BAB I
PENDAHULUAN

1.1 Latar Belakang

Minyak merupakan salah satu komoditas energi yang penting di dunia saat ini. Berdasarkan *Statistical Review of World Energy* tahun 2024 oleh *Energy Institute*, konsumsi minyak dunia saat ini sebesar 33% dari total konsumsi energi dunia. Minyak merupakan sumber energi paling praktis pada sektor-sektor penting saat ini untuk mesin pembakaran internal, seperti pada industri, transportasi, bahkan pembangkit listrik (Trencher, 2022). Sebelum digunakan secara massal, komoditas ini berada dalam wujud awalnya berupa cairan kental pekat. Wujud awal tersebut disebut sebagai minyak mentah, karena masih baru dikeluarkan dari dalam perut bumi. Minyak mentah yang diperoleh nantinya akan diolah untuk menghasilkan produk-produk yang biasa digunakan dalam keberlangsungan hidup umat manusia, seperti bahan bakar kendaraan, pelumas mesin industri, dan lain-lain. Minyak mentah ada berbagai jenis yang tersedia di pasar minyak dan memiliki karakteristik serta harga yang berbeda-beda (Obite, 2021). Namun, di antara berbagai jenis tersebut terdapat salah satu yang dijadikan sebagai patokan atau *benchmark* karena kualitasnya diakui secara internasional, yaitu *Brent Crude Oil* atau minyak mentah *brent*.

Minyak mentah *brent* merupakan minyak mentah yang diekstraksi dari Laut Utara dekat Eropa di Samudra Atlantik. Secara spesifik, ladang minyak *brent* terletak di antara Norwegia dan Shetland di Eropa. Berdasarkan *Corporate Finance Institute* dan *IG International Trading*, minyak *brent* tergolong sebagai minyak ringan karena kepadatannya yang relatif rendah serta kandungan sulfurnya yang rendah. Karena sifatnya tersebut, minyak *brent* lebih disukai dalam penyulingan untuk menghasilkan produk-produk akhir seperti bensin, solar, dan lain-lain. Minyak mentah *brent* dijadikan patokan harga oleh pasar global. Akibatnya, fluktuasi dari harga minyak mentah *brent* dapat memberikan dampak kepada pasar minyak dunia.

Ma'arif et al. (2022) mengatakan bahwa, harga minyak mentah sangat mempengaruhi pasar karena pasar minyak merupakan pasar terdepan dalam perekonomian dunia. Selain itu fluktuasi harga minyak mentah juga menjadi pengaruh yang krusial dalam makroekonomi, bahkan sering disebut sebagai urat nadi perekonomian modern sebab harga minyak membentuk inflasi, nilai tukar, dan aktivitas ekonomi agregat (Mwanjilinji, 2025). Pada umumnya, yang diperjualbelikan adalah kontrak berjangka minyak mentah. Artinya, minyak mentah tidak langsung diterima oleh pembeli, melainkan pada titik waktu yang telah ditentukan oleh jenis kontrak tersebut. Hal tersebut menjadi acuan penelitian ini, dimana data yang digunakan adalah data minyak mentah *brent* dari 1 Juli 2021 hingga 30 Juni 2025. Oleh karena itu, perlu dilakukan peramalan atau prediksi untuk masa yang akan datang agar tercipta kebijakan-kebijakan yang dapat menjaga stabilitas ekonomi, perencanaan investasi, dan pengambilan keputusan strategis di berbagai industri.

Alruqimi & Persio (2024) mengatakan, prediksi harga minyak yang akurat sangat penting karena memiliki peran vital dalam ekonomi global, akan tetapi prediksi harga minyak terkenal dengan kompleksitasnya karena fluktuasi dari harganya. Mengacu pada hal tersebut, dibutuhkan model yang bisa menangkap pola kompleks dari fluktuasi harga minyak.

Penelitian-penelitan terdahulu telah membahas berbagai macam model yang baik digunakan untuk melakukan peramalan, salah satunya harga minyak mentah *brent*. Dengan berbasis *machine learning*, pengaplikasikan berbagai jenis algoritma seperti *Random Forest*, *SVR*, *ANN*, *DNN*, model *boosting*, dan bahkan sampai melakukan *hybrid* dilakukan untuk membuat model yang baik dalam memprediksi. Obite (2021) dalam penelitiannya menggunakan model *Random Forest*, *SVR*, *ANN*, dan *DNN* dalam memprediksi harga minyak mentah *brent* dengan dataset dari 7 Juni 2016 sampai 7 Juni 2021. Fitur yang digunakan merupakan dua hari terakhir dengan pendekatan estimasi *rolling window*, dengan pengoptimalan *hyperparameter* menggunakan *GridSearch*.

Dalam skripsi ini akan dilakukan peramalan *time series* berbasis *machine learning* pada harga penutupan minyak mentah *brent* menggunakan model *Random Forest*, *XGBoost*, dan *Support Vector Regression* (SVR) dengan menggunakan fitur harga penutupan tiga hari sebelumnya dan menggunakan pendekatan *supervised learning* serta optimasi *hyperparameter* menggunakan *optuna*. Ketiga model ini dipilih karena karakteristik dari masing-masing model, seperti *Random Forest* yang tahan terhadap *overfitting* dan juga jika ada data yang hilang (*Missing Value*), *XGBoost* bekerja dengan memperbaiki kesalahan dari pohon-pohon sebelumnya dan juga memiliki regularisasi yang membantu mencegah *overfitting*, dan *SVR* cocok pada dataset yang tergolong kecil sampai menengah dan punya generalisasi yang baik. Fitur yang diambil hanya sampai tiga hari terakhir karena diasumsikan bahwa pergerakan harga minyak saat ini tidak jauh berbeda dengan harga tiga hari terakhirnya. Pengambilan hari yang lebih banyak sebagai fitur beresiko ketidakrelevanan harga terhadap harga minyak saat ini, karena karakteristik dari harga minyak yang bisa berubah dengan cepat, selain itu pengecekan korelasi menggunakan *pearson correlation* antara harga tiga hari terakhir terhadap harga saat ini juga dilakukan untuk melihat bagimana hubungan antara variabel prediktor dan variabel respon. Perubahan harga yang cepat ini terjadi karena berbagai macam faktor, misalnya keadaan geopolitik maupun wabah yang sedang terjadi sehingga menghambat produksi minyak, dan sebagainya. Oleh karena itu, dipilih tiga hari terakhir untuk menangkap pergerakan jangka pendek dari harga minyak. Kemudian, untuk pengoptimalan *hyperparameter* dilakukan oleh kerangka kerja bernama *optuna*. *Optuna* sudah tersedia secara umum serta dikenal karena efisiensi dan performanya. Berbeda dengan *GridSearch* yang mencoba semua kombinasi *hyperparameter* yang telah ditentukan lalu menetapkan *hyperparameter* terbaik. Pendekatan yang digunakan *optuna* yaitu *Bayessian Optimization*, dimana untuk awalnya diambil kombinasi acak dari interval *hyperparameter* yang ditentukan lalu dihitung akurasinya dan pada percobaan

berikutnya akan belajar berdasarkan kesalahan pada percobaan sebelumnya, sehingga proses ini lebih efisien.

Berdasarkan penjabaran latar belakang di atas, penulis bermaksud melakukan penelitian dengan judul

"Peramalan Harga Penutupan Minyak Mentah *Brent* Menggunakan *Random Forest*, *XGBoost*, dan *Support Vector Regression* dengan Optimasi *Hyperparameter* Berbasis *Optuna*"

dengan fokus pada membandingkan kinerja model *Random Forest*, *XGBoost*, dan *SVR* dalam melakukan prediksi harga minyak mentah *brent*.

1.2 Rumusan Masalah

Penelitian ini berfokus pada pertanyaan berikut:

1. Bagaimana model *Random Forest*, *XGBoost*, dan *SVR* memprediksi harga minyak mentah *brent* berdasarkan harga pada tiga hari sebelumnya?
2. Antara model *Random Forest*, *XGBoost*, dan *SVR*, manakah yang menunjukkan hasil prediksi paling akurat untuk peramalan deret waktu minyak mentah *brent*?

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini meliputi:

1. Data yang digunakan merupakan data historis dari harga minyak mentah *brent* berjangka yang diperoleh melalui website *Investing.com*.
2. Periode data yang digunakan adalah 1 Juli 2021 hingga 30 Juni 2025, dengan pencatatan yang mengacu pada hari kerja internasional.
3. Target yang digunakan adalah data historis harga penutupan (*Price*) pada dataset dengan fitur yang digunakan untuk memprediksi target adalah harga penutupan tiga hari sebelumnya.
4. Model yang digunakan yaitu *Random Forest*, *XGBoost*, dan *SVR*.
5. Tuning *hyperparameter* dioptimisasi menggunakan *Optuna* untuk meningkatkan kinerja model.
6. Bahasa pemrograman yang digunakan adalah *Python* dengan editor kodenya menggunakan *VSCode*.
7. Metrik yang digunakan dalam penelitinan ini yaitu *Mean Absolute Error* (MAE), *Mean Absolute Error Percentage* (MAPE), *Mean Squared Error* (MSE), *Roor Mean Squared Error* (RMSE), dan *R2 Score*.

1.4 Tujuan Penelitian

Merujuk pada rumusan masalah, penelitian ini bertujuan untuk:

1. Melihat hasil prediksi dari masing-masing model dalam memprediksi harga minyak mentah *brent* berdasarkan harga tiga hari sebelumnya menggunakan kelima metrik.

2. Membandingkan hasil prediksi masing-masing model dan melihat yang mana menghasilkan skor metrik terbaik di antara ketiga model.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah memberikan pengalaman dalam mengembangkan pengetahuan pengaplikasian *machine learning* untuk melakukan peramalan harga minyak mentah *brent* menggunakan algoritma *Random Forest*, *XGBoost*, dan *SVR*. Selain itu, penelitian ini juga bisa menjadi referensi bagi akademisi lain yang mungkin tertarik pada pengaplikasian *machine learning* untuk melakukan peramalan terhadap harga minyak mentah.

1.6 Landasan Teori

1.6.1 Peramalan

Peramalan merupakan kegiatan memprediksi kejadian di masa depan berdasarkan data masa lalu. Peramalan berperan sebagai masukan untuk perencanaan terhadap masa depan, tapi hasil dari prediksi dapat berbeda dari rencana yang diperkirakan (Jaelani et al. 2022). Dalam dunia investasi, melakukan prediksi merupakan hal yang biasa. Hal ini dikarenakan, sebagian besar dunia investasi menggunakan manajemen investasi kuantitatif modern dan manajemen investasi kuantitatif setara dengan pengambilan keputusan berbasis data pasar (Maung & Swanson, 2025).

Dalam penelitian Jaelani et al. (2022), metode peramalan terdapat 3 jenis yakni metode *time series* dimana teknik statistika yang menggunakan data historis, metode regresi yang menghubungkan antara nilai produksi dan penyebabnya, dan metode kualitatif dimana proses prediksi dilakukan dengan penilaian ahli, manajemen dan pendapat untuk memperkirakan hasil dari prediksi tersebut.

1.6.2 Peramalan *Time Series*

Peramalan *time series* atau peramalan deret waktu merupakan teknik *machine learning* yang memprediksi nilai target berdasarkan riwayat nilai target yang telah diketahui. Peramalan deret waktu merupakan bentuk regresi khusus yang dikenal sebagai pemodelan *auto-regressive* (Surampudi, 2022). Variabel input pada peramalan deret waktu adalah nilai target lampau. Regunath (2021) mengatakan ada empat komponen umum yang membentuk model peramalan deret waktu, di antaranya:

1. Tren
   Peningkatan atau penurunan dalam rangkaian data jangka waktu yang lebih lama.
2. Musiman
   Fluktuasi dalam pola karena penentu musiman selama periode seperti hari, minggu, bulan, atau musim.
3. Variasi siklus
   Terjadi ketika data menunjukkan kenaikan dan penurunan pada interval yang tidak teratur.

4. Variasi acak atau tidak teratur
   Ketidakstabilan karena faktor acak yang tidak berulang dalam pola.

1.6.3 *Machine Learning* & Regresi

Machine learning merupakan aplikasi dari kecerdasan buatan untuk menemukan pola dan keteraturan dalam data (Josic et al., 2022). Hal ini membuat *machine learning* memiliki banyak aplikasi di bidang ekonomi terkhususnya dalam pasar perminyakan. Hall (2025) mengatakan, dalam beberapa tahun terakhir, model berbasis *machine leaarning* telah menunjukkan keberhasilan tertinggi dalam peramalan *time series* dan mampu menggeneralisasi dengan baik ke data yang tidak terlihat. *Machine leraning* unggul dalam skenario dependensi nonlinear yang kompleks dalam data. Karena kemampuannya ini, *machine learning* sangat aplikatif untuk permasalahan dunia nyata yang memiliki karakteristik yang beragam dan dinamis.

Dikutip dari laman *IBM*, *machine learning* memiliki beberapa jenis pembelajaran, yaitu *Supervised Learning*, *Unsupervised Learning*, dan *Reinforcement Learning*. Dalam pasar perminyakan tentunya tak lepas dengan peramalan yang berhubungan dengan numerik, sehingga tergolong kepada *Supervised Learning*. *Supervised Learning* memiliki dua model, salah satunya yaitu model regresi.

Dikutip dari laman *IBM* dan *Investopedia*, regresi adalah metode statistik yang menganalisis hubungan antara variabel dependen dengan satu atau lebih variabel independen. Regresi memprediksi nilai numerik kontinu, contohnya seperti harga, durasi, suhu, dan sebagainya. Model-model regresi ada banyak, regresi yang paling umum diketahui adalah regresi linear. Namun, beberapa kasus dalam dunia nyata memiliki hubungan antara variabel independen dan variabel dependen yang non-linear, sehingga diperlukan model selain regresi linear. Beberapa model seperti *Random Forest*, *XGBoost*, atau *SVR* bisa digunakan untuk kasus regresi dan juga menawarkan kemampuannya dalam menangkap hubungan linear maupun non-linear antara variabel independen dan variabel dependen serta terregulasi, sehingga membantu dalam menghindari *overfitting*.

1.6.4 *Data Transformation*

Data transformation atau transformasi data merupakan proses mengubah data mentah menjadi format atau struktur yang terpadu. Tujuannya adalah untuk memastikan data kompatibel dan membantu saat proses pengolahan data. Pada prakteknya, melakukan transformasi data saat *preprocessing* merupakan aspek penting dalam analisis data. Contoh transformasi data misalnya, mengubah format kolom tanggal pada dataset, mengurutkan data, standarisasi data, normalisasi data, menghapus baris atau kolom dataset, dan sebagainya.

1.6.5 *Outlier*

Outlier adalah nilai yang jauh berbeda dari nilai lainnya dala kumpulan data. Nilai *outlier* bisa lebih tinggi atau lebih rendah dibanding nilai lainnya dalam kumpulan data. Hal ini dapat disebabkan karena berbagai faktor, misalnya kesalahan

pengukuran, suatu peristiwa, atau faktor-faktor yang tidak terduga. *Outlier* bisa berdampak negatif pada karakteristik data riil (Junsawang, 2021). Namun, ada kalanya *outlier* justru data yang informatif karena memang datanya seperti itu adanya sebagai mana seperti pada penelitian Dharmayanti et al. (2024) yang mempertahankan *outlier* karena hasil identifikasinya menunjukkan bahwa data inputnya bukan hasil kesalahan. Ada beberapa cara untuk mengatasi *outlier*, seperti mengubah menjadi nilai median, mengubah menjadi nilai rata-rata, melakukan transformasi, atau bahkan dipertahankan. Hal ini bergantung pada kebutuhan, sehingga mendeteksi *outlier* menjadi langkah awal yang penting sebelum mengolah data agar data dapat dipahami dengan lebih tepat dan ditangani dengan baik.

1.6.6 *Interquartile Range* (IQR)

IQR merupakan salah satu cara untuk mendeteksi *outlier* dalam kumpulan data. Konsep IQR adalah dengan menentukan selisih antara kuartil ketiga (Q_3) dan kuartil pertama (Q_1), secara rumusan dituliskan seperti berikut (Dastjerdy, 2023):

$$IQR = Q_3 - Q_1 \tag{1}$$

Kemudian, penentuan *lower bound* (batas bawah) dan *upper bound* (batas atas) berdasarkan:

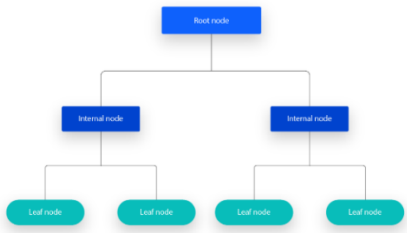
$$Lower\ Bound = Q_1 - 1.5 \times IQR \tag{2}$$

$$Upper\ Bound = Q_3 + 1.5 \times IQR \tag{3}$$

Suatu data yang nilainya kurang dari batas bawah atau lebih dari batas atas disebut sebagai *outlier*.

1.6.7 *Decision Tree* (*CART*)

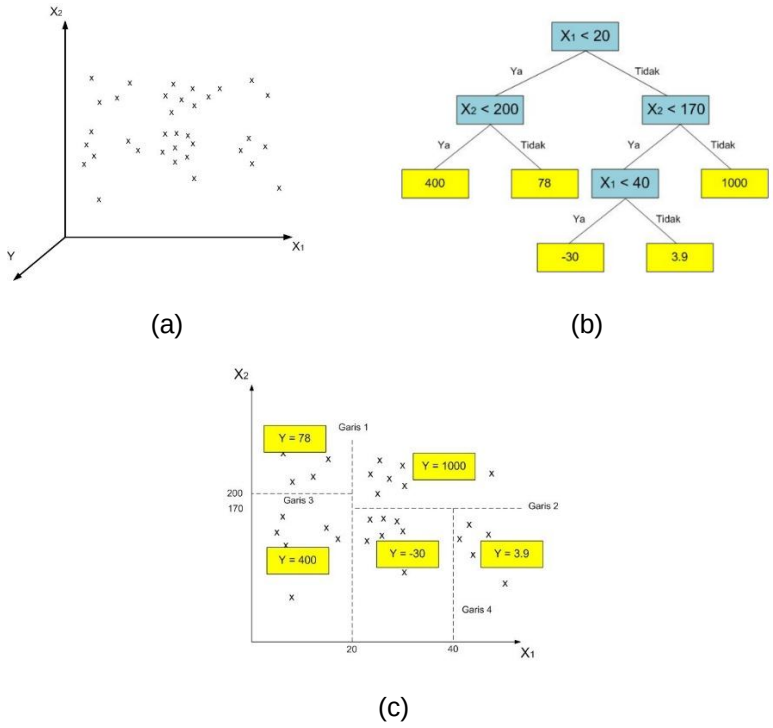
Decision Tree merupakan algoritma *machine learning* berbentuk pohon yang digunakan untuk mengambil keputusan berdasarkan data input. Algoritma ini memiliki struktur yang terdiri dari *root node* (simpul akar), *decision node* atau *internal node* (simpul keputusan/simpul internal), dan *leaf node* (simpul daun). Simpul akar mewakili pertanyaan pertama untuk membagi data. Kemudian, simpul keputusan merupakan pertanyaan lanjutan untuk membagi data lebih lanjut lagi. Yang terakhir yaitu simpul daun yang merupakan hasil akhir.



Gambar 1. Struktur Decision Tree

Sumber: IBM, https://www.ibm.com/id-id/think/topics/decision-trees

Salah satu jenis *Decision* Tree yaitu *Classification and Regression* Trees (*CART*). *CART* merupakan pohon yang bisa digunakan . Pada kasus regresi, hasil akhir pada masing-masing daun ini berupa nilai rata-rata dari data aktual variabel target dari masing-masing daerah yang terbentuk. Sebagai contoh misalkan dataset terdiri dari dua variabel bebas x_1 dan x_2 serta satu variabel tak bebas y seperti pada **Gambar 2** berikut:



Gambar 2. Ilustrasi cara kerja Decision Tree

Sumber: Herlambang, https://www.megabagus.id/machine-learning-decision-tree-regression/

Gambar 2 menunjukkan cara kerja dari *CART* dalam kasus regresi, dimana dilakukan split pada dataset yang ditunjukkan dengan letak splitnya pada garis 1 hingga garis 5. Dari hasil tersebut terbentuk lima daerah yang masing-masing daerah ini merupakan output dari *CART*. Sebagai contoh, apabila ada data input baru dan setelah melalui proses dari algoritma *CART* ternyata berada di daerah dengan $x_1 > 20$, $x_2 < 170$, dan $x_1 < 40$, maka output prediksi menghasilkan $y = -30$. Dapat dilihat bahwa keputusan yang dibentuk berdasarkan pada fitur pada data. Split data yang terjadi tidak berdasarkan pada fitur yang acak, melainkan semua fitur dievaluasi dan yang menghasilkan $Skor_{split}$ terkecil akan digunakan.

Misal diberikan kumpulan data latih sebanyak m sampel dan n fitur, sehingga $(\boldsymbol{X}_i, y_i)$ dengan $i = 1, 2, \dots, m$, dimana $\boldsymbol{X}_i \in \mathbb{R}^n$ adalah vektor input dengan $\boldsymbol{X}_1 =$

$(x_{11}, x_{12}, x_{13}, \dots, x_{1n})$, $\boldsymbol{X}_2 = (x_{21}, x_{22}, x_{23}, \dots, x_{2n})$, $\boldsymbol{X}_3 = (x_{31}, x_{32}, x_{33}, \dots, x_{3n})$ hingga $\boldsymbol{X}_m = (x_{m1}, x_{m2}, x_{m3}, \dots, x_{mn})$ dan $y_i \in \mathbb{R}$ adalah hasil prediksi (output). Misalkan juga data pada daun ke-k dituliskan sebagai R_k. R_k merupakan *region* pada daun ke-k yang dituliskan $R_k = R_{k_L} \cup R_{k_R}$ dimana $R_{k_L}(j, t_k) = \{x \mid x_j \leq t_k\}$ dan $R_{k_R}(j, t_k) = \{x \mid x_j > t_k\}$, j merupakan fitur ke-j dan t_k merupakan *threshold* pada daun ke-k. Merujuk pada penelitian Klusowski (2023), Wu et al. (2024), Rayarao dan Donikena (2025), serta website resmi *Scikit-learn* perhitungan $Skor_{split}$ terkecil dinyatakan pada rumus berikut:

$$Skor_{split} = \frac{|R_{k_L}|}{|R_k|} \cdot MSE_{k_L} + \frac{|R_{k_R}|}{|R_k|} \cdot MSE_{k_R} \tag{4}$$

$$MSE_{k_L} = \frac{1}{|R_{k_L}|} \sum_{i \in R_{k_L}} \left(y_{k_{L_i}} - \overline{y_{k_L}}\right)^2 \tag{5}$$

$$MSE_{k_R} = \frac{1}{|R_{k_R}|} \sum_{i \in R_{k_R}} \left(y_{k_{R_i}} - \overline{y_{k_R}}\right)^2 \tag{6}$$

dengan,

- $|R_{k_L}|$: Banyak data di region kiri pada daun ke-k
- $|R_{k_R}|$: Banyak data di region kanan pada daun ke-k
- $|R_k|$: Banyak data di seluruh region pada daun ke-k
- $y_{k_{L_i}}$: Data aktual ke-i di region kiri pada daun ke-k
- $y_{k_{R_i}}$: Data aktual ke-i di region kanan pada daun ke-k
- $\overline{y_{k_L}}$: Rata-rata data aktual di region kiri pada daun ke-k
- $\overline{y_{k_R}}$: Rata-rata data aktual di region kanan pada daun ke-k

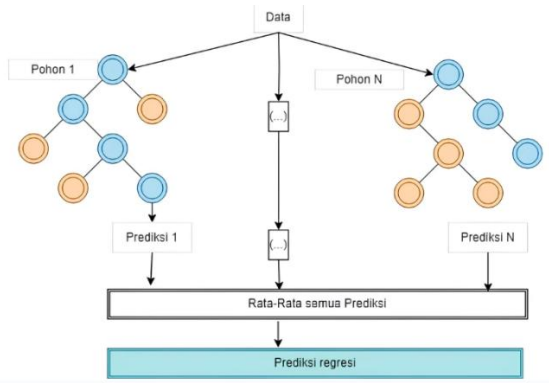
substitusikan persamaan (5) dan (6) ke persamaan (4), sehingga diperoleh

$$Skor_{split} = \frac{1}{|R_k|}\left(\sum_{i \in R_{k_L}} \left(y_{k_{L_i}} - \overline{y_{k_L}}\right)^2 + \sum_{i \in R_{k_R}} \left(y_{k_{R_i}} - \overline{y_{k_R}}\right)^2\right) \tag{7}$$

Split dengan nilai $Skor_{split}$ terkecil akan dipilih menjadi split yang fix. Selanjutnya proses ini berlanjut lagi pada simpul-simpul yang terbentuk. Proses ini akan berhenti ketika jumlah minimum split pada region, jumlah minimum sampel pada daun, dan kedalaman pohon sudah mencukupi batas yang ditentukan. *Hyperparameter* yang mengontrol hal tersebut adalah *min_samples_split, min_samples_leaf*, dan *max_depth*.

1.6.8 *Random Forest*

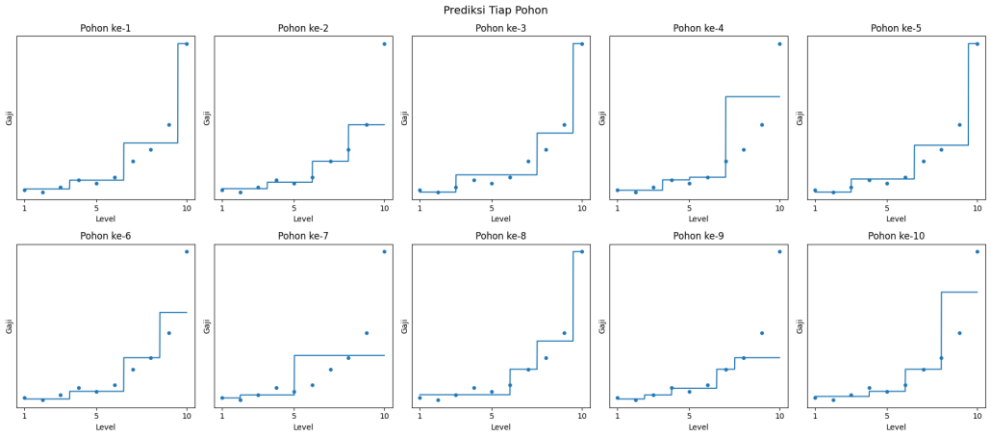
Random Forest merupakan pembelajar ensemble yang menggabungkan beberapa prediksi dari beberapa *CART* untuk menghasilkan prediksi yang lebih akurat dan stabil. Strobl et.al (2007) mengatakan, *Random Forest* memiliki kemampuan untuk diterapkan pada berbagai permasalahan prediksi, bahkan jika bersifat nonlinear dan melibatkan efek interaksi tingkat tinggi yang kompleks. Algortima ini merupakan pembelajaran terawasi yang dapat menjalankan tugas untuk klasifikasi ataupun regresi (GeeksforGeeks, 2025). Dalam kasus regresi, algoritma tersebut disebut *Random Forest Regression*. Untuk tugas regresi, algoritma akan memprediksi nilai numerik. Teknik ini memprediksi nilai kontinu dengan merata-ratakan hasil prediksi dari semua pohon keputusan.



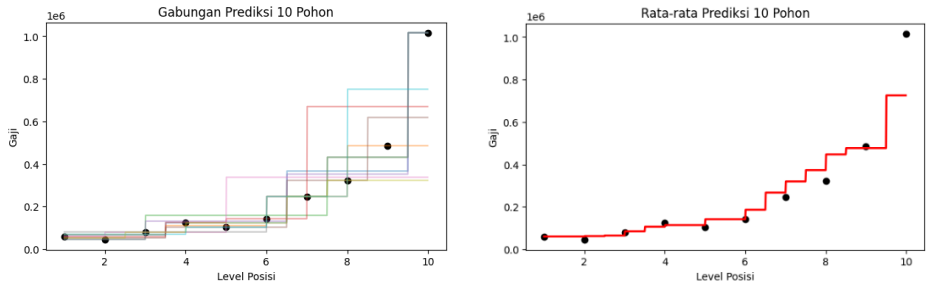
Gambar 3. Skema cara kerja Random Forest

Sumber: Ramadhani, (2024), exsight.id, https://exsight.id/blog/2024/10/14/random-forest-regression/

Adapun karakteristik *Random Forest* yaitu, stabil, tahan overfitting, dan bisa digunakan pada datset besar. *Random Forest* bekerja menggunakan teknik *bagging*. *Bagging* merupakan teknik yang melatih model *Decision Tree* dengan subset data yang berbeda. Subset data ini diambil secara acak dengan pengembalian atau disebut *bootstrap sampling*. Kemudian, prediksi akhirnya digabungkan dengan cara dirata-ratakan untuk memperoleh nilai numerik dalam kasus regresi (*aggregating*). Dengan teknik ini, random forest rentan terhadap overfitting dikarenakan kombinasi dari beberapa pohon keputusan akan meningkatkan kinerja prediksi (Hwang, 2023). Perhatikan ilustrasi berikut:



(a)

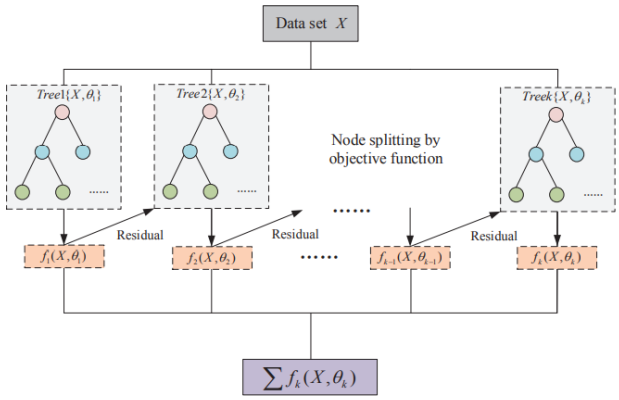


(b) (c)

Gambar 4. Ilustrasi cara kerja Random Forest

Misalkan pada kasus memprediksi besar gaji karyawan berdasarkan level posisinya dalam suatu perusahaan. Variabel level posisi menunjukkan level posisi dengan tanggungjawab yang besar, sedangkan variabel gaji merupakan variabel yang nilainya meningkat ketika level posisi semakin besar. **Gambar 4** (a) menunjukkan prediksi gaji karyawan berdasarkan level posisinya. Terdapat 10 pohon yang menampilkan hasil berbeda-beda dalam memprediksi hasil gaji berdasarkan level posisi. Kemudian, pada **Gambar 4** (b) menampilkan hasil prediksi masing-masing pohon dalam satu gambar. Hasil akhir ditunjukkan oleh **Gambar 4** (c) dimana karena kasus tersebut adalah kasus regresi, maka diambil nilai rata-rata dari hasil prediksi tiap pohon. Ilustrasi ini menunjukkan kinerja dari model *Random Forest* pada kasus regresi.

1.6.9 *XGBoost*



Gambar 5. Skema cara kerja XGBoost

Sumber: Bele, Hrishikesh, (2024), medium.com, https://medium.com/@hrishikeshbele1/xgboost-95b3cebfad4b

XGBoost merupakan singkatan dari *eXtreme Gradient Boosting*. *XGBoost* merupakan algoritma yang dikembangkan oleh Chen dan Guestrin pada tahun 2014 untuk mengatasi keterbatasan pada algoritma *Gradient-Boosted Decision Trees* (GBDT), khususnya dalam efisiensi komputasi dan skalabilitas.

XGBoost bekerja dengan cara menggabungkan beberapa model lemah untuk menjadi sebuah model yang kuat. *XGBoost* menggunakan pendekatan iteratif yang bertujuan untuk meningkatkan model sebelumnya dengan menambah model baru. Proses ini terus berulang sampai model yang dihasilkan memenuhi kriteria tertentu, dalam hal ini nilai *loss function* atau fungsi kerugiannya cukup kecil. *XGBoost* dapat diaplikasikan untuk menyelesaikan masalah klasifikasi maupun regresi. Pada masalah regresi, *loss function* yang digunakan secara default adalah *Squared Error*. Adapun karakteristiknya yaitu dapat menangani *missing value*, punya regularisasi bawaan sehingga tahan overfitting, mampu bekerja pada dataset yang besar, dan memungkinkan proses komputasi yang cepat.

Misal diberikan kumpulan data latih sebanyak m sampel dan n fitur. Merujuk pada Chen & Guestrin (2016), Guo et al. (2020), Liu & Liu (2022), Tarwidi et al. (2023), dan *medium.com*, *XGBoost* terdiri dari tiga hal berikut:

1. Inisialisasi

   Model diawali dengan prediksi awal (inisialisasi $k = 0$) yang meminimalkan *loss function* dengan mengasumsikan nilainya adalah nol.

$$\hat{y}_i^{(0)} = 0 \tag{8}$$

dimana,

- $\hat{y}_i^0$: Prediksi awal data ke-i

2. Model untuk iterasi ke-k

XGBoost membangun model prediksi dari rumusan berikut:

$$\begin{aligned}
\hat{y}_i^{(1)} &= \hat{y}_i^{(0)} + f_1(\boldsymbol{X}_i) = f_1(\boldsymbol{X}_i) \\
\hat{y}_i^{(2)} &= \hat{y}_i^{(1)} + f_2(\boldsymbol{X}_i) = f_1(\boldsymbol{X}_i) + f_2(\boldsymbol{X}_i) \\
\hat{y}_i^{(3)} &= \hat{y}_i^{(2)} + f_3(\boldsymbol{X}_i) = f_1(\boldsymbol{X}_i) + f_2(\boldsymbol{X}_i) + f_3(\boldsymbol{X}_i) \\
&\vdots \\
\hat{y}_i^{(k)} &= \hat{y}_i^{(k-1)} + f_k(\boldsymbol{X}_i) = \sum_{k=1}^{K} f_k(\boldsymbol{X}_i)
\end{aligned} \tag{9}$$

$$\hat{y}_i^{(k)} = \sum_{k=1}^{K} f_k(\boldsymbol{X}_i) \tag{10}$$

dimana,

- $\hat{y}_i^{(k)}$: Prediksi data ke-i untuk iterasi ke-k
- $f_k(\boldsymbol{X}_i)$: Pembelajaran ke-k pada data ke-i

3. Fungsi Objektif

Tujuan dari *XGBoost* adalah meminimalkan fungsi objektif yang terdiri atas dua bagian, yaitu *Loss Function* dan *Regularization Function* yang dirumuskan seperti berikut:

$$Z^{(k)} = \sum_{i=1}^{m} L\left(y_i, \hat{y}_i^{(k)}\right) + \Omega\left(f_k(\boldsymbol{X}_i)\right) \tag{11}$$

$$\Omega\left(f_k(\boldsymbol{X}_i)\right) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{12}$$

dimana,

- $L\left(y_i, \hat{y}_i{}^{(k)}\right)$: *Loss function*

- y_i : Data ke-i
- $\hat{y}_i^{(k)}$: Prediksi data ke-i pada iterasi ke-k
- $\Omega\left(f_k(\boldsymbol{X}_i)\right)$: *Regularization function*
- γ : Koefisien penalti simpul daun
- T : Jumlah simpul daun
- λ : Koefisien regular yang memastikan skor simpul daun tidak terlalu besar
- w_j : Skor simpul daun ke-j

Selanjutnya, berdasarkan persamaan (9) dapat diperoleh

$$Z^{(k)} = \sum_{i=1}^{m} L\left(y_i, \hat{y}_i^{(k-1)} + f_k(\boldsymbol{X}_i)\right) + \Omega\left(f_k(\boldsymbol{X}_i)\right) \tag{13}$$

Untuk memperoleh nilai optimal, *XGBoost* menggunakan perluasan deret taylor dari *loss function* hingga orde kedua untuk mengaproksimasi, sehingga fungsi objektifnya menjadi seperti berikut:

$$Z^{(k)} \approx \sum_{i=1}^{m} \left[L\left(y_i, \hat{y}_i^{(k-1)}\right) + g_i f_k(\boldsymbol{X}_i) + \frac{1}{2} h_i f_k^2(\boldsymbol{X}_i)\right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{14}$$

dengan $g_i = \frac{\partial L}{\partial \hat{y}_i^{(k-1)}}$ dan $h_i = \frac{\partial^2 L}{\partial \hat{y}_i^{(k-1)^2}}$. Selanjutnya, bagian $L\left(y_i, \hat{y}_i^{(k-1)}\right)$ pada persamaan (14) dapat dihilangkan karena sebagai konstanta, hal ini bertujuan untuk menyederhanakan proses. Dari sini diperoleh:

$$Z^{(k)} \approx \sum_{i=1}^{m} \left[g_i f_k(\boldsymbol{X}_i) + \frac{1}{2} h_i f_k^2(\boldsymbol{X}_i)\right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{15}$$

Kemudian, karena setiap titik data $\boldsymbol{X}_i$ pada $f_k(\boldsymbol{X}_i)$ akan jatuh pada tepat satu simpul daun pada pohon, didefinisikan $q(\boldsymbol{X}_i)$ sebagai fungsi yang memetakan titik data ke indeks simpul daun, sehingga dapat dituliskan seperti persamaan (16) dimana w merupakan skor dari daun tempat titik data tersebut berada.

$$f_k(\boldsymbol{X}_i) = w_{q(\boldsymbol{X}_i)} \tag{16}$$

Substitusikan persamaan (16) ke persamaan (15), sehingga diperoleh:

$$Z^{(k)} \approx \sum_{i=1}^{m} \left[g_i w_{q(\boldsymbol{X}_i)} + \frac{1}{2} h_i w_{q(\boldsymbol{X}_i)}^2\right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{17}$$

Dengan mendefinisikan $I_j = \{i \mid q(\boldsymbol{X}_i) = j\}$ sebagai kumpulan sampel pada daun ke-j, selanjutnya titik data dikelompokkan berdasarkan daun tempat titik data tersebut berada (Chen & Guestrin, 2016).

$$Z^{(k)} \approx \sum_{j=1}^{T}\left[\left(\sum_{i\in I_j} g_i\right)w_j + \frac{1}{2}\left(\sum_{i\in I_j} h_i + \lambda\right)w_j^2\right] + \gamma T \tag{18}$$

Agar minimum, akan dicari nilai w_j^* yang membuat $Z^{(k)}$ minimum dengan mengambil turunan parsial fungsi objektif tersebut terhadap w_j:

$$\frac{\partial Z^{(k)}}{\partial w_j} = \left(\sum_{i\in I_j} g_i\right) + \left(\sum_{i\in I_j} h_i + \lambda\right)w_j = 0 \tag{19}$$

$$w_j^* = -\frac{\left(\sum_{i\in I_j} g_i\right)}{\left(\sum_{i\in I_j} h_i + \lambda\right)} \tag{20}$$

Dengan mensubstitusikan persamaan (20) ke persamaan (18), diperolehlah

$$Z^{(k)} \approx -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j + \lambda} + \gamma T \tag{21}$$

dimana,

- $G_j = \sum_{i\in I_j} g_i$
- $H_j = \sum_{i\in I_j} h_i$

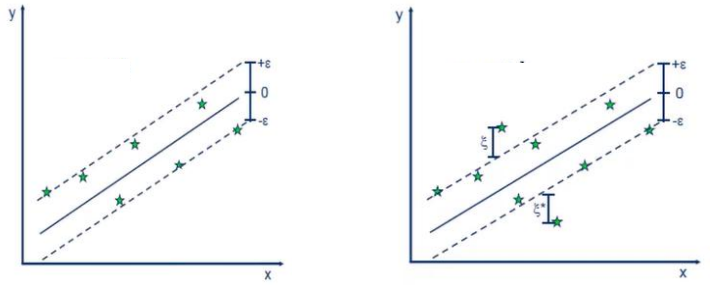
Selain tiga hal di atas, *XGBoost* memiliki cara untuk melakukan split data. Split yang dilakukan *XGBoost* tidak seperti CART klasik. Merujuk pada Chen & Guestrin (2016), dimisalkan $I = I_L \cup I_R$ dimana I merupakan daun induk, I_L merupakan daun kiri, dan I_R merupakan daun kanan. Cara *XGBoost* melakukan split yaitu berdasarkan *loss function*-nya sebagai berikut:

$$Gain = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma \tag{22}$$

1.6.10 *Support Vector Regression*

Support Vector Regression (*SVR*) merupakan algoritma yang diadaptasi dari *Support Vector Machine* (*SVM*) (Isnaeni et al., 2022). *SVM* menjalankan tugas klasifikasi, sementara *SVR* menjalankan tugas regresi. Isnaeni et al. (2022) mengatakan, konsep algoritma *SVR* dapat menghasilkan nilai peramalan atau prediksi yang bagus karena *SVR* mempunyai kemampuan menyelesaikan masalah overfitting, selain itu beberapa karakteristik *SVR* juga seperti, cocok pada dataset kecil sampai menengah, kemampuan generalisasi yang baik, dan juga proses komputasi yang

rendah. *SVR* bertujuan membuat regresi yang dapat menentukan *hyperplane* pada data latih dengan meminimalkan kesalahan. Kesalahan disebut juga sebagai *residual* yang artinya selisih antara hasil prediksi dengan nilai aktual. Pada penelitian ini digunakan *SVR* linear dalam memprediksi harga minyak mentah *brent*.



(a) (b)

Gambar 6. Ilustrasi model SVR

Sumber: Sayad, Saed, https://www-saedsayad-com.translate.goog/support_vector_machine_reg.htm?_x_tr_sch=http&_x_tr_sl=en&_x_tr_tl=id&_x_tr_hl=id&_x_tr_pto=imgs

Pada **Gambar 6** memperlihatkan ilustrasi dari *SVR* dimana terdapat garis lurus dan dua buah garis putus-putus sejauh ε dari garis lurus tersebut. Ruang yang terbentuk di antara garis lurus dan garis putus-putus disebut margin. Garis lurus ini secara umum disebut sebagai *hyperplane*. *SVR* linear bertujuan mencari fungsi *hyperplane* yang dapat membuat titik data berada di dalam ruang margin seperti pada **Gambar 6** (a). Namun, hal tersebut diperumum karena titik data tidak selalu berada di dalam ruang tersebut. Oleh karena itu, diperkenalkan variabel *slack* ξ_i dan ξ_i^* seperti pada **Gambar 6** (b) dimana $\xi_i, \xi_i^* \geq 0$.

Misal diberikan kumpulan data latih sebanyak *m* sampel dan n fitur. $(\boldsymbol{X}_i, y_i)$ dengan $i = 1, 2, \ldots, m$, dimana $\boldsymbol{X}_i \in \mathbb{R}^n$ adalah vektor input dan $y_i \in \mathbb{R}$ adalah hasil prediksi (output). Merujuk pada penelitian Bagaskara (2024), Safitri (2024), dan Vapnik (2000), perumusan dari *SVR* linear dituliskan sebagai berikut:

$$f(\boldsymbol{X}) = \boldsymbol{w}^T \boldsymbol{X} + b = \hat{y} \tag{23}$$

dengan,

- $\hat{y}$: Hasil prediksi
- $\boldsymbol{w}$: Vektor pembobot berukuran $n \times 1$
- $\boldsymbol{X}$: Vektor variabel input berukuran $n \times 1$
- b : Bias

Karena *SVR* bertujuan untuk mencari fungsi *hyperplane* yang dapat membuat titik data berada di dalam ruang margin, dengan kata lain *SVR* betujuan untuk memaksimalkan margin. Hal ini dapat dilakukan dengan meminimalkan $\left|\left|\boldsymbol{w}\right|\right|^2$. Dari sini terbentuk suatu permasalahan optimasi sebagai berikut:

Meminimalkan,

$$\frac{1}{2}\left|\left|\boldsymbol{w}\right|\right|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) \tag{24}$$

dengan syarat,

$$y_i - \boldsymbol{w}^T\boldsymbol{X}_i - b \leq \varepsilon + \xi_i \tag{25}$$

$$\boldsymbol{w}^T\boldsymbol{X}_i + b - y_i \leq \varepsilon + \xi_i^* \tag{26}$$

$$\xi_i, \xi_i^* \geq 0 \tag{27}$$

Merujuk pada penelitian Bagaskara (2024) solusi dari permasalahan ini dapat diselesaikan menggunakan metode kondisi *Karush-Kuhn-Tucker* (KKT) seperti berikut:

$$L(\boldsymbol{w}, b, \xi, \xi^*, \alpha, \alpha^*, \eta, \eta^*) = \frac{1}{2}\left|\left|\boldsymbol{w}\right|\right|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) + F + G + H \tag{28}$$

$$F = -\sum_{i=1}^{m} a_i(\varepsilon + \xi_i + \boldsymbol{w}^T\boldsymbol{X}_i + b - y_i) \tag{29}$$

$$G = -\sum_{i=1}^{m} \alpha_i^*(\varepsilon + \xi_i^* + y_i - \boldsymbol{w}^T\boldsymbol{X}_i - b) \tag{30}$$

$$H = -\sum_{i=1}^{m} (\eta_i\xi_i + \eta_i^*\xi_i^*) \tag{31}$$

dengan α_i, α_i^*, η_i, dan η_i^* merupakan pengali *Langrange* yang nilainya riil non-negatif. Berikut kondisi KKT-nya:

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{m}(a_i - a_i^*)\boldsymbol{X}_i = 0 \tag{32}$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{m}(a_i - a_i^*) = 0 \tag{33}$$

$$\frac{\partial L}{\partial \xi} = C - a_i - \eta_i = 0 \tag{34}$$

$$\frac{\partial L}{\partial \xi^*} = C - a_i^* - \eta_i^* = 0 \tag{35}$$

$$a_i(\varepsilon + \xi_i + \boldsymbol{w}^T \boldsymbol{X}_i + b - y_i) = 0 \tag{36}$$

$$\alpha_i^*(\varepsilon + \xi_i^* + y_i - \boldsymbol{w}^T \boldsymbol{X}_i - b) = 0 \tag{37}$$

$$\eta_i \xi_i = (C - a_i)\xi_i = 0 \tag{38}$$

$$\eta_i^* \xi_i^* = (C - a_i^*)\xi_i^* = 0 \tag{39}$$

dari persamaan (32) diperolehlah,

$$\boldsymbol{w} = \sum_{i=1}^{m} (a_i - a_i^*)\boldsymbol{X}_i \tag{40}$$

sehingga dengan mensubstitusi persamaan (36) ke persamaan (23), maka akan diperoleh:

$$\hat{y} = \sum_{i=1}^{m} (a_i - a_i^*)\boldsymbol{X}_i^T \boldsymbol{X}_i + b \tag{41}$$

dimana $a_i, a_i^* \in [0, C]$ berdasarkan persamaan (34) dan (35). Selanjutnya, untuk memperoleh solusi optimal untuk b diperoleh dari peninjauan kasus terhadap kondisi KKT yang dibuat. Berdasarkan Vapnik (2000) yang meninjau kasus ketika $0 < a_i, a_i^* < C$, diperoleh $(\varepsilon + \xi_i + \boldsymbol{w}^T \boldsymbol{X}_i + b - y_i) = 0$ dan $(\varepsilon + \xi_i^* + y_i - \boldsymbol{w}^T \boldsymbol{X}_i - b) = 0$. Selain itu, karena $0 < a_i, a_i^* < C$ berarti $C - a_i > 0$ dan juga $C - a_i^* > 0$, maka dari persamaan (38) dan (39) diperoleh bahwa ξ_i dan ξ_i^* bernilai nol. Dari sini didapat bahwa:

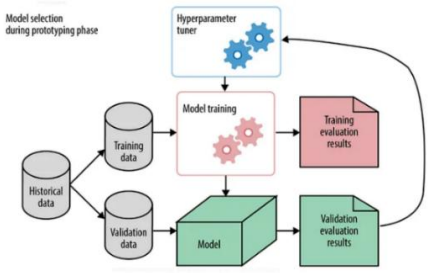
$$b = y_i - \boldsymbol{w}^T \boldsymbol{X}_i - \varepsilon \tag{42}$$

$$b = y_i - \boldsymbol{w}^T \boldsymbol{X}_i + \varepsilon \tag{43}$$

1.6.11 *Optuna Hyperparameter Optimization*

Pencarian *hyperparameter* merupakan salah satu tugas yang paling rumit dalam pembelajaran mesin. Terdapat beberapa metode dalam mencari *hyperparameter* yang membuat model bekerja dengan baik. Salah satu metode yang dapat digunakan adalah dengan *optuna*. *Optuna* merupakan salah satu *framework open-source* yang dikembangkan oleh perusahaan riset AI asal Jepang pada tahun 2019, yaitu Preferred Network. Konsep dasar *optuna* adalah melakukan studi berdasarkan fungsi objektif dan uji coba kepada fungsi objektif tersebut, dengan kata lain *optuna* melakukan iterasi *hyperparameter*.

Akiba (2019) mengatakan, *optuna* merumuskan optimasi *hyperparameter* sebagai proses meminimalkan/memaksimalkan fungsi objektif yang mengambil serangkaian *hyperparameter* sebagai input dan mengembalikan skor validasinya. Fungsi objektif merupakan inti dari proses optimasi dalam *optuna*.



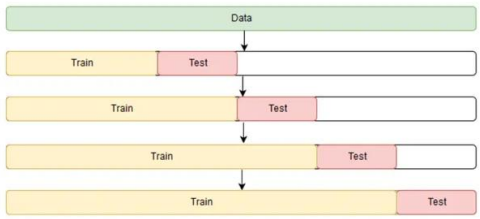
Gambar 7. Skema optimalisasi hyperparameter optuna

Sumber: *Avhale, K., (2021), medium, https://medium.com/@kalyaniavhale7/understanding-of-optuna-a-machine-learning-hyperparameter-optimization-framework-ed31ebb335b9*

Proses *optuna* menggunakan algoritma optimasi Bayesian seperti *Tree-structured Parzen Estimator* (TPE). Pada iterasi awal menggunakan sampling acak untuk memilih *hyperparameter* awal. Kemudian pemilihan *hyperparameter* uji coba berikutnya berdasarkan hasil percobaan sebelumnya. *Optuna* tidak menebak secara acak, akan tetapi belajar dari riwayat percobaan yang telah dilakukan.

1.6.12 *Time Series Cross Validation* (TSCV)

Cross validation merupakan sebuah teknik dalam *machine learning* untuk menilai performa model dengan melatih dan mengujinya pada berbagai subset data. Ini bertujuan untuk memastikan model tergeneralisasi dengan baik kepada data yang belum pernah dilihat sebelumnya. Dalam masalah *time series* validasinya sedikit berbeda. Dalam melakukan *cross validation* tidak boleh secara acak subset data yang digunakan, melainkan harus mempertahankan keterurutan waktunya. Hal ini dikarenakan dalam masalah *time series* tujuannya adalah memprediksi masa depan berdasarkan data historis yang ada, sehingga dalam melakukan *cross validation* harus memperhatikan urutan waktunya. Sebutan untuk *cross validation* pada masalah *time series* adalah *time series cross validation* (TSCV).



Gambar 8. Time series cross validation

Sumber: *Shrisvastava, S. (2020), Medium, https://medium.com/@soumyachess1496/cross-validation-in-time-series-566ae4981ce4*

Perlu diketahui bahwa dalam data dalam *cross validation* merupakan gabungan data latih dan data validasi. Kemudian, untuk TSCV dilakukan secara bergulir seperti pada **Gambar 8**. Prosesnya adalah memulai dengan subset data kecil untuk pelatihan, lalu memprediksi untuk data selanjutnya, kemudian periksa akurasinya. Selanjutnya, data yang diprediksi tadi dimasukkan sebagai set data pelatihan berikutnya, lalu lakukan prediksi lagi dan periksa akurasinya. Hal ini terus berlanjut sesuai dengan berapa *fold* atau lipatan yang kita inginkan. Jika mengacu pada **Gambar 1.8**, maka terdapat 4-*fold* atau 4 lipatan.

1.6.13 Metrik Evaluasi Model

Evaluasi kinerja pada model yang telah dibangun merupakan tahap yang sangat penting dalam *machine learning*. Hal ini dilakukan untuk menilai keakuratan dari model tersebut. Evaluasi kinerja model dalam penelitian ini menggunakan lima metrik berikut:

1. *Mean Absolute Error* (MAE)
   MAE adalah salah satu metode evaluasi yang umum digunakan. Metode ini menghitung rata-rata dari nilai absolut antara nilai aktual dan prediksi. Hal ini berarti semakin kecil nilai MAE, maka semakin bagus kualitas model.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{44}$$

   dimana,

   - n : Jumlah data
   - y_i : nilai aktual data ke-i, $i = 1, 2, 3, \dots, n$
   - $\hat{y}_i$: nilai prediksi data ke-i, $i = 1, 2, 3, \dots, n$

2. *Mean Absolute Percentage Error* (MAPE)
   MAPE merupakan salah satu metode evaluasi lain yang umum digunakan. Metode ini hanya mengubah metode MAE ke dalam persentase.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{45}$$

   Adapun kriterian kualitatif dari MAPE yaitu sebagai berikut (Vivas, 2020):

Tabel 1. Kriteria kualitatif MAPE

MAPE (%)	Keterangan
< 10	Sangat akurat
$10 - 20$	Baik
$20 - 50$	Tidak terlalu akurat
> 50	Tidak akurat

3. *Mean Squared Error* (MSE)
   MSE merupakan salah satu metode evaluasi lain yang umum digunakan. Metode ini menghitung rata-rata kuadrat antara nilai aktual dan prediksi. Hal ini berarti semakin kecil nilai MSE, maka semakin bagus kualitas model.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{46}$$

4. *Root Mean Squared Error* (RMSE)
   RMSE merupakan salah satu metode evaluasi lain yang umum digunakan. Metode ini merupakan bentuk akar dari MSE. Hal ini berarti semakin kecil nilai RMSE, semakin bagus juga kualitas model.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{47}$$

5. *R2 Score* (r^2)
   R2 Score merupakan metrik yang digunakan untuk mengukur proporsi varians dalam variabel dependen yang dapat diprediksi atau dijelaskan oleh variabel independen. Nilai *R2 Score* berada pada kisaran 0 sampai 1. Semakin nilainya dekat ke 0, ini menunjukkan model tidak bisa menjelaskan hubungan antara variabel dependen dengan variabel independen atau dengan kata lain modelnya tidak bagus dalam melakukan prediksi. Semakin nilainya dekat ke 1, ini menunjukkan dapat menjelaskan hubungan hubungan antara variabel dependen dengan variabel independen atau dengan kata lain modelnya bagus dalam melakukan prediksi.

$$r^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{48}$$

   dimana,

   - $\bar{y}$: Rata-rata dari nilai aktual

   Adapun interpretasi bagus atau tidaknya nilai dari r^2 dinyatakan dalam tabel berikut (Nurani et al., 2023):

Tabel 2. Kriteria nilai r^2

Interval	Hubungan
$0{,}8 - 1$	Sangat kuat
$0{,}6 - 0{,}79$	Kuat
$0{,}4 - 0{,}59$	Cukup kuat
$0{,}2 - 0{,}39$	Lemah
$0 - 0{,}19$	Sangat lemah

Kelima metrik tersebut yang akan digunakan dalam penelitian ini karena dianggap cukup untuk menggambarkan tingkat akurat dari model yang digunakan.

BAB II
METODE PENELITIAN

2.1 Sumber Data

Penelitian ini menggunakan data sekunder, yaitu data yang dikumpulkan oleh pihak kedua. Data pada penelitian ini merupakan data historis harga minyak mentah *brent* berjangka pada 1 Juli 2021 hingga 30 Juni 2025 yang diperoleh secara online melalui website https://www.investing.com/commodities/brent-oil-historical-data.

2.2 Metode Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan memanfaatkan tiga model *machine learning*, yaitu *Random Forest*, *eXtreme Gradient Boosting* (XGBoost), dan *Support Vector Regression* (SVR) untuk memprediksi harga penutupan minyak mentah *brent*. Jenis pembelajaran yang digunakan adalah *supervised learning* dengan split 80:20. Fitur yang digunakan adalah harga penutupan tiga hari terakhir dan targetnya adalah harga penutupan saat ini. Ketiga model ini akan dibandingkan untuk melihat kinerja dari masing-masing model dalam melakukan prediksi dan melihat model mana yang memiliki performa paling baik. Evaluasi dari kinerja model dilakukan dengan lima metrik evaluasi, yaitu *Mean Absolute Error*, *Mean Absolute Percentage Error*, *Mean Squared Error*, Root *Mean Squared Error*, dan *R2 Score*.

2.3 Prosedur Penelitian

2.3.1 Identifikasi Masalah

Penelitian dimulai dengan mengidentifikasi masalah terkait perminyakan yang dilakukan melalui observasi selama magang dan studi literatur. Tahap ini berperan penting agar penelitian lebih terarah dan terstruktur dalam menyelesaikan penelitian.

2.3.2 Studi Literatur

Studi literatur dilakukan sebelum dan sesudah mengidentifikasi masalah untuk memahami masalah yang terjadi serta bagaimana perkembangan penelitian yang telah. Melalui beberapa sumber seperti berita online, artikel, dan skripsi, serta pengalaman magang, peneliti dapat memperoleh wawasan tentang teori dan metode pendekatan yang dapat dilakukan.

2.3.3 Pengumpulan Data

Data historis harga minyak mentah *brent* berjangka diperoleh dari *Investing.com*, yang mencakup informasi seperti harga penutupan (*Price*), harga pembukaan (*Open*), harga tertinggi (*High*), harga terendah (*Low*), volume (*Vol.*), dan persentase perubahan harga (*Change %*). Data yang diambil adalah data lima tahun terakhir, yaitu mulai dari 1 Juli 2021 hingga 30 Juni 2025. Pemilihan interval tersebut karena data tergolong terbaru dan masa ini mencakup masa pandemi *Covid-19* dan masa invasi Rusia terhadap Ukraina, sehingga menyediakan kondisi fluktuasi harga.

2.3.4 *Preprocessing* Data

Data yang dikumpulkan akan diproses terlebih dahulu agar dapat digunakan dan terjamin kualitasnya. Tahapan dalam *preprocessing* terdiri dari empat tahap, yaitu *data transformation*, pengecekan *missing value*, pengecekan grafik, dan pengecekan outlier. Tahap *data transformation* dilakukan untuk mengubah tipe datanya agar bisa diolah dengan python. Tahap pengecekan *missing value* dilakukan untuk mendeteksi data yang hilang, sehingga jika ada data yang kosong akan diisi agar tidak mengganggu proses pelatihan model yang beresiko menurunkan akurasi prediksi. Kemudian pengecekan grafik dan *outlier* dilakukan untuk melihat bagaimana pergerakan harga penutupan minyak mentah *brent* berubah terhadap waktu dan mendeteksi pergerakan harga yang tidak wajar, hal ini agar data dapat diberikan *treatment* atau perlakuan tertentu sebelum diolah lebih lanjut.

2.3.5 *Split* Data

Tahap ini merupakan tahap membagi dataset. Dalam penelitian ini dataset dibagi menjadi dua bagian, yaitu *data train* (data latih) dan *data test* (data uji). Data latih digunakan untuk melatih model dalam mengenali pola data yang diberikan dan juga digunakan dalam proses pencarian *hyperparameter* terbaik. Kemudian, data uji digunakan untuk menguji model yang dibangun dengan menggunakan *hyperparameter* terbaik yang telah diperoleh. Data uji tidak dilibatkan dalam proses pelatihan dan proses tuning *hyperparameter* atau dengan kata lain data uji merupakan data yang belum pernah dilihat sama sekali sebelumnya, peran data uji sebagai evaluasi terakhir untuk mengukur seberapa baik model melakukan prediksi.

2.3.6 Implementasi Model

Penelitian ini menggunakan bahasa pemprograman *python* dengan berbagai *library* untuk menerapkan algoritma *Random Forest*, *XGBoost*, dan *SVR*. Model-model tersebut awalnya dibangun menggunakan *hyperparameter* default yang berdasarkan pada notebook-notebook dari *kaggle*. Selanjutnya, akan dilakukan pengoptimalan *hyperparameter* menggunakan *optuna* untuk meningkatkan performa dari model-model tersebut.

2.3.7 Evaluasi Kinerja Model

Pada tahap ini dilakukan perbandingan antara nilai aktual dan nilai hasil prediksi model. Evaluasi kinerja model dilakukan menggunakan lima metrik, yaitu *Mean Absolute Error* (MAE), *Mean Absolute Percentage Error* (MAPE), *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), dan *R2 Score* (r^2). *MAE* berfungsi untuk melihat seberapa jauh kesalahan rata-rata dalam peramalan, *MAPE* berfungsi untuk melihat persentase kesalahan pada peramalan, *MSE* dan *RMSE* digunakan karena sangat umum dimanfaatkan untuk mengukur seberapa besar selisih kuadrat antara hasil prediksi dengan nilai aktualnya, dan *R2 Score* berfungsi untuk melihat seberapa sesuai hasil prediksi dengan data aktualnya.

2.3.8 Penarikan Kesimpulan

Penarikan kesimpulan dalam penelitian ini berdasarkan hasil dan pembahasan yang telah dilakukan sebagai jawaban dari rumusan masalah penelitian. Kesimpulan yang

diperoleh merupakan model mana yang menghasilkan tingkat keakuratan terbaik dalam melakukan prediksi harga minyak mentah *brent* setelah dilakukan pengoptimalan *hyperparameter*.

2.4 Alur Penelitian