PERBANDINGAN METODE *RANDOM FOREST* DAN *XGBOOST* PADA CUACA DI SUMATERA UTARA

SKRIPSI

DISUSUN OLEH

ROYHAN UMRI SIBUEA

2009020085

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

MEDAN

2024

PERBANDINGAN METODE *RANDOM FOREST* DAN *XGBOOST* PADA CUACA DI SUMATERA UTARA

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) dalam Program Studi Teknologi Informasi pada Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara

DISUSUN OLEH

ROYHAN UMRI SIBUEA

2009020085

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

MEDAN

2024

LEMBARAN PENGESAHAN

Judul Skripsi	: PERBANDINGAN METODE *RANDOM FOREST* DAN *XGBOOST* PADA CUACA DI SUMATERA UTARA
Nama Mahasiswa	: ROYHAN UMRI SIBUEA
NPM	: 200902085
Program Studi	: TEKNOLOGI INFORMASI

Menyetujui Komisi pembimbing

(Halim Maulana, S.T, M.Kom)
NIDN. 0121119102

Ketua Program Studi

(Fatma Sari Hutagalung, S.Kom, M.Kom)
NIDN. 0117019301

Dekan

(Dr. Al-Khowarizmi, S.Kom, M.Kom)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

PERBANDINGAN METODE *RANDOM FOREST* DAN *XGBOOST* PADA CUACA DI SUMATERA UTARA

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, 16 Juli 2024

Yang membuat pernyataan

ROYHAN UMRI SIBUEA

NMP : 2009020085

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini :

Nama : Royhan Umri Sibuea
N[pm : 2009020085
Program Studi : Teknologi Informasi
Karya Ilmiah : Skripsi

Dengan Pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak bebas Royalti Non-Eksekutif *(Non-Exclusive Royalty Free Right)* atas penelitian skripsi saya yang berjudul :

PERBANDINGAN METODE *RANDOM FOREST* DAN *XGBOOST* PADA CUACA DI SUMATERA UTARA

Beserta perangkat yang ada (jika diperlukan). Dengan hak bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuar dengan sebenarnya.

Medan, 16 Juli 2024
Yang membuat pernyataan

Royhan Umri Sibuea
NPM.2009020085

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap	: Royhan Umri Sibuea
Tempat dan Tanggal Lahir	: Anggoli, 28 Juli 2002
Alamat Rumah	: Desa Anggoli
Telepon/Faks,HP	: 085261675895
E-mail	: royhansibuea@gmail.com
Instansi Tempat Kerja	: -
Alamat Kantor	: -

DATA PENDIDIKAN

SD	: SDN 153073 Anggoli 1	TAMAT : 2014
SMP	: SMP N1 Sibabangun	TAMAT : 2017
SMA	: SMA N1 Pinangsori	TAMAT : 2020

KATA PENGANTAR

Assalamu'alaikum warahmatullahi wabarakatuh

Alhamdulillah, segala puji dan Syukur penulis panjatkan kehadiran Allah Subhanahu wa ta ala yang telah melimpahkan Rahmat dan karunianya yang penuh dengan ilmu kepada penulis, sehingga penulis dapat menyelesaikan tugas akhir ini yang berjudul tentang " Perbandingan Metode *Random Forest* dan *XGBoost* pada Cuaca di Sumatera Utara" untuk memenuhi persyaratan dalam jenjang strata satu dan mencapai gelar Sarjana Komputer dijurusan Teknologi Informasi, Fakultas Teknologi Informasi dan Ilmu Komputer, Universitas Muhammadiyah Sumatera Utara. Sholawat serta salam selalu tercurahkan kepada junjungan Nabi besar Muhammad Shalallahu alaihi wassalam, keluarga dan sahabatnya yang syafaatnya kita nantikan diakhir zaman nanti. Dalam penyusunan Skripsi ini, penulis telah mendapatkan banyak bantuan dan bimbingan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis temtunya berterimakasih kepada pihak dalam dukungan serta doa dalam penyelesaian skripsi. Penulis juga berterimakasih kepada :

1. Tuhan Yang Maha Esa, yang telah memberikan seluruh rahmat, hidayah, dan nikmat sehat-Nya kepada penulis sehingga dapat menyelesaikan Tugas Akhir ini dengan lancar.
2. Yang tercinta, teristimewa, dan tersayang buat orangtua saya Andi Sibuea dan Nilawati Limbong yang mana selalu memberikan penulis dukungan motivasi yang tiada henti, kasihsayang, dan semangat yang tulus serta doa restu dan

nasehatan yang tiada habis nya. Serta pengorbanan yang keras dalam mencari nafkah untuk kesuksesan anak-anak nya yang tak ternilai.

3. Bapak Prof. Dr. Agussani, M.AP selaku Rektor Universitas Muhammadiyah Sumatera Utara.
4. Bapak Dr. Al-Khowarizmi,M.Kom selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara.
5. Ibu Fatma Sari Hutagalung, S.Kom., M.Kom. selaku Ketua Program Studi Teknologi Informasi.
6. Bapak Mhd. Basri, S.Si., M.Kom selaku Plt.Kaprodi Teknologi Informasi
7. Bapak Halim Maulana, ST., M.Kom selaku Dosen Pembimbing yang selalu memberikan motivasi serta pelajaran kepada penulis dan telah sabar memberikan bimbingan dari awal hingga selesainya Skripsi ini.
8. Serta sahabat-sahabat penulis yang selalu mendukung dan memberikan motivasi serta perhatiannya.

Penulis menyadari bahwa Skripsi ini masih jauh dari sempurna dikarenakan keterbatasan pengetahuan dan kemampuan yang dimiliki oleh penulis. Oleh karena itu, penulis mengharapkan kritik serta saran yang bersifat membangun untuk menyempurnakan penulisan Skripsi ini. Semoga Skripsi ini dapat bermanfaat bagi penulis khususnya dan bagi semua yang membutuhkan.

Wassalamu'alaikum warahmatullahi wabarakaatuh

Medan, 16 Juli 2024

Royhan Umri Sibuea

Perbandingan Metode *Random Forest* dan *XGBoost*

pada Cuaca di Sumatera Utara

ABSTRAK

Prediksi cuaca yang akurat sangat penting untuk berbagai sektor, termasuk pertanian, transportasi, dan manajemen bencana. Data cuaca yang digunakan mencakup variabel yaitu kelembaban, temperature, dan kecepatan angin yang dikumpulkan dari stasiun cuaca di seluruh Sumatera Utara. Metode *Random Forest* adalah algoritma ensemble berbasis pohon keputusan yang terkenal dengan kemampuannya mengatasi overfitting dan memberikan hasil yang akurat. Di sisi lain, *XGBoost* adalah teknik boosting yang meningkatkan kinerja model melalui pembelajaran bertahap, memperbaiki kesalahan yang dilakukan oleh model sebelumnya. Dari hasil penelitian menunjukkan bahwa kedua metode memiliki keunggulan masing-masing dalam hal akurasi dan kecepatan prediksi. Metode *Random forest* menghasilkan nilai *Root Mean Squared Error* (*RMSE*) sebesar 0.753732 dan *Coefficient of Determination* (R^2) sebesar 0.736315. Di sisi lain, *XGBoost* menunjukkan nilai *RMSE* sedikit lebih rendah yaitu 0.737818 dan R^2 lebih tinggi mencapai 0.747332. Disimpulkan bahwa *XGBoost* memiliki kinerja yang sedikit lebih baik dalam hal meminimalkan kesalahan prediksi (RMSE) dan meningkatkan kecocokan model terhadap data (R^2) dibandingkan *random forest*.

Kata Kunci : *Machine Learning*; Prediksi Cuaca; *Random Forest*; *XGBoost*

Perbandingan Metode *Random Forest* dan *XGBoost* pada Cuaca di Sumatera Utara

ABSTRACT

Accurate weather forecasting is crucial for various sectors, including agriculture, transportation, and disaster management. The weather data used includes variables such as humidity, temperature, and wind speed collected from weather stations across North Sumatra. The Random Forest method is an ensemble algorithm based on decision trees known for its ability to handle overfitting and provide accurate results. On the other hand, XGBoost is a boosting technique that improves model performance through iterative learning, correcting errors made by previous models. Research results show that both methods have their respective advantages in terms of accuracy and prediction speed. The Random Forest method yields a Root Mean Squared Error (RMSE) of 0.753732 and a Coefficient of Determination (R^2) of 0.736315. In contrast, XGBoost shows a slightly lower RMSE of 0.737818 and a higher R^2 of 0.747332. It is concluded that XGBoost performs slightly better in minimizing prediction errors (RMSE) and improving model fit to the data (R^2) compared to Random Forest.

Keywords: Machine Learning; Weather Prediction; Random Forest; XGBoost

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR ISTILAH

SVM	= *Support Vector Machine*
K-NN	= *K-Nearest Neighbor*
MLP	= *Multilayer Perceptron*
BMKG	= Badan Meteorologi Klimatologi dan Geofisika
SCM	= *Successive Correction Methods*
OI	= *Optimal Interpolation*
3DVAR	= *The Three Dimensional Variational*
4DVAR	= *The Four Dimensional Variational*
KF	= *Kalman Filter*
API	= *Application Programming Interface*
Max Defth	= Maksimum Kedalaman
P	= Penjelas (*predictor)*
S	= Himpunan dataset
C	= Jumlah Kelas
P_i	= Probabilitas Frekuensi Kelas ke-i dalam dataset
T,X	= Atribut T dan atribut X
P(c)	= Probabilitas kelas atribut
E(c)	= Nilai entropy kelas atribut
S	= Himpunan Dasar
A	= Atribut
|Si|	= Jumlah sampel untuk nilai i
|S|	= Jumlah seluruh sampel data
Entropy (S_i)	= Entropy untuk sampel yang memiliki nilai
Humidity	= Jumlah uap air yang terkandung di udara
Windy	= kondisi cuaca di mana angin bertiup dengan kecepatan yang cukup kencang
$\hat{y}_i^{(t)}$	= Model tree terakhir

$\hat{y}_i^{(t)}$	= Model tree yang dihasilkan sebelumnya
$f_k(x_i)$	= Model baru yang dibuat
t	= Jumlah total model tree yang dibangun dari base tree models.
y_i	= nilai aktual
l	= Fungsi loss
Ω	= fungsi regularisasi
λ	= parameter regulasi yang memili nilai default sebesar 1
T	= jumlah leaves yang ada pada tree
Ω	= Menentukan kompleksitas model dengan menentukan nilai γ
ω	= bobot leaves yang digunakan output value
Previous $f_i(x)$	= Probabilitas sebelumnya
Forecasting	= Proses memperkirakan atau memprediksi apa yang akan terjadi di masa depan
X_i	= nilai hasil prediksi
Y_i	= nilai aktual
m	= jumlah data
RMSE	= Root Mean Squared Error
R^2	= Coefficient of Determination
Fluktuasi	= Perubahan yang terjadi secara naik-turun atau tidak stabil dalam suatu hal contohnya suhu
Weak learn	= Prediksi lemah
Strong learner	= Prediksi Kuat

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Sejumlah faktor, termasuk suhu, tekanan, volume air awan, arah dan kecepatan angin, tekanan udara, dan kelembaban berpadu membentuk cuaca. Perkiraan cuaca merupakan salah satu hal yang sangat dibutuhkan orang-orang di seluruh dunia agar tidak mengganggu kegiatan yang telah direncanakan. (Mursianto et al., 2021).

Prediksi cuaca adalah prediksi kondisi atmosfer sebuah tempat dengan menggunakan sains dan teknologi (Suma & Pasundan, 2021). Pola dan korelasi kompleks antara berbagai variabel cuaca, termasuk suhu, kelembapan, waktu, kecepatan angin, dan wilayah, dapat dipelajari menggunakan teknik machine learning dalam prediksi cuaca. (Dwiyanti & Prianto, 2023).

Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) bertugas memantau dan mengukur kejadian alam yang memengaruhi cuaca. Namun, dalam meramalkan cuaca, ada beberapa faktor yang perlu diperhatikan. Pertama-tama, ramalan cuaca sering kali bergantung pada keahlian dan pengalaman peramal cuaca, yang dapat menghasilkan ramalan yang bervariasi. Hal ini dikarenakan banyaknya sumber data yang dibutuhkan, termasuk hasil pemindaian radar, citra satelit kondisi awan, dan pengamatan langsung. (Rakhmat & Mutohar, 2023).

Peneliti sering menggunakan *machine learning* untuk kasus klasifikasi atau prediksi. Penggunaan komputer dan algoritma matematika untuk belajar dari data dan meramalkan masa depan dikenal sebagai *Machine Learning*. *Machine learning* juga digunakan untuk menganalisis dan memprediksi kondisi cuaca secara akurat (Suma & Pasundan, 2021). *Supervised Learning* dan *unsupervised learning* adalah

dua kategori metodologi *Machine Learning*. Kedua metode tersebut bertujuan untuk menghasilkan pemahaman dari data yang dihasilkan oleh kecerdasan buatan yang telah diprogram. Salah satu pendekatan metode *Supervised learning* adalah klasifikasi merupakan teknik untuk smengelompokkan data berdasarkan data yang telah dilatih sebelumnya (Iman et al., 2022). Algoritma *Supervised learning* terdiri dari *random forest, XGBoost*, *logistic regression*, *support vector machine, decision tree, adaboost, naive bayes,* dan *K-nearest neighbour* (Dridi et al., 2021). Penelitian ini membahas tentang perbandingan antara metode *random forest* dengan *XGBoost*.

Algoritma *machine learning* yang menggunakan beberapa pohon keputusan disebut *Random Forest*. Pendekatan ini termasuk dalam algoritma *machine learning* teratas yang digunakan dalam berbagai disiplin ilmu dan telah menunjukkan keefektifannya dalam beberapa tahun terakhir baik dalam masalah regresi maupun klasifikasi. (YEŞİLKANAT, 2020). Sedangkan Salah satu teknik peningkatan yang membangun serangkaian pohon keputusan di mana pembuatan pohon berikutnya bergantung pada pohon sebelumnya disebut *XGBoost.* Inisialisasi probabilitas yang dipilih peneliti akan menyebabkan pohon pertama di XGBoost berkinerja buruk dalam klasifikasi. Setelah itu, setiap pohon yang dibangun akan mendapatkan pembaruan bobot, menghasilkan serangkaian pohon klasifikasi yang kuat. Semua bobot pohon dijumlahkan dan kemudian dimasukkan ke dalam fungsi logistik untuk menghasilkan prediksi. (Syukron et al., 2020).

Penelitian sebelumnya, Mdegela (2023) meneliti tentang klasifikasi curah hujan ekstrem menggunakan *Machine learning* seperti *random forest*, *XGBoost*, *SVM*, *K-NN*, dan *MLP*. Hasil dari penelitian ini model *random forest* dan *XGBoost* mencapai skor tertinggi dalam memprediksi kejadian hujan lebat (Mdegela et al.,

2023). Asselman (2021) meneliti tentang meningkatkan analisis faktor kinerja untuk memperkirakan tingkat siswa yang lebih akurat. Metode yang digunakan adalah *random forest*, *adaboost*, dan *XGBoost*. Hasil dari penelitian tersebut adalah *XGBoost* model terbaik diantara yang lainnya untuk meningkatkan prediksi kinerja.

Berdasarkan uraian penelitian sebelumnya, penilitian ini akan membahas tentang klasifikasi cuaca dan memprediksi cuaca pada tahun 2024 di Sumatera Utara dengan menggunakan *random forest* dan *XGBoost*. Hal yang berbeda dari penelitian sebelumnya adalah penelitian hanya fokus kedua metode yaitu *random forest* dan *XGBoost* sedangkan penelitian sebelumnya menggunakan beberapa metode. Kemudian penelitian ini menambahkan hasil prediksi cuaca di Sumatera Utara pada tahun 2024. Oleh karena itu peneliti menarik judul "**Perbandingan Metode *Random Forest* dan *XGBoost* pada Cuaca di Sumatera Utara"**.

1.2 Rumusan Masalah

Berdasarkan latar belakang, rumusan masalah dalam penelitian adalah :

1. Bagaimana hasil akurasi antara metode *random forest* dan *XGBoost* pada cuaca di Sumatera Utara ?
2. Bagaimana hasil prediksi cuaca di Sumatera Utara dengan menggunakan metode *random rorest* dan *XGBoost* ?

1.3 Batasan Masalah

Berikut adalah batasan penelitian tersebut :

1. Data pada penelitian ini menggunakan data sekunder yang didapatkan dari situs BMKG dari tanggal 01 April 2021 sampai dengan 31 Maret 2024.
2. Metode pada penelitian ini menggunakan metode *random forest* dan *XGBoost*.

3. Sofware untuk membantu penelitian ini menggunakan Pyhton dan Microsoft Excel.
4. Penelitian ini menggunakan tanda koma (.) menunjukkan desimal.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

1. Mengetahui hasil akurasi antara metode *random forest* dan *XGBoost*.
2. Mengetahui hasil prediksi cuaca menggunakan metode *random forest* dan *XGBoost*.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat bagi peneliti sebagai berikut :

1. Bagi peneliti untuk mengetahui metode mana yang lebih baik anatara *random forest* dan *XGBoost* dalam memprediksi cuaca di Sumatera utara.
2. Bagi masyarakat untuk memperiapkan diri untuk menghadapi keadaan cuaca ke depannya.

BAB II

LANDASAN TEORI

2.1 Prediksi Cuaca

Perubahan, perkembangan, serta munculnya atau menghilangnya fenomena udara merupakan bagian dari kondisi atmosfer yang dipantau secara kompleks yang dikenal sebagai cuaca. Cuaca dapat memengaruhi aktivitas sehari-hari seperti menjemur pakaian dan aktivitas luar ruangan lainnya. Keakuratan prakiraan cuaca dengan mempertimbangkan potensi perubahan yang berfluktuasi merupakan salah satu penelitian yang paling signifikan. Faktor ini memotivasi para ilmuwan untuk terus menciptakan model prakiraan yang semakin akurat. Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) mengeluarkan prakiraan cuaca untuk memprediksi cuaca Indonesia. Hal ini memungkinkan setiap orang untuk mempersiapkan diri dengan baik menghadapi kondisi cuaca yang tidak terduga. (Luthfiarta et al., 2020).

Praktik pengumpulan informasi dari kondisi atmosfer, seperti suhu, kelembapan, curah hujan, arah dan kecepatan angin, dikenal sebagai perkiraan cuaca. Untuk memperoleh prakiraan cuaca yang paling akurat, variabel-variabel ini kemudian diperiksa dan dibandingkan dengan kondisi sehari sebelumnya, sebulan sebelumnya, atau bahkan setahun sebelumnya. *Machine learning* merupakan subbidang ilmu komputer yang mencakup pendekatan ini. (Luthfiarta et al., 2020).

Pola dan korelasi yang kompleks antara berbagai variabel cuaca, termasuk suhu, kelembaban, waktu, dan wilayah, dapat dipelajari menggunakan teknik machine learning dalam prediksi cuaca. Metode ini memperkirakan kondisi cuaca di masa mendatang dengan menggunakan metode dan algoritma statistik yang dapat

membangun model prediktif berdasarkan data yang ada. (Dwiyanti & Prianto, 2023). Setiap ramalan cuaca memerlukan pengumpulan data meteorologi secara metodis dari berbagai lokasi dan analisis data yang tepat. (Suma & Pasundan, 2021). Ada tiga langkah untuk memprediksi cuaca yaitu:

a. Observasi dan Pengumpulan data

Pada proses ini, ahli meteorologi mengukur unsur-unsur atmosfer tertentu untuk mengumpulkan data. Temperatur, tekanan, arah dan kecepatan angin, kelembaban, tutupan awan, curah hujan, dan faktor-faktor lainnya merupakan contoh unsur cuaca. Dengan meneliti bagaimana unsur-unsur ini berevolusi dari waktu ke waktu dan membandingkan pola perubahan dengan tren sebelumnya, seseorang dapat lebih memahami kondisi cuaca yang dapat diharapkan.

b. Asimilasi, Pemrosesan, dan Analisis

Ahli meteorologi menganalisis data yang telah mereka kumpulkan, sebuah proses yang dikenal sebagai asimilasi data. Asimilasi data adalah teknik estimasi yang menggabungkan data terukur dengan hasil model prakiraan cuaca numerik. Berikut ini adalah beberapa metode untuk asimilasi data, baik saat ini maupun di masa mendatang:

1. *Successive Correction Methods* (*SCM*)
2. *Optimal Interpolation* (*OI*)
3. *The Three Dimensional Variational* (3*DVAR*)
4. *The Four Dimensional Variational* (4*DVAR*)
5. *Kalman Filter* (*KF*)

c. Prediksi kondisi atmosfer di masa depan

Pada proses ini, ahli meteorologi akan memperkirakan perubahan kondisi atmosfer di masa mendatang. Mirip dengan interpolasi, ekstrapolasi melibatkan estimasi nilai di luar rentang pengamatan aktual dan membandingkannya dengan nilai yang ada.

Teknik peramalan cuaca modern membuat peta cuaca sinoptik dengan menggabungkan informasi dari beberapa pengamatan yang dilakukan di berbagai lokasi. Peta-peta ini menampilkan tren suhu, curah hujan, angin, tekanan, dan tutupan awan di lokasi tertentu. Peramal cuaca dapat membuat peta cuaca sinoptik dari atmosfer bagian atas dengan menggunakan pengamatan radiosonde dua kali sehari. Tingkat keparahan badai ditentukan oleh pengamatan radar terhadap asal-usul, pergerakan, dan fitur badai. Satelit dan pesawat terbang digunakan untuk mengukur cuaca. (Suma & Pasundan, 2021).

2.2 *Machine Learning*

Machine learning juga dikenal sebagai pembelajaran mesin adalah metode *AI* yang digunakan untuk meniru fungsi manusia dalam berbagai tugas, seperti memecahkan masalah. Secara singkat, *Machine learning* adalah mesin yang dapat belajar dan melakukan tugas-tugasnya sendiri tanpa bantuan pengguna. Menurut Arthur Samuel, seorang pelopor Amerika dalam bidang permainan komputer dan kecerdasan buatan, *Machine learning* adalah bidang ilmu yang mempelajari bagaimana membuat komputer dapat belajar tanpa diprogram secara eksplisit (Syuhada et al., 2021). Adapun keungulan yang dimiliki *machine learning* adalah sebagai berikut :

1. Kemampuan untuk mengembangkan pengetahuan tambahan menggunakan data yang telah dikumpulkan
2. Kemampuan untuk menyelesaikan masalah tidak konsisten (Suma & Pasundan, 2021).

2.2.1 *Tahapan dalam Machine Learning*

Ada enam tahapan dalam metodologi *Machine learning* antara lain adalah tahap ide atau masalah bisnis, tahap penemuan dan pengumpulan data, tahap pemilihan model atau kelompok, tahap pelatihan model, tahap peluncuran model, dan tahap evaluasi dan retrain model (Suma & Pasundan, 2021).

2.2.1.1 Ide / Masalah Bisnis

Machine learning harus menjadi pendorong utama pada titik ini agar algoritma yang tepat dapat mengatasinya (Suma & Pasundan, 2021). Semua Permasalahan atau motivasi yang akan diselesaikan oleh tahap *machine learning* termasuk saat informasi ataupun data yang dikumpulkan. Tanpa adanya data, *machine learning* tidak dapat menyelesaikan masalah. Data data dapat berupa file seperti *microsoft excel* atau *access*. Pembelajaran dimasa mendatang dimulai dengan langkah ini. Anda dapat memecahkan masalah saat ini dengan melakukan pernyataan – pernyataan berikut ini :

1. Apakah tujuan Anda? Apakah yang harus Anda ramalan?
2. Fitur (bobot) apa yang Anda rencanakan untuk menggunakan?
3. Data jenis apa yang ingin Anda masukan? Apakah data tersebut tersedia?
4. Apa masalah yang sedang kita hadapi? Binari Klasifikasi ? (Syuhada et al., 2021)

2.2.1.2 Pengumpulan Data

Tahap selanjutnya dari *machine learning* adalah pengumpulan dan persiapan data. *Machine learning* bekerja lebih baik ketika lebih banyak data berkualitas tinggi dikumpulkan (Syuhada et al., 2021).

Misalnya, jika digunakan secara langsung sebagai fitur *machine learning*, atribut nama dan alamat yang disertakan dalam data mentah akan dianggap sebagai fitur kategoris dengan domain yang relatif luas. Karena karakteristik ini pun tidak dapat digeneralisasi, dan nama atau alamat memiliki makna yang sangat luas. Agar *machine learning* dapat menggunakan data, data tersebut harus terlebih dahulu diubah menjadi vektor fitur (juga disebut rekayasa fitur, ekstraksi fitur, atau representasi ekstraksi). Rekayasa fitur ini memerlukan banyak transformasi data, bergantung pada sifat data dan daya prediksi berbagai variabel data. (Suma & Pasundan, 2021).

2.2.1.3 Pemilihan Model

Pemilihan model memerlukan penentuan algoritma dan penyajian data sebagai model yang sesuai (Syuhada et al., 2021). Pemilihan algoritma dan penyetelan *hyperparameter* adalah dua proses utama yang terlibat dalam pemilihan model. Proses memilih algoritma digunakan untuk menentukan ruang hipotesis mana yang cocok untuk aplikasi. Selain akurasi prakiraan, campuran kompleks faktor teknis dan nonteknis memengaruhi pemilihan algoritme. Waktu proses, biaya sumber daya, aksesibilitas dan kegunaan alat pelatihan, serta pendapat pengguna mengenai "interpretabilitas" fungsi prediksi adalah beberapa contohnya. (Suma & Pasundan, 2021).

Faktor-faktor tertentu, seperti:

1. Sifat permasalahan
2. Jumlah data yang dapat diakses untuk latihan
3. Dukungan keputusan atau kasus kognitif ?
4. Membutuhkan kemampuan untuk menjelaskan hubungan antara input dan keputusan (Suma & Pasundan, 2021).

2.2.1.4 *Processing* Data

Sebelum penggunaan algortima *machine learning* data dibagi menjadi dua bagian, 80% data pelatihan dan 20% data pengujian (Agustiningsih et al., 2023).

2.2.1.5 Peluncuran Model

Prediksi berdasarkan input masa depan, model terlatih dapat dimasukkan ke *Application Programming Interface* (*API*), antarmuka pengguna, atau aplikasi seluler (Suma & Pasundan, 2021).

2.2.1.6 Evaluasi dan Pengembangan Model

Langkah terakhir setelah mengembangkan model *machine learning* yang dapat memperkirakan data uji adalah memanfaatkan data uji untuk menilai akurasi dan performa model. Ini akan mengevaluasi akurasi algoritme yang dipilih pada langkah sebelumnya (Syuhada et al., 2021).

2.2.2 *Jenis – Jenis Machine Learning*

Machine learning terbagi dua pendekatan yaitu *supervised learning* dan *unsupervised learning*. Kedua metode tersebut bertujuan untuk menghasilkan pemahaman dari data yang dihasilkan oleh kecerdasan buatan yang telah diprogram (Iman et al., 2022).

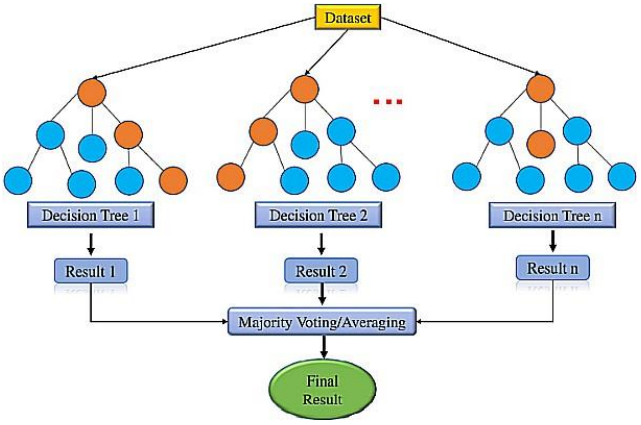
2.2.2.1 *Supervised Learning*

Supervised learning adalah kumpulan pelatihan contoh yang memiliki tanggapan (tujuan) yang tepat. Dengan pelatihan ini, sumber daya energi biasanya menanggapi semua input yang mungkin (Suma & Pasundan, 2021).

Dalam pembelajaran terbimbing, gagasan tentang *supervesid learning* pada dasarnya mengajarkan algoritma untuk memilih fungsi yang paling sesuai dengan input, di mana X tertentu menghasilkan estimasi y terbaik. Namun, dalam praktiknya, banyak orang kesulitan mengidentifikasi fungsi terbaik. Hasil dari fakta bahwa algoritma bergantung pada asumsi yang digunakan. Jika asumsi tidak terpenuhi, tidak jarang hasil pengolahan data akan bias. Oleh karena itu, algoritma ini membutuhkan data latihan yang tepat agar sistem dapat mempelajari polanya serta regresi, klasifikasi, K-NN, naive bayes, hutan kebetulan, XGBoost, pohon keputusan, linear regression, SVM, dan neural network (Syuhada et al., 2021).

2.2.2.2 *Random Forest*

Random forest adalah kumpulan dari beberapa (*decision tree*) atau pohon keputusan digunakan untuk membuat prediksi dengan memecahkan data menjadi beberapa kategori berdasarkan karakteristik tertentu dan menggunakan perbandingan nilai tertentu untuk membuat keputusan (Melvin & Soraya, 2023). Algoritma *random forest* mempunyai parameter yang dapat ditentukan termasuk jumlah decision tree yang akan dibuat, kriteria untuk mengukur jarak antara setiap node dan batas maksimal kedalaman pohon keputusan (Akbar & Sanjaya, 2023), dapat diliat gambar dibawah ini.



Gambar 2.1 Visualisasi *Random Forest*

a. Jumlah Pohon keputusan

Jumlah pohon keputusan merupakan parameter dalam metode hutan acak yang menentukan berapa banyak pohon yang disertakan dalam model hutan acak untuk membantu prediksi klasifikasi dan pengambilan keputusan. Jumlah pohon pada model hutan acak akan memengaruhi akurasi dan proses komputasi serta potensi model untuk peningkatan kinerja (Akbar & Sanjaya, 2023).

b. Kriteria Pemisahan Data

Menurut kriteria pemisahan, yang menilai kemurnian node (klasifikasi) atau kualitas kesesuaian model node (regresi), nilai yang besar untuk kriteria (C) menunjukkan klasifikasi node yang tidak akurat atau kesesuaian model yang buruk (regresi) untuk setiap keluaran. (Akbar & Sanjaya, 2023).

c. *Max Defth* (Maksimum Kedalaman)

Parameter ini digunakan dalam algoritma *random forest* untuk menentukan kedalaman pohon yang paling dalam. Kedalaman pohon maksimum suatu pohon keputusan membatasi setiap derajat kerumitan komputasi, dan seiring

bertambahnya kedalaman pohon, maka bertambah pula biaya komputasi (Akbar & Sanjaya, 2023).

Set data pelatihan pada algoritma *random forest* dapat dirumuskan sebagai berikut: $S = \{(x_i, y_i)\}$; $i = 1, 2, \dots, N$; $j = 1, 2, \dots, M$, dimana x adalah sampel dan y adalah variabel fitur S. N adalah jumlah sampel pelatih, dan ada variabel fitur M di setiap sampel (Yoga Religia et al., 2021).

Sebuah klaster data dengan n observasi dan p faktor penjelas (prediktor) berfungsi sebagai dasar untuk *random forest*. Berikut ini adalah langkah-langkah yang terlibat dalam kompilasi dan estimasi menggunakan algoritma *random forest*:

1. Tahapan *Boostrapping*

   Menggunakan penggantian sampel untuk memilih sampel acak berukuran n dari kumpulan data asli.

Sumber: dataversity.net

Gambar 2.2 Contoh Penggunaan Sampel Menggunakan *Bootstrapping*

2. Tahapan Pemilihan Fitur Acak

   Tahapan proses ini, pohon dibangun hingga ukurannya tidak dipotong. m variabel prediktor dipilih secara acak, di mana < p, dan pemilih optimal ditentukan oleh m prediktor yang dipilih.. Berikut adalah contoh membangun *decion tree* yaitu:

a. Menentukan pohon keputusan adalah menjumlahkan nilai *entropy* sebagai tergantung pada tingkat ketidakmurnian atribut dan nilai data. Ini dapat dilakukan dengan memakai rumus dengan persamaan (2.1) pada satu atribut, persamaan (2.2) pada dua atribut dengan memakai tabel frekuensi, dan persamaan (2.3):

$$\boldsymbol{Entropy\ (S) = \sum_{i=1}^{c} - p_i \log_2 p_i} \qquad \textbf{(2.1)}$$

S : Kumpulan dataset

C : Penjumlahan Kelas

p_i : Potensi frekuensi kelas ke-i dalam kumpulan data

$$\boldsymbol{Entropy\ (T, X) = \sum_{c \epsilon X} P(c) E(c)} \qquad \textbf{(2.2)}$$

T, X : Atribut T dan X

$P(c)$: Probabilitas atribut kelas

$E(c)$: Nilai entropy atribut kelas

$$\boldsymbol{Gain\ (A) = Entropy\ (S) - \sum_{i=1}^{k} \frac{|Si|}{|S|} x\ Entropy\ (S_i)} \qquad \textbf{(2.3)}$$

S : Kumpulan dataset

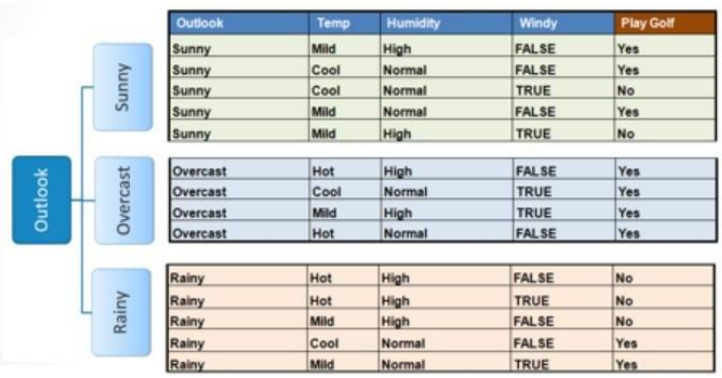
A : Atribut

$|Si|$: Jumlah contoh pada nilai i

$|S|$: Jumlah keseluruhan pada contoh data

$Entropy\ (S_i)$: *Entropy* pada contoh yang mempunyai nilai

Kumpulan data dengan atribut seperti Outlook, Suhu, Kelembaban, dan Angin adalah contoh kumpulan data yang digunakan untuk membuat pohon keputusan. Kelas Play Golf juga memiliki karakteristik "Tidak" dan "Ya".



Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

Gambar 2.3 Contoh pembentukan decision tree

Proses membangun *decision tree* adalah sebagai berikut:

1. Langkah pertama nilai *entropy* dari seluruh data dihitung pada 14 data dengan nilai "Tidak" 5 dan "Ya" 9.

Atribut play golf

Play Golf	
Ya	Tidak
9	5

$$Entropy\ (PlayGolf) = Entropy(5,9)$$

$$= Entropy\ (0.36,0.64)$$

$$= -0.36\ log_2 0.36 + (-0.64\ log_2 0.64)$$

$$= 0.53 + 0.64 = 0.94$$

2. Langkah selanjutnya adalah menentukan nilai entropi setiap atribut setelah menentukan nilai entropi atribut target. Berikut ini adalah ilustrasi cara menentukan nilai entropi karakteristik Outlook dan Play Golf.

Tabel 2.1 Hasil dari Dua Karakteristik

		Play Golf		
		Ya	Tidak	
Outlook	*Sunny*	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

$$Entropy\ (PlayGolf, Outlook) = (P(Sunny)x\big(E(3,2)\big) + (P(Overcast)x$$

$$(E(4,0)) + \ (P(Rainy)xE(2,3))$$

$$Entropy\ (PlayGolf, Outlook) = \left(\frac{5}{14}x\ 0.971\right) + \left(\frac{4}{14}x\ 0.0\right) + \left(\frac{5}{14}x\ 0.917\right)$$

$$= 0.693$$

3. Langkah selanjutnya adalah menghitung nilai informasi untuk setiap variabel setelah menghitung nilai *entropy*. Menghitung nilai Informasi untuk atribut *Outlook.*

$$Gain\ (Outlook) = E(PlayGolf) - E(PlayGolf, Outlook)$$

$$= 0.94 - 0.693$$

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

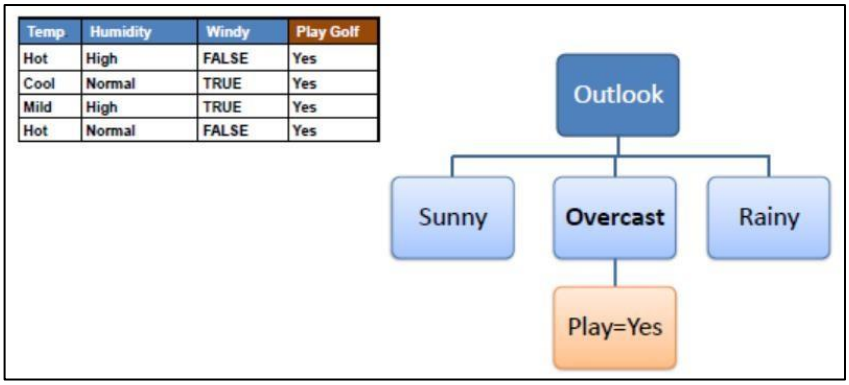
		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

Gambar 2.4 Frekuensi data dari nilai masing – masing atribut

4. Ulangi prosedur yang sama untuk setiap cabang setelah membagi kumpulan data berdasarkan cabangnya. Karena nilai informasinya paling tinggi, properti Outlook dipilih sebagai simpul akar.

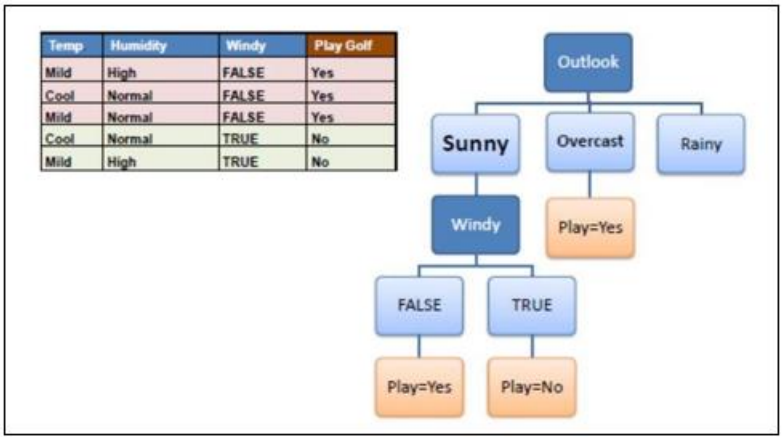
Gambar 2.5 Pembentukan *Root Node* dan Terminal

5. Periksa hasil percabangan simpul akar berikutnya, simpul daun adalah cabang dengan nilai entropi 0.



Gambar 2.6 Contoh Pembentukan *Leaf Node*

6. Hingga semua node berbentuk *leaf node*, lakukan pemisahan cabang seperti langkah langkah sebelumnya.



Gambar 2.7 Pembentukan Internal Node dan *Leaf Node*

7. Untuk mendapatkan k buah *decision tree*, lakukan perulangan Langkah 1 dan langkah 2 sebanyak k kali
8. Hitunglah estimasi gabungan dari kelompok suara mayoritas dari k keputusan.

2.2.2.3 *XGBoost*

EXtreme Gradient Boosting, juga dikenal sebagai *XGBoost*, adalah algoritma pembelajaran kelompok dengan metode peningkatan yang dikembangkan oleh Tianqi Chen pada tahun 2014 (Melvin & Soraya, 2023). *XGBoost* adalah metode kelompok yang didasarkan pada pohon peningkatan gradient. Pohon regresi, *node* bagian dalam menunjukkan nilai-nilai tes atribut, dan node cabang menunjukkan skor keputusan (Muslim Karo Karo, 2020).

Model yang lebih teratur digunakan oleh XGBoost untuk membangun struktur pohon regresi, meningkatkan kinerja dan memungkinkan pengurangan kompleksitas model untuk mencegah *overfitting*. Hasil prediksi *XGBoost* terakhir adalah jumlah hasil prediksi dari setiap pohon regresi. Data dengan kelas tidak seimbang, algoritma yang didasarkan pada pohon keputusan kurang efektif (Herni Yulianti et al., 2022). Rumus nilai prediksi pada t sebagai berikut:

$$\hat{\boldsymbol{y}}_{\boldsymbol{i}}^{(\boldsymbol{t})} = \sum_{\boldsymbol{k=1}}^{\boldsymbol{t}} \boldsymbol{f}_{\boldsymbol{k}}(\boldsymbol{x}_{\boldsymbol{i}}) \quad \textbf{(2.4)}$$

$\hat{y}_i^{(t)}$: model tree terakhir

$\hat{y}_i^{(t)}$: model tree yang dihasilkan sebelumnya

$f_k(x_i)$: Model baru telah dibuat.

t : berapa banyak model pohon yang dibangun secara keseluruhan dari model pohon dasar.

Dengan mencari klasifikasi baru yang dapat menurunkan fungsi kerugian dengan fungsi kerugian target yang mewakili fungsi pembelajaran (Obj), masalah penentuan algoritma terbaik dapat diubah.

$$\boldsymbol{obj}(\boldsymbol{t}) = \sum_{\boldsymbol{i=1}}^{\boldsymbol{t}} \boldsymbol{l}\left(\hat{\boldsymbol{y}}_{\boldsymbol{i}}^{(\boldsymbol{t})}, \boldsymbol{y}_{\boldsymbol{i}}\right) + \sum_{\boldsymbol{i=1}}^{\boldsymbol{t}} \Omega(\boldsymbol{f}_{\boldsymbol{i}}) \quad \textbf{(2.5)}$$

dimana,

y_i : nilai aktual

l : fungsi loss

Ω : fungsi regularisasi

$$\Omega(\boldsymbol{f}) = \boldsymbol{\gamma T} + \frac{\mathbf{1}}{\mathbf{2}} \boldsymbol{\lambda}\|\boldsymbol{\omega}\|^{\mathbf{2}} \quad \textbf{(2.6)}$$

λ : parameter regulasi dengan nilai 1 secara defaul

T : jumlah daun yang dimiliki pohon tersebut

Ω : menghitung nilai γ untuk memastikan kompleksitas model.

ω : nilai keluaran untuk berat daun yang digunakan (Nugraha & Irawan, 2023).

Berikut merupakan bagian dari proses penyusunan ataupun tahapan pada Algoritma *XGBoost*. Mempertimbangkan himpunan data yang berisi dua variabel, X dan Y.

Tabel 2.2 *Dataset* untuk Membangun *XGBoost Tree*

X	Y
2	0
8	1
12	1
18	0

Pengaturannya adalah min_child_weight = 0, reg_lambda = 0, max_depth = 2, learning_rate = 1, gamma = 2, n_estimators atau model yang dibangun sebanyak dua, dan base_score = 0,5.

a. Prediksi awal

Prediksi awal, atau base_score, sebesar 0,5 ditetapkan untuk setiap titik data dalam kumpulan data.

dimana :

$$f_0(X) = h_0(X) = 0.5$$

b. Perhitungan residual atau kesalahan

Tentukan sisa Y setiap titik data berdasarkan prediksi sebelumnya.

Tabel 2.3 Perhitungan Nilai kesalahan

X	Y	$f_0(X)$	$\widehat{Y} = Y - f_0(X)$
2	0	0.5	-0.5
8	1	0.5	0.5

X	Y	f_0(X)	$\widehat{Y} = Y - f_0(X)$
12	1	0.5	0.5
18	0	0.5	-0.5

c. Latih Model

Dengan menggunakan data [X, Y], model pembangunan pohon, atau M1, dilatih dalam pelatihan model awal. Berbeda dengan pohon Seleksi, model tersebut merupakan pohon XGboost unik yang dibangun secara berbeda. Istilah-istilah berikut harus dipahami sebelum membuat pohon XGBoost dengan menentukan rumus untuk menyelesaikan masalah pengoptimalan XGBoost.

$$Gain = \left(Left_{similarity} + Right_{similarity}\right) - Root_{similarity}$$

$$Similarity\ Score = \frac{(\sum \hat{Y}_i)^2}{\sum[Previous\ f_i(x)\ .(1 - Previous\ f_i(x)] + \lambda}$$

$$OutputValue = \frac{(\sum \hat{Y}_i)}{\sum[Previous\ f_i(x)\ .\ (1 - Previous\ f_i(x)] + \lambda}$$

$\hat{Y}_i$: Residual ke-i

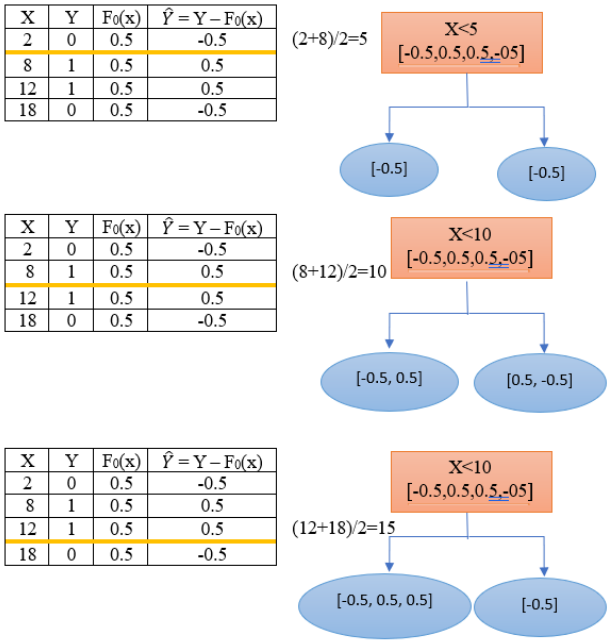
λ : Reg_lambda

$Previous\ f_i(x)$: Probabilitasi sebelumnya

- Perhitungan nilai hanya untuk menghitung akar pohon
- Perhitungan similarity untuk semua node
- Perhitungan output value hanya untuk leaf node
- Lambda merupakan parameter, apabila nilai lambda meningkat akan menghasilkan pruning lebih banyak node pada pohon yang dibangun.

Berikut adalah tahapan dalam membangun pohon Algoritma *XGBoost* :

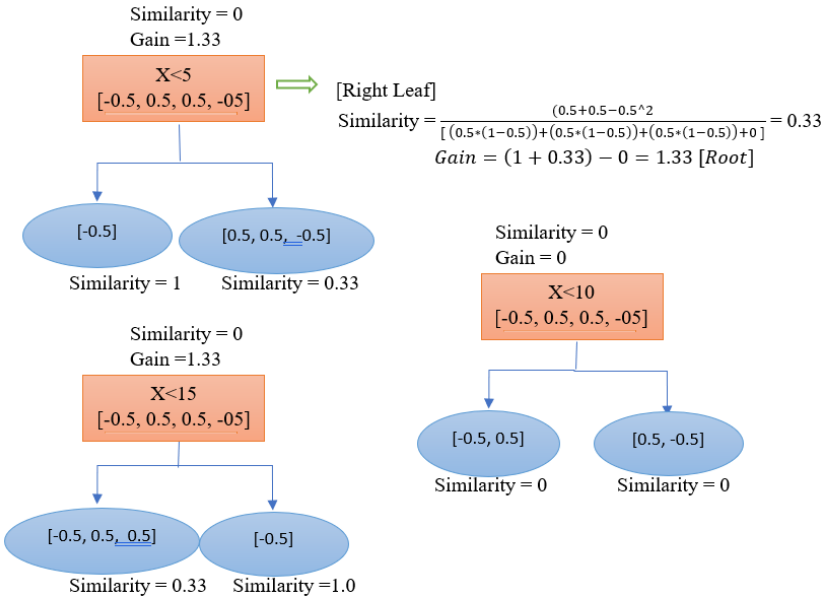
1. Data dibagi menjadi dua partisi menggunakan berbagai pemisahan potensial untuk membangun pohon.



Gambar 2.8 Contoh Membangun *XGBoost Tree*

Untuk mengidentifikasi batas-batas batang, nilai rata-rata antara dua lokasi percabangan atau pemisahan dihitung, dan nilai-nilai yang tersisa dikirimkan ke setiap simpul cabang. Dalam satu set data dengan n data, total pohon n-1 dapat dibangun.

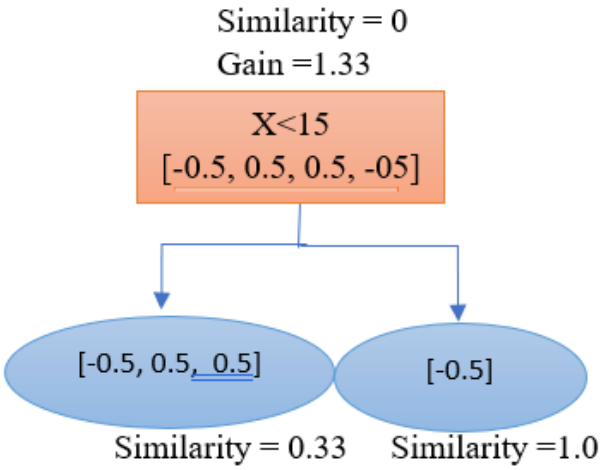
2. Tentukan pemisahan terbaik, hitung skor kesamaan, dan lihat keseluruhan pohon yang dibangun.



Gambar 2.9 Contoh Perhitungan *Similitar*

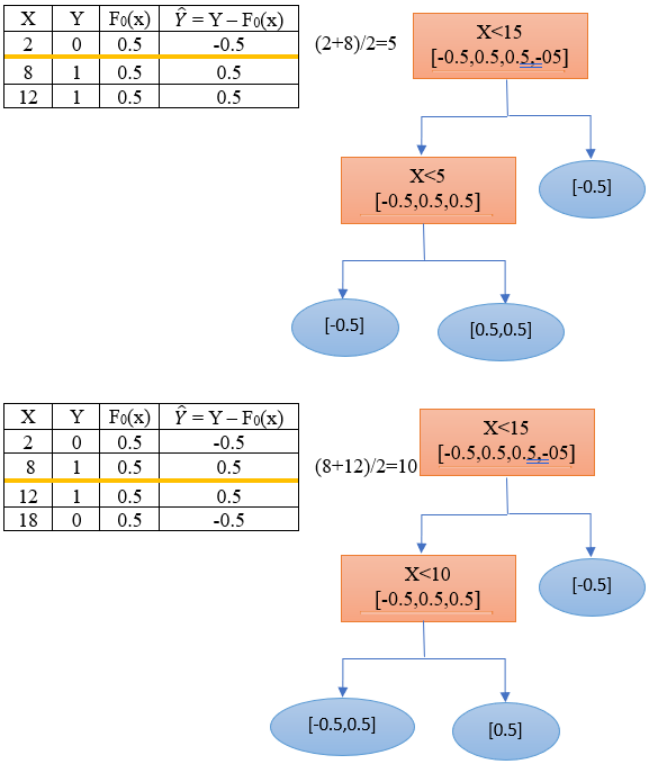
Pohon dengan x < 15 pada Gambar 2 memiliki nilai maksimum 1,33; pilihlah pohon dengan nilai terbaik

3. Membuat pohon yang memiliki nilai tertinggi, lakukan pemisahan kembali hingga kedalaman maksimal.



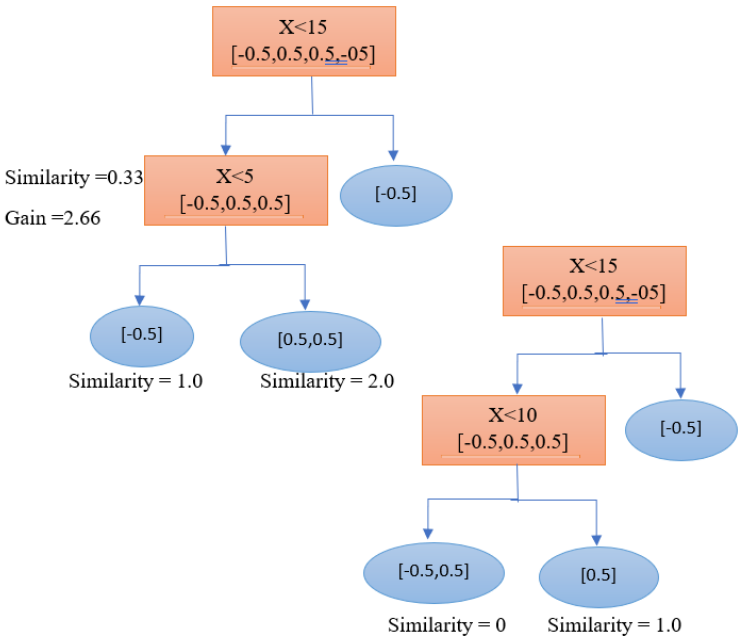
Gambar 2.10 Contoh Split pada XGBoost Tree

Bangun kembali pohon dengan memisahkan data dari daun kiri karena nilai max_depth digunakan dua kali.



Gambar 2.11 Contoh *Split* pada Turunan Percabangan

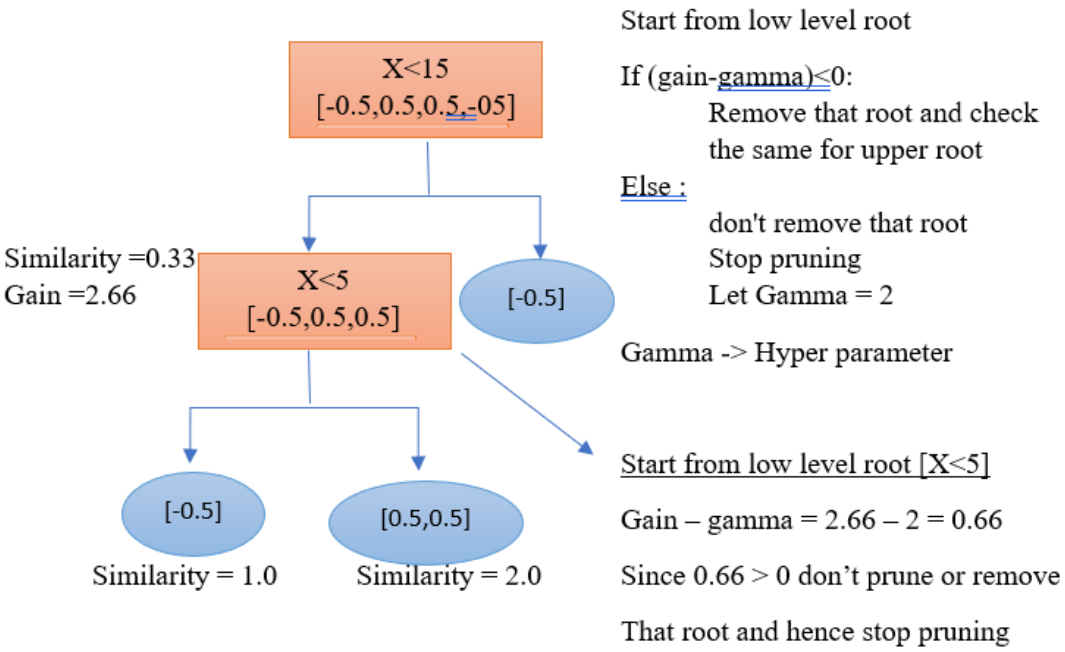
Hitung ulang nilai kesamaan, pilih akar internal dengan nilai tertinggi, lalu pisahkan sekali lagi pada cabang berikut.



Gambar 2.12 Contoh Perhitungan Simitary pada Split Lanjutan

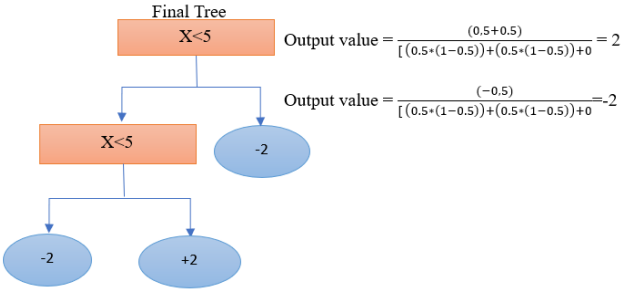
Karena menghasilkan nilai maksimal sebesar 2,66, akar internal X< 5 dipilih untuk konstruksi pohon *XGBoost*.

4. Pemangkasan adalah proses mengurangi ukuran pohon keputusan dengan menghilangkan bagian-bagian yang kurang kokoh setelah pohon dibangun.



Gambar 2.13 Proses *Prunning* pada Pohon

5. Untuk memperoleh pohon akhir dalam model 1, kita harus menghitung nilai keluaran untuk setiap cabang karena beberapa cabang memiliki beberapa residual.



Gambar 2.14 Contoh Perhitungan *Output Value*

6. Semua titik data harus melalui pohon akhir untuk menghitung prediksi dari model 1 dan memperoleh h_1(x). Prediksi h_1(x) dan nilai residual kemudian harus ditentukan.

   Diberikan nilai *learning_rate* = 1.0

   Maka :

$$f_i(x) = \sigma\left[\left(\frac{h_0(x)}{1-h_0(x)}\right) + (\eta\, x\, h_1(x))\right]$$

   Memecahkan $f1(x)$ pada klasifikasi didapatkan:

$$f_1(x) = \sigma(0 + 1.x\, h_1(x))$$

   Fungsi sigmoid:

$$\sigma(x) = \frac{1}{1+\exp(-x)}$$

Tabel 2.4 Tabel Perhitungan Nilai Prediksi pada Model-1

X	Y	$h_1(x)$	$f_1(x) = \sigma(0 + 1\,x(h_1(x)))$	$\hat{Y} = y - f_1(x)$
2	0	-2	-0.5	-0.11
8	1	2	0.5	0.12
12	1	2	0.5	0.12
18	0	-2	-0.5	-0.11

d. Untuk membangun pohon tambahan, ulangi Langkah 3.

   Sebagai hasil dari optimasi pada algoritma sebelumnya, *XGBoost* dapat melakukan berbagai fungsi seperti regresi, klasifikasi, dan *ranking* yang membuatnya lebih efisien dan dapat diskalakan dalam berbagai situasi.

2.3 *Python*

Python adalah salah satu bahasa pemrograman yang paling banyak digunakan oleh para *developer* dan perusahaan besar untuk membuat aplikasi berbasis desktop, web, dan *mobile*. Guido van Rossum dari Belanda dan pertama kali ditunjukkan

pada tahun 1991 dalam versi 1.0. Pada tanggal 16 Oktober 2000, dirilis versi 2.0, yang memiliki beberapa peningkatan besar, seperti pengidentifikasi *garbage cycle* dan dukungan untuk *unicode*. Pada tanggal 3 Desember 2008, Python 3.0 akhirnya tersedia. Ukuran Python telah meningkat tiga kali lipat pada tahun 2020, dan versi terbarunya adalah 3.8.2. (Suma & Pasundan, 2021). Namanya diambil dari acara televisi favoritnya, *Monty Python's Flying Circus*. *Python* menjadi bahasa pemrograman yang banyak digunakan dalam pendidikan dan industri karena Van Rossum mengembangkannya sebagai hobi(Muhammad Romzi & Kurniawan, 2020).

Python mendukung banyak hal, termasuk pengembangan game, perhitungan ilmiah, akses basis data, dan GUI desktop. Selain itu, python memiliki pustaka yang memungkinkan penulisan kode interaktif menggunakan *jupyter notebook*, yang dimaksudkan untuk pekerjaan konteks analisis data. Alternatif lain adalah Google Colab dengan layanan yang mempermudah ketersediaan GPU dan TPU dalam proses *machine learning* (Gusliana, 2021).

2.4 Data Time Series

Serangkaian pengamatan yang terurut berdasarkan waktu dengan jarak yang sama dikenal sebagai Data Time Series. Karena data dikumpulkan dalam interval waktu harian, mingguan, atau bulanan, jenis data ini sering ditemui dalam keseharian. Ada pola yang dapat dilihat dalam data yang dikumpulkan. Urutan waktu, ada tiga pola: pola tren, pola siklis, dan pola musiman. Pola yang berulang berulang pada interval tertentu disebut pola musiman. Interval data waktu terbagi menjadi dua domain: domain waktu dan domain frekuensi. Daerah waktu melihat autokorelasi yang signifikan, kestasioneran data, penaksiran parameter model

regresi deret waktu, dan peramalan (*forecasting*). Namun, daerah frekuensi menyelidiki frekuensi tersembunyi dalam data musiman yang sulit diperoleh dalam daerah waktu tertentu. Tujuannya adalah untuk mengetahui ciri-ciri data tertentu (Al'afi et al., 2020).

2.5 Metode Evaluasi

2.5.1 *Root Mean Squared Error (RMSE)*

RMSE yaitu metode alternatif untuk mengevaluasi teknik peramalan yang digunakan untuk tingkat akurasi hasil perkiraan suatu model. Nilai yang dihasilkan RMSE merupakan nilai rata-rata kuadrat dari jumlah kesalahan pada model prediksi(Sanjaya & Heksaputra, 2020). Adapun rumus *RMSE* tersebut sebagai berikut(Chicco et al., 2021).

$$\boldsymbol{RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(X_i - Y_i)^2}} \qquad \textbf{(2.6)}$$

di mana,

X_i : nilai hasil prediksi

Y_i : nilai aktual

m : jumlah data

Semakin kecil (mendekati 0) nilai *RMSE* maka hasil prediksi semakin akurat sedangkan semakin besar (mendekati $+\infty$) maka hasil prediksi semakin buruk(Suprayogi et al., 2014).

2.5.2 *Coefficient of Determination (R2)*

Coefficient of Determination mengambil nilai pada rentang $(-\infty, 1]$ menurut hubungan antara nilai kebenaran dan prediksi. Berikut kasus-kasus utamanya yaitu:

- $R^2 \geq 0$: regresi linier tanpa batasan, R^2 non-negatif dan bersesuaian dengan kuadrat koefisien korelasi berganda.

- $R^2 = 0$: garis yang dipasang (*hyperplane*) adalah horizontal jika variabel independen tidak berkorelasi.

Adapun rumus *Coefficient of Determination* (R^2) sebagai berikut:

$$\boldsymbol{R^2 = 1 - \frac{\sum_{i=1}^{m}(X_i - Y_i)^2}{\sum_{i=1}^{m}(\bar{Y} - Y_i)^2}} \qquad \textbf{(2.7)}$$

Coefficient of Determination dapat diartikan sebagai proporsi varians variabel respon yang dapat diprediksi dari variabel prediktor. Nilai R^2 semakin mendekati $-\infty$ maka hasil prediksi semakin buruk sedangkan nilai R^2 semakin mendekati 1 maka hasil prediksi semakin akurat (Suprayogi et al., 2014).

2.6 Penelitian Terdahulu

Tabel 2.5 Penelitihan Terdahulu

NO	Nama Peneliti (Tahun)	Judul	Hasil Penelitian
1.	Farhanuddin, Sarah Ennola Karina Sihombing, Yahfizham (2024)	Komparasi Multiple Linear Regression dan Random Forest Regression Dalam Memprediksi Anggaran Biaya Manajemen Proyek Sistem Informasi	Dalam penelitian ini membahasa tentang pengujian terhadap dua model algoritma, yaitu multiple linear regression dan random forest regression. Hasil dari penelitian ini Hasil pengujian menunjukkan bahwa model random forest regression memiliki nilai akurasi yang lebih tinggi, sebesar 81,5%, dibandingkan dengan nilai akurasi model multiple linear regression sebesar 62,9%. Hal ini menunjukkan bahwa regresi hutan acak lebih berhasil dalam memperkirakan biaya anggaran proyek sistem informasi berdasarkan data historis..

2.	Fadil Indra Sanjaya, Dadang Heksaputra (2020)	Prediksi Rerata Harga Beras Tingkat Grosir Indonesia dengan Long Short Term Memory	Pendekatan jaringan saraf LSTM, yang merupakan alat yang berguna untuk meramalkan harga beras untuk tahun mendatang, dibahas dalam artikel ini. Hasil yang diperoleh dari evaluasi hasil prediksi pada epoch 20 hingga 1000 menghasilkan nilai Root Mean Square Error (RMSE) minimum sebesar 0,43, yang menunjukkan bahwa pendekatan LSTM dapat digunakan untuk meramalkan harga beras di tingkat grosir Indonesia dengan cukup berhasil.
3.	Muhamad Fadli, Rizal Adi Saputra (2023)	Klasifikasi dan Evaluasi Performa Model Random Forest untuk Prediksi Stroke	Penelitian ini membahas tentang mengembangkan dan mengevaluasi kinerja model klasifikasi untuk prediksi stroke menggunakan algoritma Random Forest. Hasil dari penelitian ini bahwa dengan akurasi sebesar 93,6%, presisi sebesar 91,4%, recall sebesar 96,1%, dan F1-Score sebesar 93,7%, model Random Forest menunjukkan bahwa teknik Machine Learning seperti Random Forest dapat digunakan sebagai cara yang efektif untuk memprediksi stroke berdasarkan data klinis dan faktor risiko.
4.	Yoga Religia, Agung Nugroho, Wahyu Hadikristanto (2021)	Analisis Perbandingan Algoritma Optimasi pada Random Forest untuk Klasifikasi Data Bank Marketing	Penelitian ini menguji tentang algoritma RF dengan optimasi GA dan Bagging. Dari hasil penelitian ini menunjukkan bahwa algoritma RF belum mampu meningkatkan akurasi untuk klasifikasi set data Bank Marketing

			dengan menggunakan baik optimasi GA maupun Bagging; tingkat akurasi yang diperoleh baik dengan optimasi maupun tanpanya adalah 88,30%.
5.	Alan Catur Nugraha, Mohammad Isa Irawan (2023)	Komparasi Deteksi Kecurangan pada Data Klaim Asuransi Pelayanan Kesehatan Menggunakan Metode Support Vector Machine (SVM) dan Extreme Gradient Boosting (XGBoost)	Penelitian ini membahas tentang bagaimana cara mendeteksi fraud pada pelayanan kesehatan dengan cara metode machine learning yaitu Support Vector Machine(SVM) dan Extreme Gradient (XGBoost). Hasil dari perbandingan dalam klasifikasi SVM dan XGBoost, ditemukan bahwa metode XGBoost memiliki nilai akurasi yang lebih baik dan recall yang lebih baik, dengan nilai 0.98 dan 0.984 pada data, sedangkan SVM memiliki nilai 0.874 dan 0.854.
6.	Herni Yulianti, Sri Elina, Oni Soesanto, and Yuana Sukmawaty (2022)	Penerapan Metode Extreme Gradient Boosting (XGBOOST) pada Klasifikasi Nasabah Kartu Kredit	Penelitian ini membahas tentang mengklasifikasi nasabah kartu kredit yang macet menggunakan teknik machine learning yaitu metode XGBoost yang mana metode ini berguna untuk analisis regresi dan klasifikasi berdasarkan Gradient Boosting Decision Tree (GBDT). Dari hasil penelitian ini memberikan bukti bahwa algoritma eXtreme Gradient Boosting dapat meningkatkan kinerjanya dalam klasifikasi pelanggan kartu kredit dengan akurasi sebesar 80,039 persen, presisi sebesar 81,338 persen, dan nilai recall sebesar 96,854 persen.

7.	Zian Asti Dwiyanti, Cahyo Prianto (2023)	Prediksi Cuaca Kota Jakarta menggunakan Metode Random Forest	Pendekatan Random Forest dan data cuaca historis terpercaya dari situs web OpenData Jakarta digunakan dalam penelitian ini untuk meramalkan cuaca di Kota Jakarta. Dengan akurasi, presisi, dan recall sebesar 0,71, skor F1 sebesar 0,70, dan ROC-AUC sebesar 0,92, temuan penelitian menunjukkan bahwa model Random Forest secara efektif mengklasifikasikan cuaca dengan mempertimbangkan akurasi, presisi, dan keseimbangan antara recall dan presisi.
8.	Ardytha Luthfiarta, Aris Febriyanto, Heru Lestiawan, Wibowo Wicaksono (2020)	Analisa Prakiraan Cuaca dengan Parameter Suhu, Kelembaban, Tekanan Udara, dan Kecepatan Angin Menggunakan Regresi Linear Berganda	Memprediksi prakiraan cuaca untuk berfungsi sebagai sistem peringatan dini jika terjadi perubahan cuaca yang tiba-tiba atau bahkan parah merupakan tujuan dari pekerjaan ini. Koefisien regresi sebesar 0,147 (14,7%) memiliki dampak yang menguntungkan dan signifikan terhadap temuan penelitian mengenai suhu udara. Koefisien regresi 0,078 (7,8%) memiliki dampak positif dan signifikan terhadap tekanan udara. Koefisien regresi sebesar -0,356 (-35,6%), yang menunjukkan dampak negatif tetapi masih substansial pada kelembaban udara. Curah hujan memiliki efek positif tetapi tidak signifikan pada variabel kecepatan angin. Fakta bahwa nilai P = 0,523 lebih tinggi dari 0,05

			berfungsi sebagai bukti. Menurut uji hipotesis, curah hujan (Y) dipengaruhi oleh empat variabel: suhu udara (X1), tekanan udara (X2), kelembaban udara (X3), dan kecepatan angin (X4).
9.	Hafiz Akbar, Wisnu Karya Sanjaya (2023)	Kajian Performa Metode Class Weight Random Forest pada Klasifikasi Imbalance Data Kelas Curah Hujan	Kontrol variabel seperti kecepatan angin rata-rata, suhu permukaan dan suhu udara maksimum, suhu udara minimum, rata-rata radiasi matahari maksimum dan rata-rata, dan penguapan yang dapat digunakan sebagai data masukan untuk klasifikasi kelas cerah-hujan diperiksa dalam penelitian ini. Temuan penelitian membandingkan nilai metrik yang diterapkan pada model dengan berbagai pengaturan metode bobot kelas, termasuk Seimbang, {0: 0,5, 1: 1, 2: 1, 3: 1, 4: 1}, {0: 0,5, 1: 1,5, 2: 1,5, 3: 1,5, 4: 1,5}, dan {0: 0,5, 1: 1,5, 2: 1,5, 3: 1,5, 4: 1,5}. Nilai bobot kelas {0: 0,5, 1: 1,5, 2: 1,5, 3: 1,5, 4: 1,5} memiliki hasil terbaik pada data pelatihan dan akurasi 73% pada data pengujian, tetapi tidak mampu mengatasi ketidakseimbangan data secara efektif.
10.	Ghaitsa Amany Mursianto, Isma'il Muhammad Falih, Muhammad Irfan, Tiara Sakinah, Desta Sandya Prasvita (2021)	Perbandingan Metode Klasifikasi Random Forest dan XGBoost Serta Implementasi Teknik SMOTE pada Kasus Prediksi Hujan	Kategorisasi prakiraan curah hujan untuk beberapa hari mendatang menggunakan Random Forest dan XGBoost dibahas dalam studi ini. Kelas terbaik atau paling akurat akan diidentifikasi menggunakan metode ini. Temuan studi menunjukkan bahwa klasifikasi Random Forest dengan resampling memiliki

			akurasi terbaik sebesar 95,59%, sedangkan klasifikasi Random Forest paling akurat tanpa resampling adalah 89,54%. Random Forest dan XGBoost meningkatkan akurasi klasifikasinya masing-masing sebesar 6,05% dan 5,38%, dibandingkan dengan temuan akurasi klasifikasi tanpa resampling. Kelas tersebut dapat diidentifikasi secara efektif dengan resampling SMOTE, sebagaimana dibuktikan oleh peningkatan recall sebesar 88,05% pada random forest dan peningkatan sebesar 78,75% pada XGBoost.
11.	Afikah Agustiningsih, Yulian Findawati, Irwan Alnarus Kautsar (2023)	*Classification Of Vocational High School Graduates' Ability in Industry Using Extrame Gradient Boosting (XGBoost), Random Forest, and Logistic Regression*	Konstruksi sistem yang dapat mengkategorikan lulusan sekolah menengah kejuruan sebagai data penilaian untuk lembaga pendidikan dibahas dalam penelitian ini. Tiga algoritma pembelajaran mesin—XGBoost, Random Forest, dan Regresi Logistik—digunakan dalam penyelidikan ini. Algoritma XGBoost menghasilkan skor pelatihan sebesar 91,70%, skor ujian sebesar 66,88%, dan skor akurasi sebesar 67% untuk ketiga pendekatan tersebut. 97,36% untuk pelatihan, 68,71% untuk pengujian, dan 67% untuk ketepatan adalah hasil dari algoritma Random Forest. Sebaliknya, Regresi Logistik menghasilkan tingkat akurasi 50%, skor pelatihan

			51,14%, dan skor ujian 50,43%.
12.	Davide Chicco, MatthijsJ.Warrens, Giuseppe Jurman	*The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation*	Dalam penelitian ini membahas tentang membandingkan beberapa tingkat statistik umum digunakan dalam literatur ilmiah untuk evaluasi tugas regresi, dan menjelaskan keunggulan R-squared dibandingkan SMAPE, MAPE, MAE, MSE dan RMSE. Penelitian ini berfokus pada dua tingkat yang benar-benar menghasilkan skor tinggi hanya jika sebagian besar elemen kelompok kebenaran dasar telah diprediksi dengan benar: koefisien determinasi (juga dikenal sebagai R-squared atau R^2) dan kesalahan persentase absolut rata-rata simetris (SMAPE). Hasil dari penelitian ini menunjukkan bahwa koefisien determinasi (R-kuadrat) lebih informatif dan jujur dibandingkan SMAPE, dan tidak memiliki keterbatasan interpretasi seperti MSE, RMSE, MAE, dan MAPE.
13.	Cafer Mert Yes¸ilkanat	*Spatio-temporal estimation of the daily cases of COVID-19 in worldwide using random forest machine learning algorithm*	Penelitian ini membahas tentang kinerja algoritma machine learningn metode Random Forest (RF) diselidiki dalam memperkirakan jumlah kasus dalam waktu dekat untuk 190 negara di dunia dan dipetakan dibandingkan dengan hasil kasus terkonfirmasi aktual. Pada akhir penelitian ditemukan bahwa R^2 nilai untuk

			pengujian sub-data estimasi model RF berkisar antara 0,843 dan 0,995 (rata-rata R^2= 0,959), dan nilai RMSE antara 141,76 dan 526,18 (rata-rata RMSE = 259,38); dan itu R^2 nilai untuk memperkirakan sub-data berkisar antara 0,690 dan 0,968 (rata-rata R^2 = 0,914), dan nilai RMSE antara 549,73 dan 2500,79 (rata-rata RMSE = 909,37). Hasil ini menunjukkan bahwa algoritma pembelajaran mesin hutan acak berkinerja baik dalam memperkirakan jumlah kasus dalam waktu dekat jika terjadi epidemi seperti Novel Virus Corona, yang mewabah secara tiba-tiba dan menyebar dengan cepat.
14.	Galih Ashari Rakhmat, Wisnu Mutohar (2023)	Prakiraan Hujan menggunakan Metode Random Forest dan Cross Validation	Untuk mencapai kinerja model terbaik, studi ini menguji pendekatan random forest menggunakan komponen hiperparameter e_estimator dan max_depth serta strategi validasi silang. Menurut temuan studi, model random forest yang berkinerja terbaik adalah model dengan nilai n_estimator 100, max_depth None, dan validasi silang 3. menghasilkan matriks penilaian dengan nilai-nilai berikut: MAE: 0,186, RMSE: 0,290, dan MSE: 0,086. Selain itu, tingkat keberhasilan 60% diperoleh dalam pengujian penerapan identifikasi kondisi hujan dari 30 kasus data.
15.	Muhamad Syukron, Rukun Santoso,	Perbandingan Metode Smote	Metode Smote Random Forest dan Smote XGBoost

	Tatik Widiharih (2020)	*Random Forest* dan Smote *XGBoost* untuk Kalsifikasi tingkat Penyakit Hepatitis C pada *Imbalance Class* Data	untuk Mengklasifikasikan Tingkat Penyakit Hepatitis C dalam Data Kelas Ketidakseimbangan dibandingkan dalam makalah ini. Menurut temuan penelitian, XGboost dan random forest memiliki nilai recall kurang dari 2% dan akurasi sekitar 74%. Akurasi dan recall lebih dari 75% dicapai melalui random forest SMOTE dan SMOTE XGboost. Sementara SMOTE XGboost berkinerja lebih baik dalam kelas sirosis, random forest SMOTE memiliki akurasi yang lebih tinggi dalam memprediksi kelas fibrosis. Variabel hasil uji laboratorium adalah variabel yang memiliki dampak terbesar dalam mendefinisikan stadium hepatitis C.

BAB III

METODOLOGI PENELITIAN

3.1 Pendekatan Penelitian

Metode yang digunakan dalam penelitian ini adalah penelitian kuantiatif yang bertujuan untuk menguji hipotesis dengan menggunakan teori – teori yang sudah ada. Metode penelitian kuantitatif ini menggunakan matematika untuk menjelaskan proses dan hasil penelitian. Metode penelitian ini digunakan untuk memberikan garis besar tindakan yang akan diambil oleh peneliti dalam menyelesaikan permasalahan membandingkan metode *random forest* dan *XGBoost* pada cuaca. Penelitian ini dirancang untuk memberikan kejelasan dalam setiap langkah, serta untuk menjamin bahwa pengumpulan data, analisis, dan interpretasi hasil dilakukan dengan benar dan teratur.

3.2 Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan dalam penelitian ini adalah teknik skunder. Data sekunder banyak digunakan untuk penelitian dengan tujuan mengetahui perspektif alternatif dari pertanyaan riset yang telah dilakukan sebelumnya. Data sekunder adalah pengumpulan data yang dapat diperoleh melalui kajian literatur seperti kajian jurnal, buku, hasil penelitian, situs web, artikel jurnal ataupun catatan-catan yang ada di internet. Oleh karena itu, pada penelitian ini memakai data sekunder yang diambil dari BMKG (https://dataonline.bmkg.go.id/data_iklim) yaitu data cuaca yang ada di Sumatera Utara dari 01 April 2021 sampai dengan 31 Maret 2024. Adapun data variabel tersebut adalah :

Tabel 3. 1 Data Cuaca Sumatera Utara

No	Tanggal	Temperatur	Kelembapan	Kecepatan angin
1	01-04-2021	29.8	70	2
2	02-04-2021	29.2	76	1
3	03-04-2021	27.8	81	2
4	04-04-2021	28.8	78	1
5	05-04-2021	28.3	82	2
6	06-04-2021	27.8	84	1
7	07-04-2021	25.9	91	1
8	08-04-2021	28.1	77	2
9	09-04-2021	28.7	75	1
10	10-04-2021	28.2	79	1
11	11-04-2021	28.3	82	1
12	12-04-2021	28.2	79	2
13	13-04-2021	26.8	85	1
14	14-04-2021	26.9	84	2
15	15-04-2021	27.2	84	1
16	16-04-2021	26.9	84	1
17	17-04-2021	27.9	80	1
18	18-04-2021	28.8	77	1
19	19-04-2021	27	80	2
20	20-04-2021	28.5	79	1
...	...	...	...	...
1092	27-03-2024	28.4	83	1
1093	28-03-2024	31.4	66	0
1094	29-03-2024	29.1	78	1
1095	30-03-2024	28.5	80	2
1096	31-03-2024	31.5	67	2

3.3 Alat Bantu Penelitian

Penelitian ini pastinya memerlukan alat bantu yang ada dalam proses pengerjaannya, terdapat beberapa alat yang digunakan dalam pengerjaan penelitian ini adalah sebagai berikut :

1. Perangkat Keras (*Hardware*)

   Satu unit laptop acer dengan spesifikasi yaitu :

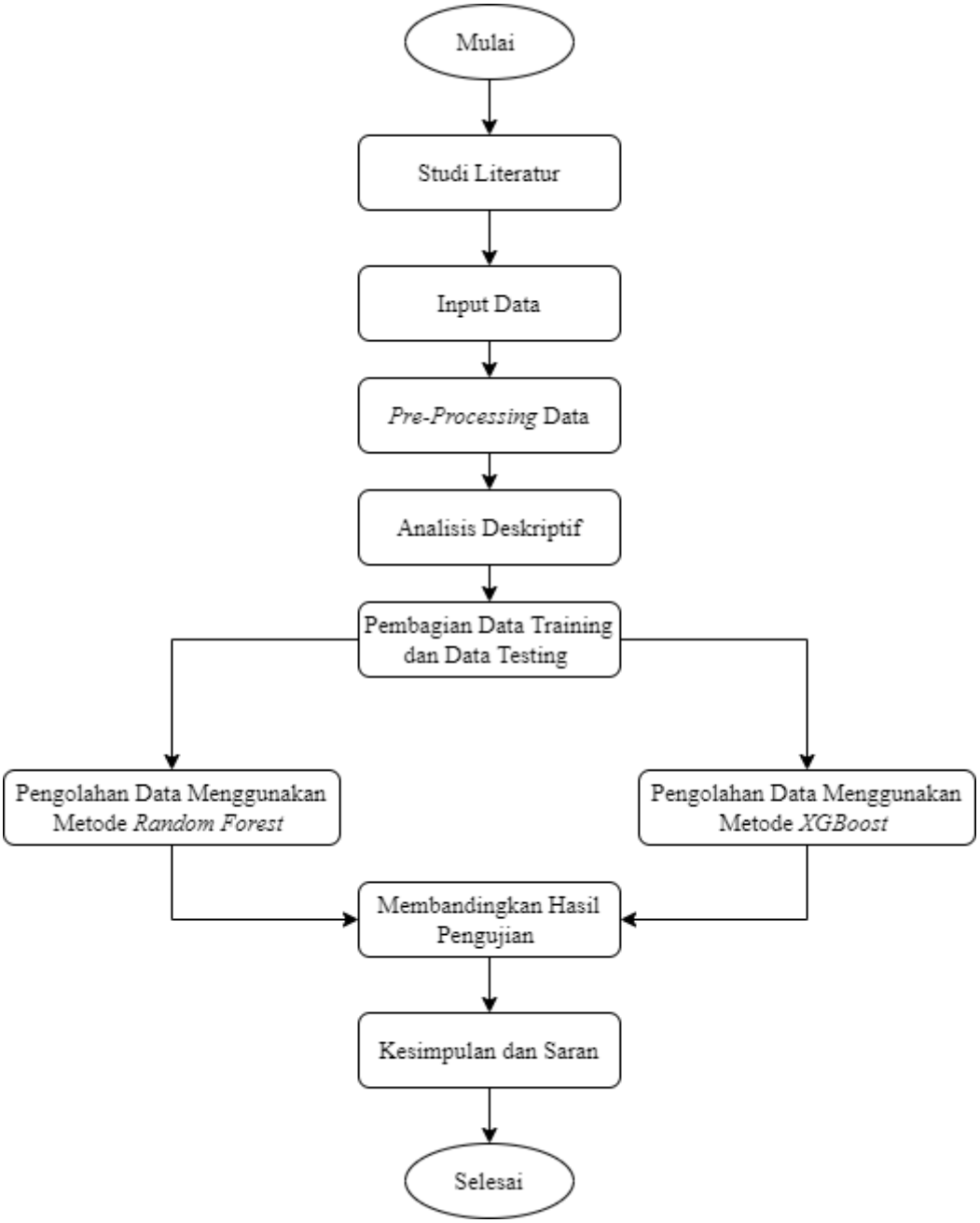
   - *Processor* 11*th Gen Intel*(*R*) *Core*(*TM*) *i*5-1155G7
   - *RAM* : 8 GB
   - *Storage*: 256 *GB SSD*

2. Perangkat Lunak (*Software*)

   - *Windows* 11 *Home*
   - *Microsoft Word* 2011
   - *Python*

3.4 Perancangan Analisis

Tujuan untuk menghasilkan akurasi perbandingan metode *random forest* dan *XGBoost* yang lebih akurat pada cuaca yang sesuai dengan kebutuhan pengguna, maka dari itu, peneliti menggunakan diagram *flowchart* untuk menggambarkan alur perbandingan metode tersebut guna memberi gambaran mengenai proses perbandingan dari metode yang satu ke metode yang lain menggunakan bahasa pemrograman python agar lebih mudah dipahami dan mudah dimengerti. Dibawah ini merupakan alur perbandingan metode *random forest* dan *XGBoost* untuk menentukan mana metode yang lebih akurat pada cuaca dalam bentuk *flowchart* :



Gambar 3. 1 Alur Penelitian

Adapun penjelasan mengenai perancangan analisis pada *flowchart* diatas adalah sebagai berikut :

1. Proses pertama yang dilakukan Studi Literatur yaitu untuk mencari referensi tentang masalah penelitian. Referensi yang bisa didapatkan dari jurnal

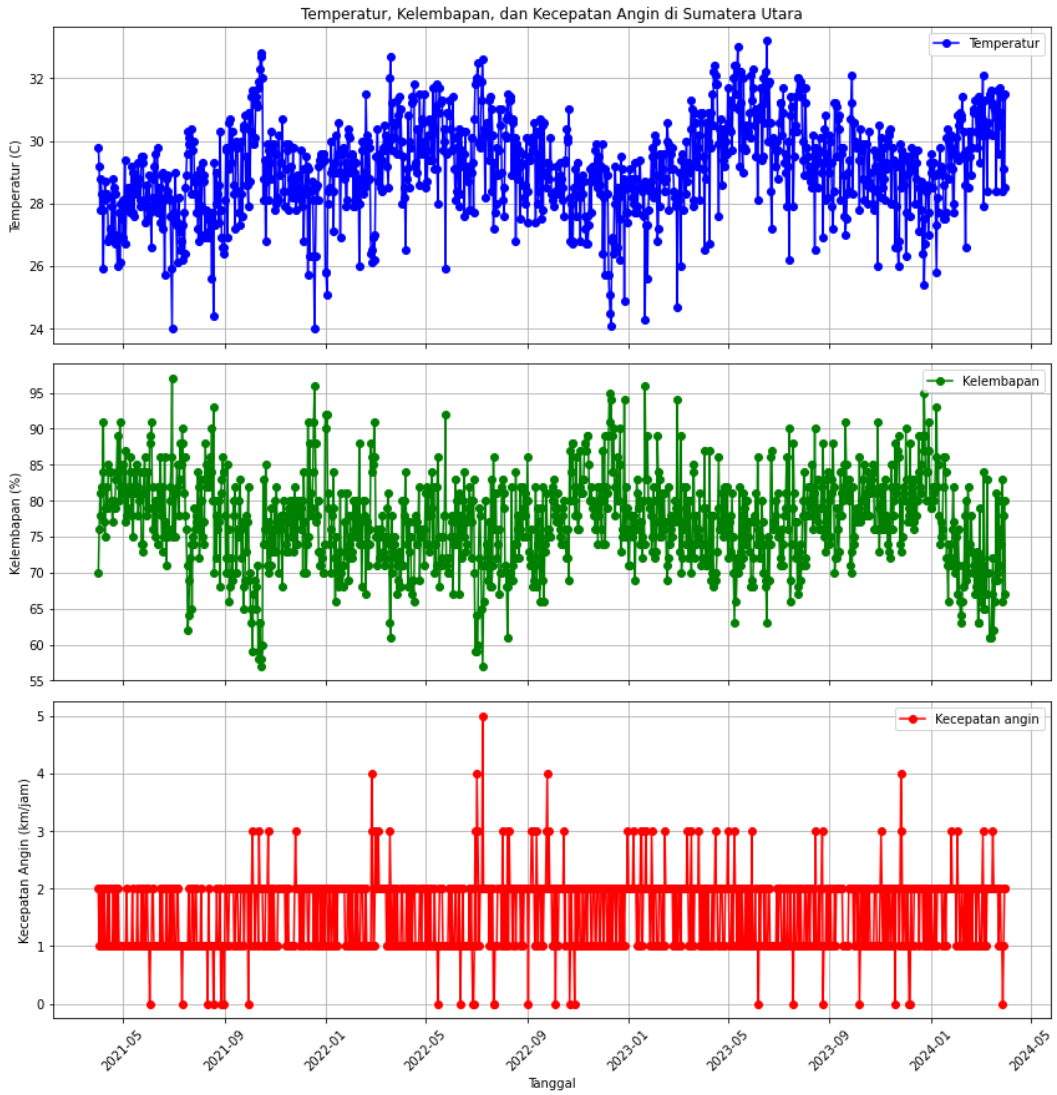
penelitian terdahulu, buku yang berhubungan dengan permasalahan penelitian dan informasi yang di dapat dari artikel di internet

2. Proses kedua yang dilakukan adalah memasukkan data ke dalam program yang akan digunakan untuk melakukan proses analisis data.
3. Proses ketiga yang dilakukan adalah *Pre-Processing* Data tujuannya untuk membagi data numerik dan data kategorik.
4. Analisis deskriptif prosedur keempat yang digunakan, bertujuan untuk menentukan bagaimana data yang akan digunakan dijelaskan.
5. Pemisahan data pelatihan dan pengujian, dengan 80% dari total data merupakan data pelatihan dan 20% merupakan data pengujian, merupakan prosedur keenam yang dilakukan.
6. Proses keenam yang dilakukan adalah pada pengolahan data menggunakan metode random forest dan xgboost guna untuk membuktikan metode yang lebih akurat pada cuaca dengan menggunakan data variabel yang diambil dari BMKG sebelumnya, dengan menggunakan bantuan bahasa pemrograman yaitu python.
7. Proses ketujuh yang dilakukan adalah membandingkan dari hasil kedua metode *Random Forest* dan *XGBoost*
8. Kemudian melakukan inrterpretasi dari metode terbaik yang didapatkan.
9. Proses selanjutnya menentukan kesimpulan dan saran. Kesimpulan akan diambil dari keseluruhan dalam penelitian ini kemudian memberikan saran untuk penelitian selanjutnya.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Grafik Cuaca di Sumatera Utara



Gambar 4.1 Grafik Cuaca di Sumatera Utara.

Pada gambar 4.1, menunjukkan grafik data cuaca di Sumatera Utara yang terdiri dari tiga bagian yaitu temperatur, kelembapan, dan kecepatan angin yang diukur dari tanggal 01 April 2021 hingga 31 Maret 2024. Berikut penjelasan detail dari masing-masing grafik:

1. Temperatur (^oC)

Grafik ini menunjukkan perubahan temperatur harian di Sumatera Utara. Temperatur berfluktuasi antara 24 ^oC hingga 33 ^oC dengan tren umum yang menunjukkan peningkatan suhu selama periode pengamatan. Bulan awal memiliki suhu lebih rendah dan cenderung meningkat hingga pertengahan tahun sebelum sedikit menurun menjelang akhir periode.

2. Kelembapan (%)

Grafik ini menggambarkan persentase kelembapan harian di Sumatera Utara. Kelembapan berkisar 57% hingga 97% dengan kebanyakan data berada di sekitar 70% hingga 90%. Kelembapan relatif stabil sepanjang periode pengamatan, meskippun terdapat beberapa fluktuasi signifikansi di beberapa hari tertentu.

3. Kecepatan angin (km/jam)

Grafik ini memperlihatkan kecepatan angin harian di Sumatera Utara. Kecepatan angin sebagian besar berada 1 dan 2 km/jam, namun terdapat beberapa hari tidak memiliki angin bahkan ada yang meningkat tajam hingga 4 atau 5 km/jam. Ini menunjukkan bahwa kondisi cuaca umumnya tenang, tetapi terdapat beberapa hari dengan angin kencang yang signifikan.

4.2 Pre-Processing Data

Pre-processing data dilakukan dengan membagi data yaitu *training* dan *testing* dengan perbandingan 80 : 20 dari banyak data. Sehingga diperoleh data *training* sebanyak 876 dan data *testing* sebanyak 220.

4.2.1 Import Library

Sebelum memulai mengolah data, perlu mengimport library yang diperlukan untuk mengakses metode *random forest*. Berikut *library* yang digunakan dalam *python*

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from datetime import datetime, timedelta
import numpy as np
import xgboost as xgb
```

Gambar 4.2 Import Library

Berdasarkan dari gambar 4.2 dapat dijelaskan fungsi dari library tersebut

a. *Pandas* digunakan untuk manipulasi dan analisis data.
b. *Matplotlib* untuk membuat visualisasi data yang informatif dan menarik.
c. *Sklearn* menyediakan berbagai macam algoritma *machine learning* yang dapat digunakan berbagai macam tugas.
d. *Datetime* untuk manipulasi tanggal dan waktu sedangkan timedelta adalah kelas yang menyediakan perbedaan antara dua tanggal
e. atau waktu yang berbeda.
f. *Numpy* untuk melakukan operasi matematika dan manipulasi *array* dengan effisien dan mudah.
g. *XgBoost* untuk implementasi algoritma *gradient boosting* yang digunakan pada *machine learning* seperti klasifikasi dan regresi(Farhanuddin et al., 2024).

4.2.2 Import Dataset

Dataset yang digunakan merupakan berformat file excel. Dimana untuk membaca file Excel menggunakan library pandas dan kemudian mencetak data yang telah dibaca. File Excel ini berisi data cuaca Sumatera Utara. Data yang dibaca dari file ini disimpan dalam variabel ‘data’ sebagai DataFrame pandas dapat dilihat pada gambar 4.3

```
data = pd.read_excel("D:/SKRIPSI/DATA CUACA SUMUT.xlsx")
print(data)
```

Gambar 4.3 Import Dataset

4.2.3 Casting

Casting dilakukan untuk mengonversi kolom 'Tanggal' dalam DataFrame ‘data’ menjadi tipe data datetime dengan format tertentu. Tujuannya untuk memastikan bahwa data dalam kolom 'Tanggal' diubah menjadi tipe datetime sehingga dapat dengan mudah digunakan untuk analisis waktu, seperti pengurutan tanggal, perhitungan selisih waktu yang terlihat pada gambar 4.4

```
data['Tanggal'] = pd.to_datetime(data['Tanggal'], format='%d-%m-%Y')
```

Gambar 4.4 Mengubah kolom ke dalam format Datatime

4.2.4 Cleaning

Cleaning dilakukan untuk menghapus spasi ekstra di awal dan akhir setiap nama kolom dalam DataFrame ‘data’ diliat pada gambar 4.5

```
data.columns = data.columns.str.strip()
print(data.columns)
```

Gambar 4.5 Menghapus spasi ekstra dari nama kolom

4.2.5 Creating a figure with multiple subplots

Selanjutnya dilakukan *creating a figure with multiple subplots*, teknik yang sering digunakan dalam visualisasi data untuk menampilkan beberapa plot secara bersamaan dalam satu gambar dapat dilihat pada gambar 4.6

```
# Membuat satu figur dengan tiga sumbu (subplots)
fig, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(12, 12), sharex=True)

# Plot untuk temperatur
ax1.plot(data['Tanggal'], data['Temperatur'], marker='o', linestyle='-', color='b', label='Temperatur')
ax1.set_title('Temperatur, Kelembapan, dan Kecepatan Angin di Sumatera Utara')
ax1.set_ylabel('Temperatur (C)')
ax1.grid(True)
ax1.legend()

# Plot untuk kelembapan
ax2.plot(data['Tanggal'], data['Kelembapan'], marker='o', linestyle='-', color='g', label='Kelembapan')
ax2.set_ylabel('Kelembapan (%)')
ax2.grid(True)
ax2.legend()

# Plot untuk kecepatan angin
ax3.plot(data['Tanggal'], data['Kecepatan angin'], marker='o', linestyle='-', color='r', label='Kecepatan angin')
ax3.set_xlabel('Tanggal')
ax3.set_ylabel('Kecepatan Angin (km/jam)')
ax3.grid(True)
ax3.legend()

# Menyesuaikan layout
plt.tight_layout()
plt.xticks(rotation=45)

# Menampilkan grafik
plt.show()
```

Gambar 4.6 Menampilkan Plot secara bersamaan dalam satu gambar

4.2.6 Feature-target dan predictors split

Selanjutnya melakukan *feature-target split* sebelum melatih model tujuannya untuk mengetahui fitur yang akan digunakan untuk melakukan prediksi (predictors) dan nilai yang ingin diprediksi (target dapat diliat pada gambar 4.7

```
X = data[['Kelembapan', 'Kecepatan angin']]
y = data['Temperatur']
```

Gambar 4.7 Memisahkan fitur prediksi dan target

4.2.7 Split Data

Split data digunakan untuk membagi dataset menjadi subset data pelatihan dan data uji. Dalam pengujian ini, test_size = 0.2 berarti 20% dari total data akan digunakan sebagai data uji, dan sisanya (80%) akan digunakan sebagai data

pelatihan dan random_state = 42 digunakan untuk menetapkan seed acak dapat diliat pada gambar 4.8

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
panjang_X_train = len(X_train)
print(f'Panjang X_train: {panjang_X_train}')
panjang_X_test = len(X_test)
print(f'Panjang X_test: {panjang_X_test}')
```

Gambar 4.8 Pembagian Data Training dan Testing

4.2.8 Modeling Random Forest dan XGBoost

Sebelum melakukan modeling, harus menentukan jumlah panjang x_train dan x_text dari keseluruhan data. Dimana jumlah panjang x_train = 876 dan x_text 220. Setelah jumlah training dan testing ditentukan, selanjutnya membangun model Random Forest dan XGBoost dapat diliihat pada gambar 4.9 dan 4.10

```
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

Gambar 4.9 Model *Random Forets*

```
model = xgb.XGBRegressor(objective ='reg:squarederror', n_estimators=100, learning_rate=0.1, max_depth=3, random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

Gambar 4.10 Model *XGBoost*

4.3 Analisis Data dengan Metode *Random Forest*

Random forest merupakan teknik *mechine learning* yang dapat digunakan untuk tugas klasifikasi maupun regresi. Berdasarkan tujuan penelitian di awal adalah memprediksi cuaca ke depannya maka dilakukan *random forest* regresi di mana variabel prediktor (*X*) adalah kelembapan dan kecepatan angin dan variabel respon (*Y*) merupakan temperatur.

4.3.1 Pengolahan data

Data diolah dengan menggunakan metode *random forest* untuk menentukan hasil prediksi dari model dapat dilihat pada gambar 4.11

```
hasil_prediksi = pd.DataFrame({'Actual': y_test, 'Prediksi': y_pred})
print(hasil_prediksi)
```

Gambar 4.11 Script evaluasi kinerja model

Berikut ini perbandingan beberapa hasil prediksi dan data aktual :

Tabel 4. 1 Data aktual dan Prediksi Metode *Random Forest*

No	Aktual	Prediksi	Selisih
1.	28.6	28.278718	-0,321282
2.	31.0	30.923164	-0,076836
3.	27.4	27.273671	-0,126329
4.	28.7	29.616175	0,916175
5.	31.0	30.465685	-0,534315
...	...	...	....
216.	30.0	29.445380	-0,55462
217.	29.3	29.296954	-0,003046
218.	31.0	31.401435	0,401435
219.	27.9	28.471565	0,571565
220.	30.2	30.888383	0,688383

Tabel 4.1 menunjukkan bahwa nilai prediksi cukup dekat dengan nilai aktual menunjukkan bahwa model tersebut memiliki performa yang cukup baik. Perbedaan antara nilai aktual dan prediksi digunakan untuk mengukur akurasi

model prediksi yang digunakan. Adapun evaluasi model dilakukan dengan menentukan nilai *Root Mean Squared Error* (*RMSE*) dan *Coefficient of Determination* (R^2).

a. *Root Mean Squared Error* (*RMSE*)

Untuk mengevaluasi kinerja model prediksi akan menghitung kuadrat *error*, *mean squared error* (MSE), dan root mean squared error (RMSE) dapat dilihat pada gambar 4.12

```
kuadrat_error = (y_pred - y_test)**2
jlh_kuadrat_error = np.sum(kuadrat_error)
print(f"Nilai Jumlah Kuadrat error: {jlh_kuadrat_error}")
mse = jlh_kuadrat_error / len(y_test)
print(f"Nilai mse :{mse}")
rmse = np.sqrt(mse)
print(f"Nilai Root Mean Squared Error (RMSE): {rmse}")
```

Gambar 4.12 Menghitung nilai *Root Squaread Error (RMSE)*

Untuk memastikan hasil nilai output pada script diatas dapat dilakukan perhitungan dengan rumus RMSE (Root Mean Squared Error) yaitu :

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(X_i - Y_i)^2}$$

$$= \sqrt{\frac{1}{220}\left((28.278718 - 28.6)^2 + \cdots + (30.888383 - 30.2)^2\right)}$$

$$= \sqrt{\frac{1}{220}\left(124.984927\right)}$$

$$= \sqrt{0.568113}$$

$$= 0.753732$$

Nilai *RMSE* sebesar 0.753732 menunjukkan bahwa rata-rata kesalahan prediksi model *random forest* untuk temperatur di Sumatera Utara adalah sekitar 0.753732 ^{o}C. Kesalahan sebesar ini dapat dianggap cukup rendah tergantung pada rentang

dan variabilitas dari nilai temperatur sebenarnya. Jika rentang temperatur di *dataset* memiliki variasi yang luas, maka *RMSE* sebesar 0.753732 cukup baik. Namun, jika variasi temperatur relatif sempit, kesalahan prediksi sebesar ini bisa jadi lebih signifikan. Selain itu, nilai *RMSE* ini perlu dibandingkan dengan rata-rata nilai temperatur sebenarnya. Jika *RMSE* relatif kecil dibandingkan dengan rata-rata tersebut, maka model memiliki performa yang baik. Misalnya, jika data cuaca ini memiliki nilai rata-rata temperatur 29.184090 maka kesalahan sebesar 0.753732 ^{o}C adalah 2.582681 % dari rata-rata bisa dianggap akurat untuk banyak aplikasi prediksi cuaca. Oleh karena itu, model *random forest* memiliki performa yanng cukup baik dalam memprediksi temperatur di Sumatera Utraa sebagai acuan untuk melakukan aktivitas sehari-hari.

b. *Coefficient of Determination* (R^2)

Menghitung *Coefficient of Determination* (R^2), yang merupakan metrik evaluasi model dalam konteks regresi. R^2 mengukur seberapa baik model menjelaskan variasi dalam data dibandingkan dengan model rata-rata (mean) dapat dilihat pada gambar 4.13

```
rata_y = np.mean(y_test)
print(f"Nilai y_mean :{rata_y}")
kuadrat_error = (y_pred - y_test)**2
jlh_kuadrat_error = np.sum(kuadrat_error)
print(f"Nilai Jumlah Kuadrat Error: {jlh_kuadrat_error}")
jlh_selisih_mean_aktual = np.sum((rata_y - y_test)**2)
print(f"Nilai Jumlah Selisih Rata-Rata dan Aktual: {jlh_selisih_mean_aktual }")
r2 = 1 - (jlh_kuadrat_error / jlh_selisih_mean_aktual)
print(f"Nilai Coefficient of Determination (R²): {r2}")
```

Gambar 4.13 Menghitung Nilai *Coefficient of Determination (R^2)*

Untuk memastikan hasil nilai output pada script diatas dapat dilakukan perhitungan *Coefficient of Determination* (R^2) dengan rumus yaitu :

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(X_i - Y_i)^2}{\sum_{i=1}^{m}(\bar{Y} - Y_i)^2}$$

$$= 1 - \frac{((28.278718-28.6)^2+\cdots+(30.888383-30.2)^2)}{((29.184090-28.278718)^2+\cdots+(29.184090-30.888383)^2)}$$

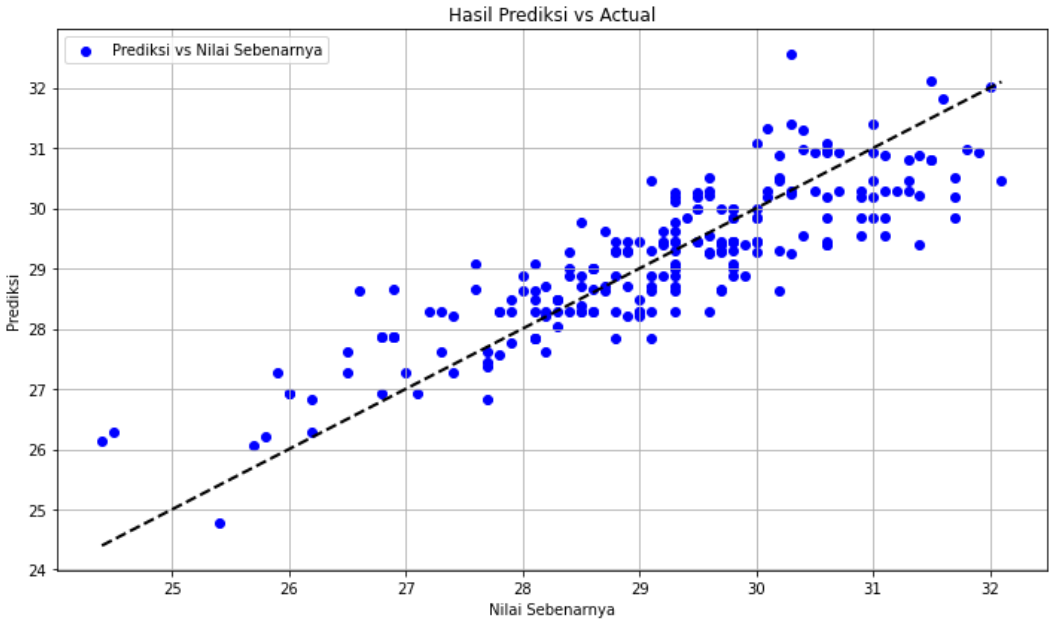
$$= 1 - \frac{(124.984927)}{(473.994318)}$$

$$= 1 - 0.263684$$

$$= 0.736315$$

Nilai R^2 sebesar 0.736315 menunjukkan bahwa model *random forest* yang digunakan mampu menjelaskan sekitar 73.6315% dari variasi dalam data temperatur berdasarkan kelembapan dan kecepatan angin. Nilai R^2 dianggap cukup baik karena menunjukkan bahwa model memiliki kemampuan yang kuat untuk menangkap hubungan antara variabel prediktor (kelembapan dan kecepatan angin) dan variabel respon (temperatur). secara keseluruhan, model *random forest* sudah menunjukkan performa model yang cukup baik dalam banyak aplikasi praktis.

4.3.2 Grafik Hasil Data Prediksi dengan Aktual



Gambar 4.14 Hasil Prediksi dengan Aktual Metode *Random Forest*

Gambar 4.14 menunjukkan bahwa grafik tersebut memiliki hubungan antara nilai prediksi dan aktual dari sebuah model prediksi. Garis hitam putus-putus

mewakili garis y=x yang mengindikasikan kesempurnaan prediksi jika semua titik berada tepat di atasnya. Sebagian besar titik biru tersebar di sekitar garis ini, menunjukkan bahwa model prediksi cukup akurat meskipun tidak sempurna. Sebaran titik lebih padat di sekitar nilai tengah pada sumbu x (sekitar 28-30) menunjukkan bahwa sebagian besar nilai aktual dan prediksi berada dalam rentang tersebut. Ada kecenderungan yang jelas bahwa prediksi model meningkat seiring dengan meningkatnya nilai aktual menunjukkan bahwa model mampu menangkap tren data dengan baik meskipun terdapat beberapa variasi. Secara keseluruhan, model ini cukup baik dalam memprediksi nilai aktual, namun masih ada ruang untuk perbaikan guna mengurangi penyimpangan dan meningkatkan akurasi prediksi.

4.4 Analisis Data dengan Metode *XGBoost*

XGBoost (Extreme Gradient Boosting) adalah salah satu metode *machine learning* yang sangat populer dan kuat, khususnya untuk masalah regresi dan klasifikasi. Metode ini merupakan bagian dari keluarga *gradient boosting* yang menggabungkan beberapa model prediksi lemah (weak learners) untuk membentuk model prediksi kuat (strong learner).

4.4.1 Pengolahan Data

Data diolah menggunakan metode *XGBoost* untuk menentukan hasil prediksi model dapat dilihat pada gambar 4.15

```
df_predictions = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
print(df_predictions)
```

Gambar 4.15 Model *XGBoost*

Berikut ini adalah perbandingan prediksi dan data aktual.

Tabel 4. 2 Data aktual dan Prediksi Metode XGBoost

No	Aktual	Prediksi	Selisih
1.	28.6	28.240177	-0,359823
2.	31.0	30.840273	-0,159727
3.	27.4	27.326345	-0,073655
4.	28.7	29.536985	0,836985
5.	31.0	30.482248	-0,517752
...	...	...	....
216.	30.0	29.336044	-0,663956
217.	29.3	29.149357	-0,150643
218.	31.0	31.102798	0,102798
219.	27.9	28.497131	0,597131
220.	30.2	31.048227	0,848227

Tabel 4.2 secara keseluruhan menunjukkan bahwa model prediksi ini bekerja cukup baik dengan sebagian besar mendekati nilai aktual. Nilai tersebut dapat digunakan untuk mengevaluasi kinerja model dapat digunakan metrik seperti *Root Mean Squared Error (RMSE)* dan *Coefficient of Determination* (R^2).

a. *Root Mean Squared Error* (*RMSE*)

Untuk mengevaluasi kinerja model prediksi akan menghitung kuadrat *error*, *mean squared error* (MSE), dan root mean squared error (RMSE) dapat dilihat pada gambar 4.16

```
kuadrat_error = (y_pred - y_test)**2
jlh_kuadrat_error = np.sum(kuadrat_error)
print(f"Nilai Jumlah Kuadrat error: {jlh_kuadrat_error}")
mse = jlh_kuadrat_error / len(y_test)
print(f"Nilai mse :{mse}")
rmse = np.sqrt(mse)
print(f"Nilai Root Mean Squared Error (RMSE): {rmse}")
```

Gambar 4.16 Menghituung nilai *Root Squaread Error (RMSE)*

Untuk memastikan hasil nilai output pada script diatas dapat dilakukan perhitungan dengan rumus RMSE (Root Mean Squared Error) yaitu :

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(X_i - Y_i)^2}$$

$$= \sqrt{\frac{1}{220}\left((28.240177 - 28.6)^2 + \cdots + (31.048227 - 30.2)^2\right)}$$

$$= \sqrt{\frac{1}{220}\,(119.762817)}$$

$$= \sqrt{0.544376}$$

$$= 0.737818$$

Nilai *RMSE* sebesar 0.737818 menunjukkan bahwa rata-rata kesalahan antara nilai aktual dan prediksi. Nilai tersebut memiliki makna model prediktif yang digunakan cukup akurat dalam memperkirakan nilai aktual karena *RMSE* mendekati 0 mengindikasikan prediksi yang sangat baik.

b. *Coefficient of Determination* (R^2)

Menghitung *Coefficient of Determination* (R^2), yang merupakan metrik evaluasi model dalam konteks regresi. R^2 mengukur seberapa baik model menjelaskan variasi dalam data dibandingkan dengan model rata-rata (mean) dapat dilihat pada gambar 4.17

```
rata_y = np.mean(y_test)
print(f"Nilai y_mean :{rata_y}")
kuadrat_error = (y_pred - y_test)**2
jlh_kuadrat_error = np.sum(kuadrat_error)
print(f"Nilai Jumlah Kuadrat Error: {jlh_kuadrat_error}")
jlh_selisih_mean_aktual = np.sum((rata_y - y_test)**2)
print(f"Nilai Jumlah Selisih Rata-Rata dan Aktual: {jlh_selisih_mean_aktual }")
r2 = 1 - (jlh_kuadrat_error / jlh_selisih_mean_aktual)

print(f"Nilai Coefficient of Determination (R²): {r2}")
```

Gambar4. 17 Menghitung Nilai *Coefficient of Determination* (R^2)

Untuk memastikan hasil nilai output pada script diatas dapat dilakukan perhitungan *Coefficient of Determination* (R^2) dengan rumus yaitu :

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(X_i - Y_i)^2}{\sum_{i=1}^{m}(\bar{Y} - Y_i)^2}$$

$$= 1 - \frac{\left((28.240177 - 28.6)^2 + \cdots + (31.048227 - 30.2)^2\right)}{\left((29.184090 - 28.240177)^2 + \cdots + (29.184090 - 31.048227)^2\right)}$$

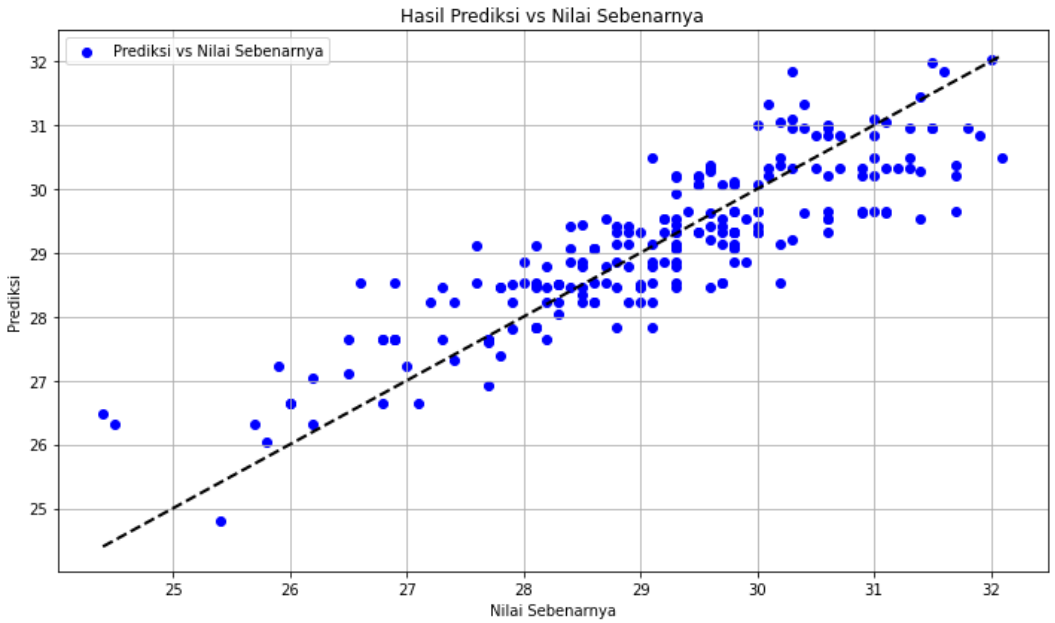
$$= 1 - \frac{(119.762817)}{(473.994318)}$$

$$= 1 - 0.252667$$

$$= 0.747332$$

Nilai R^2 sebesar 0.747332 berarti bahwa model prediksi berhasil menjelaskan sekitar 74,7332 % variabelitas data aktual. Kata lain, sebagian besar variasi dalam data aktual dapat dijelaskan oleh model prediksi. Nilai ini menunjukkan bahwa model memiliki tingkat akurasi yang cukup baik meskipun ada 25,2667% variabilitas data yang tidak dapat dijelaskan oleh model.

4.4.2 Grafik Hasil Data Prediksi dengan Aktual



Gambar 4.18 Hasil Prediksi dengan Aktual Metode XGBoost

Gambar 4.4 menunjukkan plot perbandingan antara nilai prediksi dan aktual. Setiap titik biru pada grafik mewakili pasangan nilai prediksi dan aktual. Garis putus-putus hitam merupakan gris referensi di mana nilai prediksi sama dengan aktual. Sebagian besar titik-titik data tersebar di sekitar garis referensi menunjukkan bahwa model prediksi memiliki kecenderungan yang baik meskipunada beberapa penyebaran di sekitaran garis. Secara umum, pola yang dihasilkan menunjukkan korelasi positif kuat antara nilai prediksi dan aktual.

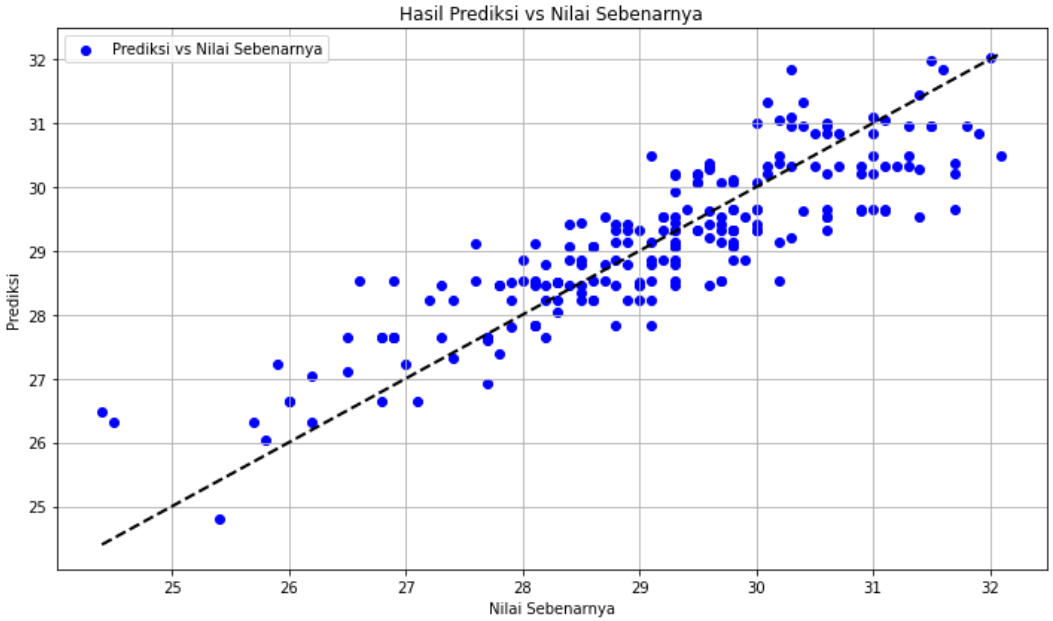
BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Adapun kesimpulan dari penelitian ini adalah

1. Metode *random forest* menghasilkan nilai *Root Mean Squared Error* (*RMSE*) sebesar 0.753732 dan *Coefficient of Determination* (R^2) sebesar 0.736315. Di sisi lain, *XGBoost* menunjukkan nilai *RMSE* sedikit lebih rendah yaitu 0.737818 dan R^2 lebih tinggi mencapai 0.747332. Jika Semakin kecil (mendekati 0) nilai *RMSE* maka hasil prediksi semakin akurat sedangkan semakin besar (mendekati $+\infty$) maka hasil prediksi semakin buruk, dan Jika Nilai R^2 semakin mendekati $-\infty$ maka hasil prediksi semakin buruk sedangkan nilai R^2 semakin mendekati 1 maka hasil prediksi semakin akurat. Disimpulkan bahwa *XGBoost* memiliki kinerja yang sedikit lebih baik dalam hal meminimalkan kesalahan prediksi (RMSE) dan meningkatkan kecocokan model terhadap data (R^2) dibandingkan *random forest*. Meskipun perbedaan tidak signifikan, hasil ini menunjukkan *XGBoost* lebih cocok untuk mendapatkan prediksi yang lebih akurat dan sesuai data aktual.
2. Melihat hasil nilai prediksi dengan metode *XGBoost* lebih dekat dengan nilai aktualnya daripada metode *random forest*. Sehingga *XGBoost* memiliki performa yang lebih baik dalam menangkap pola dan variasi data yang kompleks. Dibawah ini adalah grafik hasil data prediksi dengan aktual metode *XGBoost.*



5.2 Saran

1. Pada penelitian selanjutnya diharapkan bisa memprediksi cuaca menggunakan metode ini atau metode machine learning yang lain untuk tahun berikutnya.
2. Pada penelitian selanjutnya diharapkan dapat mengembangkan penelitian ini kedalam bentuk aplikasi atau website

DAFTAR PUSTAKA

Agustiningsih, A., Findawati, Y., & Alnarus Kautsar, I. (2023). Classification of Vocational High School Graduates' Ability in Industry Using Extreme Gradient Boosting (Xgboost), Random Forest, and Logistic Regression. *Jurnal Teknik Informatika (Jutif)*, *4*(4), 977–985. https://doi.org/10.52436/1.jutif.2023.4.4.945

Akbar, H., & Sanjaya, W. K. (2023). Kajian Performa Metode Class Weight Random Forest pada Klasifikasi Imbalance Data Kelas Curah Hujan. *Jurnal Sains, Nalar, Dan Aplikasi Teknologi Informasi*, *3*(1). https://doi.org/10.20885/snati.v3i1.30

Al'afi, A. M., Widiart, W., Kurniasari, D., & Usman, M. (2020). Peramalan Data Time Series Seasonal Menggunakan Metode Analisis Spektral. *Jurnal Siger Matematika*, *1*(1), 10–15. https://doi.org/10.23960/jsm.v1i1.2484

Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, *7*, 1–24. https://doi.org/10.7717/PEERJ-CS.623

Dridi, S., Machine, V., Tree, D., Forest, R., & Regression, L. (2021). *S l - a s l r*.

Dwiyanti, Z. A., & Prianto, C. (2023). Prediksi Cuaca Kota Jakarta Menggunakan Metode Random Forest. *Jurnal Tekno Insentif*, *17*(2), 127–137. https://doi.org/10.36787/jti.v17i2.1136

Farhanuddin, Sarah Ennola Karina Sihombing, & Yahfizham. (2024). Komparasi Multiple Linear Regression dan Random Forest Regression Dalam Memprediksi Anggaran Biaya Manajemen Proyek Sistem Informasi. *Journal of Computers and Digital Business*, *3*(2), 86–97. https://doi.org/10.56427/jcbd.v3i2.408

Gusliana, I. (2021). Bab II Landasan Teori. *Journal of Chemical Information and Modeling*, *53*(9), 1689–1699.

Herni Yulianti, S. E., Oni Soesanto, & Yuana Sukmawaty. (2022). Penerapan Metode Extreme Gradient Boosting (XGBOOST) pada Klasifikasi Nasabah Kartu Kredit. *Journal of Mathematics: Theory and Applications*, *4*(1), 21–26. https://doi.org/10.31605/jomta.v4i1.1792

Iman, A. H., Permana, F. R., Wardana, G. P., & Rachmansyah, R. K. (2022). *Perbandingan Algoritma Klasifikasi Random Forest dan Extreme Gradient Boosting pada Dataset Cuaca Provinsi DKI Jakarta Tahun 2018*. 773–782.

Luthfiarta, A., Febriyanto, A., Lestiawan, H., & Wicaksono, W. (2020). Analisa Prakiraan Cuaca dengan Parameter Suhu, Kelembaban, Tekanan Udara, dan Kecepatan Angin Menggunakan Regresi Linear Berganda. *JOINS (Journal of Information System)*, *5*(1), 10–17. https://doi.org/10.33633/joins.v5i1.2760

Mdegela, L., Municio, E., De Bock, Y., Luhanga, E., Leo, J., & Mannens, E. (2023). Extreme Rainfall Event Classification Using Machine Learning for Kikuletwa River Floods. *Water (Switzerland)*, *15*(6), 1–14. https://doi.org/10.3390/w15061021

Melvin, J., & Soraya, A. (2023). Analisis Perbandingan Algoritma XGBoost dan Algoritma Random Forest Ensemble Learning pada Klasifikasi Keputusan Kredit. *Jurnal Riset Rumpun Matematika Dan Ilmu Pengetahuan Alam (JURRIMIPA)*, *2*(2), 87–103.

Muhammad Romzi, & Kurniawan, B. (2020). Pembelajaran Pemrograman Python Dengan Pendekatan Logika Algoritma. *JTIM: Jurnal Teknik Informatika Mahakarya*, *03*(2), 37–44.

Mursianto, G. A., Falih, I. M., Irfan, M., Sakinah, T., & Prasvita, D. S. (2021). Perbandingan Metode Klasifikasi Random Forest dan XGBoost Serta Implementasi Teknik SMOTE pada Kasus Prediksi Hujan. *Jurnal Senamika*, *2*(2), 41–50.

Muslim Karo Karo, I. (2020). Implementasi Metode XGBoost dan Feature Importance untuk Klasifikasi pada Kebakaran Hutan dan Lahan. *Journal of Software Engineering, Information and Communication Technology*, *1*(1), 11–18.

Nugraha, A. C., & Irawan, M. I. (2023). Komparasi Deteksi Kecurangan pada Data Klaim Asuransi Pelayanan Kesehatan Menggunakan Metode Support Vector Machine (SVM) dan Extreme Gradient Boosting (XGBoost). *Jurnal Sains Dan Seni ITS*, *12*(1). https://doi.org/10.12962/j23373520.v12i1.107032

Rakhmat, G. A., & Mutohar, W. (2023). Prakiraan Hujan menggunakan Metode Random Forest dan Cross Validation. *Journal MIND*, *8*(2), 173–187. https://doi.org/10.26760/mindjournal.v8i2.173-187

Sanjaya, F. I., & Heksaputra, D. (2020). Prediksi Rerata Harga Beras Tingkat Grosir Indonesia dengan Long Short Term Memory. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, *7*(2), 163–174. https://doi.org/10.35957/jatisi.v7i2.388

Suma, B., & Pasundan, U. (2021). *BANDUNG SEPTEMBER 2020*. *January*. https://doi.org/10.13140/RG.2.2.16086.47680

Suprayogi, I., Trimaijon, & Mahyudin. (2014). Model Prediksi Liku Kalibrasi Menggunakan Pendekatan Jaringan Saraf Tiruan (JST) (Studi Kasus: Sub DAS Siak Hulu). *Jurnal Online Mahasiswa Fakultas Teknik Universitas Riau*, *1*(1), 1–18. http://ce.unri.ac.id

Syuhada, A. S., Simanullang, A. M., Lewa, D. S., & Marthin, S. J. (2021). *Fakultas teknik universitas maritim raja ali haji 2021*.

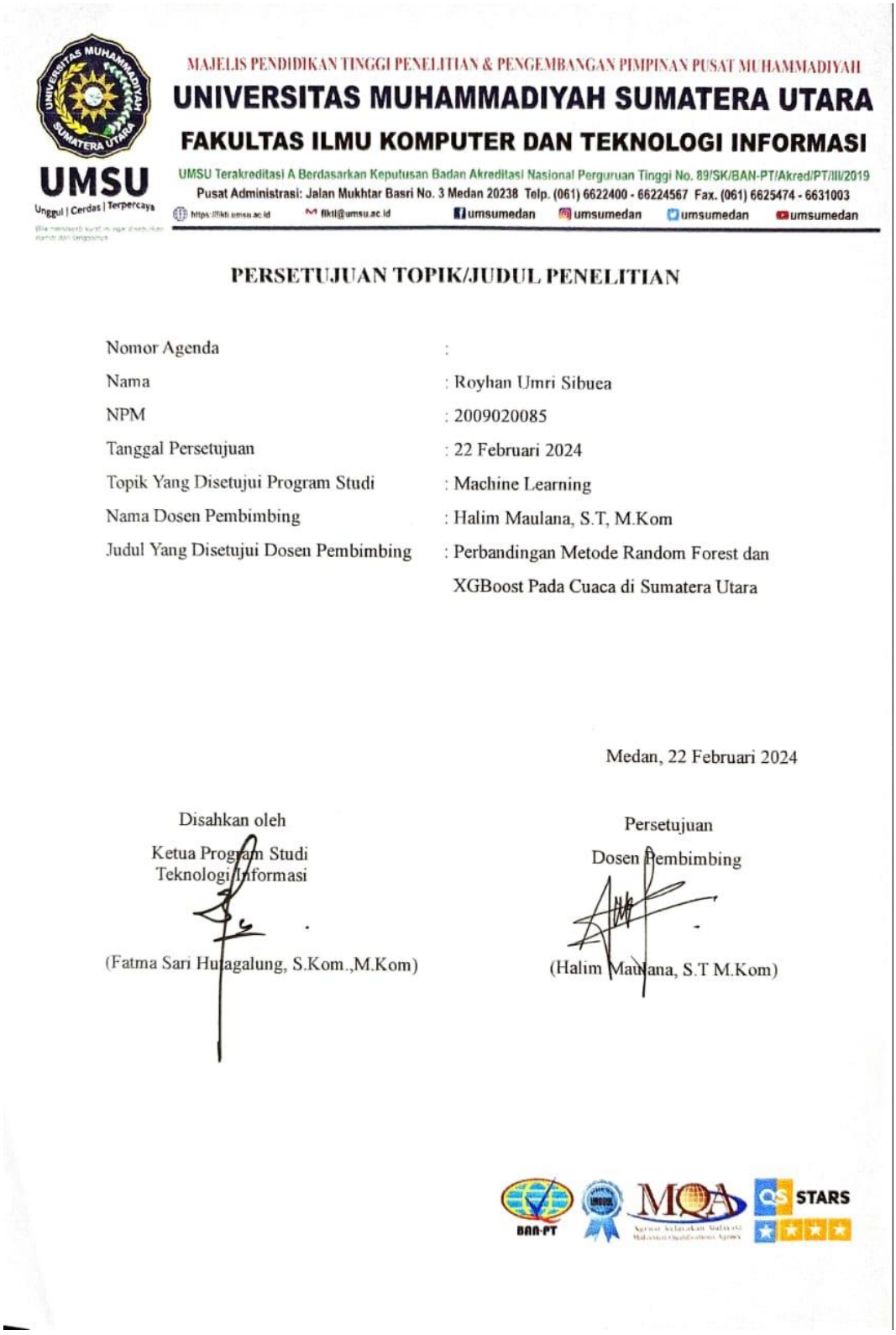
Syukron, M., Santoso, R., & Widiharih, T. (2020). Perbandingan Metode Smote Random Forest Dan Smote Xgboost Untuk Klasifikasi Tingkat Penyakit Hepatitis C Pada Imbalance Class Data. *Jurnal Gaussian*, *9*(3), 227–236. https://doi.org/10.14710/j.gauss.v9i3.28915

YEŞİLKANAT, C. M. (2020). Spatio-temporal estimation of the daily cases of COVID-19 in worldwide using random forest machine learning algorithm. *Chaos, Solitons and Fractals*, *140*. https://doi.org/10.1016/j.chaos.2020.110210

Yoga Religia, Agung Nugroho, & Wahyu Hadikristanto. (2021). Klasifikasi Analisis Perbandingan Algoritma Optimasi pada Random Forest untuk Klasifikasi Data Bank Marketing. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *5*(1), 187–192. https://doi.org/10.29207/resti.v5i1.2813

LAMPIRAN

Lampiran 1 : SK-1 Persetujuan Topik /Judul Penelitian



MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/III/2019

Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003

https://fikti.umsu.ac.id fikti@umsu.ac.id umsumedan umsumedan umsumedan umsumedan

UMSU Unggul | Cerdas | Terpercaya

PERSETUJUAN TOPIK/JUDUL PENELITIAN

Nomor Agenda	:
Nama	: Royhan Umri Sibuea
NPM	: 2009020085
Tanggal Persetujuan	: 22 Februari 2024
Topik Yang Disetujui Program Studi	: Machine Learning
Nama Dosen Pembimbing	: Halim Maulana, S.T, M.Kom
Judul Yang Disetujui Dosen Pembimbing	: Perbandingan Metode Random Forest dan XGBoost Pada Cuaca di Sumatera Utara

Medan, 22 Februari 2024

Disahkan oleh
Ketua Program Studi
Teknologi Informasi

(Fatma Sari Hutagalung, S.Kom.,M.Kom)

Persetujuan
Dosen Pembimbing

(Halim Maulana, S.T M.Kom)

Lampiran 2 : SK-2 Surat Penetapan Dosen Pembimbing

MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/III/2019
Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003
https://fikti.umsu.ac.id fikti@umsu.ac.id umsumedan umsumedan umsumedan umsumedan

UMSU
Unggul | Cerdas | Terpercaya

PENETAPAN DOSEN PEMBIMBING
PROPOSAL/SKRIPSI MAHASISWA
NOMOR : 267/II.3-AU/UMSU-09/F/2024

Assalamu'alaikum Warahmatullahi Wabarakatuh

Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara, berdasarkan Persetujuan permohonan judul penelitian Proposal / Skripsi dari Ketua / Sekretaris.

Program Studi	**: Teknologi Informasi**
Pada tanggal	**: 22 Februari 2024**

Dengan ini menetapkan Dosen Pembimbing Proposal / Skripsi Mahasiswa.

Nama	**: Royhan Umri Sibuea**
NPM	**: 2009020085**
Semester	**: VIII (Delapan)**
Program studi	**: Teknologi Informasi**
Judul Proposal / Skripsi	**: Perbandingan Metode Random Forest Dan Xgboost Pada Cuaca Disumatera Utara**
Dosen Pembimbing	**: Halim Maulana, S.Kom, M.Kom**

Dengan demikian di izinkan menulis Proposal / Skripsi dengan ketentuan

1. Penulisan berpedoman pada buku panduan penulisan Proposal / Skripsi Fakultas Ilmu Komputer dan Teknologi Informasi UMSU
2. Pelaksanaan Sidang Skripsi harus berjarak 3 bulan setelah dikeluarkannya Surat Penetapan Dosen Pembimbing Skripsi.
3. **Proyek Proposal / Skripsi** dinyatakan " **BATAL** " bila tidak selesai sebelum Masa Kadaluarsa tanggal : **22 Februari 2025**
4. Revisi judul……

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Ditetapkan di : Medan
Pada Tanggal : 12 Sya'ban 1445 H
22 Februari 2024 M

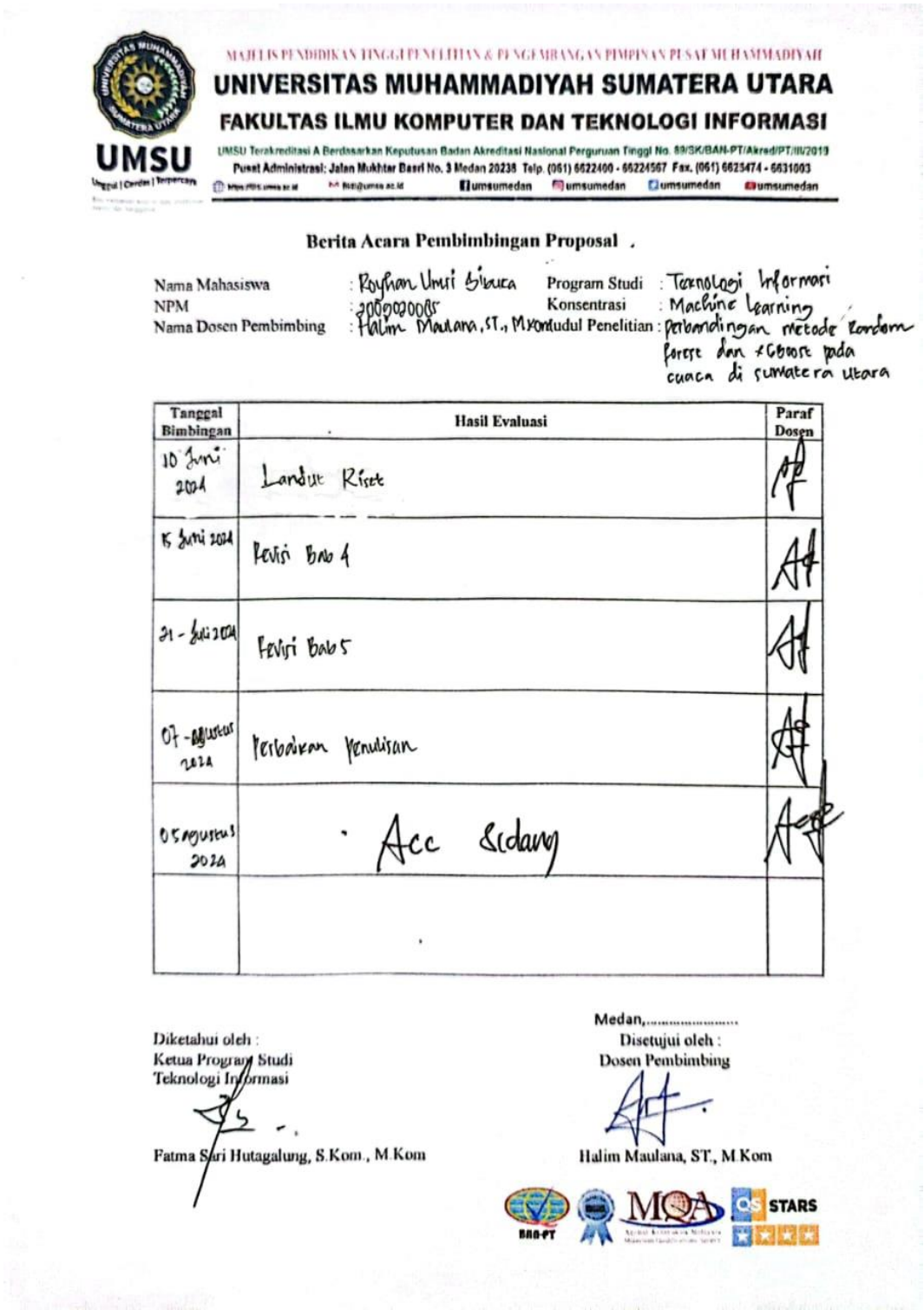
Dekan

Dr. Al-Khowarizmi,S.Kom.,M.Kom
NIDN : 0127099201

Cc. File

BAN-PT MQA QS STARS

Lampiran 3 : SK-3 Berita Acara Bimbingan



MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/III/2019

Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003

umsumedan umsumedan umsumedan umsumedan

Berita Acara Pembimbingan Proposal

Nama Mahasiswa : Rayhan Umri Sibuea
NPM : 2009020005
Nama Dosen Pembimbing : Halim Maulana, ST., M.Kom

Program Studi : Teknologi Informasi
Konsentrasi : Machine Learning
Judul Penelitian : perbandingan metode random forest dan xgboost pada cuaca di sumatera utara

Tanggal Bimbingan	Hasil Evaluasi	Paraf Dosen
10 Juni 2024	Landasan Riset	
15 Juni 2024	Revisi Bab 4	
21 - Juli 2024	Revisi Bab 5	
07 - Agustus 2024	Perbaikan penulisan	
05 agustus 2024	Acc Sidang	

Medan,.....

Diketahui oleh :
Ketua Program Studi
Teknologi Informasi

Fatma Sari Hutagalung, S.Kom., M.Kom

Disetujui oleh :
Dosen Pembimbing

Halim Maulana, ST., M.Kom

Lampiran 4 : SK-4 Surat Permohonan Seminar Proposal

MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UMSU
Unggul | Cerdas | Terpercaya
UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/III/2019
Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003
https://fikti.umsu.ac.id fikti@umsu.ac.id umsumedan umsumedan umsumedan umsumedan

PERMOHONAN
SEMINAR PROPOSAL SKRIPSI

Kepada Yth.
Bapak Dekan FIKTI UMSU
Di
Medan

Medan,.....2024

Assalamu'alaikum Warahmatullahi Wabarakatuh

Dengan hormat, saya yang bertanda tangan di bawah ini mahasiswa Fakultas Ilmu Komputer dan Teknologi Informasi UMSU :

Nama Lengkap	: Royhan Umri Sibuea
NPM	: 2009020085
Program Studi	: Teknologi Informasi

Mengajukan permohonan Mengikuti **Seminar Proposal Skripsi** yang ditetapkan dengan Surat Penetapan Judul Skripsi dan Pembimbing Nomor .....II.3-AU/UMSU-09/F/2024 Tanggal 02 April 2024 dengan judul sebagai berikut :

PERBANDINGAN METODE RANDOM FOREST DAN XGBOOST PADA CUACA DI SUMATERA UTARA

Bersama permohonan ini saya lampirkan :

1. Surat Penetapan Judul Skripsi (SK-1),
2. Surat Penetapan Pembimbing (SK-2),
3. DEKAM yang telah disahkan,
4. Kartu Hasil Studi Semester 1 s/d terakhir **ASLI**,
5. Tanda Bukti Lunas Beban SPP tahap berjalan,
6. Tanda Bukti Lunas Biaya Seminar Proposal Skripsi,
7. Proposal Skripsi yang telah disahkan oleh Pembimbing (rangkap-3),
8. Semua berkas dimasukan ke dalam MAP warna **BIRU**.

Demikian permohonan saya untuk pengurusan selanjutnya. Atas perhatian Bapak saya ucapkan terima kasih.

Wassalamualaikum Warahmatullahi Wabarakatuh.

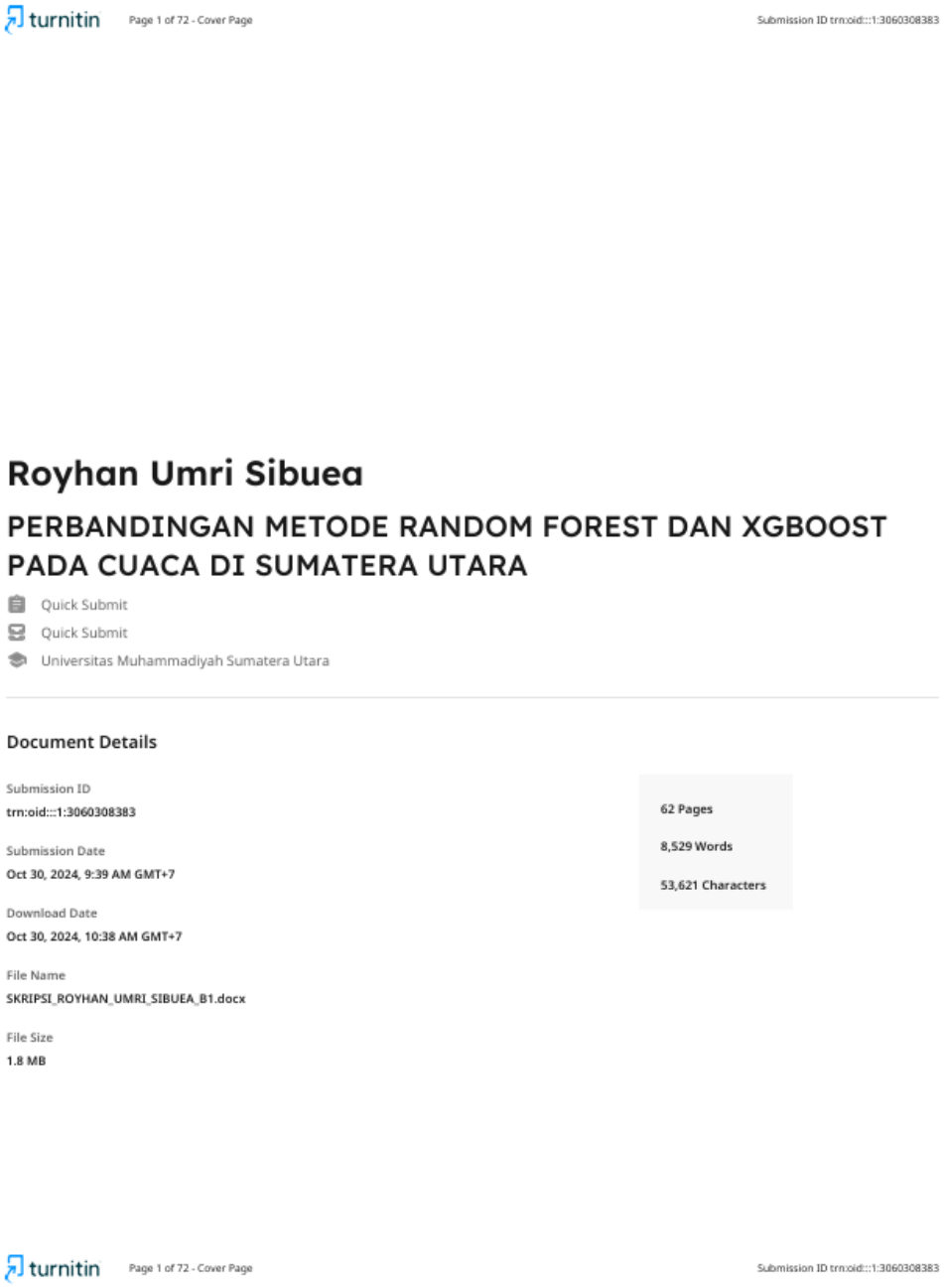
Menyetujui :
Pembimbing

(Halim Maulana, S.T M.Kom)

Pemohon

(Royhan Umri Sibuea)

Lampiran 5 : SK-5 Surat Plagiasi



turnitin Page 1 of 72 - Cover Page Submission ID trn:oid:::1:3060308383

Royhan Umri Sibuea

PERBANDINGAN METODE RANDOM FOREST DAN XGBOOST PADA CUACA DI SUMATERA UTARA

Quick Submit

Quick Submit

Universitas Muhammadiyah Sumatera Utara

Document Details

Submission ID
trn:oid:::1:3060308383

Submission Date
Oct 30, 2024, 9:39 AM GMT+7

Download Date
Oct 30, 2024, 10:38 AM GMT+7

File Name
SKRIPSI_ROYHAN_UMRI_SIBUEA_B1.docx

File Size
1.8 MB

62 Pages

8,529 Words

53,621 Characters

turnitin Page 1 of 72 - Cover Page Submission ID trn:oid:::1:3060308383

24% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Top Sources

22% Internet sources
9% Publications
9% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

turnitin Page 2 of 72 - Integrity Overview Submission ID trn:oid:::1:3060308383

Top Sources

22% Internet sources
9% Publications
9% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	dspace.uii.ac.id 2%
2	Internet	www.researchgate.net 1%
3	Internet	ojs.unsulbar.ac.id 1%
4	Publication	Zian Asti Dwiyanti, Cahyo Prianto. "Prediksi Cuaca Kota Jakarta Menggunakan Me... 1%
5	Internet	ejurnal.its.ac.id 1%
6	Internet	jurnal.iaii.or.id 1%
7	Internet	ejurnal.itenas.ac.id 1%
8	Internet	journal.uii.ac.id 1%
9	Internet	jurnal.umt.ac.id 1%
10	Internet	text-id.123dok.com 0%
11	Internet	ejurnal.stmik-budidarma.ac.id 0%

turnitin Page 3 of 72 - Integrity Overview Submission ID trn:oid:::1:3060308383

Lampiran 6 : *Source code*

program untuk menampilkan grafik data cuaca di Sumatera Utara dengan menggunakan metode *random forest*

```
#!/usr/bin/env python
# coding: utf-8
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from datetime import datetime, timedelta
import numpy as np
# Membaca data dari file CSV
data = pd.read_excel("D:/SKRIPSI/DATA CUACA SUMUT.xlsx")
print(data)
```

```
          Tanggal  Temperatur   Kelembapan   Kecepatan angin
0      01-04-2021         29.8           70                 2
1      02-04-2021         29.2           76                 1
2      03-04-2021         27.8           81                 2
3      04-04-2021         28.8           78                 1
4      05-04-2021         28.3           82                 2
...           ...          ...          ...               ...
1091   27-03-2024         28.4           83                 1
1092   28-03-2024         31.4           66                 0
1093   29-03-2024         29.1           78                 1
1094   30-03-2024         28.5           80                 2
1095   31-03-2024         31.5           67                 2

[1096 rows x 4 columns]
```

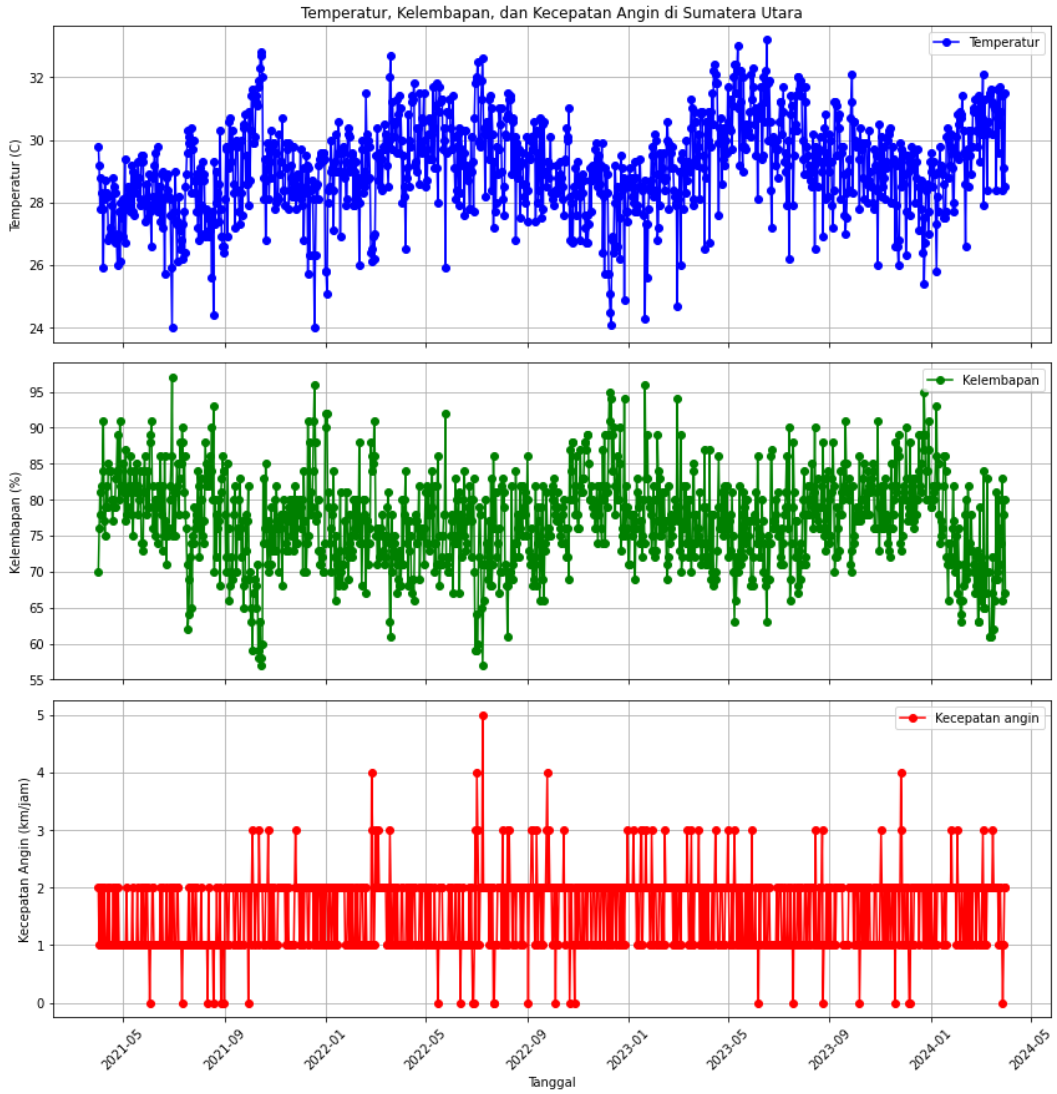
```
# Ubah kolom 'Tanggal' ke dalam format datetime
data['Tanggal'] = pd.to_datetime(data['Tanggal'], format='%d-%m-%Y')
# Hapus spasi ekstra dari nama kolom (jika ada)
```

```
data.columns = data.columns.str.strip()
print(data.columns)
```

```
Index(['Tanggal', 'Temperatur', 'Kelembapan', 'Kecepatan angin'], dtype='object')
```

```
# Membuat satu figur dengan tiga sumbu (subplots)
fig, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(12, 12), sharex=True)
# Plot untuk temperatur
ax1.plot(data['Tanggal'], data['Temperatur'], marker='o', linestyle='-', color='b',
label='Temperatur')
ax1.set_title('Temperatur, Kelembapan, dan Kecepatan Angin di Sumatera
Utara')
ax1.set_ylabel('Temperatur (C)')
ax1.grid(True)
ax1.legend()
# Plot untuk kelembapan
ax2.plot(data['Tanggal'], data['Kelembapan'], marker='o', linestyle='-',
color='g', label='Kelembapan')
ax2.set_ylabel('Kelembapan (%)')
ax2.grid(True)
ax2.legend()
# Plot untuk kecepatan angin
ax3.plot(data['Tanggal'], data['Kecepatan angin'], marker='o', linestyle='-',
color='r', label='Kecepatan angin')
ax3.set_xlabel('Tanggal')
ax3.set_ylabel('Kecepatan Angin (km/jam)')
```

```
ax3.grid(True)

ax3.legend()

# Menyesuaikan layout

plt.tight_layout()

plt.xticks(rotation=45)

# Menampilkan grafik

plt.show()
```




```
# Memisahkan fitur (predictor) dan target

X = data[['Kelembapan', 'Kecepatan angin']]
```

```
y = data['Temperatur']

# Pembagian data menjadi data pelatihan dan data uji

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,

random_state=42)

# Menentukan panjang X_test

panjang_X_train = len(X_train)

print(f'Panjang X_train: {panjang_X_train}')

panjang_X_test = len(X_test)

print(f'Panjang X_test: {panjang_X_test}')
```

```
Panjang X_train: 876
Panjang X_test: 220
```

```
# Membuat model Random Forest

model = RandomForestRegressor(n_estimators=100, random_state=42)

# Melatih model

model.fit(X_train, y_train)

# Membuat prediksi pada data uji

y_pred = model.predict(X_test)

# Menyimpan data aktual dan hasil prediksi dalam DataFrame

hasil_prediksi = pd.DataFrame({'Actual': y_test, 'Prediksi': y_pred})

# Menampilkan tabel data aktual dan prediksi

print(hasil_prediksi)
```

```
     Actual   Prediksi
44     28.6  28.278718
568    31.0  30.923164
56     27.4  27.273671
636    28.7  29.616175
486    31.0  30.465685
..      ...        ...
757    30.0  29.445380
713    29.3  29.296954
365    31.0  31.401435
299    27.9  28.471565
286    30.2  30.888383

[220 rows x 2 columns]
```

```
# Menghitung Error untuk setiap prediksi

kuadrat_error = (y_pred - y_test)**2

jlh_kuadrat_error = np.sum(kuadrat_error)

print(f"Nilai Jumlah Kuadrat error: {jlh_kuadrat_error}")

# Menghitung rata-rata dari Error Kuadrat (MSE)

mse = jlh_kuadrat_error / len(y_test)

print(f"Nilai mse :{mse}")

# Mengambil akar kuadrat dari MSE untuk mendapatkan RMSE

rmse = np.sqrt(mse)

print(f"Nilai Root Mean Squared Error (RMSE): {rmse}")
```

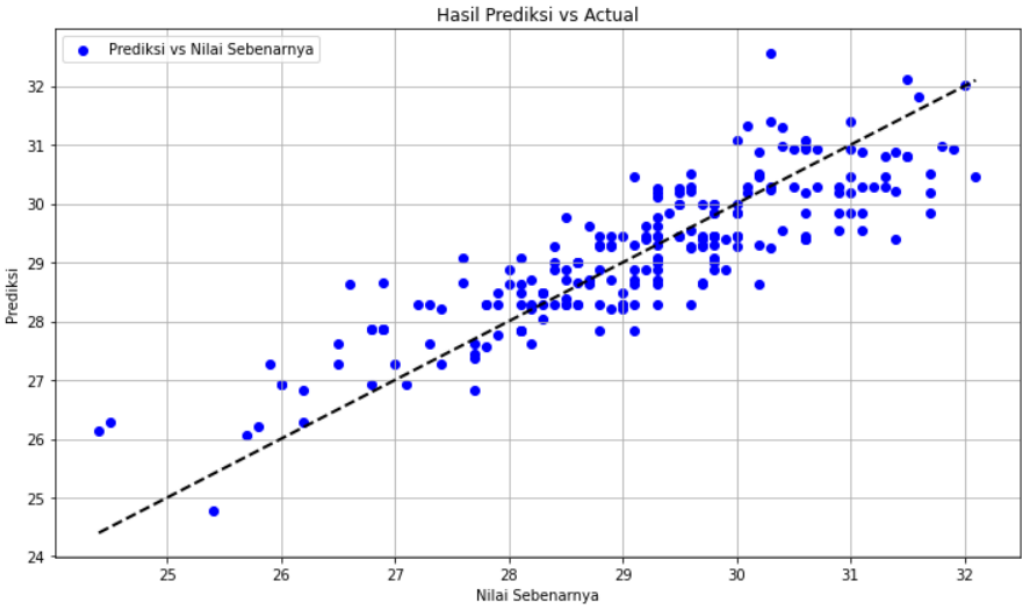
```
Nilai Jumlah Kuadrat error: 124.98492732271716
Nilai mse :0.5681133060123507
Nilai Root Mean Squared Error (RMSE): 0.7537329142424063
```

```
# Menghitung rata-rata dari nilai sebenarnya

rata_y = np.mean(y_test)

print(f"Nilai y_mean :{rata_y}")

# Menghitung Error untuk setiap prediksi

kuadrat_error = (y_pred - y_test)**2
```

```
jlh_kuadrat_error = np.sum(kuadrat_error)

print(f"Nilai Jumlah Kuadrat Error: {jlh_kuadrat_error}")

# Menghitung jumlah kuadrat dari selisih rata-rata dengan nilai aktual

jlh_selisih_mean_aktual = np.sum((rata_y - y_test)**2)

print(f"Nilai Jumlah Selisih Rata-Rata dan Aktual:
{jlh_selisih_mean_aktual }")

# Menghitung R²

r2 = 1 - (jlh_kuadrat_error / jlh_selisih_mean_aktual)

print(f"Nilai Coefficient of Determination (R²): {r2}")
```

```
Nilai y_mean :29.18409090909094
Nilai Jumlah Kuadrat Error: 124.98492732271716
Nilai Jumlah Selisih Rata-Rata dan Aktual: 473.99431818181824
Nilai Coefficient of Determination (R²): 0.7363155579540628
```

```
# Plot hasil prediksi vs nilai sebenarnya pada data uji

plt.figure(figsize=(10, 6))

plt.scatter(y_test, y_pred, color='b', label='Prediksi vs Nilai Sebenarnya')

plt.xlabel('Nilai Sebenarnya')

plt.ylabel('Prediksi')

plt.title('Hasil Prediksi vs Actual')

plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw=2)

plt.legend()

plt.grid(True)

plt.tight_layout()

plt.show()
```



2. *Source code* program dengan metode *XGBoost*

```
#!/usr/bin/env python

# coding: utf-8

import pandas as pd

from sklearn.model_selection import train_test_split

import numpy as np

import xgboost as xgb

import matplotlib.pyplot as plt

# Membaca data dari file CSV

data = pd.read_excel("D:/SKRIPSI/DATA CUACA SUMUT.xlsx")

print(data)
```

```
         Tanggal  Temperatur   Kelembapan   Kecepatan angin
0     01-04-2021         29.8           70                 2
1     02-04-2021         29.2           76                 1
2     03-04-2021         27.8           81                 2
3     04-04-2021         28.8           78                 1
4     05-04-2021         28.3           82                 2
...          ...          ...          ...               ...
1091  27-03-2024         28.4           83                 1
1092  28-03-2024         31.4           66                 0
1093  29-03-2024         29.1           78                 1
1094  30-03-2024         28.5           80                 2
1095  31-03-2024         31.5           67                 2

[1096 rows x 4 columns]
```

```
# Ubah kolom 'Tanggal' ke dalam format datetime

data['Tanggal'] = pd.to_datetime(data['Tanggal'], format='%d-%m-%Y')

# Hapus spasi ekstra dari nama kolom (jika ada)

data.columns = data.columns.str.strip()

print(data.columns)
```

```
Index(['Tanggal', 'Temperatur', 'Kelembapan', 'Kecepatan angin'], dtype='object')
```

```
# Memisahkan fitur (predictor) dan target

X = data[['Kelembapan', 'Kecepatan angin']]

y = data['Temperatur']

# Pembagian data menjadi data pelatihan dan data uji

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,

random_state=42)

# Membuat model XGBoost

model = xgb.XGBRegressor(objective ='reg:squarederror', n_estimators=100,

learning_rate=0.1, max_depth=3, random_state=42)

# Melatih model

model.fit(X_train, y_train)
```

```
# Memprediksi pada set pengujian

y_pred = model.predict(X_test)

# Menampilkan beberapa prediksi vs nilai sebenarnya

df_predictions = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})

print(df_predictions)
```

	Actual	Predicted
44	28.6	28.240177
568	31.0	30.840273
56	27.4	27.326345
636	28.7	29.536985
486	31.0	30.482248
..	...	...
757	30.0	29.336044
713	29.3	29.149357
365	31.0	31.102798
299	27.9	28.497131
286	30.2	31.048227

[220 rows x 2 columns]

```
# Menghitung Error untuk setiap prediksi

kuadrat_error = (y_pred - y_test)**2

jlh_kuadrat_error = np.sum(kuadrat_error)

print(f"Nilai Jumlah Kuadrat error: {jlh_kuadrat_error}")

# Menghitung rata-rata dari Error Kuadrat (MSE)

mse = jlh_kuadrat_error / len(y_test)

print(f"Nilai mse :{mse}")

# Mengambil akar kuadrat dari MSE untuk mendapatkan RMSE

rmse = np.sqrt(mse)

print(f"Nilai Root Mean Squared Error (RMSE): {rmse}")
```

```
Nilai Jumlah Kuadrat error: 119.76281703166461
Nilai mse :0.544376441053021
Nilai Root Mean Squared Error (RMSE): 0.7378187047324166
```

```
# Menghitung rata-rata dari nilai sebenarnya
rata_y = np.mean(y_test)
print(f"Nilai y_mean :{rata_y}")
# Menghitung Error untuk setiap prediksi
kuadrat_error = (y_pred - y_test)**2
jlh_kuadrat_error = np.sum(kuadrat_error)
print(f"Nilai Jumlah Kuadrat Error: {jlh_kuadrat_error}")
# Menghitung jumlah kuadrat dari selisih rata-rata dengan nilai aktual
jlh_selisih_mean_aktual = np.sum((rata_y - y_test)**2)
print(f"Nilai Jumlah Selisih Rata-Rata dan Aktual: {jlh_selisih_mean_aktual }")
# Menghitung R²
r2 = 1 - (jlh_kuadrat_error / jlh_selisih_mean_aktual)
print(f"Nilai Coefficient of Determination (R²): {r2}")
```

```
Nilai y_mean :29.18409090909094
Nilai Jumlah Kuadrat Error: 119.76281703166461
Nilai Jumlah Selisih Rata-Rata dan Aktual: 473.99431818181824
Nilai Coefficient of Determination (R²): 0.7473328003359629
```

```
# Plot hasil prediksi vs nilai sebenarnya pada data uji
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, color='b', label='Prediksi vs Nilai Sebenarnya')
plt.xlabel('Nilai Sebenarnya')
plt.ylabel('Prediksi')
plt.title('Hasil Prediksi vs Nilai Sebenarnya')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw=2)
plt.legend()
```

```
plt.grid(True)

plt.tight_layout()

plt.show()
```