelitian ini dimulai dengan membagi 120 data yang dimiliki menjadi dua *subset* yaitu data latih dan data uji. Adapun proporsi pembagian data latih dan data uji yang dicoba pertama kali adalah 50:50 sehingga masing-masing *subset* berisi 60 data. Pembagian ini dilakukan secara *stratified* untuk menjamin proporsi jumlah kelas pada data latih dan data uji tetap sama. Setelah data dibagi menjadi dua *subset*, data latih digunakan untuk membentuk pohon keputusan menggunakan algoritma *decision tree* C4.5 yang kemudian dijadikan sebagai *baseline model* pada penelitian ini. Adapun parameter *decision tree* lain yang digunakan pada *baseline model* antara lain adalah: maksimal kedalaman pohon = 10, *apply prepruning* = TRUE, minimal gain = 0,01, minimal jumlah daun = 2. Adapun pohon keputusan yang dihasilkan oleh *baseline model* memiliki tingkat akurasi 66,67%.

Setelah mendapatkan hasil dari *baseline model*, penelitian berlanjut dengan mengelaborasi proporsi *split* data antara *subset* data latih dan data uji. Selain proporsi 50:50, penelitian ini juga mencoba proporsi *split* data lain seperti 60:40, 70:30, 80:20, dan 90:10. Adapun pengaruh perubahan *split* data terhadap akurasi dapat dilihat pada Tabel 2.

Tabel 2. Pengaruh Nilai Split Data Terhadap Akurasi

No	Split Data	Jumlah Data Latih	Jumlah Data Uji	Akurasi
1	50:50	60	60	61,67%
2	60:40	72	48	72,92%
3	70:30	84	36	69,44%
4	80:20	96	24	62,50%
5	90:10	108	12	66,67%

Berdasarkan Tabel 2, proporsi *split* data 60:40 menjadi proporsi yang mampu menghasilkan nilai akurasi tertinggi yaitu 72,92% atau meningkat 11,25% dari versi *baseline model* yang menggunakan *split* data 50:50. Sedangkan untuk *split* data lainnya, terjadi penurunan akurasi antara 3,48% hingga 10,42% dibandingkan versi terbaik. Fenomena ini dapat disebabkan karena pada *split* 50:50, jumlah data latih lebih sedikit dibandingkan 60:40, sehingga model tidak memiliki cukup data untuk mempelajari pola data tersebut. Sedangkan pada *split* dengan proporsi data latih yang lebih besar dari 60:40, seperti 70:30, 80:20, atau 90:10, model memiliki lebih banyak data latih tetapi kehilangan jumlah data uji yang memadai, sehingga evaluasi performa menjadi kurang representatif atau cenderung *overfitting* pada data latih. Pada percobaan selanjutnya, *split* data 60:40 akan digunakan untuk pencarian nilai akurasi tertinggi.

Parameter selanjutnya yang diuji coba pada penelitian ini adalah maksimal kedalaman pohon. Rentang nilai maksimal kedalaman pohon yang digunakan pada percobaan kali ini adalah

2 hingga 10. Adapun hasil eksperimen untuk maksimal kedalaman pohon terbaik dapat dilihat pada Tabel 3.

Tabel 3. Perbandingan Nilai Maksimal Kedalaman Pohon Terhadap Akurasi

No	Maksimal Kedalaman Pohon	Akurasi
1	2	77,08%
2	3	79,17%
3	4	77,08%
4	5	77,08%
5	6	72,92%
6	7	72,92%
7	8	72,92%
8	9	72,92%
9	10 (<i>baseline</i>)	72,92%

Berdasarkan data pada Tabel 3, diketahui bahwa nilai maksimal kedalaman pohon = 10 yang digunakan pada *baseline model* memiliki tingkat kerumitan *decision tree* yang cukup tinggi sehingga mendapatkan hasil akurasi 72,92%. Menariknya ketika nilai maksimal kedalaman pohon dibuat lebih kecil, yang berarti *decision tree*-nya dibuat lebih sederhana, model mampu memberikan hasil akurasi yang lebih baik. Hal ini dapat dilihat saat nilai maksimal kedalaman pohon = 5 yang menghasilkan peningkatan akurasi sebesar 4,16% menjadi 77,08%. Adapun akurasi tertinggi dari uji coba kali ini didapatkan saat nilai maksimal kedalaman pohon = 3 dengan akurasi mencapai 79,17%. Berdasarkan uji coba ini, dapat diketahui bahwa nilai maksimal kedalaman pohon sangat berpengaruh terhadap akurasi model. Tingkat kerumitan suatu *decision tree* harus dibuat sedemikian rupa sehingga tidak terlalu kompleks tetapi juga tidak terlalu sederhana, untuk menghasilkan performa prediksi yang terbaik. Nilai maksimal kedalaman pohon = 3, sebagai nilai yang menghasilkan performa akurasi terbaik, kemudian akan digunakan pada uji coba selanjutnya.

Pada pengujian selanjutnya, nilai parameter *minimal gain* akan dielaborasi untuk melihat pengaruhnya terhadap akurasi. Seperti yang sudah diketahui, di dalam suatu *decision tree*, parameter *minimal gain* berfungsi untuk mengatur apakah suatu *node* perlu di-*split* atau tidak. Proses *split* sendiri adalah proses di dalam pembentukan pohon keputusan, di mana suatu *node* perlu di pecah menjadi cabang yang lebih kecil dan lebih homogen sehingga dapat menghasilkan performa yang lebih baik. Semakin besar nilai parameter *minimal gain*, maka akan semakin sedikit proses *split* selama pembentukan pohon keputusan dan pohon menjadi lebih sederhana. Sedangkan jika nilai parameter *minimal gain* semakin kecil maka proses *split* akan semakin sering dilakukan selama pembentukan pohon keputusan dan berdampak pada struktur pohon yang menjadi lebih rumit. Dengan kata lain, nilai *minimal gain* akan sangat mempengaruhi kedalaman pohon dimana nilai *minimal gain* yang besar akan menghasilkan pohon yang tidak terlalu dalam dan nilai *minimal gain* yang kecil akan menghasilkan pohon yang dalam. Uji coba ini menggunakan 10 nilai parameter *minimal gain* yang berada pada rentang 0 hingga 1. Hasil uji coba dari parameter *minimal gain* dapat dilihat pada Tabel 4 di bawah ini.

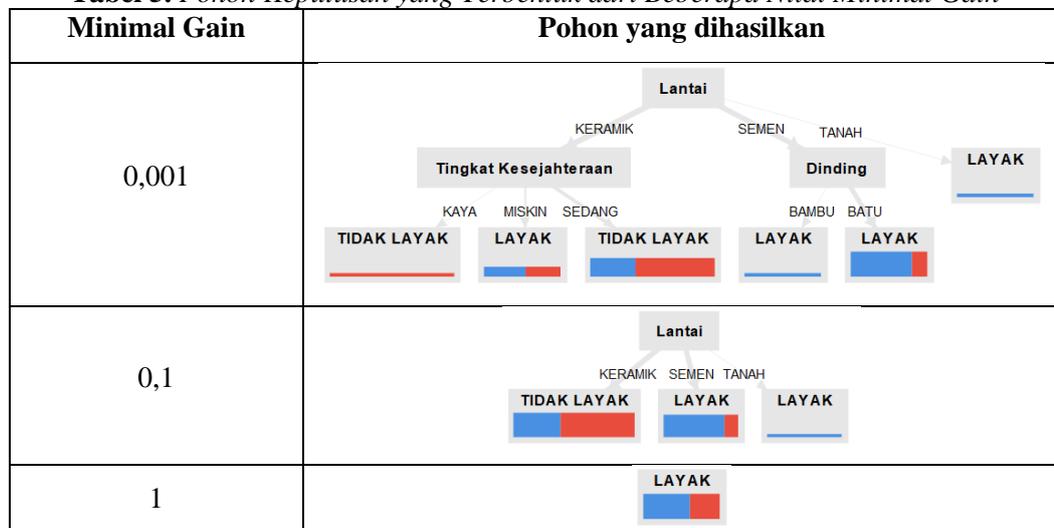
Tabel 4. Perbandingan Nilai Minimal Gain Terhadap Akurasi

No	Minimal Gain	Akurasi
1	0	79,17%
2	0,001	79,17%
3	0,005	79,17%
4	0,01	79,17%
5	0,05	79,17%

6	0,1	77,08%
7	0,5	60,42%
8	0,7	60,42%
9	1	60,42%

Berdasarkan hasil uji coba pada Tabel 4, diketahui bahwa nilai *minimal gain* yang menghasilkan skor akurasi terbaik berada pada rentang 0 hingga 0,05. Sedangkan mulai nilai *minimal gain* = 0,1 hingga 1, akurasi terus mengalami penurunan. Dengan nilai maksimal kedalaman pohon yang sama yaitu 3, hasil ini menjadi masuk akal karena, saat nilai *minimal gain* besar, tidak banyak terjadi *split* selama pembentukan pohon dan menyebabkan pohon menjadi terlalu sederhana. Hal inilah yang menyebabkan penurunan performa model dalam memprediksi data. Tabel 5 menunjukkan gambar pohon yang terbentuk pada nilai *minimal gain* yang berbeda-beda.

Tabel 5. Pohon Keputusan yang Terbentuk dari Beberapa Nilai Minimal Gain



Berdasarkan hasil uji coba ini, nilai *minimal gain* = 0,05 dipilih sebagai nilai *minimal gain* terbaik. Nilai 0,05 dipilih karena tidak terlalu kecil, sehingga proses *split* tidak perlu terlalu sering dilakukan tetapi juga masih memberikan performa akurasi terbaik yaitu 79,17%. Adapun bentuk pohon keputusan yang dibentuk pada saat nilai *minimal gain* = 0,05 sama persis seperti pohon yang terbentuk saat nilai *minimal gain* = 0,001 pada Tabel 5 di atas.

Pohon keputusan yang terbentuk juga memberikan informasi pola prediksi penerima Raskin yang digunakan pada penelitian ini. Atribut lantai menjadi atribut yang paling menentukan apakah seseorang layak menerima Raskin atau tidak. Hal ini dapat dilihat dari pohon keputusan yang memiliki *root node* yaitu lantai. Adapun pola yang terbentuk adalah jika atribut lantai bernilai “Tanah” maka data tersebut akan masuk ke dalam golongan yang layak mendapatkan Raskin. Sedangkan jika nilai atribut lantai bernilai “Semen” maka penelusuran akan berlanjut ke atribut dinding dan jika nilai atribut lantai bernilai “Keramik” maka penelusuran akan berlanjut ke atribut tingkat kesejahteraan. Pola penelusuran ini juga sangat sesuai dengan kondisi nyata dimana saat lantai rumah seseorang sudah keramik tetapi masih masuk ke dalam daftar penerima Raskin, maka perlu dilihat kembali tingkat kesejahteraannya. Penelitian ini sekaligus menguatkan keunggulan dari metode *decision tree* C4.5 yang mampu menghasilkan pohon keputusan dengan struktur sederhana dan jelas, sehingga memungkinkan pengguna memahami alur pengambilan keputusan dan mudah menginterpretasikan hasilnya. Hal ini menjadi poin yang sangat bermanfaat, terutama di dalam konteks yang membutuhkan transparansi atau interpretasi yang tinggi.

Setelah memahami pola dari pohon keputusan yang terbentuk, penelitian ini melakukan peninjauan yang lebih dalam untuk melihat kualitas prediksi yang dihasilkan oleh model dengan

akurasi tertinggi yaitu 79,17%. Kualitas prediksi model ditinjau lebih dalam menggunakan *confusion matrix*. Adapun *confusion matrix* untuk model terbaik dapat dilihat pada Tabel 6.

Tabel 6. *Confusion Matrix Hasil Prediksi Model Terbaik*

	True LAYAK	True TIDAK LAYAK	Precision
Pred. LAYAK	28	9	75,68%
Pred. TIDAK LAYAK	1	10	90,91%
Recall	96,55%	52,63%	

Berdasarkan hasil *confusion matrix*, diketahui bahwa model yang dihasilkan mampu melakukan prediksi dengan cukup baik. Kesalahan prediksi untuk kasus data yang seharusnya layak menerima tetapi diprediksi sebagai tidak layak menerima hanya terjadi pada 1 data. Sedangkan kesalahan prediksi jenis sebaliknya, yaitu sebenarnya tidak layak tetapi diprediksi sebagai layak terjadi pada 9 data. Dengan kata lain, model yang dihasilkan lebih sensitif saat menentukan data ke dalam kelas tidak layak. Adapun kesalahan prediksi untuk kelas layak yang terjadi pada 9 data, kemungkinan terjadi karena adanya faktor-faktor teknis, seperti kuota penerima Raskin untuk daerah tersebut yang sudah terpakai semuanya. Kemungkinan tersebut muncul, karena jika 9 data tersebut dicermati, seharusnya 9 data tersebut berhak menerima Raskin. Adapun 9 data yang salah diprediksi sebagai layak menerima Raskin dapat dilihat pada Tabel 7.

Tabel 7. *Data Misklasifikasi Layak seharusnya Tidak Layak*

Data ke-	Prediksi	Label	Pekerjaan	Tingkat Kesejahteraan	Sanitasi	Sumber Air	Atap	Dinding	Lantai
1	Layak	Tidak Layak	PETANI	MISKIN	KAKUS/CEMPLUNG	KERAN UMUM	GENTENG LAMA	BATU	SEMEN
2	Layak	Tidak Layak	PETANI	MISKIN	KAKUS/CEMPLUNG	KERAN UMUM	GENTENG	BATU	KERAMIK
3	Layak	Tidak Layak	WIRASWASTA	SEDANG	WC LEHER ANGSA	KERAN UMUM	GENTENG LAMA	BATU	SEMEN
4	Layak	Tidak Layak	TRANSPORTASI	MISKIN	WC LEHER ANGSA	KERAN UMUM	GENTENG	BATU	SEMEN
5	Layak	Tidak Layak	PETANI	MISKIN	WC LEHER ANGSA	KERAN UMUM	GENTENG	BATU	KERAMIK
6	Layak	Tidak Layak	JASA	MISKIN	WC LEHER ANGSA	PDAM	GENTENG	BATU	KERAMIK
7	Layak	Tidak Layak	PETANI	SEDANG	WC LEHER ANGSA	KERAN UMUM	GENTENG LAMA	BATU	SEMEN
8	Layak	Tidak Layak	PETANI	SEDANG	WC LEHER ANGSA	KERAN UMUM	GENTENG	BATU	SEMEN
9	Layak	Tidak Layak	PETANI	MISKIN	KAKUS/CEMPLUNG	SUMUR POMPA	GENTENG LAMA	BATU	SEMEN

Tabel 7 di atas menunjukkan bahwa kesembilan data tersebut berasal dari keluarga miskin dan sedang. Hal ini menjadi salah satu syarat untuk mendapatkan beras Raskin. Selain itu, 7 dari 9 data masih menggunakan keran umum sebagai sumber air. Adapun yang menggunakan PDAM dan sumur pompa berprofesi sebagai petani dan bidang jasa dari golongan tingkat kesejahteraan miskin. Berdasarkan hal inilah sangat besar kemungkinan 9 data ini memiliki label “Tidak Layak” karena adanya faktor lain yang tidak tertangkap di data.

5. KESIMPULAN

Penelitian ini berfokus pada pencarian model terbaik untuk memprediksi kelayakan seseorang dalam menerima bantuan Raskin menggunakan metode *decision tree* C4.5. Pencarian model terbaik dilakukan melalui serangkaian eksperimen dengan mengelaborasi nilai

parameter seperti *split* data, kedalaman maksimal pohon, dan *minimal gain* untuk memperoleh model dengan akurasi tertinggi. Dari hasil pengujian, sistem mampu mencapai tingkat akurasi tertinggi sebesar 79,17% pada perbandingan data latih banding data uji adalah 60:40. Adapun faktor yang paling menentukan dalam menentukan kelayakan penerima Raskin pada penelitian ini adalah kondisi lantai tempat tinggal. Pohon keputusan yang dihasilkan dari penelitian ini menunjukkan bahwa metode yang digunakan cukup efektif dalam memprediksi kelayakan penerima beras miskin (Raskin). Namun, peningkatan akurasi masih diperlukan agar sistem dapat digunakan secara lebih luas dan andal dalam konteks yang lebih beragam. Beberapa hal yang dapat dilakukan untuk meningkatkan performa metode *decision tree* C4.5 sekaligus menjadi fokus pada penelitian selanjutnya. Hal pertama yang akan dilakukan adalah menggunakan teknik *ensemble learning*, seperti *bagging* atau *boosting* untuk meningkatkan akurasi dengan menggabungkan prediksi dari beberapa model. Selain itu, metode *pruning* juga dapat diterapkan untuk mengurangi *overfitting*, sehingga kemampuan generalisasi model menjadi lebih baik dan mampu memprediksi data baru secara lebih akurat.

DAFTAR PUSTAKA

- [1] S. Sarmila, U. B. Sore, and N. Tompo, "Implementasi Distribusi Bantuan Sosial (Raskin) Bagi Masyarakat Miskin Di Kelurahan Bontoala Parang Kecamatan Bontoala Kota Makassar," *Publician: Journal of Public Service, Public Policy, and Administrastion*, vol. 1, no. 1, pp. 14-21, 2022, doi: <https://doi.org/10.56326/jp.v1i1.1371>.
- [2] D. E. Pratama, S. M. Lamata, and D. H. Zahro, "Program Raskin Sebagai Dakwah Pemberdayaan Masyarakat Proletar di Indonesia," *Dakwatun: Jurnal Manajemen Dakwah*, vol. 1, no. 2, pp. 106-117, 2022, doi: <https://doi.org/10.58194/jdmd.v1i2.121>.
- [3] B. Rachman and A. Agustian, "Efektivitas dan perspektif pelaksanaan program beras sejahtera (Rastra) dan bantuan pangan non-tunai (BPNT)," *Analisis Kebijakan Pertanian*, vol. 16, no. 1, pp. 1-18, 2018, doi: <http://dx.doi.org/10.21082/akp.v16n1.2018.1-18>.
- [4] Y. Komalasari, D. Firmansyah, R. R. Agatha, and D. Wijayanti, "Perancangan Sistem Informasi Penyaluran Beras (Raskin) Pada Desa Balonggandu Jatisari," *Jurnal Teknologi Dan Open Source*, vol. 3, no. 1, pp. 114-130, 2020, doi: <https://doi.org/10.36378/jtos.v3i1.549>.
- [5] Z. A. Karim, "Evaluasi Program Raskin (Beras untuk Keluarga Miskin) di Kelurahan Tanjung Ayun Sakti Kecamatan Bukit Bestari Kota Tanjung Pinang," *Jurnal Ilmu Sosial dan Ilmu Politik*, vol. 1, no. 1, pp. 34-50, 2019, doi: <https://doi.org/10.56552/jisipol.v1i1.4>.
- [6] U. Nain, "Transformasi Lumbung Desa: Evaluasi Program Raskin dan Rastra di Kabupaten Bulukumba," *Jurnal Pemikiran Sosiologi Volume*, vol. 5, no. 1, 2018, doi: <http://dx.doi.org/10.22146/jps.v5i1.35404>.
- [7] R. Rosihan, M. Fhadli, and A. A. H. Usman, "Klasifikasi Kelayakan Pemberian Kredit Menggunakan Metode Decision Tree dengan Seleksi Fitur (Studi Kasus: PT. Adira Finance Cabang Kota Ternate)," *Jurnal Pendidikan Tambusai*, vol. 7, no. 3, pp. 21517-21524, 10/05 2023, doi: 10.31004/jptam.v7i3.9915.
- [8] P. Andriani Hartanto, "Penerapan Algoritma Decision Tree Untuk Seleksi Penerima Beasiswa (Studi Kasus: Smpn 1 Soreang)," *Journal of Comprehensive Science (JCS)*, vol. 2, no. 7, pp. 1924-1302, 07/26 2023, doi: 10.59188/jcs.v2i7.452.
- [9] N. Attamami, A. Triayudi, and R. T. Aldisa, "Analisis Performa Algoritma Klasifikasi Naive Bayes dan C4. 5 untuk Prediksi Penerima Bantuan Jaminan Kesehatan," *Jurnal Jtik (Jurnal Teknologi Informasi Dan Komunikasi)*, vol. 7, no. 2, pp. 262-269, 2023, doi: <https://doi.org/10.35870/jtik.v7i2.756>.
- [10] E. Mardiani *et al.*, "Komparasi Metode Knn, Naive Bayes, Decision Tree, Ensemble, Linear Regression Terhadap Analisis Performa Pelajar SMA," *Innovative: Journal Of Social Science Research*, vol. 3, no. 2, pp. 13880-13892, 2023, doi: <https://doi.org/10.31004/innovative.v3i2>.

- [11] I. Nawawi and Z. Fatah, "Penerapan Decision Trees dalam Mendeteksi Pola Tidur Sehat Berdasarkan Kebiasaan Gaya Hidup," *Jurnal Ilmiah Sains Teknologi dan Informasi*, vol. 2, no. 4, pp. 34-41, 2024, doi: <https://doi.org/10.59024/jiti.v2i4.969>.
- [12] D. A. Pratama, I. R. Mutaqin, and K. R. Manuela, "Analisis Terjadinya Kanker Paru-Paru Pada Pasien Menggunakan Decision Tree: Penerapan Algoritma C4. 5 Dan RapidMiner Untuk Menentukan Risiko Kanker Pada Gejala Pasien," *Jurnal Teknik Mesin, Industri, Elektro dan Informatika*, vol. 2, no. 4, pp. 156-170, 2023, doi: <https://doi.org/10.55606/jtmei.v2i4.3004>.
- [13] A. H. Nasrullah, "Implementasi algoritma Decision Tree untuk klasifikasi produk laris," *Jurnal Ilmiah Ilmu Komputer Fakultas Ilmu Komputer Universitas Al Asyariah Mandar*, vol. 7, no. 2, pp. 45-51, 2021, doi: <https://dx.doi.org/10.35329/jiik.v7i2.203>.
- [14] P. B. N. Setio, D. R. S. Saputro, and B. Winarno, "Klasifikasi Dengan Pohon Keputusan Berbasis Algoritme C4. 5," in *PRISMA, Prosiding Seminar Nasional Matematika*, 2020, vol. 3, pp. 64-71.
- [15] P. Kasih, "Pemodelan Data Mining Decision Tree Dengan Classification Error Untuk Seleksi Calon Anggota Tim Paduan Suara," *Jurnal Innovation in Research of Informatics (INNOVATICS)*, vol. 1, no. 2, pp. 63-69, 2019, doi: <https://doi.org/10.37058/innovatics.v1i2.918>.
- [16] R. Rusito and M. Firmansyah, "Implementasi Metode Decision Tree Dan Algoritma C4. 5 Untuk Klasifikasi Data Nasabah Bank," *Jurnal Ilmiah Infokam*, vol. 12, no. 2, pp. 1-12, 2016, doi: <https://doi.org/10.53845/infokam.v12i2.103>.
- [17] G. L. Pritalia, "Penerapan Algoritma C4. 5 untuk Penentuan Ketersediaan Barang E-commerce," *Indonesian Journal of Information Systems*, vol. 1, no. 1, pp. 47-56, 2018, doi: <https://doi.org/10.24002/ijis.v1i1.1727>.
- [18] S. F. Damanik, A. Wanto, and I. Gunawan, "Penerapan Algoritma Decision Tree C4. 5 untuk Klasifikasi Tingkat Kesejahteraan Keluarga pada Desa Tiga Dolok," *Jurnal Krisnadana*, vol. 1, no. 2, pp. 21-32, 2022, doi: <https://doi.org/10.58982/krisnadana.v1i2.108>.
- [19] H. Nugroho, G. E. Yuliasuti, and A. F. Pradana, "Klasifikasi Diagnosis Diabetes Melitus Menggunakan Metode Naïve Bayes dengan Seleksi Fitur Backward Elimination," *NERO (Networking Engineering Research Operation)*, vol. 8, no. 2, pp. 97-106, 2023, doi: <https://doi.org/10.21107/nero.v8i2>.
- [20] D. Anggara, N. Suarna, and Y. A. Wijaya, "Analisa Perbandingan Performa Optimizer Adam, SGD, dan MSProp Pada Model H5," *NERO (Networking Engineering Research Operation)*, vol. 8, no. 1, pp. 53-64, 2023, doi: <https://doi.org/10.21107/nero.v8i1>.
- [21] A. R. Pratama and A. F. Cobantoro, "Klasifikasi Citra Pneumonia Menggunakan Arsitektur Convolutional Neural Network (CNN)," *NERO (Networking Engineering Research Operation)*, vol. 8, no. 2, pp. 133-144, 2023, doi: <https://doi.org/10.21107/nero.v8i2>.