

BAB II TINJAUAN PUSTAKA

A. Penelitian Terkait

Berikut adalah beberapa penelitian terkait dapat dilihat pada tabel dibawah.

Tabel 1. Penelitian Terkait

No	Nama (Tahun)	Judul
(1)	(2)	(3)
1	R. R. Hallana, I. N. Fajri (2025)	Judul : Prediksi Harga Rumah Menggunakan Machine Learning Algoritma Regresi Linier Metode : Linear Regression Software : Python, Jupiter Notebook Hasil Penelitian : Hasil penelitian mendapatkan nilai R-squared (R^2) sebesar 0,81 yang menunjukkan model mampu menjelaskan 81% variasi harga rumah. Nilai RMSE sebesar 35.830,40 mengindikasikan kesalahan prediksi yang rendah dan konsisten pada data berbeda.
2	Evita Fitri (2023)	Judul : Analisis Perbandingan Metode Regresi Linier, Random Forest Regression dan Gradient Boosted Trees Regression Method untuk Prediksi Harga Rumah Metode : Regresi Linear, Random Forest & GBT Software : RapidMiner Hasil Penelitian : Hasil penelitian menunjukkan bahwa metode Random Forest Regression memberikan hasil prediksi terbaik dengan error rate terkecil (RMSE = 0.440) dan akurasi tertinggi sebesar 81,5%. Sementara itu, Linear Regression memiliki nilai RMSE sebesar 0.515, dan Gradient Boosted Trees Regression sebesar 0.508.
3	Dhiwa Aqsa (2025)	Judul : Perbandingan Kinerja Algoritma Extreme Gradient Boosting dan Random Forest untuk Prediksi Harga Rumah di Jabodetabek Metode : Extreme Gradient Boosting, Random Forest Software : Python Hasil Penelitian :

		Hasil penelitian menunjukkan bahwa Random Forest memberikan performa terbaik dengan MAE 95.200.513,25, MSE $1,47 \times 10^{19}$, dan R^2 0,77, melampaui Extreme Gradient Boosting yang memiliki MAE 121.836.703,27, MSE $3,03 \times 10^{19}$, dan R^2 0,52.
4	A. A. Handani, A. M. Siregar, T. A. Mudzakir (2024)	Judul : Prediksi Harga Rumah di Karawang Menggunakan Algoritma Multiple Linear Regression dan Support Vector Regression Metode : Linear Regression, SVR Software : Python Hasil Penelitian : Hasil penelitian menunjukkan MLR memiliki akurasi 0,5439 dan RMSE 148,3586, sedangkan SVR memiliki akurasi 0,5104 dan RMSE 153,7563, sehingga MLR menghasilkan prediksi yang lebih baik.
5	F. Hidayah, S. J. Angesti, Y. P. Widyastuti (2024)	Judul : Prediksi Harga Rumah di Boston Menggunakan Metode Linear Regression, SVR, Decision Tree dan Random Forest Regression Metode : MLP, SVR, Decision Tree, RFR Software : Python, R studio Hasil Penelitian : Hasil penelitian menunjukkan bahwa Random Forest Regression memberikan performa terbaik dengan RMSE sebesar 3.204 dan R^2 sebesar 0.863, yang berarti mampu 5 menjelaskan 86,3% variasi dalam data pengujian. SVR menyusul dengan RMSE 3.838 dan R^2 0.804, sementara Linear Regression mencatat RMSE 4.647 dan R^2 0.713. Decision Tree mengalami overfitting dengan R^2 training 1.000 namun hanya 0.598 pada data pengujian, serta RMSE tertinggi sebesar 5.496.
6	Hafizh, M. A., Subairi, Libriawan, R. D., Maulana, N. D., Rizki, A. M. (2024)	Judul : Prediksi Harga Rumah di Jabodetabek Menggunakan Metode Artificial Neural Network Metode : Machine Learning Neural Network Software : Python Hasil Penelitian : Penelitian menggunakan metode Artificial Neural Network (ANN) dalam memprediksi harga rumah di Jabodetabek terbukti baik. Model terbaik diperoleh

		melalui konfigurasi hyperparameter yang optimal, menghasilkan MAE sebesar 0.209, MSE 0.159, dan MAPE 4.951%. Hasil ini menunjukkan ANN efektif untuk prediksi harga rumah jika data dan konfigurasi tepat.
7	N. P. Ariyanti, A.Triayudi, R. T. K. Sari (2024)	Judul : Analysis of K-NN Algorithm and Linear Regression to Predict House Prices in Jabodetabek Metode : K-NN, Linear Regression Software : Python Hasil Penelitian : Hasil penelitian menunjukkan bahwa algoritma Multiple Linear Regression (MLR) memiliki kinerja lebih baik dibandingkan K-Nearest Neighbor (KNN) dalam memprediksi harga rumah di wilayah Jabodetabek. Model MLR menghasilkan nilai MAE dan MSE yang lebih rendah serta R^2 sebesar 48,72%, menunjukkan kemampuan yang cukup baik dalam menjelaskan variasi data.
8	B. W. Sari and D. Prabowo (2025)	Judul : Analisis Perbandingan Prediksi Harga Rumah dengan Random Forest, Gradient Boosting dan XGBoost Metode : Random Forest, GBT, XGBoost Software : Python Hasil Penelitian : Hasil penelitian menunjukkan bahwa XGBoost memiliki performa terbaik dengan MSE $1,56 \times 10^{14}$, R^2 0,7746, dan RMSE sekitar 12,5 juta rupiah. Model ini terbukti paling akurat dalam menangani data kompleks dan berpotensi digunakan sebagai alat bantu penetapan harga serta analisis pasar properti.
9	K. D. Sanjaya (2025)	Judul : Prediksi Harga Rumah dengan Metode Regresi Linear dan Support Vector Regression di Daerah Tebet Metode : Regresi Linear, SVR Software : Python Hasil Penelitian : Hasil penelitian menunjukkan Regresi Linear memiliki performa lebih baik dengan nilai MAE, MSE, dan RMSE lebih rendah serta R^2 lebih mendekati 1 dibandingkan SVR. Dengan demikian, Regresi Linear dinilai lebih sesuai untuk dataset yang digunakan.

10	A. Fauzi, N. Maulidah, R. Supriyadi, H. Nalatissifa, S. Diantika (2025)	Judul : Prediksi Harga Properti di Indonesia Menggunakan Algoritma Random Forest Metode : Random Forest Software : Python Hasil Penelitian: Penelitian ini melakukan prediksi harga properti menggunakan algoritma Random Forest dengan memanfaatkan variabel seperti luas tanah, luas bangunan, jumlah kamar, jarak ke pusat kota, serta ketersediaan fasilitas di sekitarnya. Model yang dibangun menghasilkan nilai MAE sebesar Rp2,48 miliar dan RMSE sebesar Rp2,89 miliar, menandakan bahwa tingkat akurasi cukup baik.
11	U. Lathifah, R. D. Dana (2024)	Judul : Implementasi Metode Linear Regression untuk Prediksi Harga Properti Real Estate Menggunakan Rapidminer Metode : Linear Regression Software : Rapidminer Hasil Penelitian : Penelitian ini berhasil menerapkan metode Linear Regresi menggunakan Rapidminer untuk memprediksi harga real estate tahun 2021 dengan 414 data. Evaluasi model menunjukkan nilai RMSE sebesar 0,116 dan MAE sebesar 0,088, yang mengindikasikan tingkat kesalahan prediksi yang rendah dan model yang cukup baik.
12	Hartono, P. C., Widianoro, A. (2024)	Judul : Analisis Prediksi Harga Saham Unilever Menggunakan Regresi Linier dengan Rapidminer Metode : Linear Regression Software : Rapidminer Hasil Penelitian : Penelitian ini menunjukkan bahwa metode Linear Regression memiliki performa kurang baik, dengan mendapatkan nilai Root Mean Squared Error (RMSE) sebesar 57,072. Hasil ini mengindikasikan bahwa akurasi prediksi masih rendah.

Penelitian terkait juga dapat dilihat pada gambar mindmap berikut.



Gambar 1 Mindmap

B. Landasan Teori

1. Data Mining

Data mining adalah proses menemukan pola, hubungan dan informasi yang berguna dari kumpulan data besar yang dapat digunakan untuk pengambilan keputusan yang lebih baik. Tujuan utama dari *data mining* adalah untuk mengidentifikasi pola tersembunyi atau keterkaitan antar variabel yang dapat dimanfaatkan dalam perencanaan dan pengambilan keputusan dimasa mendatang [9]. Data mining memiliki kemampuan untuk memproses data dalam skala besar dan dengan tingkat kompleksitas yang tinggi. Selain itu, data mining juga dapat bekerja dengan berbagai jenis atribut data. Proses pencarian dan pengolahan data dapat dilakukan secara otomatis oleh *data mining*.

Proses *data mining* mencakup pengumpulan dan penggunaan data historis untuk menemukan keteraturan, pola, atau hubungan dalam dataset yang kompleks dan besar [10]. Dengan kata lain, *data mining* merupakan suatu pendekatan sistematis untuk menggali informasi yang tersembunyi dari data

agar dapat dimanfaatkan secara optimal. *Data mining* tidak hanya berfokus pada ekstraksi informasi, tetapi juga berperan penting dalam membangun model prediktif yang mampu memperkirakan nilai atau perilaku dimasa depan berdasarkan data yang tersedia [11]. Hal ini menjadikan *data mining* sebagai alat analisis yang sangat berdaya guna dalam berbagai bidang.

Tahapan dalam proses data mining mencakup beberapa langkah utama yang bertujuan untuk menghasilkan informasi yang akurat dan berguna. Berikut adalah beberapa tahapan dari proses *data mining*:

a. Data cleaning

Data cleaning atau pembersihan data adalah proses untuk membersihkan data dari kesalahan, data duplikat dan nilai yang hilang sehingga data menjadi lebih akurat dan konsisten.

b. Data integration

Data integration merupakan proses menggabungkan data dari berbagai sumber sehingga dapat dianalisis secara menyeluruh.

c. Data selection

Data selection bertujuan untuk memilih bagian-bagian data yang relevan dengan tujuan analisis, karena tidak semua variabel data akan digunakan dalam proses *data mining*.

d. Transformasi data

Transformasi data bertujuan untuk mengubah struktur data agar sesuai dengan format yang dibutuhkan untuk proses analisis, seperti melalui normalisasi atau konversi data.

e. Data mining

Data mining adalah proses utama untuk mengekstrak pola atau informasi tersembunyi dalam data menggunakan algoritma tertentu.

f. Evaluasi

Setelah pola data ditemukan maka dilakukan evaluasi pola untuk menilai dan memastikan bahwa pola tersebut valid, berguna, dan sesuai dengan kebutuhan. Kemudian, hasil dari proses tersebut disajikan dalam tahap penyajian pengetahuan atau *knowledge presentation* agar informasi yang diperoleh dapat dipahami dan digunakan oleh pengguna dalam pengambilan keputusan [7].

Penerapan *data mining* sangat beragam, mencakup bidang bisnis, keuangan, teknologi, ilmu pengetahuan, kesehatan, hingga sektor pemerintahan. Teknologi ini berperan dalam membantu organisasi memahami perilaku konsumen, mengenali pola pasar, meningkatkan kinerja operasional, serta mendukung pengambilan keputusan strategis secara lebih efektif. Dalam penelitian ini, *data mining* digunakan untuk menganalisis dan memprediksi harga rumah berdasarkan berbagai faktor seperti lokasi, luas tanah, jumlah kamar, dan kondisi lingkungan [12]. Selain itu, *data mining* juga membantu dalam

memahami tren pasar, menilai risiko investasi, serta mendukung pengambilan keputusan yang lebih rasional dan berbasis data. Model prediksi yang dibangun dengan pendekatan *data mining* mampu mengurangi ketidakpastian dalam penetapan harga rumah, memberikan pengetahuan tentang harga pasar, dan menjadi acuan penting bagi para pelaku di sektor properti [13].

2. Algoritma Decision Tree

Algoritma decision tree merupakan salah satu teknik dalam *data mining* yang banyak digunakan untuk menyelesaikan masalah klasifikasi dan prediksi. *Decision tree* merupakan metode non-parametrik yang bekerja dengan membagi data menjadi beberapa bagian berdasarkan aturan tertentu dan membuat prediksi sederhana disetiap bagian. Metode ini membantu dalam memodelkan hubungan yang rumit dan interaksi antara variabel yang berbeda [14]. *Algoritma decision tree* memiliki struktur yang menyerupai flowchart dan menerapkan pendekatan pembelajaran mesin terawasi, dimana data baru dikelompokkan berdasarkan sampel pelatihan yang sudah ada.

Decision Tree merupakan salah satu metode prediksi dan klasifikasi yang menggunakan struktur pohon sebagai representasi proses pengambilan keputusan. Setiap *node* dalam pohon menggambarkan atribut, cabang menunjukkan nilai dari atribut tersebut, dan daun (*leaf node*) merepresentasikan kelas dari data. *Node* akar berada pada posisi paling atas dan berfungsi sebagai titik awal pembagian data. Setiap *node* berperan sebagai pemisah yang menerima satu masukan dan menghasilkan minimal dua keluaran. *Leaf node* merupakan *node* terminal yang hanya memiliki satu masukan tanpa keluaran, dan berfungsi untuk menunjukkan label kelas. Setiap cabang pada pohon mencerminkan kondisi atau aturan yang harus dipenuhi, sedangkan puncak pohon secara keseluruhan menggambarkan hasil klasifikasi berupa nilai kelas dari data yang dianalisis [15].

Decision tree memiliki daya tarik karena fleksibilitasnya dalam menyajikan visualisasi yang jelas, sehingga hasil prediksi dapat dengan mudah dipahami. Namun, kualitas keputusan yang dihasilkan sangat bergantung pada desain dan kompleksitas model yang diterapkan [6]. Salah satu kelebihan utama *decision tree* adalah kemampuannya dalam mengolah data baik numerik maupun kategorikal serta sifatnya yang mudah dipahami. Dengan kata lain, *decision tree* memudahkan pengguna dalam memahami proses analisis yang dilakukan karena dapat secara cepat menunjukkan cara suatu keputusan atau prediksi dihasilkan, sehingga proses pengambilan keputusan menjadi lebih transparan [16].

Dalam penerapannya pada prediksi harga rumah, *algoritma decision tree* digunakan untuk membentuk model yang menggambarkan hubungan antara karakteristik properti dan harga jual. Beberapa fitur umum yang digunakan antara lain adalah harga rumah, luas tanah, luas bangunan, jumlah kamar, lokasi,

serta fasilitas yang tersedia. Dengan memanfaatkan dataset yang representatif, model *decision tree* dapat dilatih untuk menghasilkan prediksi harga berdasarkan kombinasi dari fitur-fitur tersebut [16]. Dengan menggunakan model *decision tree*, kita dapat melakukan prediksi atau klasifikasi terhadap data baru dengan mengikuti jalur pada pohon keputusan, dimulai dari akar pohon hingga mencapai simpul yang mempresentasikan hasil prediksi atau klasifikasi [17].

3. Algoritma Neural Network

Algoritma neural network merupakan algoritma yang dikembangkan berdasarkan prinsip kerja *neuron* pada otak manusia, terdiri dari sejumlah unit pemrosesan (*neuron*) yang saling terhubung dan bekerja secara bersama-sama untuk menyelesaikan berbagai permasalahan, seperti tugas klasifikasi maupun prediksi. [4]. *Algoritma neural network* merupakan model komputasi yang tersusun atas sejumlah unit pemrosesan yang disebut *neuron* buatan, yang meniru fungsi neuron biologis pada otak manusia. Neuron buatan ini saling terhubung membentuk suatu jaringan, dimana setiap neuron menerima input, mengolahnya melalui proses perhitungan, kemudian menghasilkan output. [18].

Neural network merupakan metode yang fleksibel dan berbasis pembelajaran dengan kemampuan memproses data dengan perkiraan fungsi terstruktur. Hal ini yang membuat *neural network* sesuai diaplikasikan untuk prediksi. Struktur *neural network* terdiri dari beberapa lapisan, termasuk *input layer* yang menerima data masukan, *hidden layers* yang bertugas memproses data, dan *output layer* yang menghasilkan prediksi akhir, dalam konteks ini, harga rumah. Cara kerja *neural network* mirip dengan *Multi-Layer Perceptron (MLP)*, namun dalam *neural network*, setiap neuron disajikan dalam format dua dimensi, berbeda dengan *MLP* yang hanya memiliki ukuran satu dimensi untuk setiap neuronnya [19].

Neural Network merupakan salah satu pendekatan dalam bidang kecerdasan buatan yang mengadopsi prinsip kerja otak manusia dalam mengolah informasi. Secara umum, proses pemodelan ini terdiri atas dua tahap utama, yaitu tahap training dan testing. Pada tahap training, sistem melakukan proses pembelajaran dengan mengenali pola hubungan antara data masukan dan keluaran melalui mekanisme penyesuaian bobot (*weight*) dan bias. Tahap ini bertujuan untuk membentuk suatu model yang mampu menghasilkan prediksi atau klasifikasi secara akurat. Setelah model terbentuk, dilakukan tahap testing untuk mengukur tingkat akurasi dan kehandalan model terhadap data yang tidak digunakan selama pelatihan. Dalam perspektif statistik, proses training dapat dianggap setara dengan proses estimasi, sedangkan testing memiliki kesamaan fungsi dengan *cross validation*, yaitu mengevaluasi kinerja model secara keseluruhan [20]. Dalam mengevaluasi model, metrik seperti *Mean Absolute Error (MAE)*,

Root Mean Squared Error (RMSE), dan *Mean Squared Error (MSE)* digunakan untuk menilai seberapa baik model dalam memprediksi harga rumah [21].

Kelebihan *neural network* meliputi kemampuannya untuk mengelola data besar dan kompleks, serta fleksibilitas dalam menangkap pola nonlinear. Secara keseluruhan, *algoritma neural network* merupakan alat yang sangat efektif dalam memprediksi harga rumah, mampu memberikan akurasi tinggi dengan memanfaatkan data yang kompleks. Dengan pengembangan lebih lanjut, seperti optimasi hyperparameter dan penerapan teknik regularisasi, model prediksi *neural network* dapat digunakan untuk melakukan prediksi harga rumah dengan hasil yang cukup akurat, jika data dan konfigurasi yang digunakan relevan dan optimal [21].

4. Algoritma K-Nearest Neighbor

K-Nearest Neighbors (KNN) adalah algoritma pembelajaran berbasis instance yang banyak diterapkan pada permasalahan prediksi, termasuk dalam estimasi harga rumah. Metode *KNN* merupakan salah satu pendekatan yang sederhana dan efektif yang bekerja dengan mencari sejumlah K tetangga terdekat dari data uji berdasarkan data latih. Algoritma ini dapat digunakan pada dataset kecil maupun besar, meskipun sensitif terhadap data latih yang mengandung noise [22].

Konsep utama *KNN* yaitu menentukan sejumlah tetangga terdekat dari suatu data uji dengan mengukur jarak menggunakan metode tertentu, misalnya Euclidean atau Manhattan, kemudian menghasilkan prediksi berdasarkan rata-rata atau bobot nilai dari tetangga tersebut. Dalam penerapannya pada prediksi harga rumah, setiap properti digambarkan melalui atribut seperti luas tanah, luas bangunan, jumlah kamar, lokasi, maupun aksesibilitas, sehingga harga rumah baru dapat ditaksir dari data rumah yang memiliki kemiripan karakteristik. *KNN* bekerja dengan membandingkan setiap data baru dengan sejumlah data K terdekat dalam data pelatihan [23].

Dalam penggunaannya, parameter utama yang memengaruhi performa *KNN* adalah nilai k . Pemilihan k yang terlalu kecil berpotensi membuat model sangat peka terhadap data dengan *noise* sehingga rawan *overfitting*, sedangkan k yang terlalu besar dapat menyebabkan *underfitting* karena hasil prediksi menjadi terlalu rata. Mekanisme kerja algoritma *KNN* diawali dengan penentuan training dataset, testing dataset, serta nilai K yang digunakan. Setelah itu, data latih disusun berdasarkan perhitungan jarak terdekat terhadap data uji. Pada tahap akhir, nilai prediksi ditentukan dengan menghitung rata-rata terkecil dari sejumlah tetangga sesuai nilai K , yang kemudian digunakan untuk menentukan kategori regresi [22]. Selain itu, tahap pra-pemrosesan data seperti normalisasi atau standardisasi sangat diperlukan karena *KNN* sensitif terhadap skala fitur.

Kelebihan *KNN* adalah kesederhanaan serta kemudahan interpretasi, sedangkan kelemahannya meliputi sensitivitas terhadap pemilihan parameter *K*, perbedaan skala antar variabel dan potensi masalah *curse of dimensionality* saat jumlah fitur terlalu banyak. Oleh sebab itu, untuk memperoleh hasil yang optimal, pemilihan nilai *K* serta seleksi fitur yang relevan sangat diperlukan, karena fitur yang tidak relevan dapat menurunkan kinerja model, sedangkan pemilihan fitur yang tepat akan meningkatkan efisiensi dan ketepatan prediksi.

5. Algoritma Gradient Boosted Trees

Gradient Boosted Tree merupakan algoritma regresi yang bersifat fleksibel karena menggabungkan keunggulan *regression tree* yang mudah diinterpretasikan, memiliki konsep sederhana, serta efisien secara komputasi. Melalui penerapan pendekatan boosting, metode ini mampu meningkatkan akurasi model yang dihasilkan [24]. Metode *Gradient Boosted Trees* merupakan pendekatan yang mengombinasikan sejumlah *decision tree*. Teknik ini juga dapat dipandang sebagai bentuk modifikasi dari algoritma *gradient descent* yang dipadukan dengan *boosting algorithm* untuk meningkatkan tingkat akurasi [11].

Prinsip utama *GBT* adalah membentuk model secara berurutan, di mana setiap pohon yang ditambahkan bertugas memperbaiki kesalahan prediksi atau residual error dari pohon sebelumnya. Tahapan kerja *GBT* diawali dengan pembangunan model dasar berupa pohon keputusan sederhana menggunakan data pelatihan. Setelah itu, algoritma menghitung selisih antara nilai aktual dengan prediksi model, kemudian menjadikan selisih tersebut sebagai target baru untuk melatih pohon berikutnya. Proses ini dilakukan secara berulang hingga jumlah iterasi tertentu tercapai atau hingga model mencapai titik konvergensi. *GBT* beroperasi secara bertahap dengan membangun pohon keputusan baru yang berfungsi memperbaiki kesalahan dari model sebelumnya. Pendekatan boosting ini umumnya mampu menghasilkan prediksi yang lebih akurat dibandingkan metode *bagging*, meskipun memiliki kerentanan lebih tinggi terhadap *overfitting* [25].

Metode *Gradient Boosted Tree* memiliki sejumlah keunggulan, khususnya dalam menyelesaikan permasalahan regresi yang bersifat kompleks. Algoritma ini dikenal mampu menghasilkan prediksi dengan tingkat akurasi tinggi dan seringkali menunjukkan performa yang lebih baik dibandingkan model regresi lainnya. Selain itu, metode ini cukup tangguh terhadap keberadaan *outlier* serta efektif dalam memodelkan hubungan nonlinear antara variabel fitur dan variabel target, sehingga sangat sesuai digunakan pada data yang kompleks [24]. Kelebihan utama *GBT* adalah kemampuannya dalam menangkap hubungan nonlinear antar variabel dengan tingkat akurasi tinggi, serta kemampuan melakukan seleksi fitur secara otomatis, sehingga tetap efektif meskipun diterapkan pada data berdimensi besar. Walaupun demikian,

algoritma ini memiliki keterbatasan, seperti kebutuhan komputasi yang lebih tinggi akibat proses pelatihan yang bersifat sekuensial. Selain itu, *GBT* rentan mengalami overfitting apabila jumlah pohon terlalu banyak atau parameter tidak diatur dengan tepat. Oleh karena itu, pemilihan parameter yang optimal menjadi aspek krusial agar model tidak hanya memiliki akurasi tinggi, tetapi juga mampu melakukan generalisasi dengan baik.

6. Algoritma Support Vector Regression

Algoritma *Support Vector Regression (SVR)* merupakan pengembangan dari *Support Vector Machine (SVM)* yang dirancang khusus untuk menyelesaikan masalah regresi [26]. Berbeda dengan *SVM* yang berfokus pada klasifikasi, *SVR* digunakan untuk memprediksi nilai kontinu dengan membangun fungsi regresi yang merepresentasikan hubungan antara variabel input dan output. *SVR* memiliki fleksibilitas dalam menentukan batas kesalahan yang masih dapat ditoleransi pada model, sehingga mampu menghasilkan *hyperplane* yang sesuai dengan pola data [27].

Kinerja *SVR* sangat dipengaruhi oleh pemilihan kernel dan parameter utama. *Kernel* digunakan untuk mentransformasikan data nonlinear ke ruang berdimensi lebih tinggi agar hubungan dapat dimodelkan secara linear. Kernel yang sering dipakai antara lain *linear*, *polynomial*, dan *radial basis function (RBF)*. Selain itu, parameter seperti *epsilon* (ϵ), *C* (*regularisasi*), dan *gamma* (γ) pada kernel RBF berperan penting dalam menentukan akurasi prediksi dan kompleksitas model. *Kernel Radial Basis Function (RBF)* merupakan salah satu jenis kernel yang paling umum digunakan dalam penerapan *SVR*. Kernel ini memiliki sifat yang fleksibel serta mampu memodelkan data dengan pola distribusi nonlinear, sehingga sering menghasilkan prediksi dengan tingkat akurasi yang tinggi. Atas dasar keunggulan tersebut, penerapan kernel *RBF* pada model *SVR* dianggap sebagai pilihan yang tepat untuk memperoleh hasil prediksi yang lebih optimal [28].

SVR memiliki sejumlah kelebihan, seperti mampu menangani data berdimensi tinggi, lebih tahan terhadap overfitting, dan fleksibel dalam memodelkan hubungan linear maupun nonlinear. Namun, metode ini juga memiliki keterbatasan, yaitu membutuhkan waktu komputasi yang lebih besar dan sensitif terhadap pemilihan parameter. Algoritma ini digunakan untuk menemukan fungsi regresi yang mampu merepresentasikan hubungan antara variabel input dengan output secara optimal [29].

7. Algoritma Random Forest

Algoritma *Random Forest* merupakan algoritma *machine learning* berbasis ensemble yang banyak diterapkan dalam masalah klasifikasi maupun regresi, termasuk prediksi harga rumah atau data numerik lainnya. Algoritma ini memanfaatkan banyak pohon keputusan (*decision trees*) yang dibangun secara

acak, sehingga model yang dihasilkan lebih stabil, akurat, dan mampu mengurangi risiko overfitting.

Prinsip kerja *Random Forest* adalah membentuk sejumlah pohon keputusan menggunakan teknik bootstrap aggregating (*bagging*). Setiap pohon dibuat dari subset data latih yang diambil secara acak dengan pengembalian (*bootstrap sampling*), dan pada tiap node hanya sebagian fitur yang dipilih secara acak untuk menentukan pemisahan terbaik [30]. Pendekatan ini membuat setiap pohon dalam hutan menjadi unik, mengurangi korelasi antar pohon, dan meningkatkan kemampuan generalisasi model.

Dalam proses prediksi, *Random Forest* menggabungkan output dari seluruh pohon. Pada klasifikasi, prediksi akhir ditentukan melalui *majority voting*, sedangkan pada regresi, hasil prediksi diperoleh dari rata-rata prediksi seluruh pohon. Dengan metode ini, *Random Forest* dapat memberikan prediksi yang lebih akurat dibanding pohon keputusan tunggal yang mudah dipengaruhi perubahan kecil pada data.

Random Forest memiliki kemampuan untuk mengatasi overfitting, suatu permasalahan yang umum pada pohon keputusan tunggal. Overfitting terjadi ketika model terlalu menyesuaikan diri dengan data pelatihan sehingga kehilangan kemampuan untuk menggeneralisasi data baru. Dengan menggabungkan prediksi dari berbagai pohon keputusan secara ensemble, *Random Forest* dapat meminimalkan risiko overfitting, menghasilkan prediksi yang lebih stabil, dan meningkatkan ketahanan terhadap variasi data [23].

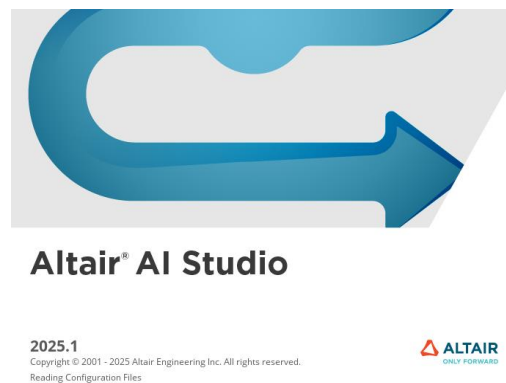
8. Altair Studio

Altair Studio adalah aplikasi open source yang dirancang untuk mempelajari *data mining*, termasuk analisis dan pemrosesan data serta pengembangan model prediktif. Altair Studio adalah perangkat lunak yang dikembangkan oleh Dr. Markus Hofmann dari Institute of Technology Blanchardstown dan Ralf Klinkenberg dari rapid-i.com. Perangkat lunak ini dirancang dengan antarmuka grafis (*Graphical User Interface/GUI*) sehingga memudahkan pengguna dalam mengoperasikannya[31]. Altair Studio sendiri sebelumnya bernama RapidMiner, kemudian pada tahun 2022 RapidMiner diakuisisi oleh perusahaan teknologi global bernama Altair Engineering Inc yang bergerak di bidang *ai* dan *data science*. Setahun setelah akuisisi pada tahun 2023, perusahaan ini melakukan rebranding RapidMiner menjadi Altair Studio dengan tetap mempertahankan fungsionalitas inti serta peningkatan integrasi dan performa.

Altair Studio menerapkan berbagai metode yang cerdas dan informatif untuk memberikan wawasan kepada *user*, sehingga mereka dapat membuat pilihan yang paling tepat. Selain itu, Altair Studio memiliki sekitar 500 operator untuk penambahan informasi, yang mencakup administrator untuk input, output, pra-pemrosesan data, dan representasi [32]. Altair Studio adalah sebuah

perangkat lunak yang berfungsi untuk mengolah dan menganalisis data. *Software* ini bekerja dengan memanfaatkan algoritma *machine learning* dan *data mining* guna menemukan pola dari kumpulan data berukuran besar [33]. Dalam prosesnya, Altair Studio mengintegrasikan berbagai pendekatan analisis, termasuk metode statistik, teknologi kecerdasan buatan (*ai*), serta manajemen basis data. Keunggulan utama dari software ini terletak pada antarmuka yang intuitif dan mudah digunakan. Proses pengembangan model tidak memerlukan pemrograman, namun dilakukan melalui pemanfaatan operator yang dapat disusun secara visual. Operator tersebut berperan dalam transformasi, pengolahan, hingga evaluasi data. Setiap operator dihubungkan melalui node, yang kemudian disambungkan ke node output untuk menampilkan hasil dari analisis yang dilakukan.

Altair Studio dirancang khusus untuk aplikasi *data mining* dan menawarkan berbagai model yang lengkap, termasuk Model *Regression*, *Bayesian*, *Modelling*, *Tree Induction*, *Neural Network*, dan lainnya. Terdapat banyak metode yang tersedia dalam *Altair Studio*, seperti *klasifikasi*, *klustering*, dan *prediction* yang memungkinkan pengguna untuk menerapkan teknik yang sesuai dengan kebutuhan analisis mereka [31].



Gambar 2 Altair Studio

Antarmuka Altair Studio yang intuitif memudahkan pengguna dalam mengoperasikannya, di mana saat aplikasi dijalankan, pengguna akan disambut dengan tampilan *welcome perspective*, diikuti oleh *design perspective* yang menunjukkan area kerja, dan *result perspective* yang menampilkan hasil analisis [34]. Altair Studio juga mendukung sekitar 22 format file, seperti *.xls*, *.csv*, dan sebagainya. Altair Studio adalah solusi untuk melakukan analisis data mining, text mining, dan analisis prediktif [35]. *Tools* ini mendukung teknik deskriptif dan prediktif secara luas.

Sebagai perangkat lunak independen untuk analisis data dan *data mining*, Altair Studio dapat dengan mudah diintegrasikan dengan berbagai bahasa pemrograman. Ditulis dalam bahasa pemrograman *Java*, Altair Studio juga

dapat beroperasi diberbagai sistem operasi [36]. Dalam konteks prediksi harga rumah, Altair Studio digunakan untuk membangun model prediktif dengan memanfaatkan variabel-variabel penting seperti lokasi, luas bangunan, jumlah kamar, tipe rumah, dan harga rumah awal. Kemampuannya dalam memodelkan relasi kompleks antar variabel membuat Altair Studio sangat cocok digunakan dalam permasalahan prediksi harga rumah.

9. Machine Learning

Machine learning adalah salah satu cabang dari *Artificial Intelligence* yang memungkinkan komputer atau mesin untuk mempelajari informasi dari data yang tersedia tanpa perlu diprogram secara eksplisit untuk setiap tugas tertentu [37]. *Machine learning* dapat berfungsi ketika tersedia data yang dianalisis dari kumpulan data berukuran besar (*Big Data*) untuk mengidentifikasi pola-pola tertentu [38]. Dalam penerapannya, *machine learning* memanfaatkan data pelatihan agar komputer mampu mengenali pola dan hubungan dalam data tersebut. Model yang dihasilkan dari proses ini kemudian digunakan untuk memetakan data input menjadi output atau perintah yang diharapkan. Dengan demikian, *machine learning* mampu memberikan prediksi dan insight yang bermanfaat untuk membantu proses pengambilan keputusan yang didasarkan pada data yang ada.

Machine Learning dapat dipandang sebagai proses pemrograman komputer yang memanfaatkan data historis untuk membangun suatu model, yang kemudian digunakan untuk mempelajari pola guna mencapai kinerja optimal dalam mengekstraksi informasi dari sekumpulan data [39]. *Machine learning* terbagi ke dalam beberapa jenis, yaitu *supervised learning*, *unsupervised learning*, *semisupervised learning*, dan *reinforcement learning*. Dalam pendekatan *supervised learning*, sistem belajar dari data yang telah diberi label, sedangkan *unsupervised learning* menganalisis data tanpa label untuk menemukan pola tersembunyi. *Semisupervised learning* memadukan keduanya, dan *reinforcement learning* melibatkan agen yang belajar dari umpan balik lingkungan untuk memperoleh hasil maksimal [40].

Proses dalam *machine learning* secara umum mencakup beberapa tahap penting, dimulai dari pengumpulan data yang relevan, dilanjutkan dengan pra-pemrosesan untuk membersihkan dan menyiapkan data, serta pembagian data ke dalam set pelatihan dan pengujian. Kemudian, model dipilih dan dilatih menggunakan data pelatihan. Setelah itu, performa model dievaluasi melalui data pengujian, dan dilakukan penyetelan (*tuning*) terhadap hyperparameter untuk meningkatkan kinerjanya. Setelah berhasil dilatih, model dapat diterapkan pada sistem atau aplikasi nyata. Beberapa algoritma yang sering digunakan dalam machine learning antara lain *regresi linier*, *decision tree*, *neural network*, *gradient boosted*, *support vector regression*, *random forest* serta *K-nearest neighbor*.

Penerapan *machine learning* sangat luas, mencakup sektor kesehatan, properti, pemasaran, transportasi, hingga teknologi informasi. Meskipun teknologi ini memiliki potensi besar, ada berbagai tantangan yang harus dihadapi, seperti risiko *overfitting* dan *underfitting*, kualitas data yang kurang baik, serta masalah bias dan etika dalam proses pengambilan keputusan. Dengan pemahaman mendalam tentang teori dan praktik *machine learning*, para profesional dapat menciptakan solusi inovatif yang mampu mengatasi persoalan di dunia nyata secara lebih efektif karena teknologi *Machine Learning* dapat memberikan solusi yang efektif untuk menganalisis data sentimen yang jumlahnya sangat besar dan kompleks [41]. Inti dari *Machine Learning* adalah menciptakan model yang dapat merepresentasikan pola-pola berdasarkan dataset yang digunakan.

10. RMSE

Root Mean Square Error (RMSE) merupakan salah satu metrik evaluasi yang banyak digunakan dalam statistik dan pembelajaran mesin untuk menilai akurasi suatu model prediktif. Metrik ini memberikan estimasi rata-rata dari kesalahan absolut antara prediksi dan nilai sebenarnya. *RMSE* menghitung rata-rata dari kuadrat selisih antara nilai aktual dengan nilai prediksi, lalu mengambil akar kuadrat dari hasil tersebut. Nilai *RMSE* yang rendah menunjukkan bahwa hasil prediksi mendekati nilai aktual, sementara nilai yang tinggi menandakan adanya kesalahan prediksi yang cukup besar. Metrik ini memberikan bobot yang lebih besar terhadap kesalahan besar karena adanya proses kuadrat dalam perhitungannya, sehingga lebih sensitif terhadap outlier. *RMSE* memberikan bobot yang lebih besar terhadap kesalahan yang signifikan, sehingga sangat efektif untuk menilai performa model dalam mendeteksi outlier atau anomali data [42]. Secara matematis, *RMSE* dirumuskan sebagai berikut:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

Keterangan :

y_i = nilai aktual ke- i

$\hat{y}_i$ = nilai prediksi ke- i

n = jumlah data yang diuji

Rumus ini digunakan dalam berbagai penelitian untuk mengevaluasi kinerja model prediktif dalam merepresentasikan data sesungguhnya. Interpretasi nilai *RMSE* perlu mempertimbangkan konteks dan skala data. Nilai *RMSE* yang

rendah mengindikasikan tingkat kesalahan prediksi yang kecil, namun karena sifatnya yang sensitif terhadap nilai ekstrem, *RMSE* dapat memberikan gambaran yang kurang representatif apabila data mengandung banyak outlier. Oleh karena itu, banyak peneliti menyarankan penggunaan *RMSE* bersama metrik lain seperti *Mean Absolute Error (MAE)*, yang menghitung rata-rata nilai absolut dari selisih antara nilai aktual dan prediksi tanpa memperbesar dampak dari error besar. Kombinasi *RMSE* dan *MAE* dapat memberikan gambaran yang lebih komprehensif terhadap akurasi model [42].

RMSE merupakan metrik evaluasi yang penting dalam pengukuran performa model prediktif, terutama karena kemampuannya menekankan kesalahan besar. Namun, penggunaannya perlu disertai dengan pemahaman terhadap karakteristik data serta pelengkap dari metrik lain agar hasil evaluasi menjadi lebih akurat dan menyeluruh. Untuk menentukan nilai *RMSE* dapat dilihat berdasarkan hasil prediksi yang didapatkan, jika hasil yang didapatkan mendekati angka 0 maka hasil prediksi atau peramalan mendekati atau sesuai dengan data aktualnya [43].

11. MAE

Mean Absolute Error (MAE) merupakan salah satu metrik yang digunakan untuk mengukur tingkat kesalahan dalam prediksi model. *MAE* menghitung rata-rata nilai absolut selisih antara nilai aktual dan nilai prediksi sehingga memberikan gambaran langsung mengenai kesalahan rata-rata yang terjadi dalam satuan yang sama dengan data asli. *MAE* digunakan untuk menghitung rata-rata nilai absolut dari perbedaan antara data aktual dan hasil prediksi, sekaligus memberikan indikasi tentang besarnya kesalahan prediksi model dibandingkan dengan nilai sebenarnya [42]. *MAE* secara umum digunakan untuk pengukuran prediksi eror pada analisis time series sama seperti halnya *RMSE* namun dengan rumus perhitungan yang sederhana. Rumus perhitungan *MAE* secara matematis sebagai berikut:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Keterangan:

y_i = nilai aktual

$\hat{y}_i$ = nilai prediksi

n = jumlah data yang diuji

Kelebihan utama *MAE* dibandingkan dengan metrik lain seperti *Root Mean Squared Error (RMSE)* terletak pada kemudahannya dalam interpretasi serta kemampuannya untuk tahan terhadap pengaruh nilai ekstrem atau outlier. *MAE* memiliki peran penting dalam evaluasi model karena memberikan bobot kesalahan yang sama untuk setiap data, sehingga mudah diinterpretasikan dan

cocok digunakan pada data yang tidak mengandung outlier besar. *MAE* tidak memberikan penalti yang berlebihan pada kesalahan besar, sehingga hasilnya lebih konsisten dan mudah dipahami. Namun, kelemahan *MAE* adalah kurangnya sensitivitas terhadap kesalahan yang signifikan, sehingga pada situasi dimana kesalahan besar perlu mendapatkan penalti lebih berat, metrik seperti *RMSE* lebih cocok untuk digunakan.

MAE banyak diaplikasikan dalam berbagai bidang seperti statistik, machine learning, dan analisis data. Semakin kecil nilai *MAE* yang diperoleh, semakin tinggi tingkat akurasi model dalam memprediksi data, karena hal ini menunjukkan bahwa selisih antara nilai prediksi dengan nilai aktual semakin kecil dan model mampu menghasilkan prediksi yang lebih mendekati kenyataan [44].