PENERAPAN TEKNIK SMOTE DAN *CROSS VALIDATION* PADA *DECISION TREE* UNTUK KLASIFIKASI TINGKAT KEMACETAN LALU LINTAS

Yajid Khoeri [1], Rudi Kurniawan [2], Yudhistira Arie Wijaya [3]
[1] Jurusan Teknik Informatika, STMIK IKMI Cirebon
[2] Jurusan Rekayasa Perangkat Lunak, STMIK IKMI Cirebon
[3] Jurusan Sistem Informasi, STMIK IKMI Cirebon
Jl. Perjuangan No.10B, Karyamulya, Kec. Kesambi, Kota Cirebon, Jawa Barat 45135
khoeriyajid@gmail.com

ABSTRAK

Kemacetan merupakan suatu keadaan dimana jumlah kendaraan yang lalu lalang melebihi kapasitas jalan yang dapat ditampung. Kemacetan lalu lintas, disebabkan oleh volume kendaraan melebihi kapasitas jalan, merugikan secara ekonomi dan sosial. Kemacetan juga mempengaruhi mobilitas penduduk, perekonomian, dan lingkungan, sehingga menimbulkan biaya ekonomi yang signifikan melalui peningkatan konsumsi bahan bakar, hilangnya waktu, dan dampak negatif terhadap produktivitas. Tujuan penelitian ini adalah untuk mengurangi kemacetan lalu lintas. Penelitian ini menerapkan metode *Decision Tree* dan *Synthetic Minority Over-sampling Technique* (SMOTE) untuk mengatasi ketidakseimbangan kelas. Proses *Knowledge Discovery in Database* (KDD) dan Cross Validation digunakan untuk evaluasi. Hasilnya menunjukkan Decision Tree dengan SMOTE memiliki akurasi 96,54%. Dengan *Cross Validation*, akurasi mencapai 96,03% +/- 0,85% (rata-rata mikro: 96,03%). Model ini dianggap baik dan efektif dalam mengklasifikasikan tingkat kemacetan lalu lintas, memberikan kontribusi penting pada pemahaman dan manajemen lalu lintas perkotaan. Implementasi ini dapat membantu mengoptimalkan solusi untuk mengurangi dampak negatif kemacetan dan meningkatkan kualitas hidup masyarakat.

Kata kunci : *Kemacetan, Decision Tree, SMOTE (Synthetic Minority Over-sampling Technique) , Cross Validation, KDD (Knowledge Discovery in Database)*

1. PENDAHULUAN

Kemacetan secara umum menjadi kekhawatiran yang semakin meningkat, terutama di kota-kota besar di negara-negara berkembang. Idealnya, seluruh pemangku kepentingan harus mengatasi masalah kemacetan lalu lintas dan memperkenalkan berbagai alternatif solusi untuk mengendalikan kemacetan lalu lintas tersebut. Karena kecelakaan ini akan menjadi sangat serius di kemudian hari dan akan mengakibatkan penumpukan pertumbuhan penduduk di perkotaan yang berdampak pada peningkatan kebutuhan lalu lintas[1]. Kemacetan merupakan suatu keadaan dimana jumlah kendaraan yang lalu lalang melebihi kapasitas jalan yang dapat ditampung[2].

Kemacetan juga mempengaruhi mobilitas penduduk, perekonomian, dan lingkungan, sehingga menimbulkan biaya ekonomi yang signifikan melalui peningkatan konsumsi bahan bakar, hilangnya waktu, dan dampak negatif terhadap produktivitas. Selain itu, terjebak kemacetan terus-menerus dapat meningkatkan stres dan kelelahan, yang dapat berdampak buruk pada kesehatan psikologis pengemudi dan penumpang. Situasi ini semakin buruk karena meningkatnya risiko kecelakaan di jalan raya, peningkatan emisi kendaraan, dan pertumbuhan perkotaan yang tidak terkendali. Seiring pesatnya kemajuan teknologi informasi, teknologi informasi berperan dalam memahami dan mengoptimalkan lalu lintas data melalui teknik data mining.

Salah satunya adalah manajemen lalu lintas, dan pengendalian manajemen lalu lintas dapat digunakan sebagai bahan untuk mengklasifikasikan dan memprediksi arus lalu lintas kemacetan.serta kepadatan. Selain itu, penerapan data mining pada data lalu lintas dapat menimbulkan banyak tantangan yang memerlukan pengelolaan yang cermat. Mengelola data lalu lintas merupakan tugas kompleks yang menghadapi banyak kendala, antara lain keterbatasan data, ketidakpastian cuaca, dan kompleksitas perilaku pengemudi. Solusi yang efektif harus menggunakan Algoritma *Decision Tree*, yang memiliki keunggulan dalam kemampuan interpretasi dan kemampuan untuk menangkap hubungan nonlinier.

Untuk mengatasi ketidakseimbangan kelas pada lalu lintas data, teknik SMOTE (*Synthetic Minority Oversampling Technique*) dapat digunakan untuk meningkatkan representasi kelas minoritas dan memastikan model berkinerja baik dalam mengklasifikasikan data. Selain itu, penerapan *Cross Validation* memastikan evaluasi model yang konsisten dan andal dengan membagi kumpulan data menjadi beberapa bagian yang seimbang untuk melatih dan menguji model secara adil. SMOTE dan *Cross Validation* dapat digunakan dengan metode *Decision Tree* untuk mengatasi kendala ini. SMOTE membantu menghasilkan sampel sintetik dan menyempurnakan model untuk mengatasi ketidakseimbangan kelas. Cross *Validation* juga mempertahankan performa model pada data yang tidak terlihat selama pelatihan.

2. TINJAUAN PUSTAKA

Pada penelitian sebelumnya dengan judul "Integrasi *Decision Tree* dan Metode SMOTE untuk Pengklasifikasian Data Kecelakaan Lalu Lintas", pengujian dilakukan dengan menggunakan tiga desain model: Split Validation *Decision Tree* dan SMOTE, dengan akurasi sebesar 69,23%. Pengujian menggunakan pohon keputusan yang divalidasi silang dan SMOTE menunjukkan akurasi sebesar 63,56%.Pengujian data split *Decision Tree* dan SMOTE menunjukkan akurasi sebesar 71,12 dengan rasio 1: 9. Setelah membandingkan ketiga desain model, Decision Tree dan SMOTE Split Data mencapai akurasi tertinggi. Selanjutnya diperoleh akurasi sebesar 89,71% (3:7) dan area under curve (AUC) sebesar 0,773 (1:.9). Penelitian ini termasuk dalam kategori cukup[3].

Penelitian selanjutnya dengan judul "Klasifikasi Pembagian Arus Lalu Lintas Menggunakan Alogaritma *Naïve Bayes* Dan *Model Linear*". hasil penelitian ini metode Naive Bayes mendapatkan accuracy lebih tinggi dari model linier yaitu untuk Naive Bayes accuracy 95.70%, precision 95.67%, dan recall 100%, sedangkan untuk Naive Bayes accuracy 92.10%, precision 95.68%, dan recall 96.20%. maka hasilnya metode Naive Bayes lebih unggul dalam proses klasifikasi data arus lalu lintas.[4]

Pada penelitian sebelumnya Dengan Judul "Implementasi Algoritma *Decision Tree* Untuk Klasifikasi Data Peserta Didik". Berdasarkan percobaan didapatkan bahwasanya algoritma *Decision Tree* yang diujicoba menunjukkan hasil yang memuaskan. Baik C4.5 maupun Random Forest telah menunjukkan kinerja yang tinggi dalam ukuran akurasi, yakni 97,63% untuk C4.5 dan 95,13% untuk Random Forest. Berdasarkan ukuran akuras ini, C4.5 mengungguli Random Forest sebesar 2,5%. Oleh karena itu, model dan rule yang dihasilkan oleh C.45 digunakan sebagai dasar pengembangan prototipe aplikasi klasifikasi.[5]

Pada penelitian sebelumnya, yang berjudul "Analisis sentimen pada rating aplikasi Shopee menggunakan metode Decision Tree berbasis SMOTE" membahas membahas analisis sentimen, cabang penelitian text mining yang melakukan proses klasifikasi dokumen teks. Analisis sentimen melibatkan penggunaan teknik pemrosesan bahasa alami untuk mengekstraksi opini, perasaan, dan penilaian orang yang menulis tentang topik tertentu. Peneliti melakukan penelitian terhadap analisis sentimen review aplikasi Shopee dengan menggunakan teknik pohon keputusan. Tujuan dari penelitian ini adalah untuk mengetahui tingkat akurasi dan mengetahui opini pengguna terhadap aplikasi Shopee. Hasil penelitian menggunakan Algoritma Decision Tree menggunakan SMOTE (Synthetic Minority Oversampling Technique) menunjukkan nilai presisi sebesar 99,98% dengan nilai presisi sebesar 99,91%, AUC (Area Under The Curve) sebesar 0,999, dan recall sebesar 99,88%.Hasil penggunaan Algoritma Decision Tree tanpa SMOTE menunjukkan nilai presisi sebesar 99,89%, AUC (Area Under The Curve) sebesar 0,950, recall sebesar 99,88%, dan nilai presisi sebesar 99,98%. Dari hasil evaluasi yang ada, SMOTE dapat mempengaruhi nilai presisi dan AUC (Area Under Curve), dan nilai recall dan presisi tidak berpengaruh atau hasilnya berbeda, terlepas digunakan atau tidaknya SMOTE, dapat kita simpulkan sama saja. Selisih nilai akurasi yang diperoleh sebesar 0,02 dengan AUC sebesar 0,049.[6]

Pada penelitian sebelumnya yang berjudul "Klasifikasi Penyakit Jantung Menggunakan Metode *Synthetic Minority Over- Sampling Technique* Dan *Random Forest Clasifier*" membahas mengenai, klasifikasi penyakit jantung dengan menggabungkan Synthetic Minority Over-sampling Technique (SMOTE) dan Algoritma Random Forest Classifier, dengan menggunakan data pasien yang mengalami diagnosis penyakit jantung dan tidak mengalami penyakit jantung. Tahapan awal yang dilakukan yaitu mengatasi masalah ketidakseimbangan kelas dengan mengaplikasikan SMOTE untuk menghasilkan sampel sintetis dari kelas minoritas, lalu dilanjutkan pada proses normalisasi data menggunakan metode min-max normalisasi, setelah itu masuk pada proses klasifikasi menggunakan Random Forest Classifier untuk melatih model dalam melakukan klasifikasi. Hasilnya menunjukkan bahwa pendekatan ini mampu meningkatkan kemampuan model dalam mengidentifikasi kasus penyakit jantung. Evaluasi model menghasilkan akurasi yang lebih baik dibandingkan hasil yang didapatkan pada penelitian yang dilakukan sebelumnya yaitu mencapai akurasi 92%, hasil terbaik ini terjadi peningkatan 2% dari hasil akurasi yang dihasilkan penelitian.[7]

3. METODE PENELITIAN

Pada bagian ini memuat metode yang digunakan pada penelitian ini. Metode penelitian ini menggunakan Algoritma *Decision Tree* berbasis SMOTE (*Synthetic Minority Over- Sampling Technique*) dan *Cross Validation*. Pada bagian data tingkat kemacetan lalu lintas. Dataset dalam penelitian ini terdiri dari 297.000 yang di ambil dari situs open data yaitu kaggle berjenis time series . Data tersebut kemudian di analisis untuk mengetahui situasi lalu lintas berdasarkan 4 kategori 1-berat(*heavy*), 2-Tinggi(*high*), 3-Normal(*normal*), dan 4-Rendah(*low)* menggunkan Algoritma *Decision Tree* berbasis SMOTE) dan *Cross Validation*. Performa algoritma *Decision Tree* dalam menganalisis data tingkat lalu lintas diukur dengan menghitung akurasi, recall, dan precision. Alur metode penelitian ini yang di tampilkan pada gambar 1.

Gambar 1. Metode penelitian

3.1. Identifikasi Masalah

Memahami dengan jelas ruang lingkup penelitian dan identifikasi tujuan yang ingin dicapai. Menganalisis secara mendalam situasi dan latar belakang yang berkaitan dengan penelitian. Memahami faktor-faktor yang mungkin mempengaruhi atau berkontribusi terhadap masalah.Menemukan informasi dari berbagai sumber, termasuk literatur, laporan, data statistik, dan studi kasus terkait topik atau masalah yang diidentifikasi. Merumuskan masalah yang ingin dipecahkan atau diselidiki.

3.2. Studi Literature

Pada penelitian ini topik yang dipilih adalah mengenai Implementasi Algoritma *Decision Tree* Berbasis SMOTE dan *Cross Validation***.** Langkah selanjutnya adalah pencarian literatur yang relevan dengan topik penelitian. Langkah ini dapat memberikan gambaran mengenai dengan penggunaan Algoritma *Decision Tree* dalam mengklasifikasikan data tingkat kemacetan lalu lintas . Proses pencarian dan pengumpulan artikel dengan menggunakan *tools publish or perish*, yang diambil dari database akademik yaitu *Google Scholar, Shinta dan Scopus* dengan kata kunci "Klasifikasi Kemacetan" dan "Algoritma *Decision Tree*".

3.3. Pengumpulan Data

Dalam penelitian ini untuk mendapatkan dataset yang diperlukan maka dari itu digunakan beberapa metode pengumpulan data. Dataset yang digunakan dalam penelitian ini adalah data arus lalu lintas yang bersifat time series dan didapatkan dari website open data kaggle dataset yang di memiliki 9 *atribut* diantaranya *Time, date, Day of the week, Carcount, Bikecount, Buscount, Truckcount, Total dan Traffic situation* dan memiliki 297.000 *record*.

3.4. Proses Tahapan KDD

Discovery of Knowledge in Databases (KDD) adalah proses yang dapat digunakan untuk menemukan pola yang berguna dan mengekstrak informasi dari kumpulan data yang besar. KDD bersifat otomatis dan dapat didefinisikan sebagai organisasi proses untuk mengidentifikasi dan menemukan pola dari kumpulan data yang besar dan kompleks secara akurat dan efektif. Data yang dikumpulkan melewati tahap pra-pemrosesan untuk memastikan kualitas dan kegunaannya untuk analisis.Pembersihan data melibatkan penghapusan data yang tidak valid atau tidak relevan. Jika terdapat nilai yang hilang pada data maka dilakukan langkah pengisian nilai yang hilang tersebut. Konversi data mungkin diperlukan jika skala atau format perlu disesuaikan.

3.5. Implementasi Algoritma *Decision Tree*

Algoritma *Decision Tree* merupakan Algoritma yang terdapat dalam metode klasifikasi dari data mining. *Decision tree* adalah konsep *flowchart* dengan struktur pohon (*tree*) dimana setiap node (node internal) mewakili atribut dan cabang menggambarkan hasil pengujian atau nilai input atribut, sedangkan daun mewakili kelas atau distribusi kelas[8] Langkah kerja algoritma *Decision tree* yaitu menghitung nilai *entropy*, nilai *information gain*, nilai *split info*, dan *nilai gain ratio* dijelaskan sebagai berikut.

3.5.1. Menghitung entropi setiap atribut menggunakan persamaan 1

$$Entropy\ (S) \sum_{i=1}^{n} -pi\ log2pi \quad (1)$$

Dimana :
S = Himpunan Kasus
n = Jumlah partisi S
p_i = Probabilitas yang didapat dari jumlah kelas dibagi total kasus.

3.5.2. Menghitung *Information gain* dari setiap atribut menggunakan persamaan 2

$$Gain\ SA = Entropy\ (S) \sum_{i=1}^{n} \frac{[Si]}{[S]} * Entropy\ Si \quad (2)$$

Dimana :
S = Himpunan Kasus
A = Atribut
n = Jumlah partisi S
[Si] = Jumlah partisi ke-i
[S] = Jumlah kasus dalam S

3.5.3. Menghitung *Split gain* dari setiap atribut menggunakan persamaan 3

$$Split\ in\ for(S,A) = Entropy\ (S) \sum_{i=1}^{n} \frac{[Si]}{[S]}\ log2\ \frac{[Si]}{[S]} \quad (3)$$

Dimana :
S = Ruang sampel data yang digunakan untuk training
A = Atribut
[Si] = Jumlah sampel atribut i

3.5.4. Menghitung Gain ratio dari setiap atribut menggunakan persamaan 4

$$Gain\ ratio\ (S,A) = \frac{Information\ Gain\ (S,A)}{Split\ Info\ (S,A)} \quad (4)$$

3.6. Evaluasi

Menentukan akurasi persisi, dan *recall* dari *Cross Validation*. Untuk mengetahui pengujian model desain yang terbaik dari integrasi metode klasifikasi Decision Tree dan teknik penyeimbang data yaitu SMOTE berdasarkan faktor yang mempengaruhi tingkat kemacetan lalu lintas, sehingga akurasi menjadi meningkat melalui persamaan 5.

$$Akurasi = \frac{tp+tn}{tp+fn+fp+tn} \quad (5)$$

$$Precision = \frac{tp}{tp+fp} \quad (6)$$

Adapun attribute/variabel yang digunakan untuk penelitian ini yaitu *Day of the week*, *CarCount*, *BikeCount*, *BusCount*, *TruckCount*, *Total*, dan *Traffic Situation.*

4. HASIL DAN PEMBAHASAN

4.1. Data Set

Data yang digunakan dalam penelitian ini diperoleh dari website kaggel https://www.kaggle.com/datasets/hasibullahaman/traffic-prediction-dataset/data. Dimana berisikan 297.000 recod dan 9 atribut. Berisi informasi yang dikumpulkan oleh model visi computer. Selain itu, kumpulan data menyertakan kolom yang menunjukkan situasi lalu lintas yang dikategorikan ke dalam empat kelas: 1-Berat(*Heavy*), 2-Tinggi*(High)*, 3-Normal*(Normal*), dan 4-Rendah*(Low).* Informasi ini dapat membantu menilai tingkat keparahan kemacetan dan memantau kondisi lalu lintas pada waktu dan hari yang berbeda dalam seminggu tools yang di gunakan dalam penelitian ini menggunakan aplikasi Rapidminer versi *Educational* 10.3000.

4.2. Data Selection

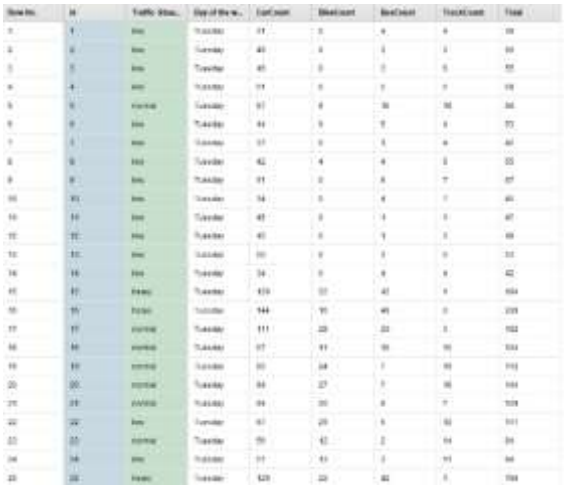
Proses pemilihan data (*data selection*) dalam KDD melibatkan identifikasi dan ekstraksi subset data yang relevan dari sumber data yang lebih besar. Tujuan utamanya adalah untuk fokus pada bagian data yang paling mungkin mengandung pola, informasi, atau relasi yang signifikan. Atribut yang digunakan tampak dalam tabel 1 berikut ini :

Tabel 1. Atribut

Atribut	Type Data
Day of the week	Polynominal
CarCount	Integer
BikeCount	Integer
BusCount	Integer
TruckCount	Integer
Total	Integer
Traffic Situation	Polynominal

4.3. Pre-Processing

Tahap selanjunya proses pelabelan dalam data set menggunkan operator *Set Role* pada RapidMiner. Pelabelan berfungsi untuk memberi identitas objek pada atribut data set. *Set Role* adalah suatu langkah yang menentukan peran atau *role* dari setiap atribut dalam dataset. Proses ini penting karena membantu RapidMiner memahami bagaimana atribut tertentu seharusnya diperlakukan selama analisis data. Atribut yang menjadi label dalam data ini adalah *Traffic situation*. Untuk mengetahui tingkat kemacetan dalam data arus lalu lintas. Hasil dari proses pelabebelan bisa di lihat dalam gambar 2 berikut ini :



Gambar 2. Hasil operator *set role*

4.4. Transformation

Tahapan Transformasi data adalah pengubahan format data tersimpan menjadi bentuk standar format file yang sesuai dengan aplikasi yang akan digunakan. Pada proses ini salah satu atribut yang di pakai yaitu atribut *day of the week* di rubah menjadi type data numerik, Operator *Nominal to Numerical* membantu mengatasi karakteristik kategorikal tersebut dengan memberikan representasi numerik pada setiap nilai kategori. Proses ini dikenal sebagai encoding, dan dalam konteks hari dalam seminggu, melibatkan pengubahan setiap hari menjadi nilai numerik. Parameter Parameter pada operator *Nominal to Numerical* yang digunakan tampak pada tabel 2 dibawah ini :

Tabel 2. *Parameter operator nominal to numerical*

Parameter	Isi
attribut filter type	*Subset*
Attributes	*Select Atributes*
	Day Of The Week

Tahapan selanjutnya adalah *SMOTE*. *SMOTE* merupakan teknik yang dimanfaatkan untuk mengatasi masalah *class imbalance problem* (CIP). *SMOTE* bekerja dengan memodifikasi dataset yang tidak seimbang dengan cara membuat data sintetik baru dari kelas minoritas dengan tujuan meningkatkan kinerja dari metode klasifikasi[9]. Pada *SMOTE* kemungkinan terjadi *overfitting y*aitu data pada kelas minoritas yang terduplikasi. Dengan tampilan parameter tampak pada gambar 3 berikut ini :

Gambar 3. *Parameter smote*

Kemudian data pada tahap selanjutnya data set di bagi 2 dengan data testing dan data traning untuk dapat di klasifikasikan dengan ratio 80% dan 20%. Menggunkan *operator Split Data*. Pemisahan data pelatihan dan pengujian dengan rasio 80% dan 20% umumnya dilakukan untuk mengoptimalkan model dan menghindari overfitting. Data pelatihan 80% digunakan untuk melatih model, sementara data pengujian 20% digunakan untuk menguji kinerja model pada data yang belum pernah dilihat sebelumnya.

4.5. Data mining

Dalam tahapan data mining menggunakaan implementasi algoritma *Decicion Tree* untuk mengklassifikasikan data lalu lintas atau *traffic data* yang di kategorikan ke dalam empat kelas: 1-Berat (*Heavy*), 2-Tinggi *(High)*, 3-Normal *(Normal*), dan 4-Rendah *(Low).* Dari hasil implementasi algoritma Decsion Tree berupa model pohon keputusan yang tampak pada gambar 4 berikut ini :

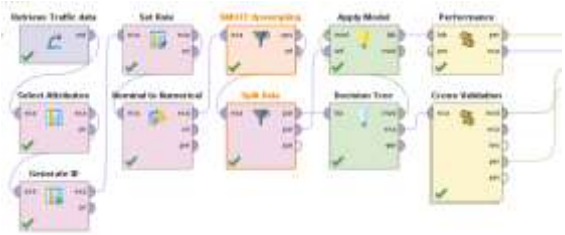
Gambar 4. Model pohon keputusan

Jika Total kendaraan melebihi 167.500, maka diklasifikasikan sebagai *'heavy'* dengan 546 kasus." "Jika Total kendaraan ≤ 167.500, dan *TruckCount* > 12.500, maka dilakukan pengecekan tambahan." "Jika Total > 111.500 dan *TruckCount* > 20.500, maka diklasifikasikan sebagai *'high'* dengan 159 kasus." "Jika Total ≤ 111.500 dan *TruckCount* > 12.500, maka *'normal'* dengan 909 kasus." "Jika *TruckCount* ≤ 12.500, dilakukan pengecekan pada Total dan *BusCount*." "Jika Total > 111.500 dan *BusCount* > 25.500, maka *'high'* dengan 72 kasus." "Jika Total > 111.500 dan *BusCount* ≤ 25.500, maka 'normal' dengan 172 kasus." "Jika Total ≤ 111.500, maka diklasifikasikan sebagai *'low'* dengan 1335 kasus." Hasil ini memberikan aturan klasifikasi berdasarkan kombinasi nilai *Total*, *TruckCount*, dan *BusCount.* Dan dengan model *attribute weights Decision Tree* seperti tampak pada tabel 3 berikut ini :

Tabel 3. *Attribute weights decision tree*

Atribut	*Weights*
Total	0,051
BusCount	0,223
TruckCountt	0,276

Hasil model *atribut weights* pada *Decision Tree* menunjukkan kontribusi masing-masing atribut terhadap keputusan klasifikasi. Dalam konteks ini, *weights* menunjukkan seberapa besar pengaruh setiap atribut terhadap keputusan akhir dalam model *Decision Tree*. Semakin tinggi nilai *weights,* semakin besar kontribusi atribut tersebut terhadap pembentukan Keputusan. Proses data mining bisa di lihat dalam gambar 5 berikut ini :



Gambar 5. Proses *data mining*

4.6. Evaluation

Pada tahap ini bertujuan untuk mengetahui pengujian model desain yang terbaik dari integrasi metode klasifikasi *Decision Tree* dan SMOTE berdasarkan faktor yang mempengaruhi tingkat kemacetan lalu lintas. Output dari evaluasi ini berupa *nilai akurasi*, *presisi*, dan *recall.* Dengan menggunakan desain mod el *Cross Validation* [10]*Decision Tree* dan SMOTE dilakukan untuk menguji kinerja model klasifikasi *yaitu Decision Tree.* Dengan menggunkan parameter Cross Validation seperti pada gambar 6 berikut ini :

Gambar 6. Parameter *cross validation*

Angka 10 fold dan *sampling type automatic* pada *cross validation* dipilih karena merupakan rekomendasi yang terbaik untuk model. Di dalam operator *Cross validation* terdapat beberapa operasi seperti pada gambar 7 berikut ini :

Gambar 7. Kumpulan operator di dalam *cross validation*

Pada kumpulan operator *di cross validation* ini yang pertama gunakan operator metode *Decision Tree* selanjutnya *apply model* atau memodelkan metode *Decision Tree*. Dan yang terakhir gunakan *performance (classification*). Pada *performance* pilih untuk memunculkan tingkat *accuracy, precision,* dan *recall.* Paramater nilai *Accuracy, Precisio*n, dan *Recall* di gunakan untuk mengukur seberapa performa Algoritma *Decision Tree* dan SMOTE dengan Cross *Validation* dalam mengklasifikasikan data *traffic.* Berikut hasil *performance Decision tree* dan SMOTE bias di lihat pada gambar 8 berikut ini :

Gambar 8. *Performa decision tree dan smote*

Model pohon keputusan menggunakan SMOTE menunjukkan kinerja yang baik dalam mengklasifikasikan data lalu lintas dengan akurasi sebesar 96,54%. Matriks konfusi menggambarkan keberhasilan model dalam memprediksi setiap kelas, seperti yang ditunjukkan oleh 334 prediksi yang benar untuk kelas "rendah". Kelas tersebut menunjukkan kemampuan model dalam mendeteksi kasus positif.

Hasil ini memberikan gambaran komprehensif tentang efektivitas model dalam menghilangkan ketidakseimbangan kelas dan kemampuannya untuk memprediksi secara akurat setiap kategori kemacetan lalu lintas. Dan hasil perfoma mengklasifikasikan data *traffic* dari metode *Decisin Tree* dan *SMOTE* dengan *Cross Validation* bias di lihat pada tabel 4 berikut ini:

Tabel 4. Hasil cross validation

Accuracy : 96.03% +/- 0,85% (rata-rata mikro: 96,03%)	**True low**	**True normal**	**True heavy**	**True high**	**Class precision**
Pred low	1335	104	0	0	92.77%
Pred normal	0	1223	0	26	97.92%
Pred heavy	0	6	546	0	98.91%
Pred high	0	2	0	231	98.21%
Class recall	100.00%	91.61%	100.00%	89.88%	99.14%

Hasil dari evaluasi *Cross Validation* mencapai akurasi sebesar 96,03% dengan deviasi sekitar +/- 0,85% (rata-rata mikro: 96,03%). Hasil ini mencerminkan hasil model. Kemampuan untuk mengklasifikasikan instance dengan akurasi tinggi dalam kumpulan data lalu lintas yang berisi empat kelas (Rendah, Normal, Kuat, Tinggi). Matriks konfusi menunjukkan bahwa semua model yang menghasilkan hasil baik memberikan kelas. Pada tingkat ketelitian yang tinggi. Ketelitian pada kelas “rendah” adalah sekitar 92,77%.

Akurasi untuk kelas 'berat' adalah sekitar 98,91%, yang menunjukkan tingkat akurasi model dalam mengidentifikasi instance yang termasuk dalam kelas tersebut. Selain itu, mendapatkan hasil recall yang baik untuk setiap kelas, termasuk recall sekitar

91,61% untuk kelas 'normal'. Hal ini menunjukkan bahwa model mampu menangkap sebagian besar data nyata dan mengenali contoh positif yang ada dalam data

5. KESIMPULAN DAN SARAN

Penelitian ini mengimplementasikan algoritma *Decision Tree* dengan memanfaatkan teknik SMOTE (*Synthetic Minority Over-sampling Technique*) dan *Cross Validation* untuk mengklasifikasikan data tingkat kemacetan lalu lintas. Hasil penelitian menunjukkan bahwa model yang dikembangkan mencapai tingkat akurasi sebesar 96.03% dengan deviasi standar sebesar 0.85% (*micro average*: 96.03%) menggunakan *Cross Validation*. Evaluasi model juga dilakukan dengan *Confusion Matrix*, yang menunjukkan tingkat presisi dan *recall* yang baik untuk setiap kelas tingkat kemacetan. Model ini memperoleh tingkat presisi tertinggi pada kelas *"heavy*" sebesar *98.91%*.

Meskipun demikian, saran untuk penelitian selanjutnya mencakup optimasi parameter *Decision Tree*, eksplorasi Algoritma klasifikasi alternatif, penanganan variabilitas data lalu lintas, dan inklusi fitur tambahan untuk meningkatkan generalitas dan kinerja model di lapangan. Implementasi di lapangan dan keterlibatan pihak terkait juga diusulkan untuk memastikan relevansi dan penerimaan solusi ini dalam konteks manajemen lalu lintas perkotaan.

DAFTAR PUSTAKA

[1] A. Pengaruh, P. Pengendara, P. Kemacetan, D. Alternatif, P. Di, and K. Makassar, "Jurnal Pendidikan dan Konseling," *J. Pendidik. DAN KONSELING Vol. 5 NOMOR 1 TAHUN 2023*, vol. 5, pp. 2784–2799, 2023.

[2] N. Fadlia and R. Kosasih, "Klasifikasi Jenis Kenda0raan Menggunakan Metode Convolutional Neural Network (CNN)," *J. Ilm. Teknol. dan Rekayasa Vol.*, pp. 207–215, [Online]. Available: https://doi.org/10.35760/tr.2019.v24i3.2397

[3] A. Franseda, "Integrasi Metode Decision Tree dan SMOTE untuk Klasifikasi Data Kecelakaan Lalu Lintas Integration of Decision Tree and SMOTE Methods for Classification of Traffic Accidents Data," *JUSTIN(Jurnal Sist. dan Teknol. Informasi) Vol. 08 , No. 3 , Juli 2020 kecelakaan*, vol. 08, no. 3, 2020, doi: 10.26418/justin.v8i3.40982.

[4] M. F. Abdi, S. Y. Qodarbaskoro, A. Alfani, and D. Maulina, "Klasifikasi Pembagian Arus Lalu Lintas Menggunakan Alogaritma Naïve Bayes dan Model Linear," *J. Ilm. "Technologia" Vol 12, No. 4, Oktober-Desember 2021*, no. 4, pp. 203–208, 2021.

[5] Imam Sutoyo, "Implementasi Algoritma Decision Tree Untuk Klasifikasi Data Peserta Didik," *J. PILAR Nusa Mandiri*, vol. 7, no. 2, pp. 45–51, 2018, doi: 10.35329/jiik.v7i2.203.

[6] C. Cahyaningtyas, Y. Nataliani, I. R. Widiasari, F. T. Informasi, U. Kristen, and S. Wacana, "Analisis sentimen pada rating aplikasi Shopee menggunakan metode Decision Tree berbasis SMOTE," *AITI J. Teknol. Informasi, Vol. 18 No. 2 Agustus 2021, 173-184 ISSN*, vol. 18, no. 2, pp. 173–184, 2021.

[7] K. Kunci, "Klasifikasi Penyakit Jantung Menggunakan Metode Synthetic Minority Over-Sampling Technique Dan Random Forest Clasifier," *J. Comput. Sci.*, vol. 12, no. 1, pp. 2995–3011, 2023.

[8] D. Septhya *et al.*, "Implementasi Algoritma Decision Tree dan Support Vector Machine untuk Klasifikasi Penyakit Kanker Paru," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 3, no. 1, pp. 15–19, 2023, doi: 10.57152/malcom.v3i1.591.

[9] R. Sistem and O. Heranova, "Synthetic Minority Oversampling Technique pada Averaged One Dependence Estimators untuk Klasifikasi Credit Scoring," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 1, no. 10, pp. 10–12, 2021.

[10] T. Untuk, P. Status, and K. Mahasiswa, "Klasifikasi algoritma k-nearest neighbor, naive bayes, decision tree untuk prediksi status kelulusan mahasiswa s1 1) 1,2,3)," *RABIT J. Teknol. dan Sist. Inf. Univrab*, vol. 8, no. 2, pp. 146–154, 2023, [Online]. Available: https://doi.org/10.36341/rabit.v8i2.3434