BAB II
TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Belum banyak penelitian yang berkaitan langsung dengan prediksi harga jual cabai rawit khususnya melakukan perbandingan pada beberapa model regresi untuk mengetahui model yang akurat dalam melakukan prediksi harga jual. Penelitian ini menggunakan beberapa jurnal yang sedikit banyaknya memiliki kaitan seperti perbandingan model regresi, prediksi harga jual cabai rawit di wilayah selain Kota Semarang maupun pengaruh iklim terhadap harga jual. Dengan memanfaatkan penelitian-penelitian sebelumnya diharapkan dapat menambah wawasan yang berharga dalam mengimplementasikan teori serta konsep yang berkaitan dengan penelitian ini. Berikut merupakan beberapa penelitian terdahulu yang menjadi rujukan pada penelitian ini.

Penelitian [1] yang berfokus pada permasalahan keterkaitan antara perubahan iklim terhadap produktivitas tanaman cabai rawit menggunakan data iklim BMKG dan data produktivitas cabai rawit di kabupaten Malang. Penelitian tersebut menerapkan algoritma *Linear Regression* yang menghasilkan kesimpulan bahwa perubahan pada iklim dan luas panen memiliki hubungan yang berpengaruh terhadap produktivitas tanaman cabai rawit sebesar 85%.

Konsep yang sama dengan penelitian ini juga pernah dilakukan sebelumnya yaitu memprediksi harga cabai berdasarkan iklim, namun memiliki *output* yang berbeda. Penelitian [13] menggunakan data iklim dari BMKG dan harga jual cabai di daerah kabupaten Bandung untuk memprediksi harga cabai. Harga cabai yang telah diprediksi kemudian diklasifikasi menggunakan algoritma *SVM* dengan optimasi *ANFIS* untuk menentukan apakah harga cabai tersebut menguntungkan bagi petani atau tidak. Dari penelitian tersebut didapat bahwa algoritma *SVM* dengan optimasi *ANFIS* dapat memprediksi harga cabai dengan akurasi yang tinggi ketika semua atribut dimasukkan tanpa normalisasi data, dengan rata-rata akurasi sebesar 92,68%.

Masih menggunakan data yang sama dengan penelitian [13], penelitian [10] menggunakan algoritma *KNN* untuk mengkalsifikasi harga jual cabai rawit, namun permasalahannya data yang diperoleh mengalami *imbalance class*. Penanganan *imbalance class* memanfaatkan salah satu metode *oversampling* yaitu *Adaptive Synthetic (ADASYN)* menghasilkan bahwa algoritma *KNN* yang dikombinasikan dengan metode ini efektif dalam menangani dataset yang tidak seimbang dan membuat prediksi yang akurat. Performa yang dihasilkan dari kombinasi metode tersebut yaitu akurasi mencapai 100% dan *F1-Score* mencapai 100%.

Salah satu model regresi yaitu *Random Forest Regression* yang digunakan pada penelitian ini sama dengan penelitian [19] untuk memprediksi harga rumah di daerah Jakarta Selatan dan Tebet. Data yang digunakan memiliki karakteristik yang non-*linear* serta berfluktuasi. Beberapa model regresi seperti *Linear Regression*, *Random Forest Regression* dan *Gradient Boosted Trees* dibandingkan untuk mendapatkan model yang akurat, didapat bahwa model *Random Forest Regression* merupakan model terbaik. Model ini memiliki akurasi tertinggi yaitu 81,5% dengan tingkat kesalahan prediksi 0,440.

Model regresi selanjutnya yang digunakan pada penelitian ini yaitu *KNN Regression* pernah digunakan juga pada penelitian [20] untuk memprediksi kekuatan generator turbin. Menggunakan data keadaan iklim seperti kecepatan angin, suhu udara dan kekuatan angin dengan distribusi data yang tidak normal. Tujuan dari penelitian [20] untuk membandingkan performa dari beberapa model regresi dan hasilnya algoritma *KNN Regression* dapat memprediksi kekuatan turbin dengan akurasi 94% dan diketahui bahwa kecepatan angin merupakan parameter yang memiliki pengaruh terbesar terhadap kekuatan generator turbin.

XGBoost Regression yang merupakan salah satu metode pada penelitian ini juga pernah digunakan pada penelitian [16]. Tujuan dari penelitian tersebut untuk memprediksi hasil panen tumbuhan biji-bijian menggunakan data ramalan cuaca khususnya curah hujan. Bukan hanya algoritma *XGBoost Regression* saja, namun penelitian tersebut juga menggunakan algoritma *Linear Regression* dan *D-Tree Regression* untuk dibandingkan sehingga mendapatkan model yang memiliki performa terbaik. Menggunakan data hasil panen serta ramalan cuaca, penelitian ini

menghasilkan bahwa *XGB Regressor* mampu memprediksi hasil panen biji-bijian di Kazakhstan dengan akurasi yang tinggi dibandingkan dengan algoritma lainnya.

Seperti yang diketahui bahwa penelitian ini bertujuan untuk membandingkan beberapa model regresi, tujuan yang sama pernah dilakukan pada penelitian [22]. Penelitian [22] bertujuan untuk membandingkan 6 model regresi yang berbeda sehingga ditemukan model terbaik dalam memprediksi rating restoran serta mengidentifikasi variabel yang mempengaruhi rating sebuah restoran. Menggunakan data restoran yang dikumpulkan dari situs web *Zomato* dengan variabel berupa nama restoran, alamat restoran, jumlah menu dan lain sebagainya. Hasilnya yaitu model *Random Forest Regression* dengan akurasi 92% memiliki performa terbaik dibanding dengan model lainnya, diketahui juga bahwa variabel yang paling berpengaruh adalah harga makanan yang lebih tinggi cenderung memiliki rating yang lebih tinggi.

Penelitian lain yang memiliki tujuan membandingkan beberapa model regresi juga pernah dilakukan pada penelitian [21]. Menggunakan *Graduate Admission Data* yang berasal dari web Kaggle dengan variabel berupa *TOEFL score*, rating universitas dan lain sebagainya. Dari 4 model regresi yang dibandingkan, didapat bahwa model *Random Forest Regression* memiliki nilai absolut *error* terendah dan memberikan hasil prediksi yang akurat serta memiliki kemampuan generalisasi yang lebih baik.

Pengolahan *missing value* pada penelitian ini memiliki metode yang sama dengan penelitian [17] dengan menggunakan metode *KNN imputation*. Penggunaan metode ini bertujuan untuk mengatasi masalah *missing values* dalam dataset BMKG yang digunakan untuk memprediksi durasi hujan. Data dari BMKG penelitian [17] memiliki beberapa variabel iklim diantaranya tekanan udara, kecepatan angin, curah hujan dan lain sebagainya dengan distribusi data yang tidak normal dan berkarakteristik non-*linear*. Penggunaan metode tersebut mampu menghasilkan prediksi yang lebih akurat dalam mengatasi *missing values* dalam dataset dengan akurasi model sebesar 71% dan persentase *missing value* sebesar 90%.

Model *Random Forest Regression* pada penelitian ini pernah digunakan pada penelitian [18] dengan tujuan untuk mengembangkan metode estimasi kapasitas baterai secara otomatis yang akurat dan efisien. Fokus dari penelitian [18] yaitu dengan menggunakan model *Random Forest Regression* yang diharapkan mampu dalam mengestimasi masa penuaan dari suatu baterai. Model regresi yang dijadikan sebagai metode utama pada penelitian [18] berhasil digunakan untuk estimasi kapasitas baterai secara otomatis dengan *error* yang rendah yaitu 1,3%, metode ini juga efisien dalam hal komputasi dan tidak memerlukan pre-seleksi fitur.

Teknik transformasi pada penelitian ini pernah dilakukan pada penelitian [23]. Penelitian [23] menggunakan beberapa teknik transformasi yaitu *BoxCox, Winsorizing* dan *Trimming data.* Adanya *outlier* dan ketidaknormalan data dapat menyebabkan bias dalam mengestimasi parameter serta mengganggu distribusi data. Dari penggunaan tiga teknik transformasi tersebut, didapat bahwa teknik transformasi *BoxCox* mampu menangani *outlier* serta ketidaknormalan pada data.

Perlunya penskalaan pada penelitian ini dijelaskan pada penelitian [24]. Menggunakan teknik penskalaan dapat mengurangi kesalahan prediksi yang berakibat pada meningkatnya akurasi pada model. Penelitian [24] menyatakan bahwa dari 11 algoritma yang digunakan, algoritma CART yang dikombinasikan dengan metode *StandardScaler* menunjukkan kinerja terbaik dengan akurasi, presisi, *recall*, dan skor F1 yang tinggi dan stabil.

Pembuatan suatu model prediksi bukan hanya sekedar menghasilkan keakuratan model yang tinggi serta meminimalkan kesalahan dalam prediksi saja. Interpretasi hasil prediksi serta faktor-faktor yang mempengaruhi juga tidak kalah penting dalam suatu penelitian mengenai model prediksi. Penelitian [25] yang membandingkan 6 algoritma regresi menghasilkan bahwa *XGBoost Regression* merupakan model yang memiliki akurasi yang baik dalam melakukan prediksi harga emas. Interpretasi mengenai faktor-faktor apa saja yang berpengaruh terhadap fluktuasi harga emas dijelaskan secara baik oleh metode *SHAP*

Tabel 2. 1 Literatur *Review*

Literatur	**Latar Belakang Penelitian**		**Desain Riset dan Metodologi**		
Penulis, Tahun, Judul	**Masalah Penelitian/ Rumusan Masalah**	**Tujuan**	**Metode**	**Data**	**Hasil/Temuan/Kesimpulan**
[1], 2020	Perubahan iklim yang terjadi dapat berpengaruh terhadap hasil tanaman cabai rawit	Mengetahui pengaruh perubahan iklim dan luas panen terhadap produktivitas cabai rawit	*Linear regression*	Data iklim BMKG, data luas panen dan data produktivitas cabai rawit kabupaten Malang tahun 2003-2018	Adanya pengaruh signifikan yaitu 85% antara perubahan iklim serta waktu panen dengan produktivitas tanaman cabai rawit.
[13], 2019	Harga cabai merah mengalami inflasi dari tahun ke tahun, hal tersebut dipengaruhi oleh faktor	Mengklasifikasi harga cabai apakah harga tersebut menguntungkan	*KNN* dan *ADASYN*	Data historis harga cabai per bulan dan data cuaca dari	Dalam pengklasifikasian harga menggunakan *KNN* dan *ADASYN*, penelitian ini mencapai akurasi 100% dan F1-Score 100%.

	permintaan dan iklim. Kedua faktor tersebut berakibat terhadap penurunan produktivitas sehingga berdampak pada naik turunnya harga	atau tidak bagi petani serta mengatasi masalah ketidakseimbangan kelas dalam dataset		BMKG selama empat tahun (2014-2017) di Kabupaten Bandung	
[10], 2019	*Missing values* dalam dataset menyebabkan penurunan dari performa kinerja model dalam melakukan prediksi durasi hujan	Mengatasi masalah *missing values* dalam dataset BMKG yang digunakan untuk memprediksi durasi hujan	*K-Nearest Neighbors Regression*	Data iklim dari Badan Meteorologi, Klimatologi, dan Geofisika (BMKG)	*KNN imputation* menghasilkan *R2-Score* sebesar 0.71 pada 90% data yang mengalami *missing value*.
[19], 2023	Metode *machine learning* yang digunakan untuk memantau status kesehatan baterai memerlukan ekstraksi fitur yang canggih	Mengembangkan metode estimasi kapasitas baterai secara otomatis	*Random Forest Regression*	Data dari sel baterai komersial tipe A dan tipe B	*Random Forest Regression* menghasilkan *error* pada hasil prediksi yang rendah yaitu 1,3%, metode ini juga efisien dalam hal

	sehingga mempersulit implementasi dalam sistem manajemen baterai	yang akurat dan efisien			komputasi dan tidak memerlukan pre-seleksi fitur.
[20], 2019	Fluktuasi harga cabai yang merupakan salah satu masalah ekonomi yang dihadapi oleh petani sehingga diperlukan adanya model untuk menentukan waktu yang tepat untuk menanam cabai dan memaksimalkan produksi.	Mengembangkan model prediksi harga cabai dengan mempertimbangkan faktor cuaca yang mempengaruhi harga cabai di Kabupaten Bandung	*Support Vector Machine* (*SVM*) dan *Adaptive Neuro-Fuzzy Inference System* (*ANFIS*)	Data historis bulanan harga cabai di Pasar Soreang dan data cuaca dari tahun 2014 hingga 2017 di Kabupaten Bandung	Algoritma *SVM* memprediksi harga cabai dengan akurasi yang tinggi ketika semua atribut dimasukkan tanpa normalisasi data, dengan rata-rata akurasi sebesar 92,68%.
[16], 2021	Banyak mahasiswa yang menghadapi kesulitan dalam menyusun daftar universitas karena kurangnya pengetahuan tentang peringkat universitas atau	Membandingkan beberapa model regresi dalam memprediksi penerimaan	*Linear Regression, Support Vector Regression, Decision Tree*	*Graduate Admission Data* dari Kaggle	Model *Random Forest Regression* mencapai *Mean Squared Error (MSE)* terendah, akurat dan memiliki kemampuan generalisasi yang lebih baik

	karena informasi yang salah dari senior dan rekan sesama pelamar	mahasiswa pada program magister	*Regression* dan *Random Forest Regression*.		
[22], 2020	Kebutuhan untuk melakukan prediksi daya turbin angin dengan akurasi tinggi sangat diperlukan karena sifat angin yang tidak langsung, prediksi daya turbin angin menjadi penting untuk menjaga keseimbangan sistem tenaga listrik	Membandingkan beberapa algoritma machine learning dan mengevaluasi beberapa parameter iklim terhadap parameter daya turbin angin	*Linear regression, K-Nearest Neighbor regression. Decision tree regression*	*Database on Wind Characteristics* yang disediakan oleh *Technical University of Denmark & Risø National Laboratory*	Algoritma *linear regression* memberikan *R2-Score* yang lebih tinggi, sementara algoritma *K-Nearest Neighbor regression.* menghasilkan nilai *MAE* yang lebih rendah. Kecepatan angin merupakan parameter yang memiliki pengaruh terbesar terhadap daya turbin angin.
[21], 2019	Persaingan yang tinggi dalam industri restoran di Bangalore, India memerlukan prediksi terhadap rating suatu	Menganalisis faktor-faktor yang mempengaruhi rating restoran di Bangalore	*Linear regression, K-Nearest Neighbor regression.*	Data restoran yang dikumpulkan dari situs web Zomato	*Random Forest Regression* menghasilkan tingkat kesalahan yang paling rendah. Faktor yang paling berpengaruh terhadap rating restoran adalah biaya

	restoran untuk melihat parameter apa saja yang dibutuhkan agar pelanggan memberi rating yang baik.		*Decision tree regression, Ridge Regression, Lasso Regression, Bayesian Regression*		
[17], 2022	Kebutuhan masyarakat akan informasi yang akurat dan terkini terkait prediksi harga rumah.	Mengetahui informasi prediksi harga rumah yang dapat digunakan sebagai acuan perencanaan di masa depan.	Regresi Linier, *Random Forest Regression* dan *Gradient Boosted Trees Regression Method*	Data harga rumah di Jakarta Selatan dan Tebet	Model *Random Forest Regression* menghasilkan nilai prediksi dengan margin kesalahan sekitar ±5
[18], 2018	Kebutuhan untuk memprediksi hasil panen	Membuat model prediksi hasil panen	*XGBRegressor Algorithm*	Data hasil panen produk biji-	*XGBRegressor* mampu memprediksi hasil panen biji-bijian

	secara akurat dalam sektor pertanian di Kazakhstan diperlukan digitalisasi pertanian yang menjadi penting dalam mengurangi dampak iklim	untuk memudahkan departemen pertanian dalam melakukan pemantauan		bijian dari Badan Perencanaan Strategis dan Reformasi Republik Kazakhstan dan data ramalan cuaca	di Kazakhstan dengan akurasi yang tinggi.
[23], 2023	Data *outlier* dapat menyebabkan bias dalam estimasi parameter dan mengganggu distribusi data	Mencari alternatif penyelesaian pada data *outlier* dan ketidaknormalan data.	Transformasi *BoxCox, Winsorizing* dan *Trimming data*	Data persentase kemiskinan di 34 Provinsi di Indonesia tahun 2022, Badan Pusat Statistik	Metode *BoxCox* dan *trimming* sekaligus mampu mengatasi masalah kenormalan data, sedangkan metode *Winsorizing* belum dapat mengatasi masalah kenormalan data.
[24], 2021	Ketidaksesuaian data yang dapat menyebabkan probabilitas kesalahan	Mengevaluasi pengaruh metode penskalaan data	11 algoritma *machine*	Data diagnosa penyakit jantung dari UCI	Algoritma CART dengan metode *scaling data* menunjukkan kinerja

	prediksi yang tinggi dan hasil yang tidak akurat.	terhadap kinerja model pada algoritma pembelajaran mesin	*learning* dan *scaling data*	*machine learning*	terbaik dengan akurasi, presisi, *recall*, dan skor F1 yang tinggi.
[25], 2024	Untuk menilai indikator kinerja ekonomi global di masa depan, diperlukan adanya sistem prediksi fluktuasi harga emas untuk investor, proyek pertambangan dan perusahaan terkait	Memprediksi pergerakan harga emas secara akurat dan menganalisis faktor yang mempengaruhi fluktuasi harga.	*Linear Reg, Neural Network, Random Forest, LightGBM, CarBoost, XGBoost Reg* dan *SHAP*	Faktor-faktor yang mempengaruhi fluktuasi harga emas dari berbagai sumber data *opensource* Tahun 1986-2019	Algoritma *XGBoost* dan nilai interaksi *SHAP* dapat membantu dalam memprediksi pergerakan harga emas secara akurat dan memberikan wawasan tentang faktor-faktor penting yang mempengaruhi harga emas.

2.2 Landasan Teori

Berikut merupakan beberapa landasan teori mengenai metode yang digunakan pada penelitian ini.

2.2.1 Fluktuasi Harga Jual

Harga penjualan adalah jumlah uang yang dibebankan kepada pembeli sebagai imbalan atas produk atau layanan yang diberikan. Besaran biaya ini terdiri dari total biaya produksi dan nonproduksi yang dikeluarkan, termasuk juga keuntungan yang diinginkan oleh perusahaan [26]. Adapun beberapa tujuan dari penetapan harga jual antara lain yaitu memaksimalkan keuntungan, memaksimalkan pangsa pasar, memaksimalkan harga tertinggi dan cara agar suatu perusahaan dapat bertahan dalam pasar [27]. Cara dalam menentukan harga jual yaitu mengetahui nilai atau kualitas produk, mengetahui harga eceran, menghitung total biaya produksi, mencari laba dari harga jual dan permintaan serta penawaran barang [26].

Menurut Yohanes Surya, fluktuasi merujuk pada lonjakan atau variabilitas suatu fenomena yang bisa digambarkan dalam bentuk grafik. Fluktuasi harga mengacu pada perubahan nilai atau harga suatu barang atau produk, yang terjadi saat permintaan konsumen meningkat atau menurun terhadap produk tersebut [28].

2.2.2 Iklim

Iklim merujuk pada rangkuman karakteristik perubahan dalam nilai-nilai elemen cuaca seperti suhu udara, kelembaban udara, dan tingkat sinar matahari dalam jangka waktu yang panjang di suatu lokasi. Hal ini juga dapat diartikan sebagai deskripsi kondisi cuaca yang berlaku dalam suatu daerah atau zona. Data iklim melibatkan informasi yang bersifat terpisah seperti radiasi dan lamanya waktu sinar matahari, serta informasi berkelanjutan seperti suhu, kelembaban, tekanan udara, curah hujan dan kecepatan angin [29].

Berdasarkan Perserikatan Bangsa-Bangsa (PBB), perubahan iklim mencakup transformasi kondisi suhu dan cuaca dalam periode yang lama. Perubahan ini dapat terjadi secara alami, seperti akibat perubahan dalam siklus matahari. Tetapi, sejak abad ke-19, aktivitas manusia menjadi faktor kunci dalam

perubahan iklim, terutama melalui penggunaan bahan bakar fosil seperti batu bara, minyak, dan gas [30]. Ketika iklim mengalami perubahan, efeknya dapat dirasakan pada kondisi di daratan maupun di wilayah pesisir atau lautan. Perubahan iklim dapat berdampak pada hasil dan produksi tanaman [31]. Perubahan iklim secara langsung dapat menurunkan produksi pangan dunia sehingga mengakibatkan kenaikan harga bahan pangan [32].

2.2.3 Transformasi *BoxCox*

Transformasi *BoxCox* adalah suatu teknik statistik yang digunakan untuk mengubah distribusi data yang tidak normal menjadi distribusi normal. Teknik ini dikembangkan oleh George Box dan David Cox pada tahun 1964. Transformasi *BoxCox* mempertimbangkan kelas transformasi berparameter tunggal, yaitu λ yang dipangkatkan pada variabel respons Y, sehingga diperoleh model transformasinya dengan λ sebagai parameter yang harus diduga.

Transformasi *BoxCox* dilakukan dengan cara menghitung nilai λ yang optimal untuk mengubah distribusi data menjadi normal. λ dapat berupa bilangan riil, dan nilai λ yang dipilih harus memenuhi beberapa kriteria, seperti ragam dari variabel yang baru tidak dipengaruhi oleh perubahan rata-rata, variabel yang baru hendaknya menyebar normal, skala pengukuran variabel yang baru hendaknya sedemikian sehingga pengaruh sesungguhnya bersifat linier dan aditif. Adapun perhitungan transformasi ini menggunakan persamaan (2.1) [23][33]:

$$Y^{(\lambda)} = \begin{cases} \frac{Y^{(\lambda)}-1}{\lambda} \; if \;\; \lambda \neq 0 \\ \log Y \, , if \; \lambda = 0, \end{cases} \tag{2.1}$$

dengan Y merupakan nilai yang ingin ditransformasi dan λ merupakan parameter dalam *BoxCox*.

Transformasi *BoxCox* tidak hanya berlaku pada variabel respons Y, tetapi juga dapat diterapkan pada variabel bebas X. Namun, dalam beberapa kasus, transformasi *BoxCox* hanya diberlakukan pada variabel respons Y yang memiliki tanda positif. Jika variabel X tidak memenuhi asumsi normalitas, maka transformasi *BoxCox* dapat digunakan untuk mengubah distribusi data X menjadi normal. Diketahui sampel data pada variabel X_1 dan X_2 pada Tabel 2.2 berikut:

Tabel 2. 2 Data Sampel untuk Perhitungan Manual

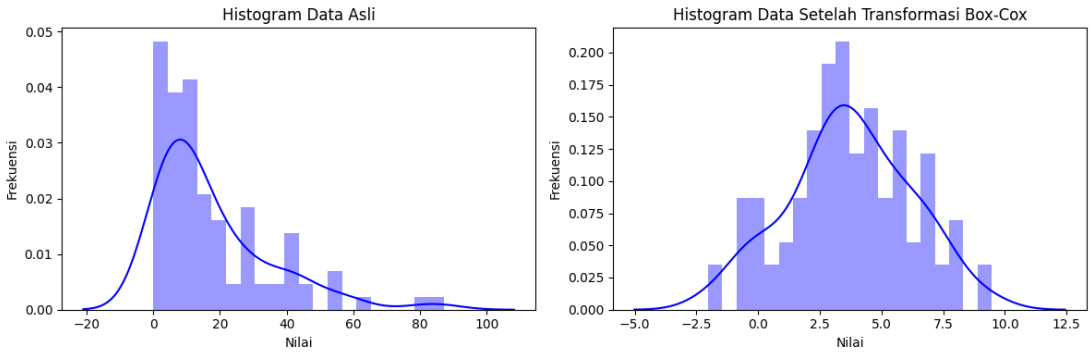
X_1	X_2
5	71
8	65
10	82
15	50
6	62

Jika dimisalkan nilai λ = 0,5, maka perhitungan menggunakan persamaan (2.1) dengan $\lambda \neq 0$ ditampilkan pada Tabel 2.3

Tabel 2. 3 Perhitungan Manual Transformasi *BoxCox* menggunakan Data Sampel

X_1	Transformasi X_1	X_2	Transformasi X_2
5	$\frac{5^{(0,5)}-1}{0,5} = 2,47$	71	$\frac{71^{(0,5)}-1}{0,5} = 14,85$
8	$\frac{8^{(0,5)}-1}{0,5} = 3,65$	65	$\frac{65^{(0,5)}-1}{0,5} = 14,12$
10	$\frac{10^{(0,5)}-1}{0,5} = 4,32$	82	$\frac{82^{(0,5)}-1}{0,5} = 16,11$
15	$\frac{15^{(0,5)}-1}{0,5} = 5,74$	50	$\frac{50^{(0,5)}-1}{0,5} = 12,14$
6	$\frac{6^{(0,5)}-1}{0,5} = 2,89$	62	$\frac{62^{(0,5)}-1}{0,5} = 13,74$

Pada gambar 2.1 terlihat bahwa plot distribusi data sebelum dan setelah transformasi *BoxCox*. Data sebelum di trasnformasi (kiri) terlihat memiliki distirbusi miring ke kanan, sedangkan setelah di transformasi (kanan) terlihat data mendekati distiribusi normal dan memiliki rentang data yang tidak jauh satu sama lain.



Gambar 2. 1 Histogram Data Sebelum Transformasi (kiri) dan Setelah Transformasi (Kanan)

2.2.4 Standarisasi *StandardScaler*

Standarisasi dengan *StandardScaler* adalah salah satu teknik *pre-processing* data yang umum digunakan dalam *machine learning*. Metode ini berguna untuk mengubah skala fitur numerik sehingga memiliki rata-rata 0 dan standar deviasi 1 serta membantu mengurangi efek *outlier* dalam data. Persamaan (2.2) merupakan cara melakukan standarisasi data menggunakan metode ini.

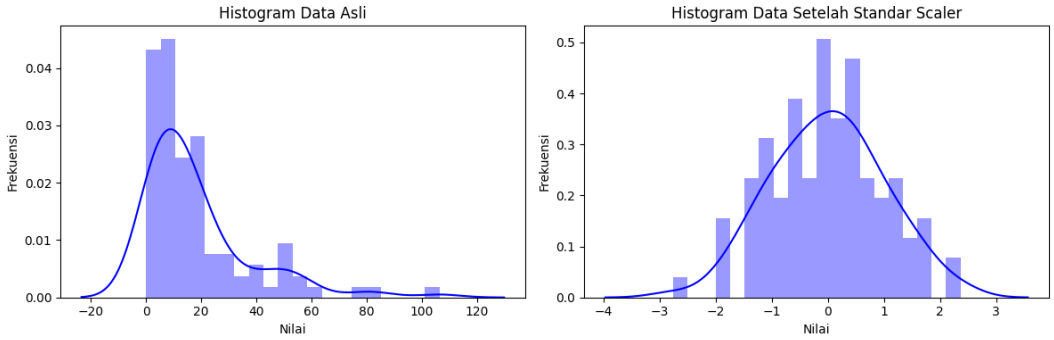
$$z = \frac{(x - \overline{x})}{\sigma}, \qquad (2.2)$$

dengan x merupakan nilai yang ingin di standarisasi, $\overline{x}$ merupakan nilai rata-rata didalam suatu variabel dan σ merupakan standar deviasi variabel [24][34]. Menggunakan data pada Tabel 2.2, berikut merupakan perhitungan manual *StandardScaler* menggunakan persamaan (2.2). Sebelum ke perhitungan *StandardScaler*, cari terlebih dahulu nilai $\overline{x}$ dan σ pada variabel X_1 dan X_2. Diketahui bahwa nilai $\overline{x}$ pada variabel X_1=8,8 dan X_2=66, sedangkan untuk nilai σ pada variabel X_1=3,54 dan $X_2 = 10{,}52$. Tabel 2.4 menampilkan hasil perhitungan menggunakan persamaan (2.2) menggunakan data sampel.

Tabel 2. 4 Perhitungan Manual Standarisasi *StandardScaler* menggunakan Data Sampel

X_1	$StandardScaler\ X_1$	X_2	$StandardScaler\ X_2$
5	$\frac{(5-8{,}8)}{3{,}54}$= -1,07	71	$\frac{(71-66)}{10{,}52}$= 0,47
8	$\frac{(5-8{,}8)}{3{,}54}$= -0,22	65	$\frac{(71-66)}{10{,}52}$= -0,09
10	$\frac{(5-8{,}8)}{3{,}54}$= 0,33	82	$\frac{(71-66)}{10{,}52}$= 1,52
15	$\frac{(5-8{,}8)}{3{,}54}$= 1,75	50	$\frac{(71-66)}{10{,}52}$= -1,52
6	$\frac{(5-8{,}8)}{3{,}54}$= -0,79	62	$\frac{(71-66)}{10{,}52}$= -0,38

Pada gambar 2.2 terlihat bahwa plot distribusi data sebelum dan setelah standarisasi *StandarScaler*. Data sebelum di standarisasi (kiri) terlihat memiliki distirbusi miring ke kanan, sedangkan setelah di transformasi (kanan) terlihat data mendekati distiribusi normal dan memiliki rentang data antara -1 sampai 1 saja.



Gambar 2. 2 Histogram Data Sebelum Standarisasi (kiri) dan Setelah Standarisasi (kanan)

2.2.5 Prediksi Menggunakan Model Regresi Non-Linear

Prediksi sama halnya dengan ramalan atau perkiraan. Menurut Kamus Besar Bahasa Indonesia (KBBI), peramalan merupakan hasil dari upaya memperkirakan nilai-nilai di masa mendatang dengan menggunakan informasi dari periode sebelumnya. Prediksi tidak selalu memberikan jawaban yang pasti terhadap suatu peristiwa, melainkan upaya untuk mendekati jawaban yang seakurat mungkin dengan apa yang sedang terjadi [35].

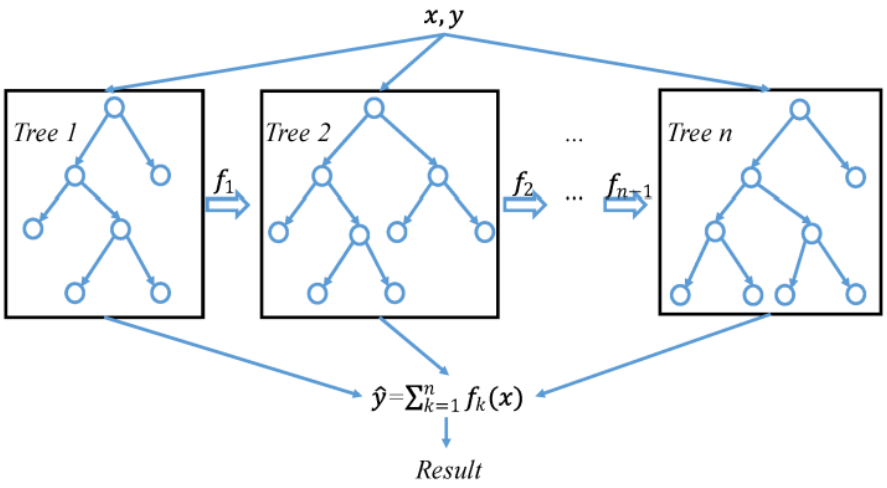
Wantono menjelaskan bahwa peramalan merupakan langkah sistematis dalam mengestimasi kemungkinan besar dari kejadian di masa mendatang [36]. Hal ini didasarkan pada informasi yang ada pada masa sebelumnya dan saat ini, dengan tujuan untuk mengurangi kesalahan atau perbedaan antara apa yang diprediksi dengan apa yang sebenarnya terjadi [37].

Regresi dalam KBBI diartikan sebagai hubungan rata-rata antarvariabel. Regresi dipakai untuk mengamati pengaruh antara dua variabel atau lebih. Analisis regresi secara umum bertujuan untuk mendapatkan prediksi dan ramalan. Regresi dikatakan linear apabila hubungan antara variabel independen dan dependen adalah linear, sedangkan apabila hubungan antara variabel independen dan dependen tidak linear, maka regresi tersebut dapat dikatakan regresi *non* linear. Hubungan antara variabel independen dan dependen dapat dikatakan linear apabila diagram pencar data dari variabel-variabel tersebut mendekati pola garis lurus [14]. Berikut merupakan beberapa algoritma regresi non-linear yang digunakan pada penelitian ini.

a) *XGBoost Regression*

XGBoost adalah sebuah algoritma yang digunakan dalam *gradient tree boosting* untuk membangun model *ensemble* yang kuat dan efisien. Algoritma ini dikembangkan oleh Tianqi Chen dan Carlos Guestrin pada tahun 2014, telah digunakan secara luas oleh para *data scientist* untuk mencapai hasil terbaik dalam berbagai tantangan *machine learning*. *XGBoost* bertujuan untuk meningkatkan akurasi dan performa dalam pembelajaran mesin, sehingga dapat menghasilkan model yang lebih baik dalam memprediksi atau mengklasifikasikan data.

XGBoost Regression adalah metode regresi yang menggunakan algoritma *XGBoost* untuk memprediksi nilai kontinu. Setiap pohon keputusan memprediksi *variabel* target dan menghitung residual yang dihasilkan oleh pohon-pohon sebelumnya. Algoritma ini mampu untuk menangani beberapa karakteristik data seperti jumlah fitur yang besar, kompleksitas antar variabel, distribusi data tidak normal dan data dengan *missing value*. Gambar 2.3 menunjukkan ilustrasi kerja dari algoritma ini.



Gambar 2. 3 Ilustrasi Kerja Algoritma *XGBoost Regression*
(Sumber : [38])

Langkah kerja dari algoritma *XGBoost Regression* yaitu dengan membuat model *tree* awal ke dalam data dengan persamaan (2.3)

$$f_1(x) = y, \qquad (2.3)$$

dengan nilai y merupakan hasil prediksi yang didapatkan dari *tree* awal. Prediksi yang telah dihasilkan dari *tree* awal masuk kedalam persamaan (2.4) untuk menghitung residual

$$h_1(x) = y - f_1(x), \qquad (2.4)$$

sehingga *tree* selanjutnya dibangun berdasarkan besarnya residual di *tree* awal, yang dihitung dengan persamaan (2.5)

$$f_2(x) = f_1(x) + h_1(x) \qquad (2.5)$$

Pembuatan *tree* baru secara rekursif menggunakan metode *additive* untuk mendapatkan *tree* akhir. *Tree* akhir didapat dari persamaan (2.6)

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_t(x_i), \qquad (2.6)$$

Notasi :

- $\hat{y}_i^{(t)}$ = *Tree* akhir
- $\hat{y}_i^{(t-1)}$ = Prediksi dari *tree* sebelumnya
- $f_t(x_i)$ = Model terbaru dari *tree* selanjutnya
- k = Banyaknya pohon

Algoritma ini mengoptimalkan fungsi objektif (*loss function*) dalam setiap langkah dengan menambahkan pohon baru secara iteratif. Dalam kasus regresi, umumnya digunakan fungsi objektif yang berhubungan dengan mengurangi kesalahan prediksi, seperti *Mean Squared Error* (*MSE*) atau fungsi objektif yang serupa untuk meminimalkan *loss function*, formulanya pada persamaan (2.7)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \qquad (2.7)$$

Notasi :

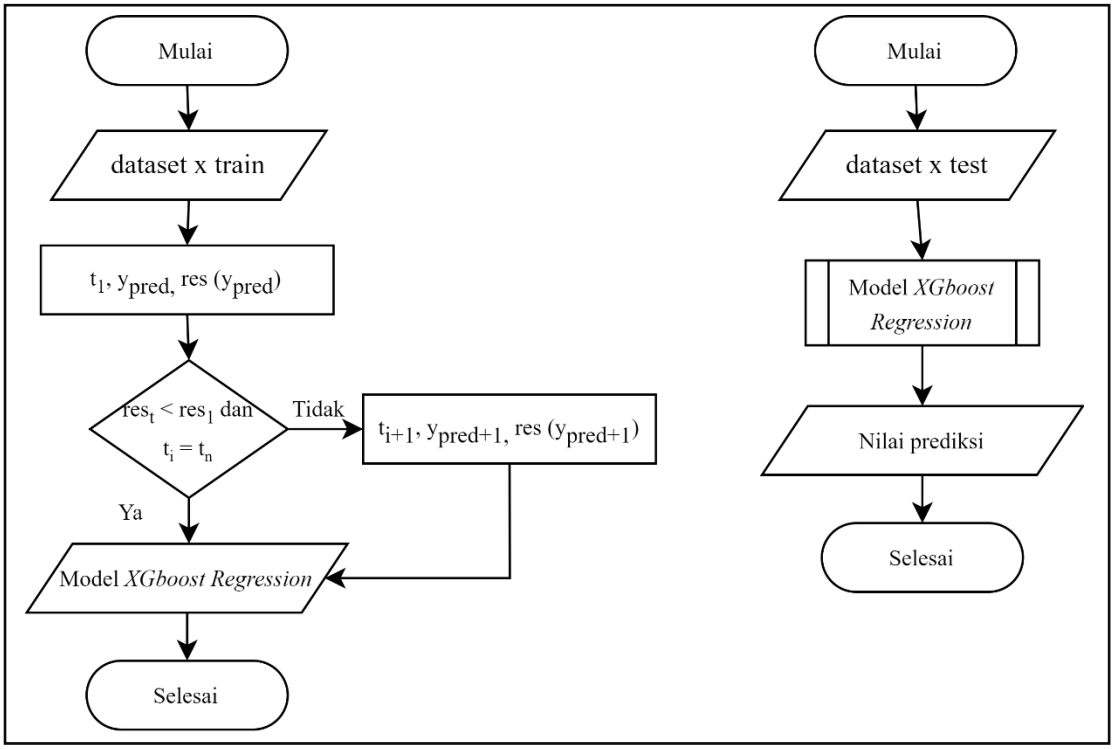
- n = Banyaknya sampel
- y_i = Nilai aktual
- $\hat{y}_i$ = Nilai prediksi

Gradien yang dihasilkan dari *MSE* terhadap prediksi dapat dihitung sebagai turunan parsial dari *MSE* terhadap prediksi, seperti persamaan (2.8)

$$\frac{\partial MSE}{\partial \hat{y}_i} = -2(y_i - \hat{y}_i), \qquad (2.8)$$

gradien ini memberikan informasi tentang arah dan seberapa besar pohon mengubah prediksi $\hat{y}_i$ agar mendekati nilai sebenarnya. Model atau *tree* baru dapat meminimalkan *loss function* yang dapat digunakan untuk melihat skor kualitas struktur pohon, semakin kecil *loss function* maka model atau *tree* tersebut semakin baik [39].

Langkah-langkah dalam melakukan *fitting* data latih menggunakan *XGBoost Regression* dapat dilihat pada Gambar 2.4



Gambar 2. 4 *Flowchart* Model *XGBoost Regression*

Berikut merupakan penjelasan *flowchart* yang terdapat pada Gambar 2.4

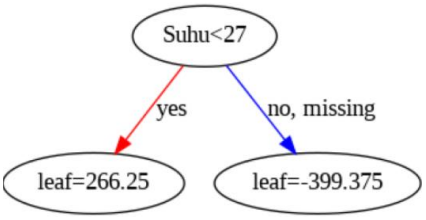
1) Langkah awal yang harus dilakukan yaitu memasukkan seluruh data beserta variabel yang telah diproses sebelumnya pada tahap *pre-processing.* Tabel 2.5 merupakan contoh beberapa data yang dianalisis.

Tabel 2. 5 Dataset Untuk Perhitungan Manual

Suhu	Kelembapan	Curah Hujan	Lama Penyinaran	Kec Angin	Harga Jual
27	88	9,6	3,6	4	49600

27,4	87	16,7	3,4	3	49600
28,4	84	2	2,5	3	44400
27,1	90	36,6	4,6	2	47000
26,8	92	3,7	0,8	2	50000
26,3	92	20,3	0,2	2	59000
26,7	94	28,8	3,2	2	55000
26,9	93	28	3,1	2	54400
26,4	90	13,5	2,1	2	56000
26,9	93	2,5	2,9	2	59200

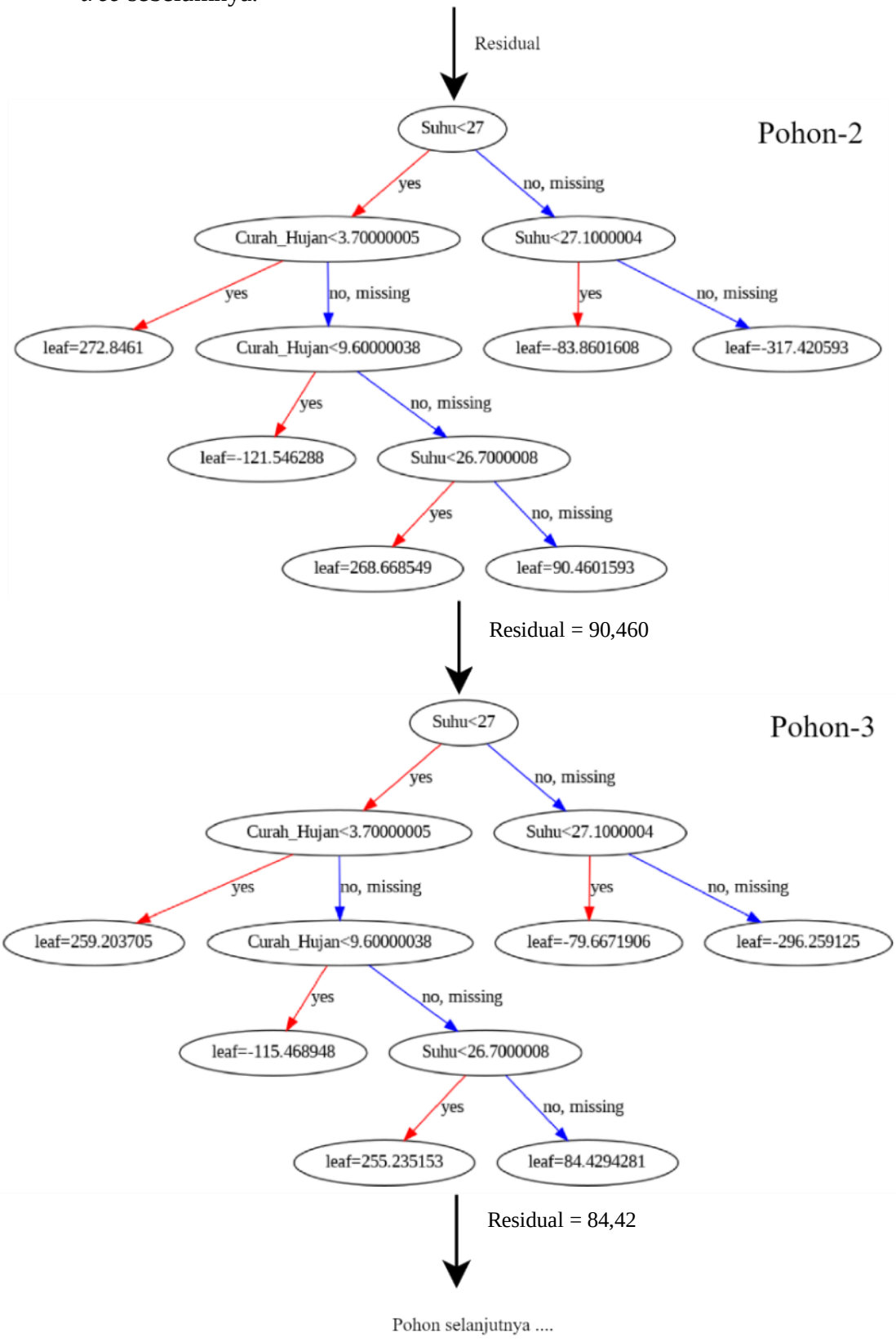
2) Membangun *tree* awal dari data yang telah di*input*kan. Gambar 2.5 menampilkan *tree* awal yang dibangun.

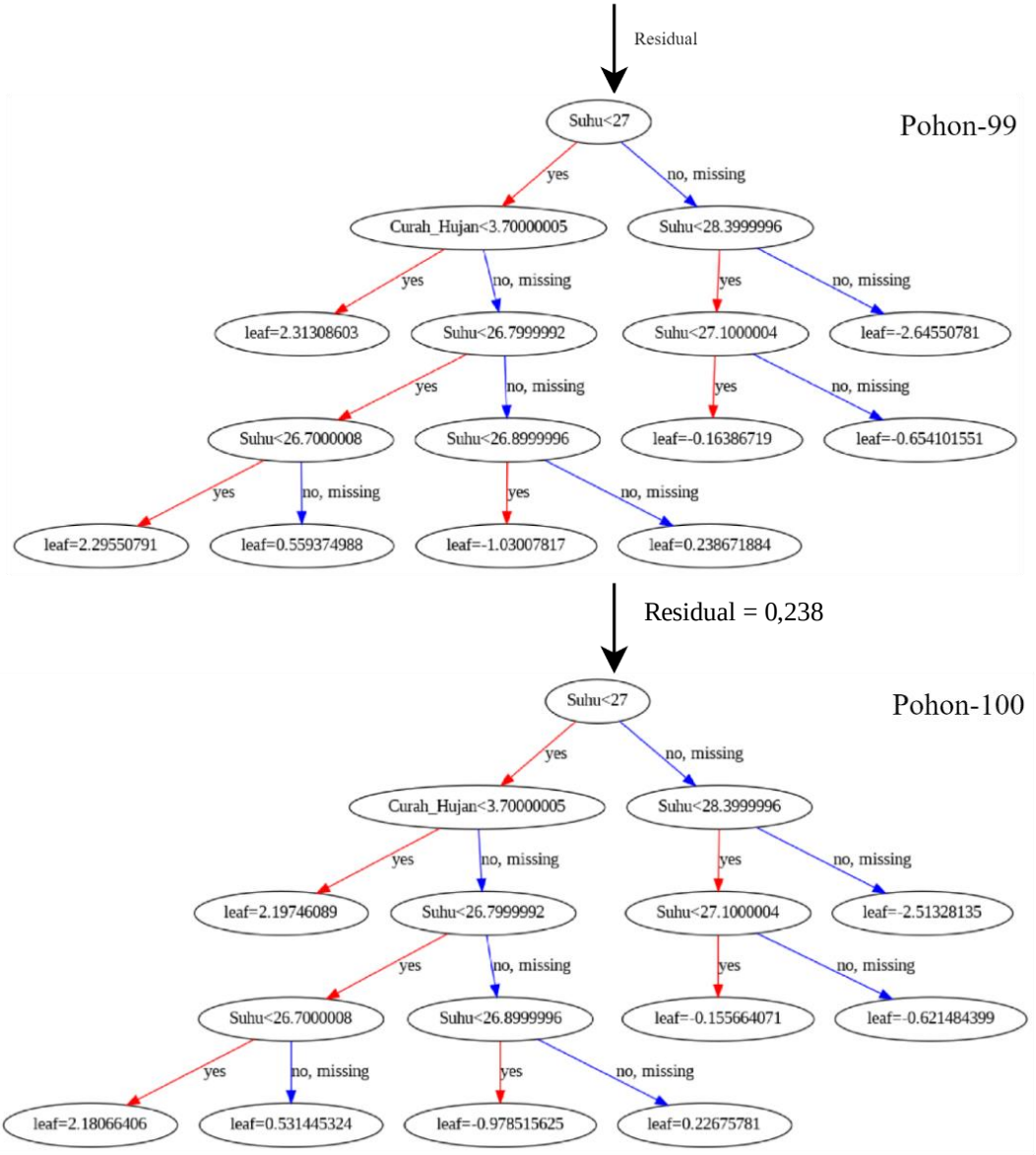


Gambar 2. 5 Pembangunan *Tree* Awal

3) Dari *tree* yang telah dibangun sebelumnya menghasilkan prediksi awal. Terlihat pada Gambar 2.5 *loss function* yang dihasilkan yaitu -399,375, sedangkan untuk nilai yang terprediksi dengan benar yaitu 266,25
4) Prediksi yang sudah dihasilkan masuk ke dalam proses perhitungan residual $\hat{y}$ menggunakan persamaan (2.4)
5) Pada tahap percabangan, terdapat kondisi di mana jika residual *tree* baru (res_t) kurang dari residual *tree* awal (res_1) dan jumlah *tree* yang telah dibangun (f_t) sudah memenuhi banyaknya *tree* (f_k). Jika kondisi tersebut tidak terpenuhi, maka *tree* dibangun secara rekursif sampai memenuhi kondisi yang telah ditentukan.
6) Pembuatan seluruh *tree* baru dilakukan hingga mencapai jumlah maksimal *tree* yang telah ditentukan serta kondisi pada langkah ke 5). Pada contoh perhitungan ini menggunakan jumlah *tree* sebanyak 100.
7) Proses pembuatan pohon baru dilakukan dengan mempertimbangkan bagaimana pohon baru tersebut dapat mengurangi kesalahan prediksi yang masih tersisa (residual) dari model sebelumnya. Gambar 2.6 merupakan *tree*-

tree yang dibangun secara *rekursif* dengan mempertimbangkan *residual* pada *tree* sebelumnya.





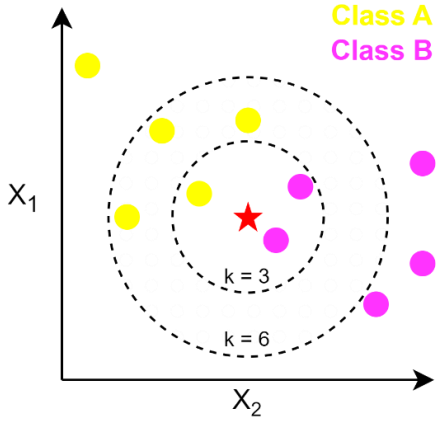
Gambar 2. 6 Pembangunan *Tree-Tree XGBoost Regression*

8) Pohon baru dipilih berdasarkan peningkatan (gain) dalam mengurangi residual atau fungsi objektif (*loss function*) yang ditetapkan seperti *Mean Squared Error* (*MSE*). Terlihat pada Gambar 2.6 *loss function* yang dihasilkan *tree* ke-100 mencapai -0,226. Hal tersebut menandakan bahwa model dapat melatih data dengan meminimalkan residual dan *loss function*
9) Menghitung nilai prediksi pada masing-masing *tree* yang baru dibangun
10) Menghitung nilai residual untuk mengetahui apakah *tree* yang baru dibangun memiliki kualitas yang baik atau tidak

11) Apabila kondisi sudah terpenuhi, maka nilai prediksi akhir muncul.

b) *K-Nearest Neighbor Regression*

K-Nearest Neighbor (KNN) adalah metode pengenalan pola yang tidak bergantung *pada* parameter, berguna untuk klasifikasi dan regresi. Algoritma ini memanfaatkan perhitungan jarak antar titik data dan menentukan tetangga terdekat untuk setiap titik data tertentu. *K-nearest neighbour* digunakan sebagai metode non-parametrik untuk analisis statistik dan pengenalan pola sejak awal tahun 1970an. Salah satu cara sederhana untuk menerapkan regresi *KNN* adalah dengan menghitung nilai rata-rata target numerik dari *K* tetangga terdekat. Gambar 2.7 menunjukkan ilustrasi kerja dari algoritma ini.



Gambar 2. 7 Ilustrasi Kerja Algoritma *KNN*
(Sumber : [40])

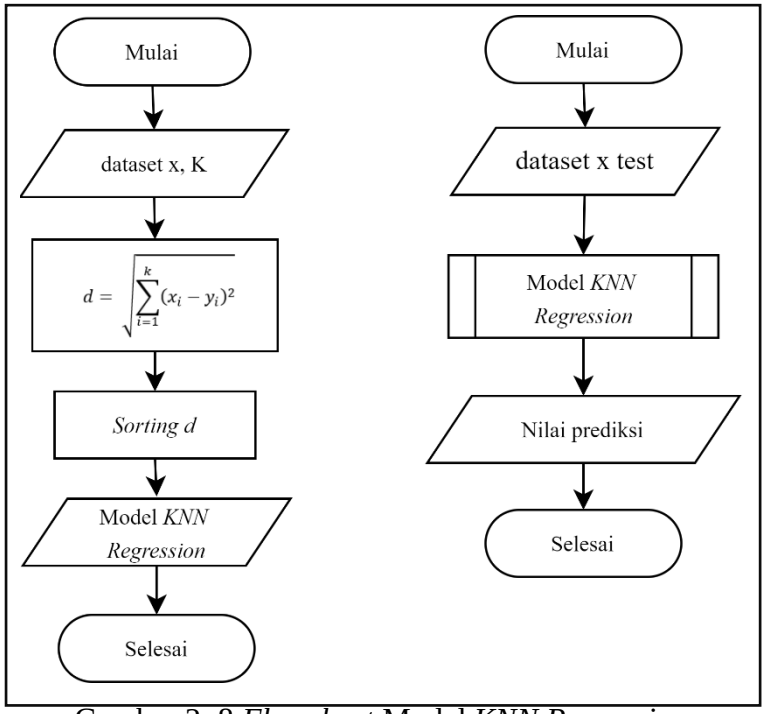
Algoritma *KNN Regression* menghitung jarak antara titik baru dan setiap titik *latihan*. Ada berbagai metode untuk menghitung jarak ini, metode yang paling umum dikenal pada data kontinyu adalah *Euclidean*. Persamaan (2.9) merupakan perhitungan dari *Euclidean*

$$d = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}, \qquad (2.9)$$

Notasi :

- d = Jarak *eucledian*
- x_i = Variabel *x* di setiap indeks
- y_i = Variabel y di setiap indeks
- k = Banyaknya data

Langkah-langkah dalam melakukan *fitting* data latih menggunakan *KNN Regression* dapat dilihat pada Gambar 2.6 [17].



Gambar 2. 8 *Flowchart* Model *KNN Regression*

Berikut merupakan penjelasan *flowchart* yang terdapat pada Gambar 2.8 serta perhitungan manual dari model *KNN Regression*

1) Langkah awal yang harus dilakukan yaitu memasukkan seluruh data beserta variabel yang telah diproses sebelumnya pada tahap *pre-processing*, Tabel 2.5 merupakan contoh beberapa data yang di analisis.
2) Menetapkan nilai *K* yang dapat ditentukan secara manual. Pada perhitungan ini menggunakan *K*=3
3) Menghitung jarak setiap data menggunakan metode *Euclidean* terhadap nilai *K* yang telah ditentukan. Pada perhitungan manual memfokuskan jarak *Euclidean* menggunakan persamaan (2.9) antara data pada *index* 1 terhadap data lainnya. Tabel 2.6 merupakan hasil dari perhitungan data menggunakan metode *Euclidean.*

Tabel 2. 6 Perhitungan Jarak *Euclidean*

Data Uji	Data Latih	Jarak *Euclidean*
(27, 88, 9,6, 3,6, 4)	(27,4, 87, 16,7, 3,4, 3)	1,44
(27, 88, 9,6, 3,6, 4)	(28,4, 84, 2, 2,5, 3)	4,24
(27, 88, 9,6, 3,6, 4)	(27,1, 90, 36,6, 4,6, 2)	5,66
(27, 88, 9,6, 3,6, 4)	(26,8, 92, 3,7, 0,8, 2)	3,06
(27, 88, 9,6, 3,6, 4)	(26,3, 92, 20,3, 0,2, 2)	6,08
(27, 88, 9,6, 3,6, 4)	(26,7, 94, 28,8, 3,2, 2)	5,32
(27, 88, 9,6, 3,6, 4)	(26,9, 93, 28, 3,1, 2)	4,76
(27, 88, 9,6, 3,6, 4)	(26,4, 90, 13,5, 2,1, 2)	4,48
(27, 88, 9,6, 3,6, 4)	(26,9, 93, 2,5, 2,9, 2)	5,28

4) Memilih nilai *K* berdasarkan jarak terdekat dari masing-masing data. Tabel 2.7 merupakan beberapa *K* yang terdekat terhadap data uji.

Tabel 2. 7 Pemilihan K Terdekat Dari Data Uji

K	Data Latih	Jarak *Euclidean*
1	(27,4, 87, 16,7, 3,4, 3)	1,44
2	(27,1, 90, 36,6, 4,6, 2)	5,66
3	(26,8, 92, 3,7, 0,8, 2)	3,06

5) Menghitung rata-rata jarak antara nilai *K* yang dipilih terhadap data terdekat, sehingga perhitungan tersebut menghasilkan nilai prediksi. Perhitungan pada langkah ke 4) menghasilkan 3 *indeks* data terdekat terhadap data uji yaitu data pada Tabel 2.5 pada baris ke-2, 4 dan 5, maka nilai target dari masing-masing data tersebut dihitung menggunakan rata-rata untuk menghasilkan nilai prediksi dari data uji.

$$Prediksi = \frac{49600 + 47000 + 50000}{3}$$

$$Prediksi = 49000$$

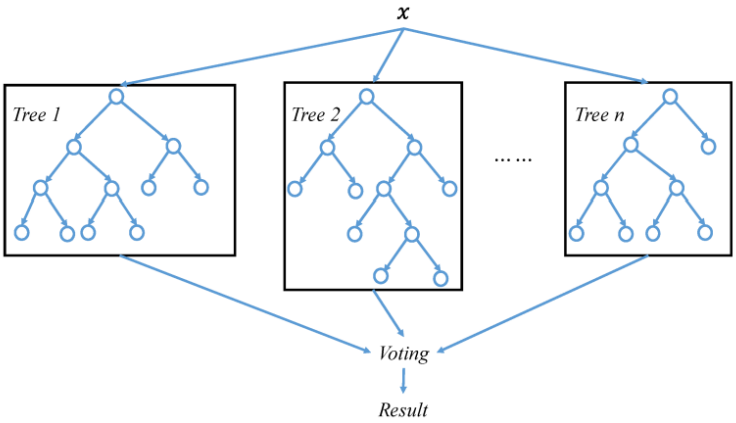
Maka didapat hasil prediksi akhir menggunakan model *KNN Regression* yaitu Rp49.000

c) *Random Forest Regression*

Konsep dari algoritma *Random Forest* adalah menggabungkan prediksi dari beberapa pohon keputusan (*decision trees*) yang dibangun secara acak untuk mencapai akurasi prediksi yang lebih tinggi. Setiap pohon keputusan dibangun dengan menggunakan subset acak dari data pelatihan dan subset acak dari fitur-fitur yang tersedia. Setelah semua pohon keputusan dibangun, hasil prediksi dari setiap

pohon diambil dan mayoritas suara digunakan untuk menentukan prediksi akhir. Algoritma ini menggabungkan kekuatan dari *ensemble learning* dan *randomization* untuk mengurangi *overfitting* dan meningkatkan generalisasi model. Perbedaan algoritma ini dengan *XGBoost Regression* yaitu bagaimana metode dalam mengambil dan menentukan prediksi akhir. Dalam algoritma *XGBoost Regression*, *error* yang dihasilkan oleh suatu hasil prediksi dipelajari pada pohon berikutnya, sehingga dapat meminimalkan kesalahan dalam prediksi, sedangkan algoritma *Random Forests* melakukan agregasi dan menghitung *vote* terbanyak dalam menentukan prediksi akhir.

Random Forests juga dapat digunakan untuk masalah regresi. Dalam metode *Random Forests* untuk regresi, pohon-pohon keputusan dibangun menggunakan vektor acak, di mana prediktor pohon menghasilkan nilai numerik sebagai hasilnya. Adapun perbedaan *Random Forests* dan *Random Forests Regression* terletak pada nilai prediksi yang dihasilkan dan metode pembagian *node*. *Random Forests Regression* menghasilkan nilai prediksi berupa kontinyu dan menggunakan *MSE* sebagai metode dalam membagi *node*. Gambar 2.9 menunjukkan ilustrasi kerja dari algoritma *Random Forests Regression*.



Gambar 2. 9 Ilustrasi Kerja Algoritma *Random Forest Regression*
(Sumber : [38])

Data pelatihan dibagi menjadi subset acak dengan pengambilan sampel dengan penggantian (*bootstrap*). Setiap subset ini digunakan untuk membangun pohon keputusan. Untuk setiap subset data, pohon keputusan dibangun dengan menggunakan algoritma CART (*Classification and Regression Trees*). Dalam

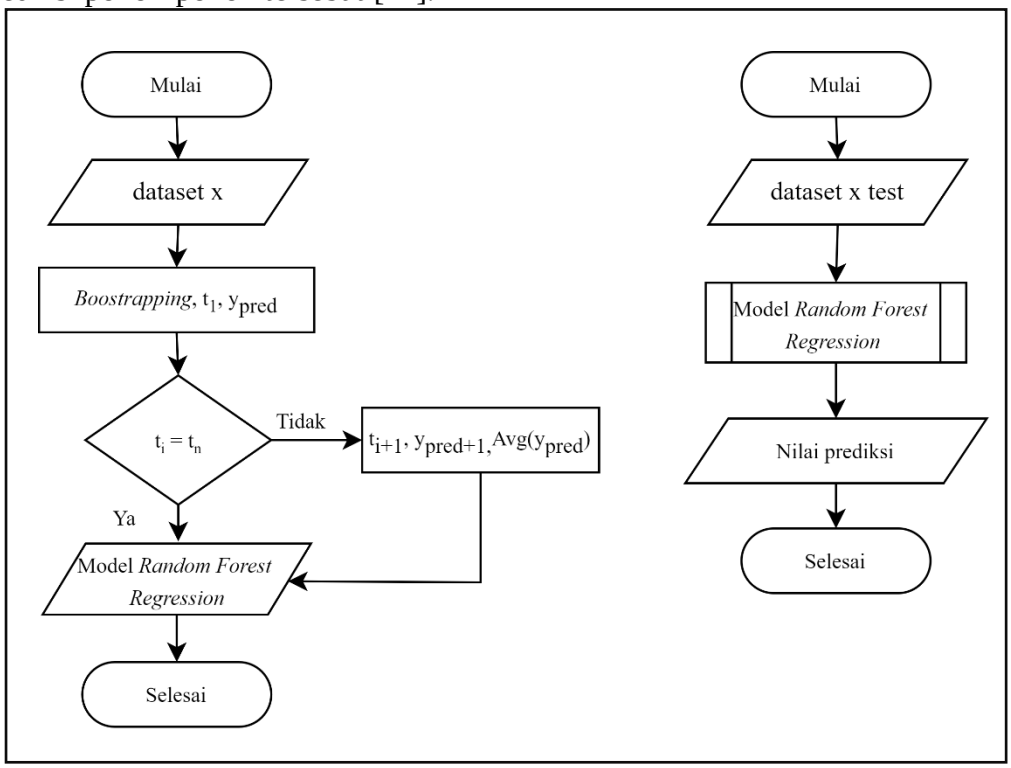
setiap *node* pada pohon keputusan, tujuan utama adalah membagi data sedemikian rupa sehingga menghasilkan prediksi target yang semakin akurat. Hal ini dilakukan dengan memilih fitur dan nilai ambang yang meminimalkan *MSE* di setiap tahap pemisahan. Setiap fitur dan nilai ambang yang dipilih untuk membagi *node* dalam pohon keputusan dihitung nilai *MSE*. Persamaan (2.10) merupakan formula untuk menghitung *MSE*

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i), \qquad (2.10)$$

Notasi :

- n = Banyaknya sampel
- y_i = Nilai aktual
- $\hat{y}_i$ = Nilai prediksi

Dalam *Random Forests*, pohon keputusan dapat berhenti dibangun jika mencapai kriteria penghentian tertentu, seperti mencapai kedalaman maksimum yang ditentukan atau jika tidak ada lagi pemisahan yang signifikan dalam data. Setelah semua pohon keputusan dibangun, prediksi dari setiap pohon diambil. Prediksi akhir diperoleh dengan mengambil mayoritas suara atau rata-rata dari prediksi pohon-pohon tersebut [41].



Gambar 2. 10 *Flowchart* Model *Random Forest Regression*

Langkah-langkah dalam melakukan *fitting* data latih menggunakan *Random Forest Regression* dapat dilihat pada Gambar 2.10 Berikut merupakan penjelasan *flowchart* yang terdapat pada Gambar 2.10 serta perhitungan manual dari model *Random Forest Regression*

1) Langkah awal yang harus dilakukan yaitu memasukkan seluruh data beserta variabel yang telah diproses sebelumnya pada tahap *pre-processing*, Tabel 2.5 merupakan contoh beberapa data yang di analisis.
2) Data yang telah di*input*kan kemudian masuk kedalam *bootstrapping* data yaitu pengambilan sampel secara acak dari dataset dengan pengembalian, dengan penentuan parameter berupa jumlah pohon=3. Tabel 2.8 merupakan hasil dari *bootstrapping* fitur-fitur yang ada di dataset.

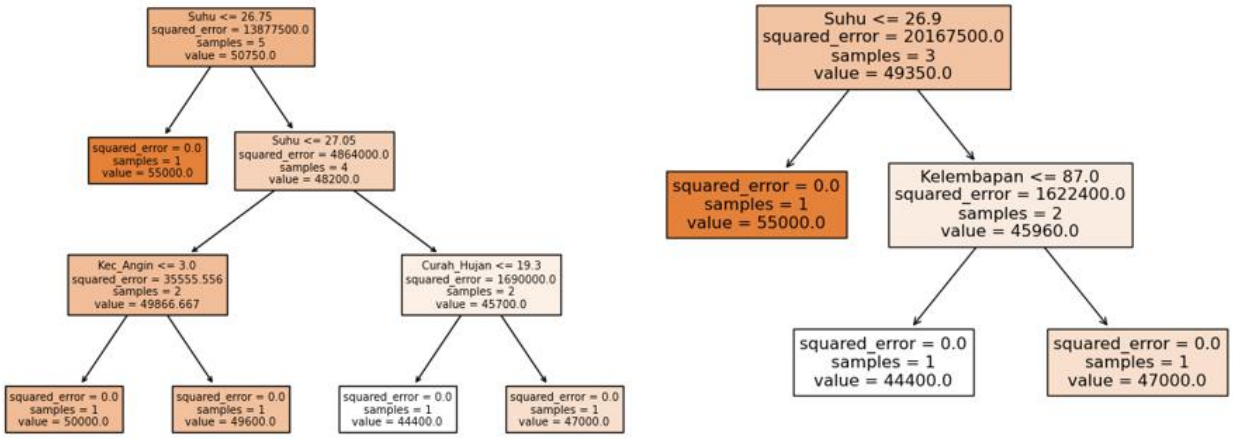
Tabel 2. 8 *Bootstrapping* Fitur-Fitur Dataset

Pohon-1				
Suhu	Kelembapan	Curah_Hujan	Lama_Penyinaran	Kec_Angin
Node-1				
26,3	92	20,3	0,2	2
26,7	94	28,8	3,2	2
26,4	90	13,5	2,1	2
Node-4				
26,8	92	3,7	0,8	2
26,9	93	28	3,1	2
26,9	93	2,5	2,9	2
Node-5				
27	88	9,6	3,6	4
Node-7				
27,4	87	16,7	3,4	3
28,4	84	2	2,5	3
Node-8				
27,1	90	36,6	4,6	2
Pohon-2				
Suhu	Kelembapan	Curah_Hujan	Lama_Penyinaran	Kec_Angin
Node-1				
26,8	92	3,7	0,8	2
26,3	92	20,3	0,2	2
26,7	94	28,8	3,2	2
26,9	93	28	3,1	2
26,4	90	13,5	2,1	2
26,9	93	2,5	2,9	2
Node-3				
27,4	87	16,7	3,4	3
28,4	84	2,0	2,5	3
Node-4				
27	88	9,6	3,6	4
27,1	90	36,6	4,6	2

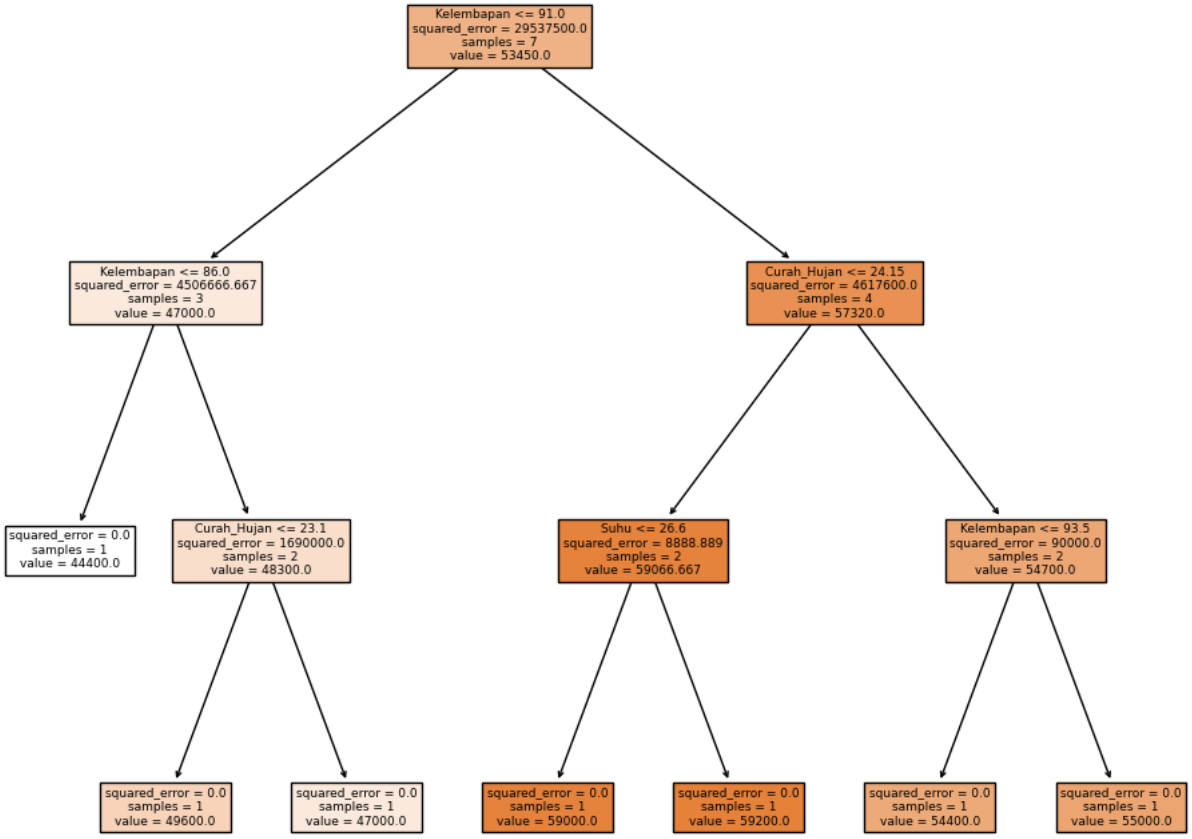
Pohon-3				
Suhu	Kelembapan	Curah_Hujan	Lama_Penyinaran	Kec_Angin
Node-2				
28,4	84	2	2,5	3
Node-4				
27	88	9,6	3,6	4
27,4	87	16,7	3,4	3
26,4	90	13,5	2,1	2
Node-5				
27,1	90	36,6	4,6	2
Node-8				
26,3	92	20,3	0,2	2
Node-9				
26,8	92	3,7	0,8	2
26,9	93	2,5	2,9	2
Node-11				
26,9	93	28	3,1	2
Node-12				
26,7	94	28,8	3,2	2

3) Tahap selanjutnya melakukan pembangunan *tree* hingga mencapai ukuran maksimum yang telah ditentukan tanpa adanya *pruning* pada *tree*

4) Setiap fitur dan nilai ambang yang dipilih untuk membagi *node* dalam pohon keputusan dihitung *MSE*-nya menggunakan persamaan (2.10). Pemisahan yang menghasilkan penurunan *MSE* yang paling signifikan dipilih. Gambar 2.11 dan Gambar 2.12 merupakan *tree-tree* yang dibangun



Gambar 2. 11 Pembangunan *Tree-Tree* Awal *Random Forest Regression*



Gambar 2. 12 Pembangunan *Tree-Tree Random Forest Regression*

5) Pada tahap percabangan, terdapat kondisi di mana jika kedalaman pohon (max_depth) sudah memenuhi nilai yang telah ditentukan dan jumlah *tree* yang telah dibangun (t) sudah memenuhi banyaknya *tree* (k). Jika kondisi tersebut tidak terpenuhi, maka *tree* dibangun secara rekursif sampai memenuhi kondisi yang telah ditentukan.

Apabila kondisi sudah terpenuhi, maka langkah terakhir yaitu melakukan agregasi terhadap prediksi yang dihasilkan semua *tree* untuk mendapatkan nilai prediksi akhir. Dari ketiga *tree* yang telah dibangun, didapat bahwa hasil akhir prediksi yaitu Rp48.900

2.2.6 Matriks Evaluasi

Penelitian ini menilai performa dari masing-masing model regresi menggunakan tiga jenis pengukuran performa seperti *MAE*, *MAPE* dan *R2-Score.*

a) *Mean Absolute Error* (*MAE*)

MAE umumnya memiliki konsep untuk menghitung rata-rata dari perbedaan absolut antara nilai yang diprediksi dengan nilai sebenarnya. Dengan

kata lain, *MAE* mengukur rata-rata kesalahan yang bersifat mutlak dari prediksi tersebut. Semakin kecil nilai *MAE*, semakin baik kualitas model yang dibuat [18][20]. Nilai *MAE* dihitung dengan persamaan (2.11)

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|, \qquad (2.11)$$

Notasi :

- n = Banyaknya sampel
- y_i = Nilai aktual
- $\hat{y}_i$ = Nilai prediksi

b) *Mean Absolute Percentage Error* (*MAPE*)

MAPE (*Mean Absolute Percentage Error*) seringkali dipakai untuk mengukur kesalahan relatif antara nilai prediksi dari suatu model dengan nilai yang diamati. *MAPE* menggambarkan persentase kesalahan prediksi terhadap data aktual selama periode tertentu. Semakin kecil nilai *MAPE*, semakin baik kualitas dari model yang digunakan [18][20]. Nilai *MAPE* didapatkan dengan persamaan (2.12)

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{\hat{y}_i - y_i}{y_i} \right| x\ 100 \qquad (2.12)$$

Notasi :

- n = Banyaknya sampel
- y_i = Nilai aktual
- $\hat{y}_i$ = Nilai prediksi

Kriteria nilai *MAPE* ditunjukkan pada Tabel 2.9

Tabel 2. 9 Kriteria Nilai *MAPE*

Nilai *MAPE*	Kriteria
< 10%	Sangat baik
10% - 20%	Baik
20% - 50%	Cukup
>50%	Buruk

c) *Goodness-of-fit* (*R2-Score*)

R2-Score adalah *ukuran* mengenai seberapa dekat nilai prediksi dari suatu model cocok dengan nilai yang diamati. Nilai *R2* ideal suatu model adalah 1 yang

menunjukkan bahwa model tersebut dapat menjelaskan seluruh variabilitas pada kelas sasaran. Nilai *R2* dihitung dengan persamaan (2.13)

$$R^2 = 1 - \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|^2}{\sum_{i=1}^{n} |y_i - \bar{y}_i|^2}, \qquad (2.13)$$

Notasi :

- n = Banyaknya sampel
- y_i = Nilai aktual
- $\hat{y}_i$ = Nilai prediksi
- $\bar{y}_i$ = Rata-rata nilai prediksi

Ketika nilai prediksi mendekati nilai sebenarnya, maka *MAE* mendekati nol, sehingga *R2-Score* yang mendekati 1 menunjukkan kecocokan yang baik antara nilai hasil prediksi dan nilai sebenarnya [18][20].

2.2.7 *SHapley Additive explanation* (*SHAP*)

SHAP merupakan salah satu metode yang membantu dalam melakukan interpretasi dari hasil prediksi model pada *machine learning*. Tujuan dari *SHAP* adalah untuk menjelaskan prediksi dari sebuah *instance* x dengan menghitung kontribusi dari setiap fitur untuk prediksi. Nilai fitur dari *instance* data bertindak sebagai *players* dalam suatu koalisi. Nilai *Shapley* memberi tahu cara mendistribusikan "prediksi" secara adil di antara fitur. Metode ini membantu meningkatkan interpretabilitas model prediksi dan memberikan wawasan yang lebih dalam tentang bagaimana model membuat keputusan.

Konsep utama dari *SHAP* yaitu untuk menghitung nilai *Shapely* bagi setiap variabel pada dataset yang diinterpretasikan, sehingga nilai *Shapely* mewakili dampak yang dihasilkan dalam prediksi untuk masing-masing variabel. Persamaan (2.14) untuk menghitung kontribusi masing-masing variabel menggunakan nilai *Shapely*.

$$\varphi_j = \sum_{S \subseteq N\{j\}} \frac{|S|!(N-|S|-1)!}{N!} \big(v(S \cup \{j\} - v(S)\big), \qquad (2.14)$$

Notasi :

- S = Subset fitur yang digunakan dalam model
- $v(S)$ = Nilai fungsi karakteristik setiap prediktor
- N = Jumlah prediktor

Kontribusi dari setiap fitur didefinisikan juga sebagai nilai rata-rata dari marginal kontribusinya terhadap seluruh permutasi yang mungkin dari set fitur. Proses ini menggunakan perhitungan nilai Shapely dengan mengukur bagaimana kontribusi setiap fitur berubah saat fitur-fitur lain juga berubah [42].

Untuk contoh perhitungan nilai *Shapely*, Tabel 2.10 merupakan contoh data sample.

Tabel 2. 10 Data Sampel untuk Perhitungan Manual Nilai *Shapely*

X_1	X_2	Y
1	5	10
2	4	15
3	3	20
4	2	25
5	1	30

Pada contoh perhitungan ini akan melihat nilai *Shapely* untuk observasi pertama di mana $X_1 = 1$ dan $X_2 = 5$. Berikut merupakan langkah-langkah *SHAP* dalam menentukan nilai *Shapely* pada masing-masing fitur:

1) Asumsikan model telah dilatih pada data di atas, anggap saja model telah mempelajari hubungan antara X_1, X_2, dan Y.
2) Perlu diketahui prediksi model untuk kombinasi variabel X_1 dan X_2, hanya menggunakan X_1, hanya menggunakan X_2 dan tidak menggunakan kedua variabel tersebut (nilai dasar). Misalkan nilai prediksi untuk setiap kasus kombinasi tersebut adalah :

   $f(X_1, X_2) = 10$

   $f(X_1, -) = 12$

   $f(-, X_2) = 8$

   $f(-, -) = 5\ (nilai\ dasar, misal\ rata - rata\ y)$

3) Menghitung nilai *Shapely* dengan mempertimbangkan semua permutasi dari fitur dan melihat kontribusi tambahan dari setiap fitur ketika ditambahkan ke subset dari fitur lain. Ada 2 permutasi untuk kedua fitur pada data sampel, yaitu X_1, X_2 dan X_2, X_1

   a. Permutasi X_1, X_2

   $f(X_1, -) - f(-, -) = 12 - 5 = 7$

   $f(X_1, X_2) - f(X_1, -) = 10 - 12 = -2$

   b. Permutasi X_2, X_1

   $f(-, X_2) - f(-, -) = 8 - 5 = 3$

   $f(X_1, X_2) - f(-, X_2) = 10 - 8 = 2$

4) Setelah nilai permutasi didapat, maka langkah terakhir menghitung rata-rata dari kontribusi tambahan untuk setiap permutasi menggunakan persamaan (2.14)

$$\varphi_{X1} = \frac{7 + 2}{2} = 4{,}5$$

$$\varphi_{X2} = \frac{-2 + 3}{2} = 0{,}5$$

Berdasarkan perhitungan di atas, nilai *Shapley* menunjukkan bahwa X_1 memiliki kontribusi lebih besar yaitu 4.5 ke hasil prediksi dibandingkan dengan X_2 yang hanya memberikan kontribusi 0.5. Ini menunjukkan bahwa fitur X_1 lebih penting untuk model dalam kasus prediksi ini.