e-ISSN 2962-6129

Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)

Jakarta-Indonesia, 20 Agustus 2022

Perbandingan Algoritma Klasifikasi Random Forest dan Extreme Gradient Boosting pada Dataset Cuaca Provinsi DKI Jakarta Tahun 2018

Agung Hot Iman [1], Fransisco Ready Permana [2], Gito Putro Wardana [3], Raihan Kemmy Rachmansyah [4], Mayanda Mega Santoni [5]
Program Studi Informatika / Fakultas Ilmu Komputer
Universitas Pembangunan Nasional Veteran Jakarta
Jl. RS. Fatmawati Raya, Pd. Labu, Kec. Cilandak, Kota Depok, Daerah Khusus Ibukota Jakarta 12450
agunghi@upnvj.ac.id [1], fransiscorp@upnvj.ac.id [2], gitopw@upnvj.ac.id [3], raikemmy@upnvj.ac.id [4], megasantoni@upnvj.ac.id [5]

Abstrak. Perkembangan teknologi semakin berkembang pesat yang menyebabkan seluruh aktivitas kehidupan manusia tidak terlepas dari penggunakan teknologi. Teknologi *machine learning* merupakan teknologi yang mengembangkan kecerdasan buatan agar dapat belajar dengan sendirinya dengan menerapkan ilmu seperti statistika, matematika serta *data mining*. Pada penelitian ini peneliti ingin membandingkan performa algoritma antara *Random Forest* dan *Extreme Gradient Boosting* pada data cuaca Provinsi DKI Jakarta tahun 2018. Setelah melakukan tahap *preprocessing*, *exploratory data analysis*, *resampling* pada label, dan melatih data untuk mendapatkan hasil akurasi yang nantinya digunakan sebagai pembanding antara algoritma *Extreme Gradient Boosting* dengan *Random Forest*. Berdasarkan hasil penelitian, peneliti menyimpulkan bahwa algoritma *Random Forest* lebih baik daripada *Extreme Gradient Boosting* jika digunakan pada data cuaca Provinsi DKI Jakarta tahun 2018 karena waktu pemrosesan yang lebih cepat dan akurasi yang dihasilkan cukup tinggi yaitu 68%.

Kata Kunci: *Machine Learning*, *Random Forest*, *Extreme Gradient Boosting*, Cuaca

1 Pendahuluan

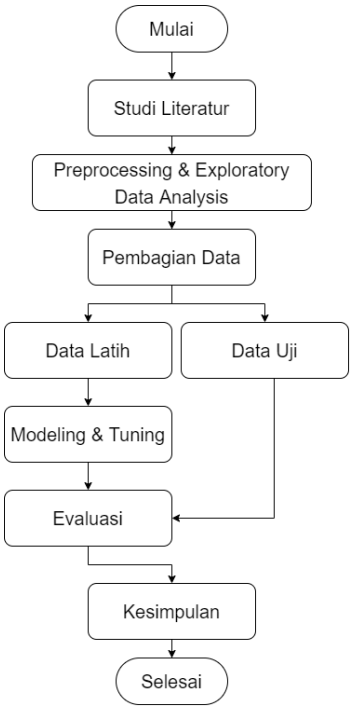
Machine learning merupakan teknologi yang bertujuan untuk mengembangkan mesin agar dapat belajar dengan sendirinya. *Machine learning* memiliki dua pendekatan yaitu *supervised learning* dan *unsupervised learning* yang keduanya bertujuan untuk menghasilkan wawasan dari sebuah data berdasarkan kecerdasan buatan yang telah di program. Salah satu pendekatan yang dapat ditangani oleh *supervised learning* adalah klasifikasi. Klasifikasi merupakan metode yang bertujuan untuk mengelompokan suatu data berdasarkan data yang telah dilatih sebelumnya.

Pada *machine learning* juga terdapat istilah *ensemble learning* pada *machine learning* yang menggunakan kombinasi dari beberapa algoritma untuk mendapatkan hasil yang lebih baik dengan tiga metode yaitu *bagging*, *boosting* dan *stacking* [1]. *Random Forest* dan *Extreme Gradient Boosting* merupakan algoritma yang menerapkan *ensemble learning*. Oleh karena itu, pada penelitian kali ini peneliti ingin membandingkan hasil akurasi algoritma *Random Forest* dan *Extreme Gradient Boosting* berdasarkan klasifikasi cuaca Provinsi DKI Jakarta pada tahun 2018. Penelitian ini dilakukan karena cuaca pada provinsi tersebut yang tidak menentu, hal ini mengakibatkan banyak kegiatan terganggu akibat perubahan cuaca secara mendadak. Peneliti juga ingin membuat sebuah model yang lebih baik diantara *Random Forest* dengan *Extreme Gradient Boosting* pada data cuaca Provinsi DKI Jakarta tahun 2018.

Terdapat beberapa penelitan sebelumnya yang terkait dengan penelitian ini seperti penelitian yang pernah dilakukan oleh Mursianto, dkk. 2021 berjudul "Perbandingan Metode Klasifikasi *Random Forest* dan XGBoost Serta Implementasi Teknik SMOTE pada Kasus Prediksi Hujan" [2]. Penelitian tersebut membuat sebuah model *machine learning* untuk mengelompokkan cuaca berdasarkan data WeatherAUS yang diperoleh dari situs Kaggle menggunakan algoritma *Random Forest* dan *Extreme Gradient Boosting*. Luaran dari penelitian tersebut adalah sebuah model dengan tingkat akurasi tertinggi sebesar 94.34%. Penelitian lainnya pernah dilakukan oleh Faqih Hamami dengan judul "Klasifikasi Cuaca Provinsi DKI Jakarta Menggunakan Algoritma Random Forest Dengan

Teknik Oversampling" [3]. Penelitian ini cukup mirip dengan penelitian yang dilakukan oleh penelitian kali ini karena data yang digunakan adalah dataset cuaca provinsi DKI Jakarta tahun 2018. Namun, perbedaannya terdapat pada algoritma yang digunakan, penelitian yang dilakukan oleh Faqih hanya menggunakan algoritma *Random Forest* sedangkan penelitian kali ini peneliti ingin membandingkan algoritma yang lebih baik antara *Random Forest* dan *Extreme Gradient Boosting* terhadap data yang dimiliki. Luaran dari penelitian yang dilakukan oleh Faqih adalah sebuah model *machine learning* dengan tingkat akurasi sebesar 70%. Penelitian lainnya juga pernah dilakukan oleh Budi, dkk yang berjudul "Klasifikasi Cuaca Menggunakan Metode Convolutional Neural Network (CNN)" [4]. Penelitian ini melakukan klasifikasi cuaca beradasarkan citra yang dimulai dengan melakukan *resizing* citra menjadi 224x224 kemudian dilanjutkan dengan *feature extraction* dan melatih data. Luaran dari penelitian ini adalah sebuah model machine learning dengan tingkat akurasi 87% yang diperoleh dari nilai *learning rate* 0.0001 pada epoch ke 75.

2. Metode Penelitian



Gambar. 1. Tahapan Metode Penelitian

2.1 Studi Literatur

Pada tahapan ini peneliti melakukan studi literatur untuk mendapatkan teori-teori yang valid terkait dengan penelitian yang dilakukan sebelumnya. Sumber literatur yang digunakan untuk dijadikan referensi antara lain jurnal, artikel, dan situs internet.

2.2 *Input Dataset*

Data yang digunakan untuk penelitian ini diambil dari *open dataset* Indonesia pada website https://katalog.data.go.id/dataset/data-prakiraan-cuaca-wilayah-provinsi-dki-jakarta-tahun-2018.

2.3 *Preprocessing Data* dan *Exploratory Data Analysis*

Tahapan selanjutnya, peneliti melakukan pra-proses dan *exploratory data analysis* pada data yang telah peroleh. Tahapan ini bertujuan untuk membersihkan data dari derau (*noise*), mendapatkan lebih banyak *insight* dari data dengan *exploratory data analysis*, dan mengekstrak informasi yang ada pada data dengan melakukan *feature engineering*. Berikut adalah beberapa hal yang peneliti lakukan pada tahapan ini, yaitu:

2.3.1 Melihat Jumlah Baris dan Kolom

Melihat jumlah keseluruhan baris dan kolom dari dataset menggunakan fungsi .shape.

2.3.2 Memproses *Missing Values*

Salah satu *noise* yang terdapat pada data ini adalah missing value. Melihat apakah terdapat data yang kosong dan menangani data kosong tersebut dengan cara menghapus data tersebut menggunakan fungsi dropna(). Setelah dilakukan penghapusan data pada *missing value*, dapat dilakukan pengecekan kembali untuk melihat apakah masih terdapat *missing value* atau tidak.

2.3.3 Memproses Kolom Cuaca dan Waktu

Tahapan ini bertujuan untuk membersihkan dan mengelompokkan data pada kolom cuaca dan waktu agar data lebih mudah untuk diproses pada tahap selanjutnya. Pada kolom cuaca akan mengelompokkan data menjadi 3 yaitu Cerah, Hujan dan Berawan sedangkan pada kolom waktu mengelompokkan data menjadi 4 yaitu Pagi, Siang, Malam dan Dini Hari. Proses ini dilakukan menggunakan fungsi *Regular Expression*.

2.3.4 Memproses Kolom kelembaban_persen dan suhu_derajat_celcius

Tahapan ini bertujuan untuk membagi data pada kolom kelembaban_persen dan suhu_derajat_celcius menjadi 2 bagian yaitu data minimal dan maksimal. Proses ini menggunakan fungsi split() untuk membagi data lalu disimpan kedalam variabel kelembaban_min dan suhu_min untuk data minimal serta kelembaban_max dan suhu_max untuk data bernilai maksimal berdasarkan kolom kelembaban_persen dan suhu_derajat_celcius.

2.3.5 Melihat dan Memperbaiki Tipe Data

Selanjutnya, kolom dataset yang masih memiliki tipe data objek tetapi memiliki nilai numerik atau angka akan diubah menjadi bertipe data integer, untuk memudahkan dalam melakukan proses *modeling*. Fitur-fitur yang diubah menjadi *integer* atau numerik yaitu, kelembaban_min, suhu_min, kelembaban_max, dan suhu_max.

2.3.6 Melihat Statistik Deskriptif

Tahapan berikutnya adalah dengan melihat statistik deskriptif pada setiap kolom untuk melihat ringkasan dari data secara keseluruhan menggunakan fungsi .describe(). Fungsi ini dapat memberikan informasi mengenai nilai rata-rata, standar deviasi dan interkuartil pada setiap kolom.

2.3.7 Menghapus Kolom

Tahapan terakhir pada pra-proses ini adalah memilih fitur yang akan digunakan dengan cara menghapus kolom yang tidak mempengaruhi proses klasifikasi menggunakan fungsi .drop(). Beberapa kolom yang dihapus dan tidak digunakan yaitu tanggal, kelembaban_persen, dan suhu_derajat_celcius.

2.3.8 Melihat Bentuk Persebaran Data

Tahap selanjutnya adalah melihat grafis visualisasi data yang bertujuan untuk menganalisis data agar lebih mudah dipahami. Proses visualisasi data yang peneliti lakukan menggunakan package Seaborn lalu memanggil fungsi .pairplot() yang nantinya akan menghasilkan grafis visualisasi dari data cuaca Provinsi DKI Jakarta tahun 2018.

2.6 Pembagian Data

Tahapan setelah *preprocessing* dan *exploratory data analysis* adalah pembagian data menjadi dua yaitu data latih dan data uji dengan perbandingan 80% data latih dan 20% data uji. Selanjutnya akan dipisahkan antara kolom label dan kolom fitur. Pada kolom label dilakukan pembagian secara stratifikasi atau pembagian data secara seimbang karena label yang digunakan bertipe kategorikal.

2.8 *Preprocessing, Modeling*, dan *Tuning*

2.8.1 *Preprocessing*

Sebelum tahapan selanjutnya, peneliti mempersiapkan data terlebih dahulu agar lebih siap untuk di proses. Peneliti melakukan tahapan pra-proses kembali setelah tahap pembagian data. Hal ini peneliti lakukan untuk menghindari kebocoran data sehingga akan mengurangi tingkat *overfitting* ataupun *underfitting*. Pada tahapan pra-proses kali ini, peneliti menggunakan *Column Transformer* dengan membagi kolom menjadi kolom numerik dan kolom kategorik. Untuk kolom numerik dilakukan pra-proses berupa *Polynomial Feature* sedangkan kolom kategorik dilakukan pra-proses berupa *One-Hot Encoder* dengan tujuan untuk membangun model yang sederhana yang dapat digunakan untuk pemahaman dan melakukan prediksi. *Polynomial Feature* adalah sebuah teknik untuk mengatasi data yang tidak linier dengan menambahkan kekuatan pada setiap fitur sebagai fitur yang baru. *Polyomial Feature* akan melatih model linier pada kumpulan fitur yang telah diperluas [5]. *One-Hot Encoder* sendiri adalah salah satu metode *encoding* yang merepresentasikan data bertipe kategori sebagai vektor biner dengan nilai integer, 0 dan 1, dimana semua elemen akan bernilai 0 kecuali satu elemen yang bernilai 1, yaitu elemen yang memiliki nilai kategori tersebut.

2.8.2 Modeling

Setelah membuat *Column Transformer*, Kemudian peneliti membuat Pipeline untuk menggabungkan *Column Transformer* yang telah dibuat sebelumnya dengan algoritma yang akan peneliti gunakan yaitu *Random Forest* dan *Extreme Gradien Boosting*. *Random Forest* adalah pengembangan dari metode pohon keputusan yang menggunakan beberapa pohon keputusan, dimana setiap pohon keputusan telah dilakukan pelatihan menggunakan sampel individu dan setiap atribut dipecah pada pohon yang dipilih antara atribut subset yang bersifat acak [6]. Sementara itu, *Extreme Gradient Boosting* adalah metode *boosting* yang menggabungkan beberapa pohon keputusan dan dibangun dengan cara membuat pohon keputusan yang baru berdasarkan residual pohon keputusan sebelumnya dengan tujuan untuk menghasilkan kesalahan prediksi (*error*) yang lebih kecil dari model sebelumnya [2]. Pipeline sendiri adalah sebuah metode yang dapat mengatur aliran data *input* dan *output* dari sebuah model atau kumpulan beberapa model *machine learning*. Salah satu keuntungan menggunakan *pipeline* adalah *pipeline* dapat memproses data mulai dari input data mentah, fitur, *output*, model *machine learning* dan parameter model, dan *output* prediksi.

2.8.3 *Tuning*

Pembuatan *Column Transformer* dan *pipeline* bertujuan untuk mempermudah proses dan memprediksi data baru dengan model telah dibuat serta memungkinkan kita dalam melakukan *tuning hyperparameter* berdasarkan algoritma yang ada di dalam pipeline. *Tuning hyperparameter* sendiri merupakan proses untuk menemukan nilai *hyperparameter* terbaik yang sesuai dengan data yang dimiliki sehingga dapat menghasilkan model dengan performa yang terbaik [7].

2.9 Evaluasi

Setelah semua tahapan selesai, selanjutnya akan dilakukan tahapan evaluasi performa dari model *Random Forest* dan *Extreme Gradient Boosting* menggunakan data uji. Evaluasi model bertujuan untuk membuat estimasi generalisasi *error* dari model yang dipilih, dan menguji seberapa baik kinerja model tersebut pada data baru yaitu data uji. Peneliti menggunakan *confusion matrix* untuk mengetahui efisiensi kinerja model agar dapat diketahui

seberapa banyak data yang salah diklasifikasi pada data uji dan *classification report* untuk mengukur tingkat akurasi, presisi, dan *recall*.

3. Hasil dan Pembahasan

Penelitian ini menggunakan data cuaca Provinsi DKI Jakarta pada tahun 2018 yang berasal dari situs katalog.data.go.id. Pada awalnya, data yang peneliti dapatkan masih terpisah berdasarkan bulan kemudian digabungkan menjadi satu file. Penjelasan lebih lanjut mengenai atribut dalam dataset cuaca Provinsi DKI Jakarta Tahun 2018 dapat dilihat pada tabel 1.

Tabel. 1. Atribut variabel dataset.

tanggal	Tanggal pengambilan data pada Provinsi DKI Jakarta tahun 2018
wilayah	Tempat pada saat pengambilan data dilakukan
waktu	Waktu pengambilan data pada saat Pagi, Siang, Malam dan lain sebagainya
cuaca	Cuaca hasil pengamatan Provinsi DKI Jakarta tahun 2018 pada tanggal tersebut
kelembaban_persen	Tingkat kelembapan udara pada wilayah tersebut ketika pengamatan data cuaca dilakukan
suhu_derajat_celcius	Suhu udara pada wilayah tersebut ketika pengamatan data cuaca dilakukan

3.1 *Pre-processing* dan *Exploratory Data Analysis*

Tahap selanjutnya peneliti melakukan pra-proses dan *exploratory data analysis* pada data yang telah peneliti peroleh. Berikut merupakan beberapa hal yang peneliti lakukan saat melakukan pra-proses, yaitu:

3.1.1 Melihat Jumlah Baris dan Kolom

Melihat seberapa besar jumlah data pada dataset yang menghasilkan total 8535 baris dan 6 kolom yaitu kolom tanggal, wilayah, waktu, cuaca, kelembapan_persen, suhu_derajat_celcius.

3.1.2 Memproses *Missing Value*

Langkah selanjutnya, peneliti mengecek apakah terdapat data yang kosong atau tidak. Pada tahap ini ditemukan bahwa data yang digunakan terdapat *noise* atau derau berupa *missing value* yang jumlahnya tidak terlalu banyak. Peneliti memutuskan untuk menghapus baris yang memiliki *missing value* tersebut dengan fungsi dropna(). Setelah itu peneliti melakukan pengecekan kembali apakah masih terdapat *missing value* menggunakan fungsi .isna().sum().

3.1.3 Memproses Kolom cuaca dan waktu

Setelah menghapus *missing value,* peneliti menggunakan fungsi *regular expression* untuk memproses kolom cuaca dan waktu kerena peneliti menemukan bahwa banyak nilai dari kolom tersebut memiliki kesalahan penulisan dan nilainya sangat luas dan beragam.

Pada kolom cuaca terdiri dari 'Hujan Lokal', 'Hujan Ringan', 'Berawan', 'Cerah Berawan', 'Cerah', 'Berawan Tebal', 'Hujan Sedang', 'Cerah Berawn', 'Cerang Berawan', 'Beawan', 'Berawan ', 'Hujan Petir', 'Hujan Lokal ','Cerah Berawan ', 'Cerah ', 'Cerah Berawah', 'Cerah ', ' Berawan', 'Hujan', 'Hujan Petir ', 'Hujan Sedang ', 'Cerah berawan',

'Hujan Ringanl', 'Berawa', 'Hujang Sedang', 'Hujan Loka', dan 'Hujan Ringan ' sedangkan pada kolom waktu terdiri dari ‘Siang’, ‘Pagi’, ‘Dini Hari’, ‘Malam’, ‘siang’, ‘pagi’, ‘malam’ dan ‘dini hari’. Dari data yang sangat luas dan beragam ini peneliti mengelompokkan data kolom cuaca menjadi 3 yaitu Cerah, Hujan, dan Berawan dan data pada kolom waktu menjadi 4 yaitu Siang, Pagi, Malam, dan Dini Hari menggunakan bantuan fungsi *regular expression*.

3.1.4 Memproses kolom kelembaban_persen dan suhu_derajat_celcius

Peneliti juga melakukan pemrosesan terhadap kolom kelembaban_persen dan suhu_derajat_celcius. Peneliti melakukan *splitting* terhadap tanda (-) dan membagi data berdasarkan nilai sebelum dan sesudah tanda (-). Kemudian membuat kolom baru yaitu kolom minimal untuk menampung nilai sebelum tanda (-) dan kolom maksimal untuk menampung nilai setelah tanda (-) sehingga terbentuk 4 kolom baru yaitu suhu_max, suhu_min, kelembaban_max, dan kelembaban_min berdasarkan kolom kelembaban_persen dan suhu_derajat_celcius.

3.1.5 Memperbaiki Tipe Data

Tahap selanjutnya peneliti memperbaiki tipe data dari setiap kolom yang belum sesuai. Peneliti menemukan bahwa semua kolom bertipe data objek termasuk kolom numerik yang akan diproses seperti suhu_max, suhu_min, kelembaban_max, dan kelembaban_min sehingga diperlukan perbaikan tipe data menjadi integer agar kolom tersebut dapat diolah dan diproses untuk tahap selanjutnya. Fungsi yang digunakan untuk melihat tipe data setiap kolom adalah fungsi .dtypes() dan untuk mengubah tipe data dari setiap kolom yang tidak sesuai peneliti menggunakan fungsi .astype(). Berikut adalah hasil yang peneliti dapatkan dari fungsi .dtypes() sebelum dan setelah dilakukan pengubahan tipe data:

Tabel. 2. Perbandingan sebelum dan sesudah perbaikan tipe data

Sebelum perbaikan tipe data

```
tanggal                object
wilayah                object
waktu                  object
cuaca                  object
kelembaban_persen      object
suhu_derajat_celcius   object
kelembaban_min         object
kelembaban_max         object
suhu_min               object
suhu_max               object
dtype: object
```

Sesudah perbaikan tipe data

```
tanggal                object
wilayah                object
waktu                  object
cuaca                  object
kelembaban_persen      object
suhu_derajat_celcius   object
kelembaban_min          int64
kelembaban_max          int64
suhu_min                int64
suhu_max                int64
dtype: object
```

3.1.6 Melihat Statistik Deskriptif

Tahap selanjutnya adalah melihat statistik seperti nilai maksimum, nilai minimum, jumlah, standar deviasi, median, kuartil pertama pada kolom numerik dan nilai unique serta frekuensi dari kolom kategorik. Selain untuk melihat statistik dari data, tahapan ini juga bertujuan untuk melihat apakah ada anomali dari data yang perlu untuk diperbaiki. Untuk melakukan tahapan ini, peneliti menggunakan fungsi .describe() dan berikut adalah hasil dari proses yang peneliti dapatkan:

Tabel. 3. Tabel *output* dari fungsi .describe()

Statistik pada data kategorik

	cuaca	waktu	wilayah	tanggal
count	8398	8398	8398	8398
unique	3	4	6	348
top	Cerah	Siang	Jakarta Utara	2018-01-02
freq	3796	2100	1400	48

Statistik pada data numerik

	kelembaban_min	kelembaban_max	suhu_min	suhu_max
count	8398.000000	8398.000000	8398.000000	8398.000000
mean	62.254704	91.727792	23.548464	32.608597
std	10.592668	5.562658	0.775947	1.246190
min	35.000000	75.000000	20.000000	28.000000
25%	55.000000	90.000000	23.000000	32.000000
50%	65.000000	95.000000	24.000000	33.000000
75%	70.000000	95.000000	24.000000	33.000000
max	85.000000	100.000000	26.000000	35.000000

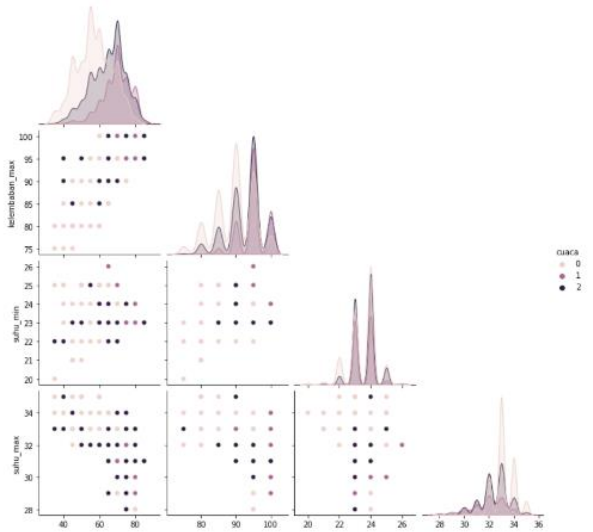
3.1.7 Menghapus Kolom

Tahapan terakhir pada tahap pra-proses adalah memakai fungsi .drop() untuk menyaring serta membuang kolom yang tidak digunakan untuk tahapan selanjutnya. Kolom yang dibuang pada tahap ini adalah kolom tanggal, kelembaban_persen, dan suhu_derajat_celcius.

3.1.8 Melihat Bentuk Persebaran Data

Tahapan selanjutnya pada *Exploratory Data Analysis* adalah membuat visualisasi persebaran data. Tujuan dibuat visualisasi persebaran data tidak hanya untuk melihat bentuk persebaran dari data yang dimiliki, tetapi juga untuk melihat korelasi dari setiap kolom yang ada pada data tersebut. Selain itu, dengan dibuatnya visualisasi persebaran data, dapat memberikan wawasan kepada peneliti mengenai algoritma apa yang sesuai untuk digunakan pada data yang peneliti miliki.

Peneliti menggunakan bantuin fungsi .pairplot() dari *package seaborn* dalam pembuatan visualisasi persebaran data. Berdasarkan hasil dari visualisasi tersebut, peneliti menemukan bahwa data peneliti memiliki persebaran yang cukup acak. Oleh karena itu, penggunaan algoritma *Extreme Gradient Boosting* dan *Random Forest* adalah pilihan yang baik untuk menangani kasus tersebut. Berikut merupakan persebaran dari data yang peneliti:



Gambar. 2. Bentuk Persebaran Data serta korelasi antar tiap kolom

3.3 Pembagian Data

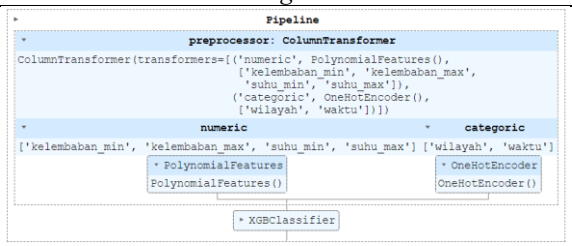
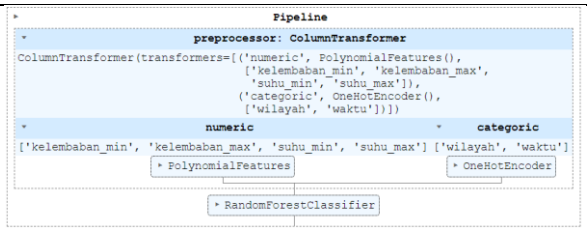
Data yang telah dilakukan tahap pra-proses dan *Exploratory Data Analysis* kemudian akan dibagi menjadi data latih dan juga data uji dengan perbandingan 80% data latih dan 20% data uji atau sebanyak 6718 baris data latih dan 1688 baris menjadi data uji. Peneliti memilih kolom cuaca sebagai label yang akan diklasifikasi serta menerapkan pembagian secara stratified atau pembagian data dengan seimbang kepada kolom cuaca.

3.5 Modeling

Modeling merupakan tahapan terakhir dalam memproses sebuah data. Pada tahap ini, peneliti membuat sebuah Pipeline yang didalamnya terdapat *Column Transformer*, dan algoritma yang akan dibandingkan yaitu *Extreme Gradient Boosting*, dan *Random Forest* untuk mempermudah dalam memproses serta memprediksi data baru dengan model telah dibuat. Sebelum membuat sebuah *Pipeline*, peneliti mempersiapkan *Column Transformer* terlebih dahulu untuk memproses data numerik dan kategorik. Ada beberapa tahapan yang peneliti lakukan menggunakan *Column Transformer* dimulai dari membagi kolom menjadi numerik dan kategorik, lalu

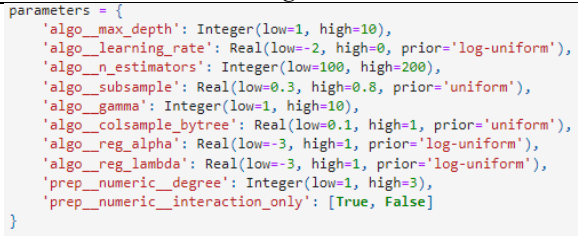
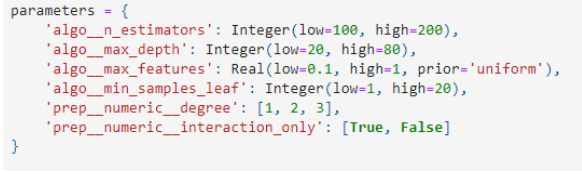
menyimpannya ke dalam variabel numerik untuk data numerik dan kategorik untuk data kategorikal, menerapkan *Polynomial Feature* pada kolom numerik untuk mengatasi data yang persebarannya tidak linier, dan menerapkan *One-Hot Encoding* pada kolom kategorik sebagai vektor biner yang bernilai integer 0 dan 1. Berikut adalah isi dari *Pipeline* yang peneliti buat:

Tabel. 4. Tabel isi dari *Pipeline*

Extreme Gradient Boosting	Random Forest
	

Selain memudahkan dalam memprediksi data, penggunaan *Pipeline* juga dapat membantu dalam proses *tuning hyperparameter* untuk memaksimalkan akurasi dari algoritma yang peneliti gunakan. Peneliti menggunakan salah satu algoritma *tuning hyperparameter* yaitu RandomSearchCV dengan mengatur nilai cv sebanyak 3, n_iter sebanyak 500, *Pipeline*, dan *hyperparameter* yang akan di *tuning*. Nilai dari *hyperparameter* yang akan di *tuning* adalah sebagai berikut:

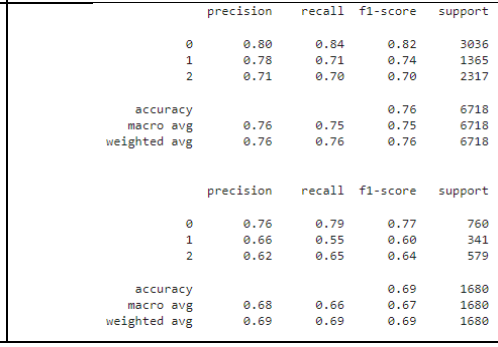
Tabel. 5. Daftar nilai *hyperparamter* yang akan dilakukan *tuning*

Extreme Gradient Boosting	*Random Forest*
	

3.6 Evaluasi

Setelah tahap mencari parameter yang terbaik, dilanjutkan dengan mengevaluasi model akhir pada data uji untuk menilai kinerja dari model yang telah dibuat menggunakan data baru. Tahapan ini akan menghasilkan parameter terbaik berdasarkan *tuning hyperparameter*, nilai *confusion matrix*, akurasi, presisi serta recall dari data uji. Berikut adalah tabel hasil evaluasi:

Tabel. 6. Hasil evaluasi

	Nilai *hyperparameter* terbaik yang didapat setelah *tuning*	*Classification Report*
Extreme Gradient Boosting	{'algo__colsample_bytree': 0.9554779473275833, 'algo__gamma': 2, 'algo__learning_rate': 0.08826616660734693, 'algo__max_depth': 9, 'algo__n_estimators': 100, 'algo__reg_alpha': 0.011907916572798662, 'algo__reg_lambda': 0.042450971980599515,	

	'algo__subsample': 0.7457400078117535, 'prep__numeric__degree': 2, 'prep__numeric__interaction_only': True}	
Random Forest	{'algo__max_depth': 39, 'algo__max_features': 0.45783495816651487, 'algo__min_samples_leaf': 5, 'algo__n_estimators': 102, 'prep__numeric__degree': 3, 'prep__numeric__interaction_only': True}	precision recall f1-score support 0 0.81 0.85 0.83 3036 1 0.78 0.70 0.74 1365 2 0.72 0.72 0.72 2317 accuracy 0.77 6718 macro avg 0.77 0.75 0.76 6718 weighted avg 0.77 0.77 0.77 6718 precision recall f1-score support 0 0.74 0.78 0.76 760 1 0.65 0.52 0.58 341 2 0.62 0.65 0.63 579 accuracy 0.68 1680 macro avg 0.67 0.65 0.66 1680 weighted avg 0.68 0.68 0.68 1680

3.7 Perbandingan Klasifikasi

Berdasarkan hasil evaluasi dari keempat metode klasifikasi yaitu *Extreme Gradient Boosting* dan *Random Forest* didapatkan hasil sebagai berikut:

Tabel. 6. Hasil akurasi dari setiap model

	Extreme Gradient Boosting	***Random Forest***
Akurasi data uji	69.226%	68.035%
Akurasi data latih	76.495%	77.284%

Berdasarkan penelitian yang telah dilakukan, peneliti mendapatkan bahwa algoritma *Extreme Gradient Boosting* memiliki performa yang lebih baik daripada *Random Forest* jika digunakan pada data cuaca Provinsi DKI Jakarta tahun 2018. Algoritma *Extreme Gradient Boosting* menghasilkan nilai akurasi sebesar 69% sedangkan *Random Forest* sebesar 68%.

Selain itu, melalui penelitian ini terdapat hal yang cukup penting yaitu waktu yang diperlukan pada saat melakukan pelatihan pada data cuaca Provinsi DKI Jakarta tahun 2018. Waktu yang dibutuhkan oleh algoritma *Extreme Gradient Boosting* untuk melakukan pelatihan data jauh lebih lama dibandingkan dengan waktu yang dibutuhkan oleh *Random Forest*. Berdasarkan penelitian peneliti, algoritma *Extreme Gradient Boosting* membutuhkan waktu kurang lebih 15 menit sedangkan *Random Forest* tidak lebih dari 10 menit.

4. Kesimpulan dan Saran

4.1 Kesimpulan

Dengan mempertimbangkan kelebihan dan kekurangan pada masing-masing algoritma, dapat disimpulkan bahwa algoritma *Random Forest* lebih baik daripada *Extreme Gradient Boosting* karena waktu yang dibutuhkan *Random Forest* lebih cepat dan juga akurasi yang dihasilkan *Random Forest* tidak berbeda jauh dengan *Extreme Gradient Boosting* yaitu sekitar 1 - 2% saja. Hal ini membuat peneliti memilih *Random Forest* sebagai algoritma yang lebih

efektif digunakan pada data cuaca Provinsi DKI Jakarta tahun 2018. Harapannya penelitian berikutnya dapat menemukan algoritma yang lebih efektif serta lebih baik dari *Random Forest* jika digunakan pada data cuaca Provinsi DKI Jakarta tahun 2018 dari segala hal.

4.2 Saran

Saran dari peneliti adalah karena tingkat akurasi dari model yang peneliti buat masih di angka 67% sampai 69% maka, perlu dilakukannya peninjauan kembali mengenai *Feature Engineering* maupun teknik pra-proses lainnya terhadap data yang peneliti miliki sehingga bisa mendapatkan tingkat akurasi yang lebih tinggi daripada yang telah peneliti lakukan.

5. Referensi

[1] X. Li, L. Wang, and E. Sung, "AdaBoost with SVM-based component classifiers," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 5, pp. 785–795, Aug. 2008, doi: 10.1016/J.ENGAPPAI.2007.07.001.

[2] G. A. Mursianto, "Perbandingan Metode Klasifikasi Random Forest dan XGBoost Serta Implementasi Teknik SMOTE pada Kasus Prediksi Hujan," 2021.

[3] F. Hamami and A. Dahlan, "KLASIFIKASI CUACA PROVINSI DKI JAKARTA MENGGUNAKAN ALGORITMA RANDOM FOREST DENGAN TEKNIK OVERSAMPLING," 2022.

[4] R. Sulistyo Budi, R. Patmasari, and S. Saidah, "KLASIFIKASI CUACA MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) WEATHER CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD."

[5] R. Manorathna, "Polynomial Regression with a Machine Learning Pipeline," 2020. [Online]. Available: https://www.researchgate.net/publication/344616388

[6] R. Supriyadi, W. Gata, N. Maulidah, A. Fauzi, I. Komputer, and S. Nusa Mandiri Jalan Margonda Raya No, "Penerapan Algoritma Random Forest Untuk Menentukan Kualitas Anggur Merah," vol. 13, no. 2, pp. 67–75, 2020, [Online]. Available: http://journal.stekom.ac.id/index.php/E-Bisnis■page67

[7] M. Radhi, S. Hamonangan Sinurat, D. Ryan Hamonangan Sitompul, E. Indra, and S. Informasi, "PREDIKSI WATER QUALITY INDEX (WQI) MENGGUNAKAN ALGORITMA REGRESSI DENGAN HYPER-PARAMETER TUNING," *Jurnal Sistem Informasi dan Ilmu Komputer Prima)*, vol. 5, no. 1, 2021.