e-ISSN 2962-6129
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 23 Agustus 2023

Implementasi Algoritma *Extra Trees* Untuk Klasifikasi Cuaca Provinsi Dki Jakarta Dengan *Oversampling* SMOTE

Raihan Kemmy Rachmansyah[1], Ria Astriratma[2]
Program Studi S1 Informatika / Fakultas Ilmu Komputer
Universitas Pembangunan Nasional Veteran Jakarta
Jl. RS. Fatmawati Raya, Pd. Labu, Kec. Cilandak, Kota Depok, Daerah Khusus Ibukota Jakarta 12450
raikemmy@upnvj.ac.id[1], astriratma@upnvj.ac.id[2]

Abstrak. Cuaca yang sulit diprediksi membuat banyak aktivitas warga Provinsi DKI Jakarta terganggu sehingga diperlukan sebuah ilmu teknologi yang diimplementasikan untuk mengklasifikasikan sebuah cuaca. Maka dari itu, penelitian ini menerapkan metode *Machine Learning* untuk klasifikasi cuaca Provinsi DKI Jakarta menggunakan algoritma *Extra Trees* dengan metode *oversampling* SMOTE. Data yang digunakan pada penelitian ini adalah data Prakiraan Cuaca Provinsi DKI Jakarta pada tahun 2017 hingga 2018 yang diperoleh dari situs https://data.jakarta.go.id/. Data yang diperoleh memiliki distribusi data yang tidak seimbang sehingga perlu diseimbangkan terlebih dahulu menggunakan metode *resampling* data menggunakan *Synthetic Minority Oversampling Technique* atau SMOTE. Berdasarkan hasil penelitian, metode *oversampling* SMOTE tidak mempengaruhi hasil evaluasi pada klasifikasi cuaca Provinsi DKI Jakarta menjadi lebih baik. Hasil evaluasi terbaik didapatkan oleh model menggunakan algoritma *Extra Trees* tanpa metode *oversampling* SMOTE pada rasio 80% data latih dan 20% data uji dengan nilai akurasi sebesar 79,8%, *precision* 63,1%, dan *recall* 56,1%.

Kata Kunci: Klasifikasi, *Extra Trees, Oversampling* SMOTE, Cuaca Provinsi DKI Jakarta

1 Pendahuluan

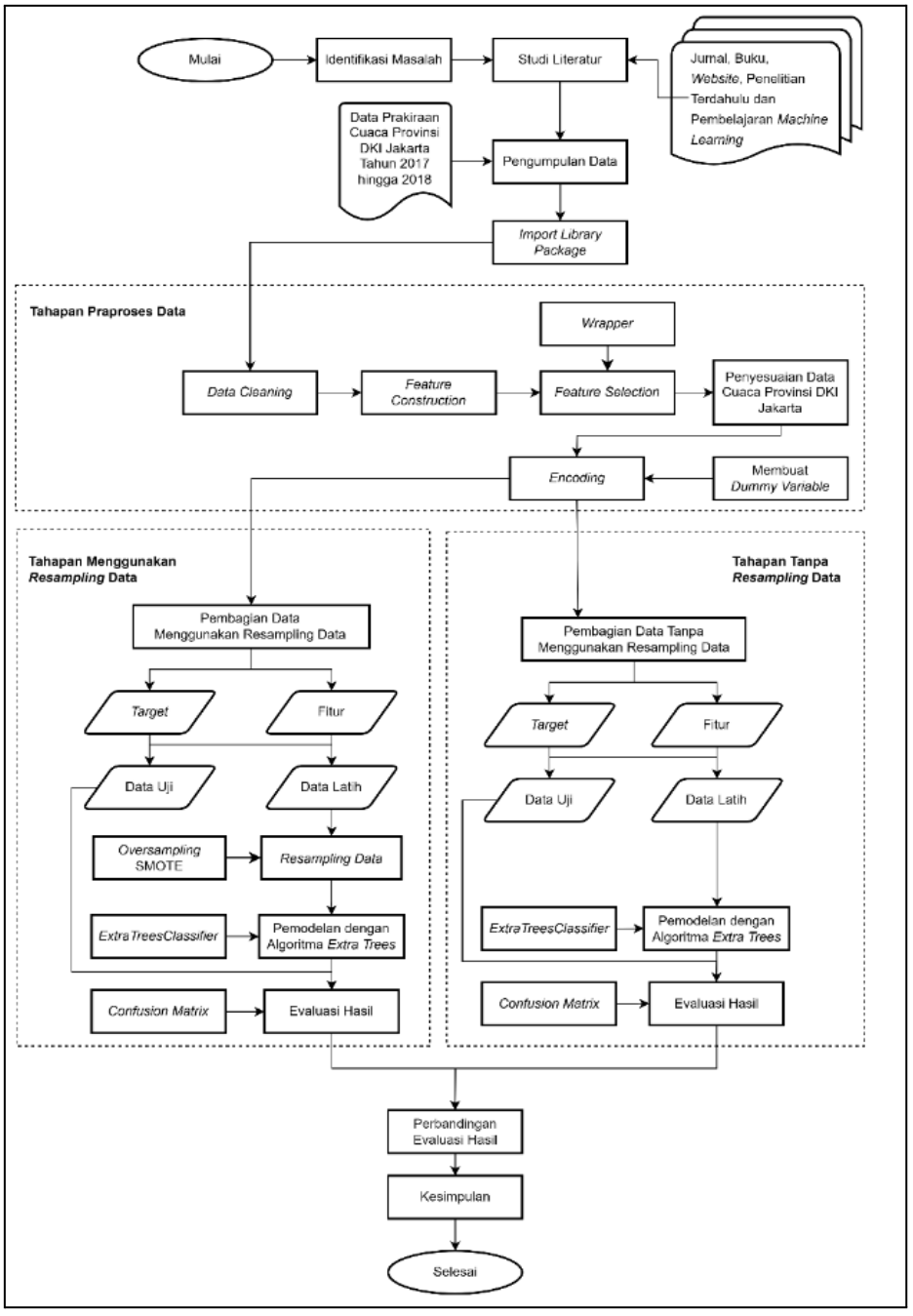
Cuaca di Indonesia khususnya di Provinsi DKI Jakarta sangat tidak menentu dan sulit untuk diprediksi. Cuaca yang sulit diprediksi membuat aktivitas warga khususnya pada Provinsi DKI Jakarta terganggu sehingga diperlukan sebuah ilmu teknologi yang diimplementasikan untuk memprediksi sebuah cuaca yaitu *Machine Leaning*. *Machine Learning* merupakan salah satu bagian dari *Artificial Intelligence* yang bekerja secara otomatis dan dapat semakin cerdas melalui pembelajarannya terhadap pengalaman yang diperoleh sebelumnya [1]. Menurut [2] prediksi cuaca adalah proses dari pengumpulan data berdasarkan kondisi atmosfer seperti suhu dan kelembaban suatu udara. Faktor tersebut diteliti lalu dibandingkan dengan kondisi cuaca pada tanggal, bulan dan tahun sebelumnya untuk mendapatkan prediksi cuaca yang akurat serta sesuai. Berdasarkan faktor tersebut maka diperoleh juga prediksi beberapa jenis cuaca seperti cerah, berawan dan hujan. Prediksi cuaca juga dilakukan untuk membuat kumpulan informasi mengenai kondisi iklim serta unsur didalamnya. Hasil prediksi cuaca di Indonesia diterbitkan melalui Badan Meteorologi Klimatologi dan Geofisika (BMKG) dengan tujuan agar masyarakat dapat mempersiapkan diri untuk menghadapi setiap kondisi cuaca yang akan datang.

Beberapa penelitian serupa telah dilakukan sebelumnya. Penelitian oleh Ghaitsa Amany Mursianto dan rekan-rekannya [3] yang melakukan klasifikasi cuaca pada data WeatherAUS menggunakan algoritma Extreme Gradient Boosting dan Random Forest serta mengimplementasikan metode oversampling SMOTE. Algoritma *Random Forest* tanpa metode *oversampling* SMOTE sebesar 89,54%, *XGBoost* tanpa metode *oversampling* SMOTE sebesar 88,96%, *Random Forest* dengan metode *oversampling* SMOTE 95,59% dan *XGBoost* dengan metode *oversampling* SMOTE sebesar 93,34%. Selain itu, penelitian oleh Faqih Hamami dan Iqbal Ahmad Dahlan [4] yang melakukan klasifikasi cuaca Provinsi DKI Jakarta tahun 2018 menggunakan algoritma Random Forest sertam mengimplementasikan metode oversampling SMOTE. algoritma *Random Forest* menggunakan metode *oversampling* SMOTE sebesar 70% dan akurasi *Random Forest* tanpa metode *oversampling* SMOTE sebesar 69%.

Berdasarkan penelitian terdahulu dan masalah yang ditemukan, peneliti mengimplementasikan metode *Ensemble Learning* yaitu algoritma *Extra Trees* dan metode *resampling* data menggunakan *oversampling* SMOTE untuk menangani distribusi data yang tidak seimbang pada kelas minoritas. Menurut [5], *Ensemble Learning* adalah metode yang baik digunakan untuk memecahkan masalah klasifikasi yang memiliki data tidak

seimbang. Penelitian ini bertujuan untuk membuktikan apakah penerapan metode *oversampling* SMOTE memiliki pengaruh pada hasil evaluasi model *Machine Learning* yang dibentuk. Oleh karena itu, penelitian ini melakukan klasifikasi cuaca Provinsi DKI Jakarta menggunakan algoritma *Extra Trees* dengan metode *oversampling* SMOTE dan algoritma *Extra Trees* tanpa metode *oversampling* SMOTE.

2 Metode Penelitian



Gambar. 1. Tahapan Metode Penelitian

2.1 Identifikasi Masalah

Perubahan cuaca secara mendadak menjadi hambatan bagi masyarakat Provinsi DKI Jakarta dalam melakukan kegiatan sehari-hari sehingga diperlukannya sebuah metode *Machine Learning* untuk memprediksi cuaca yang terjadi khususnya pada Provinsi DKI Jakarta menggunakan algoritma *Extra Trees* serta menerapkan metode *resampling* data dengan *oversampling.* Penelitian ini bertujuan untuk membuktikan apakah penerapan metode *resampling* data dengan *oversampling* SMOTE dapat mempengaruhi hasil evaluasi model menggunakan algoritma *Extra Trees* pada kasus klasifikasi cuaca Provinsi DKI Jakarta.

2.2 Studi Literatur

Studi literatur dikumpulkan untuk dijadikan sebagai referensi selama penelitian berlangsung. Beberapa sumber referensi yang digunakan antara lain jurnal, buku, *website* dan penelitian terdahulu terkait prediksi klasifikasi cuaca disertai dengan pembelajaran metode *Machine Learning*.

2.3 Pengumpulan Data

Data yang digunakan adalah data Prakiraan Cuaca Provinsi DKI Jakarta yang terdiri dari bulan Januari sampai Oktober tahun 2017 dan bulan Januari sampai Desember tahun 2018. Data tersebut diperoleh dari situs https://data.jakarta.go.id/ yang terdiri dari beberapa dokumen lalu dikumpulkan menjadi satu dokumen dengan format CSV (*Comma Separated Value*). *Dataset* tersebut terdiri dari lima fitur yaitu kolom *tanggal, waktu, wilayah, suhu_derajat_celcius* dan *kelembaban_persen* yang merupakan variabel independen dan satu target yaitu kolom *cuaca* sebagai variabel dependen. *Dataset* pada penelitian ini memiliki total 14676 data.

2.4 *Import Library Package*

Import Library Package merupakan tahapan awal dalam membangun sebuah model *Machine Learning* menggunakan bahasa pemrograman *Python*. *Library Package* yang digunakan pada penelitian ini adalah *library* dari bahasa pemrograman *Python*. Menurut [6], *library* adalah sebutan untuk kode program tambahan yang dikembangkan oleh pengembang lain untuk menyelesaikan suatu hal yang khusus. *Python* memiliki lebih dari 170000 *library* atau *packages* yang dikembangkan di suatu komunitas pada situs https://pypi.org. Dalam mempermudah pengguna, *Python* memiliki sebuah *Package System* yang disebut *pip*. Dengan adanya *Package Manager*, pengguna dapat memasang serta memperbarui *library* dengan mudah tanpa bingung dengan depedensi yang ada.

2.5 Praproses Data

Tahapan praproses data merupakan tahapan yang dilakukan terlebih dahulu sebelum membuat model klasifikasi cuaca Provinsi DKI Jakarta. Praproses merupakan tahapan yang dilakukan untuk melihat serta mengolah sebuah data yang masih memiliki *noise* atau masih kotor menjadi data yang siap digunakan untuk dalam pembuatan model [7]. Pada penelitian ini, terdapat empat tahapan praproses data yaitu *Data Cleaning*, *Feature Construction*, *Feature Selection*, penyesuaian data cuaca Provinsi DKI Jakarta dan *Encoding* dengan membuat *Dummy Variable*.

2.5.1 *Data Cleaning*

Tahapan *Data Cleaning* terdiri dari beberapa tahapan pengolahan data seperti menangani *Missing Value*, data yang duplikat, kesalahan ketik pada kolom kategorik, dan memperbaiki kesalahan tipe data pada setiap kolom. Hasil dari tahapan *Data Cleaning* adalah data cuaca Provinsi DKI Jakarta tahun 2017 dan 2018 tanpa *Missing Value*, data duplikat, kesalahan ketik, dan kesalahan tipe data.

2.5.2 *Feature Construction*

Tahapan *Feature Construction* pada penelitian ini bertujuan untuk menangani data bertipe numerik agar mempermudah dalam pembuatan model *Machine Learning* dalam memproses serta mengenali pola suatu data. Hasil dari tahapan ini dapat berupa kolom baru yang di esktrak dari satu atau lebih kolom yang dapat digunakan sebagai fitur tambahan agar model dapat mempelajari lebih banyak pola data. *Feature Constuction* bertujuan untuk memproses data pada kolom aslinya yang tidak dapat diproses oleh model *Machine Learning*.

2.5.3 *Feature Selection*

Tahapan *Feature Selection* membuang fitur yang tidak berpengaruh dalam klasifikasi cuaca Provinsi DKI Jakarta karena hanya akan memperpanjang waktu komputasi pada proses pelatihan data. Pada penelitian ini, penerapan seleksi fitur menggunakan metode *Wrapper* dengan memakai kombinasi fitur terbaik lalu membuang kolom yang tidak relevan serta yang tidak mempengaruhi hasil klasifikasi.

2.5.4 Penyesuaian Data Cuaca Provinsi DKI Jakarta

Tahapan penyesuaian data cuaca Provinsi DKI Jakarta pada masing-masing kategori cuaca agar tidak memiliki rentang nilai yang terlalu jauh. Penyesuaian jumlah data dilakukan berdasarkan setiap kategori cuaca yang mewakili setiap kategori pada kolom bertipe kategorik dengan jumlah data terbanyak. Hasil dari tahapan ini adalah data cuaca Provinsi DKI Jakarta pada tahun 2017 dan 2018 dengan jangkauan yang lebih rendah antara jumlah kategori cuaca mayoritas dengan jumlah kategori cuaca minoritas.

2.5.5 *Encoding*

Tahapan *Encoding* menggunakan metode *One-Hot Encoding* dengan membuat *Dummy Variable* pada kolom kategorik. Tahapan ini akan mengubah kolom kategorik menjadi berbentuk bilangan *biner* atau bertipe numerik agar dapat diproses oleh model *Machine Learning*. Tahapan ini menghasilkan kolom baru berdasarkan kolom kategorik yang telah dilakukan *Encoding* menggunakan metode *One-Hot Encoding* dengan membuat *Dummy Variable.*

2.6 Pembagian Data

Data yang sudah dilakukan tahapan praproses data, dilanjutkan dengan tahapan untuk membagi data menjadi target dan fitur. Target merupakan kolom *cuaca* yang merupakan variabel dependen sedangkan fitur merupakan kolom selain kolom *cuaca* yang merupakan variabel independen. Data pada fitur disimpan pada variabel *X* dan data pada target disimpan pada variabel *y*.

2.6.1 Data Latih dan Data Uji

Tahapan selanjutnya adalah memecah kembali fitur dan target menjadi dua jenis yaitu data latih (*training*) dan data uji (*testing*) pada masing-masing variabel *X* dan *y*. Lalu, variabel tersebut dipecah menjadi *X_train* dan *y_train* untuk data latih serta *X_test* dan *y_test* untuk data uji. Data latih digunakan sebagai data untuk pelatihan suatu model *Machine Learning* sedangkan data uji digunakan sebagai data untuk menguji model yang telah dibuat. Oleh karena itu, pembagian jumlah data latih harus lebih banyak dari pada data uji [8]. Peneliti melakukan percobaan menggunakan beberapa rasio antara data latih dan data uji untuk mencari komposisi data yang sesuai dengan model yang telah dibuat.

2.7 *Resampling Data*

Tahapan *resampling* data menggunakan metode *oversampling* SMOTE yang bertujuan untuk mengatasi persebaran jumlah data yang tidak seimbang pada setiap kelasnya. *Dataset* yang digunakan terdiri dari 14676 data dengan delapan jenis kategori pada kolom cuaca yaitu Cerah, Cerah Berawan, Berawan, Berawan Tebal,

Hujan Lokal, Hujan Ringan, Hujan Sedang dan Hujan Petir. Penelitian ini juga membuktikan apakah penerapan metode *oversampling* SMOTE memiliki pengaruh terhadap hasil evaluasi model *Machine Learning* pada klasifikasi cuaca Provinsi DKI Jakarta pada tahun 2017 hingga 2018.

Penelitian ini memiliki dua kondisi utama yaitu pemodelan menggunakan metode *resampling* data dengan *oversampling* SMOTE pada data latih dan pemodelan tanpa metode *resampling* data dengan *oversampling* SMOTE. Kedua kondisi tersebut bertujuan untuk mencari tahu apakah penerapan metode *resampling* data dengan *oversampling* SMOTE memiliki pengaruh terhadap hasil evaluasi model *Machine Learning* menggunakan algoritma *Extra Trees* pada klasifikasi cuaca Provinsi DKI Jakarta. Tahapan metode *resampling* data menggunakan *oversampling* SMOTE dilakukan terhadap data latih yaitu pada variabel *X_train* dan *y_train* setelah pembagian data antara fitur dan target dilakukan.

2.8 Pembuatan Model

Tahapan pembuatan model pada penelitian ini melakukan klasifikasi *Multi-Class* menggunakan algoritma dari metode *Ensemble Learning* yaitu *Extra Trees* serta menerapkan metode *resampling* data menggunakan *oversampling* SMOTE. Pemodelan dilakukan berdasarkan data latih pada *dataset* cuaca Provinsi DKI Jakarta yang sudah dilakukan tahapan praproses hingga pembagian data.

Menurut [9], *Extra Trees* atau *Extremly Randomized Trees* merupakan bagian dari metode *Ensemble Learning* yang merupakan *Ensemble* dari algoritma *Decision Tree.* Sedangkan menurut [10], *Synthetic Minority Oversampling Technique* atau SMOTE adalah metode untuk menyeimbangkan data pada kelas minoritas sebanyak data pada kelas mayoritas.

Penelitian ini menggunakan algoritma *Extra Trees* dengan *Class ExtraTreesClassifier* dan metode *oversampling* SMOTE dengan *Class SMOTE*. Parameter yang digunakan adalah parameter secara *default* yang akan menyesuaikan dengan pembelajaran pada data latih dengan nilai *random_state* adalah 42 sebagai parameter untuk menginisialisasi generator angka acak sebagai 42. Pembentukan model klasifikasi dibagi menjadi dua skenario menggunakan kombinasi rasio antara data latih data uji pada setiap skenario. Skenario yang pertama adalah pembuatan model dengan algoritma *Extra Trees* tanpa metode *oversampling* SMOTE sedangkan skenario yang kedua adalah pembuatan model dengan algoritma *Extra Trees* menggunakan metode *oversampling* SMOTE.

2.9 Evaluasi Hasil

Evaluasi hasil merupakan tahapan untuk menguji model *Machine Learning* yang telah dibuat dengan algoritma *Extra Trees* menggunakan metode *oversampling* SMOTE dan algoritma *Extra Trees* tanpa metode *oversampling* SMOTE. Peneliti menggunakan *Confusion Matrix* untuk menghitung nilai akurasi, *precision* dan *recall* pada setiap model *Machine Learning* yang telah dibuat sebagai nilai hasil evaluasi model tersebut. Menurut [11], *Confusion Matrix* adalah suatu metode yang umumnya digunakan untuk melakukan perhitungan tingkat akurasi sebuah model. Hasil evaluasi dari pemodelan algoritma yang telah dilakukan dapat ditampilkan menggunakan *Confusion Matrix.*

2.10 Perbandingan Evaluasi Hasil

Tahapan ini dilakukan untuk membandingkan hasil evaluasi model *Machine Learning* yang menggunakan algoritma *Extra Trees* dengan metode *oversampling* SMOTE dan algoritma *Extra Trees* tanpa metode *oversampling* SMOTE. Pertimbangan hasil evaluasi didasarkan pada nilai akurasi, *precision* dan *recall* pada data uji. Hasil dari tahapan ini adalah model *Machine Learning* dengan nilai akurasi, *precision*, dan *recall* data uji yang terbaik.

3 Hasil Penelitian

Data yang digunakan memiliki enam kolom dengan total 14676 data yang dimulai dari tanggal 3 Januari 2017 hingga 31 Oktober 2017 dan 1 Januari 2018 hingga 31 Desember 2018. Kolom tersebut terdiri dari kolom *tanggal*, *wilayah*, *waktu*, *cuaca*, *kelembaban_persen,* dan *suhu_derajat_celcius*. Data terdiri dari lima variabel independen sebagai fitur dan satu variabel dependen sebagai target untuk permasalahan klasifikasi. Fitur yang merupakan variabel independen terdiri dari kolom *tanggal*, *waktu*, *wilayah*, *kelembaban_persen* dan *suhu_derajat_celcius* sedangkan target yang merupakan variabel dependen adalah kolom *cuaca*. Penjelasan lebih rinci mengenai masing-masing kolom dijelaskan pada Tabel 1 sedangkan informasi mengenai data yang digunakan dapat dilihat pada Tabel 2.

Tabel. 1. Keterangan Setiap Kolom pada Data Cuaca Provinsi DKI Jakarta.

No	Kolom	Keterangan
1	*tanggal*	Tanggal pengambilan data cuaca pada Provinsi DKI Jakarta yang terdiri dari tahun 2017 hingga 2018
2	*wilayah*	Tempat saat pengambilan data cuaca dilakukan
3	*waktu*	Waktu saat pengambilan data cuaca dilakukan
4	*cuaca*	Cuaca hasil pengamatan pada Provinsi DKI Jakarta tahun 2017 hingga 2018 pada tanggal, wilayah dan waktu pengambilan data
5	*kelembaban_persen*	Tingkat kelembaban udara pada wilayah tersebut ketika pengamatan data cuaca dilakukan dalam bentuk persentase
6	*suhu_derajat_celcius*	Suhu udara pada wilayah tersebut ketika pengamatan data cuaca dilakukan dalam bentuk derajat celcius

Tabel. 2. Data Cuaca Provinsi DKI Jakarta.

index	*tanggal*	*wilayah*	*waktu*	*cuaca*	*kelembaban_persen*	*suhu_derajat_celcius*
0	2017-01-03	Jakarta Barat	Malam	Hujan Ringan	65 – 95	22 – 31
1	2017-01-03	Jakarta Barat	Pagi	Hujan Ringan	65 – 95	22 – 31
2	2017-01-03	Jakarta Barat	Siang	Hujan Petir	65 – 95	22 – 31
…	…	…	…	…	…	…
14672	2018-12-31	Kepulauan Seribu	Dini Hari	Hujan Ringan	70 – 90	24 – 32
14673	2018-12-31	Kepulauan Seribu	Malam	Hujan Lokal	70 – 90	24 – 32
14674	2018-12-31	Kepulauan Seribu	Pagi	Hujan Lokal	70 – 90	24 – 32
14675	2018-12-31	Kepulauan Seribu	Siang	Hujan Lokal	70 – 90	24 – 32

3.1 *Import Library Package*

Library Package digunakan untuk membantu peneliti selama penelitian klasifikasi cuaca Provinsi DKI Jakarta berlangsung. *Library Package* yang digunakan terdiri dari Pustaka *Pandas*, *Matplotlib*, *Seaborn*, *Regular Expression*, *Scientific Python*, *Scikit-Learn*, *Imbalanced-Learn*, *PyDotPlus*, dan *Graph Visualization Software*.

3.2 Praproses Data

3.2.1 *Missing Value*

Tahapan praproses yang pertama adalah memeriksa *Missing Value* atau nilai yang hilang pada *dataset* yang digunakan. Berdasarkan hasil analisis didapatkan informasi bahwa, pada data cuaca Provinsi DKI Jakarta tidak memiliki *Missing Value* sehingga tidak diperlukan proses untuk menangani data yang memiliki *Missing Value*.

3.2.2 Data Duplikat

Tahapan praproses yang kedua adalah memeriksa apakah terdapat data yang duplikat pada data cuaca Provinsi DKI Jakarta. Berdasarkan hasil pemeriksaan terhadap data cuaca Provinsi DKI Jakarta didapatkan bahwa, pada data cuaca Provinsi DKI Jakarta tidak terdapat data yang duplikat sehingga tidak diperlukan proses untuk menangani data duplikat.

3.2.3 Memperbaiki Kesalahan Ketik

Tahapan praproses yang ketiga adalah memperbaiki kesalahan ketik pada kolom kategorik. Sebelum memperbaiki kesalahan ketik, peneliti menganalisis terlebih dahulu setiap kategori pada kolom kategorik pada *dataset* cuaca Provinsi DKI Jakarta. Berdasarkan hasil analisis tersebut didapatkan bahwa, pada kolom *waktu* dan *cuaca* terdapat jenis kategori yang berulang dan memiliki kesalahan ketik. Oleh karena itu, tahapan ini akan memperbaiki kategori yang berulang dan kesalahan ketik tersebut untuk menyesuaikan jenis kategori pada kolom *waktu* dan *cuaca* agar lebih efektif.

Proses yang pertama adalah memperbaiki kesalahan ketik pada kolom *waktu* yang pada awalnya memiliki delapan jenis kategori waktu yaitu ”Pagi”, “Siang”, “Malam”, “Dini Hari”, “malam”, “pagi”, “siang” dan “dini hari”. Tahapan memperbaiki kesalahan ketik pada kolom *waktu* menggunakan fungsi *re* atau *Regular Expression* yang bertujuan untuk mencari dan memperbaiki kategori yang memiliki kesalahan ketik agar dapat diubah menjadi benar dan sesuai. Proses tersebut menyesuaikan dan mengubah kategori “malam” menjadi "Malam", “pagi” menjadi "Pagi", “siang” menjadi "Siang" dan “dini hari” menjadi "Dini Hari". Hasil dari tahapan ini berhasil mengurangi jumlah kategori waktu dari delapan menjadi empat kategori yaitu Pagi, Siang, Malam, dan Dini Hari.

Proses yang kedua adalah memperbaiki kesalahan ketik pada kolom *cuaca* yang pada awalnya memiliki 32 jenis kategori cuaca. Tahapan memperbaiki kesalahan ketik pada kolom *cuaca* menggunakan fungsi *map* yang bertujuan untuk memperbaiki penulisan setiap kategori cuaca agar benar dan sesuai. Hasil tahapan tersebut berhasil mengurangi jumlah kategori cuaca dari 32 menjadi 8 kategori cuaca yaitu Cerah, Cerah Berawan, Berawan, Berawan Tebal, Hujan Lokal, Hujan Ringan, Hujan Sedang dan Hujan Petir. Data dengan kategori “Udara Kabur” tidak sesuai dengan hierarki kategori cuaca yang digunakan sehingga data tersebut dibuang dan jumlah data berkurang menjadi 14667 data.

3.2.4 *Feature Construction*

Tahapan praproses yang keempat adalah *Feature Construction* yang bertujuan untuk memproses kolom numerik yaitu kolom *kelembaban_persen* dan *suhu_derajat_celcius*. Tahapan ini bekerja dengan membagi nilai interval pada kolom tersebut menjadi kolom baru untuk menampung nilai maksimal dan minimal. Kolom baru yang terbentuk yaitu *kelembaban_min* dan *suhu_min* untuk menampung nilai pada interval minimal sedangkan kolom *kelembaban_max* dan *suhu_max* untuk menampung nilai pada interval maksimal. Setelah itu, peneliti membuang kolom *kelembaban_persen* dan *suhu_derajat_celcius* karena informasi kedua kolom tersebut sudah dicakup oleh kolom baru yang telah dibuat.

3.2.5 Memperbaiki Tipe Data

Tahapan praproses yang kelima adalah melihat serta memperbaiki tipe data yang tidak sesuai pada setiap kolom. Pada tahapan ini peneliti memperbaiki tipe data sesuai dengan nilai dari setiap kolom agar dapat diproses oleh model *Machine Learning*. Tahapan tersebut memperbaiki kesalahan tipe data pada kolom *tanggal*, *kelembaban_min*, *kelembaban_max, suhu_min* dan *suhu_max* karena nilai data dari kolom tersebut tidak sesuai dengan tipe datanya sehingga diperlukan perbaikan tipe data. Kolom *kelembaban_min, kelembaban_max, suhu_min* dan *suhu_max* seharusnya bertipe data *integer* karena memiliki nilai numerik pada datanya sedangkan kolom *tanggal* seharusnya bertipe data *datetime* karena berisi tanggal pengambilan data.

3.2.6 *Feature Selection*

Tahap praproses yang keenam adalah *Feature Selection* untuk memilih fitur apa saja yang digunakan pada saat pemodelan menggunakan metode *Wrapper*. Proses seleksi fitur dilakukan dengan memilih fitur yang berpengaruh terhadap hasil klasifikasi menjadi lebih baik serta fitur yang dapat diproses oleh model *Machine Learning*. Hasil dari tahapan ini mengurangi fitur menjadi tujuh yaitu kolom *waktu*, *kelembaban_min*, *kelembaban_max*, *kelembaban_mean*, *suhu_min*, *suhu_max*, dan *suhu_mean* sebagai kolom yang digunakan untuk pemodelan.

3.2.7 Penyesuaian Data Cuaca Provinsi DKI Jakarta

Tahapan praproses yang ketujuh adalah penyesuaian data cuaca Provinsi DKI Jakarta pada setiap kategori cuaca yang mewakili kategori waktu dengan jumlah data terbanyak. Kolom *waktu* merupakan satu-satunya kolom bertipe kategorik sehingga kolom *waktu* dijadikan sebagai kolom untuk mewakili setiap kategori cuaca. Tahapan ini mengurangi jumlah data dari 14667 menjadi 5039 sehingga setiap kategori cuaca akan mewakili masing-masing kategori waktu dengan jumlah data terbanyak.

3.2.8 *Encoding* dengan Membuat *Dummy Variable*

Tahapan praproses yang terakhir adalah *Encoding* dengan membuat *Dummy Variable*. Peneliti melakukan tahapan *Encoding* menggunakan metode *One-Hot Encoding* yaitu dengan membuat *Dummy Variable* pada kolom *waktu* karena merupakan kolom kategorik. Tahapan *Encoding* dilakukan menggunakan fungsi *get_dummies*. Tahapan ini menghasilkan kolom baru yang berisi bilangan *biner* yang mewakili setiap kategori pada kolom *waktu*.

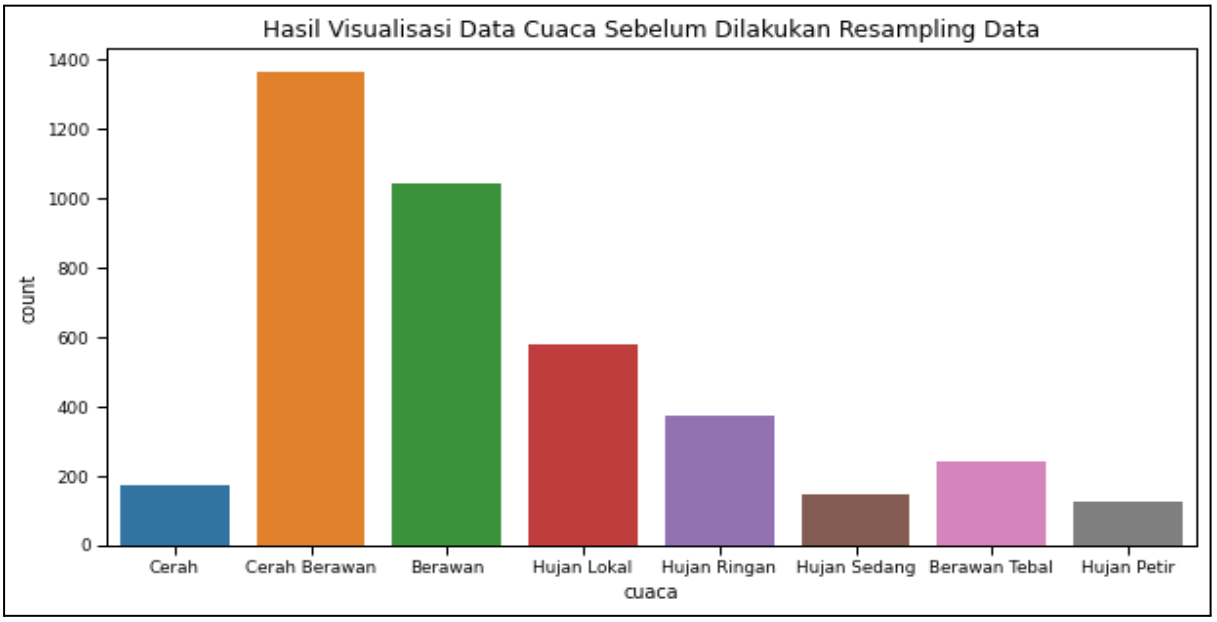
3.3 Pembagian Data

Pada tahap ini, data dibagi menjadi kolom fitur dan kolom target. Kolom fitur disimpan pada variable *X* dan kolom target disimpan pada variable *y*. Kolom *cuaca* menjadi kolom target, sedangkan kolom selain *cuaca* menjadi kolom fitur. Selanjutnya, data dibagi menjadi data latih dan data uji menggunakan fungsi *train_test_split* dengan empat rasio berbeda yaitu 90% data latih 10% data uji, 80% data latih 20% data uji, 70% data latih 30% data uji, dan 60% data latih 40% data uji. Tujuannya adalah untuk mencari rasio yang sesuai berdasarkan hasil evaluasi terbaik.

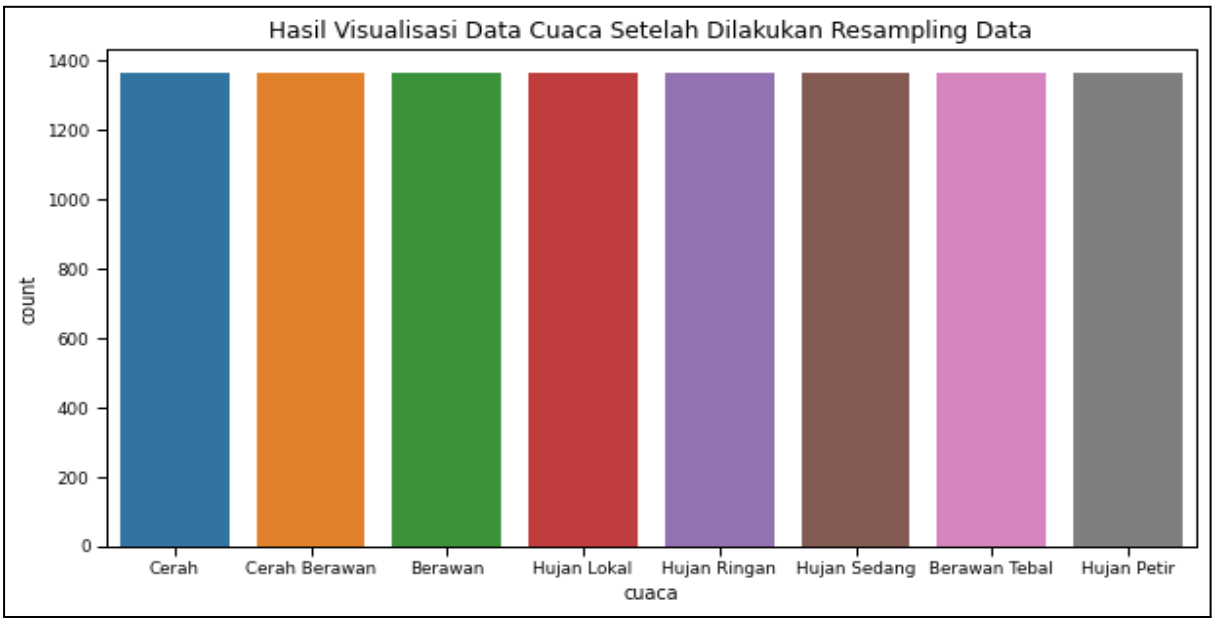
3.4 *Resampling* Data Menggunakan *Oversampling* SMOTE

Tahapan selanjutnya adalah *resampling* data menggunakan metode *oversampling* SMOTE. Sebelum tahapan *resampling* data dilakukan, peneliti melakukan analisis terlebih dahulu terhadap kolom *cuaca* dengan tujuan untuk mengetahui perbandingan jumlah data pada setiap kategori cuaca. Berdasarkan analisis, diperoleh cuaca Cerah Berawan memiliki persentase jumlah data sebesar 33,9%, Berawan sebesar 25,8%, Hujan Lokal sebesar 14,3%, Hujan Ringan sebesar 9,2%, Berawan Tebal sebesar 5,9%, Cerah sebesar 4,2%, Hujan Sedang sebesar 3,6%, dan Hujan Petir sebesar 3,0% dari total keseluruhan data. Hasil tersebut menunjukan bahwa data yang digunakan tidak seimbang sehingga diperlukan metode *oversampling* SMOTE untuk menyeimbangkan data.

Berikut merupakan hasil visualisasi persebaran data pada kolom *cuaca* berdasarkan rasio 80% data latih sebelum dilakukan metode *resampling* data menggunakan *oversampling* SMOTE yang dapat dilihat pada Gambar 2 dan sesudah dilakukan metode *resampling* data menggunakan *oversampling* SMOTE yang dapat dilihat pada Gambar 3.



Gambar. 2. Visualisasi kolom *cuaca* sebelum dilakukan metode resampling data menggunakan *oversampling* SMOTE pada rasio 80% data latih.



Gambar. 3. Visualisasi kolom *cuaca* sesudah dilakukan metode *resampling* data menggunakan *oversampling* SMOTE pada rasio 80% data latih.

3.5 Pemodelan

Tahapan selanjutnya adalah pembuatan model menggunakan algoritma *Extra Trees* dan metode *oversampling* SMOTE untuk klasifikasi cuaca Provinsi DKI Jakarta. Penelitian ini menggunakan empat rasio untuk pembagian data latih dan data uji. Pada setiap rasio tersebut, peneliti mencoba mengimplementasikan algoritma *Extra Trees* tanpa metode *oversampling* SMOTE dan *Extra Trees* menggunakan metode *oversampling* SMOTE. Oleh karena itu, pada penelitian ini terdapat delapan skenario pembuatan model *Machine Learning* yang terdiri dari pemodelan menggunakan algoritma *Extra Trees* tanpa metode *oversampling* SMOTE dan algoritma *Extra Trees* dengan metode *oversampling* SMOTE pada setiap rasio pembagian data.

3.6 Evaluasi Hasil

Tahapan selanjutnya adalah melihat evaluasi hasil setiap model *Machine Learning* yang telah dibuat dengan tujuan untuk mengevaluasi performa setiap model dalam melakukan klasifikasi terhadap cuaca Provinsi DKI Jakarta. Tahapan evaluasi hasil menggunakan *Confusion Matrix* untuk menghasilkan nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN) untuk menghitung nilai akurasi, *precision* dan *recall*. Setiap model menghasilkan nilai TP atau data yang benar diprediksi sebagai positif, nilai FP atau data yang salah diprediksi sebagai positif, nilai FN atau data yang salah diprediksi sebagai negatif dan nilai TN atau data yang benar diprediksi sebagai negatif pada setiap kategori cuaca Berawan, Berawan Tebal, Cerah, Cerah Berawan, Hujan Lokal, Hujan Petir, Hujan Ringan dan Hujan Sedang. Informasi mengenai hasil evalusi model dengan algoritma *Extra Trees* menggunakan metode *oversampling* SMOTE dapat dilihat pada Tabel 3 sedangkan model dengan algoritma *Extra Trees* tanpa metode *oversampling* SMOTE dapat dilihat pada Tabel 4.

Tabel. 3. Hasil Evaluasi Model Menggunakan Metode *Oversampling* SMOTE

No	Hasil Evaluasi	Model 90:10 SMOTE	Model 80:20 SMOTE	Model 70:30 SMOTE	Model 60:40 SMOTE
1	Akurasi Data Latih	84,6%	85,0%	86,1%	87,1%
2	*Precision* Data Latih	84,1%	85,4%	86,4%	87,5%
3	*Recall* Data Latih	84,6%	85,0%	86,1%	87,1%
4	Akurasi Data Uji	71,8%	71,9%	72,8%	74,4%
5	*Precision* Data Uji	55,1%	54,6%	56,4%	56,1%
6	*Recall* Data Uji	62,2%	61,6%	63,6%	61,9%

Tabel. 4. Hasil Evaluasi Model Tanpa Metode *Oversampling* SMOTE

No	Hasil Evaluasi	Model 90:10	Model 80:20	Model 70:30	Model 60:40
1	Akurasi Data Latih	83,0%	83,1%	83,4%	83,5%
2	*Precision* Data Latih	72,2%	72,7%	72,7%	74,3%
3	*Recall* Data Latih	62,2%	62,2%	63,1%	62,5%
4	Akurasi Data Uji	79,4%	79,8%	78,9%	79,6%
5	*Precision* Data Uji	65,1%	63,1%	61,4%	62,4%
6	*Recall* Data Uji	56,9%	56,1%	57,4%	56,3%

3.7 Perbandingan Evaluasi Hasil

Berdasarkan Tabel 3 dan Tabel 4 diperoleh hasil evaluasi model *Machine Learning* menggunakan algoritma *Extra Trees* dengan metode *oversampling* SMOTE dan algoritma *Extra Trees* tanpa metode *oversampling* SMOTE. Hasil evaluasi tersebut menunjukkan bahwa model tanpa metode *oversampling* SMOTE memiliki nilai evaluasi hasil akurasi dan *precision* yang lebih baik dari pada model yang menggunakan metode *oversampling* SMOTE pada setiap rasio pembagian data. Namun, metode yang menggunakan *oversampling* SMOTE memiliki nilai *recall* yang lebih baik dari pada model tanpa metode *oversampling* SMOTE. Hal tersebut disebabkan oleh

metode *oversampling* SMOTE bekerja pada data latih saja. Data latih digunakan hanya untuk melatih sebuah model sedangkan data uji digunakan untuk menguji model yang dibuat. Oleh karena itu, dalam pengukuran performa sebuah model hanya mempertimbangkan hasil evaluasi pada data uji.

4 Kesimpulan

4.1 Kesimpulan

Berdasarkan analisis model Machine Learning pada klasifikasi cuaca Provinsi DKI Jakarta menggunakan algoritma Extra Trees, diperoleh bahwa model dengan oversampling SMOTE menghasilkan performa terbaik pada rasio 60% data latih dan 40% data uji sedangkan model tanpa metode oversampling SMOTE menghasilkan performa terbaik pada rasio 80% data latih dan 20% data uji.

a) Model yang menggunakan metode oversampling SMOTE pada rasio 60% data latih dan 40% data uji memiliki nilai akurasi sebesar 74,4%, precision 56,4%, dan recall 63,6%.

b) Model tanpa metode oversampling SMOTE pada rasio 80% data latih dan 20% memiliki nilai akurasi sebesar 79,8%, precision 63,1%, dan recall 56,1%.

Metode *oversampling* SMOTE berhasil menangani permasalahan ketidakseimbangan data pada kelas minoritas. Lalu, berdasarkan nilai *recall*, metode *oversampling* SMOTE memberikan hasil yang lebih baik dalam memberikan informasi mengenai seberapa baik model dalam memprediksi data dengan benar. Metode *oversampling* SMOTE juga tidak memerlukan persentase data latih yang besar karena metode ini sudah menghasilkan data latih sintetis yang menyebabkan jumlah data latih bertambah

4.2 Saran

Berdasarkan hasil analisis dari penelitian yang telah dilakukan, dapat diberikan beberapa saran yang dapat dilakukan untuk penelitian selanjutnya dengan melakukan percobaan penerapan algoritma lain dari metode *Ensemble Learning* untuk membandingkan hasil evaluasi yang diperoleh dan mengetahui apakah penerapan metode *oversampling* SMOTE juga mempengaruhi hasil evaluasi terhadap algoritma tersebut. Serta melakukan percobaan penerapan metode Resampling data lainnya untuk menentukan metode yang lebih optimal dalam melakukan klasifikasi data cuaca Provinsi DKI Jakarta.

Referensi

[1] B. Santoso, A. I. S. Azis, and Zohrahayaty. (2020). *Machine Learning & Reasoning Fuzzy Logic Algoritma, Manual, Matlab, & Rapid Miner*. DEEPUBLISH

[2] A. Luthfiarta, A. Febriyanto, H. Lestiawan, and W. Wicaksono. (2020). Analisa Prakiraan Cuaca dengan Parameter Suhu, Kelembaban, Tekanan Udara, dan Kecepatan Angin Menggunakan Regresi Linear Berganda. *JOINS (Journal of Information System)*, vol. 5, no. 1, pp. 10–17. doi: 10.33633/joins.v5i1.2760.

[3] G. A. Mursianto, I. M. Falih, M. Irfan, T. Sakinah, and D. Sandya. (2021). Perbandingan Metode Klasifikasi Random Forest dan XGBoost Serta Implementasi Teknik SMOTE pada Kasus Prediksi Hujan. *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)*.

[4] F. Hamami and A. Dahlan. (2022). Klasifikasi Cuaca Provinsi Dki Jakarta Menggunakan Algoritma Random Forest Dengan Teknik Oversampling.

[5] P. Shi and Z. Wang. (2021). An Ensemble Tree Classifier for Highly Imbalanced Data Classification. *J Syst Sci Complex*, vol. 34, no. 6, pp. 2250–2266. doi: 10.1007/s11424-021-1038-8.

[6] Ibnu Daqiqil. (2021). *MACHINE LEARNING: Teori, Studi Kasus dan Implementasi Menggunakan Python*, 1st ed. UR PRESS.

[7] M. Yunus, M. Husni, and M. M. Mufadhdha. (2021). Klasifikasi Sentimen Terhadap Badan Penyelenggara Jaminan Sosial (BPJS) Pada Media Sosial Twitter Menggunakan Naive Bayes. *SMATIKA JURNAL*, vol. 11, no. 02, pp. 81–91. doi: 10.32664/smatika.v11i02.577.

[8] B. Suma. (2020). Implementasi Machine Learning Di Dalam Prediksi Cuaca. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 4, pp. 648–654. doi: 10.13140/RG.2.2.16086.47680.

[9] S. Khomsah and Agus Sasmito Aribowo. (2020). Text-Preprocessing Model Youtube Comments in Indonesian. doi: 10.29207/resti.v4i4.2035.

[10] Y. A. Sir and A. H. H. Soepranoto. (2022). Pendekatan Resampling Data Untuk Menangani Masalah Ketidakseimbangan Kelas. *Jurnal Komputer dan Informatika*, vol. 10, no. 1, pp. 31–38. doi: 10.35508/jicon.v10i1.6554.

[11] E. V. Rahcmadani, N. H. Harani, and S. F. Pane. (2020). Algoritma C4.5 dan K-Nearest Neighbors (KNN) untuk Memetakan Matakuliah dan Keterlambatan Kelulusan Mahasiswa, 1st ed. Kreatif.