BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Dalam mengembangkan kerangka penelitian untuk analisis komparatif algoritma *Random Forest* dan *Gradient Boosting* pada prediksi lama menginap tamu hotel, diperlukan tinjauan mendalam terhadap penelitian-penelitian sebelumnya yang telah mengeksplorasi penerapan *Machine Learning* dalam industri perhotelan. Kajian literatur ini mencakup delapan penelitian kunci yang dipilih berdasarkan relevansi metodologi, konteks aplikasi, dan kualitas publikasi, meliputi:

Tabel 2. 1 Penelitian Terdahulu

Referensi ke-1	
Judul	*Hotel Guest Length of Stay Prediction Using Random Forest Regressor*
Nama Penulis	Yerik Afrianto Singgalen [26].
Sumber Jurnal	*Journal of Information Systems and Informatics* (SINTA 3)
Tahun	2024
Permasalahan	Memprediksi *length of stay* (LoS) hotel berdasarkan fitur *country*, *guest type*, *room type*, dan *rating* dengan akurasi tinggi.
Framework	*Random Forest Regression*, *Feature Engineering*, *Data Preprocessing* dengan *encoding categorical variables*.
Temuan	R^2 = 0.85, MAE = 1.06. *Feature importance*: *Country* (0.5), *Guest Type* (0.2), *Room Type* (0.15), *Rating* (0.15)
Pembahasan	*Random Forest* efektif menangani data *non-linear* dengan performa tinggi. *Country* menjadi prediktor terkuat dalam menentukan *length of stay* (LoS).
Relevansi	Memberikan *baseline* kuat untuk implementasi *Random Forest* dalam prediksi *length of stay* (LoS) hotel serta *benchmark* metrik evaluasi (R^2, MAE) yang dapat dibandingkan dengan *Gradient Boosting* pada penelitian ini
Referensi ke-2	
Judul	*Big Data in Tourism and Hospitality Industry: Predictive Analytics of Hotel Room Trend*
Nama Penulis	Yerik Afrianto Singgalen [27].
Sumber Jurnal	*Indonesian Journal of Tourism and Leisure* (SINTA 3)
Tahun	2025

Permasalahan	Implementasi *XGBoost* untuk prediksi *room type* dan *trend okupansi* hotel berbasis *big data analytics*.
Framework	*XGBoost*, *Gradient Boosting*, SMOTE untuk *class imbalance*, *Grid Search hyperparameter optimization*.
Temuan	Akurasi prediksi 85% untuk *room selection patterns*. *Feature importance*: *Rating* (0.35), *Length of Stay* (0.28), *Guest Type* (0.22).
Pembahasan	*XGBoost* unggul dalam menangani dataset kompleks dan *imbalanced data*. *Gradient boosting* efektif untuk identifikasi pola tersembunyi.
Relevansi	Validasi keunggulan algoritma *Gradient Boosting* (*XGBoost*) pada dataset hotel dan memberikan *insight* tentang *feature engineering* serta *handling imbalanced* data yang relevan dan sejalan dengan penelitian ini.
Referensi ke-3	
Judul	*Booker Prediction from Requests for Quotation via Machine Learning Techniques*
Nama Penulis	Samuel Runggaldier, Gabriele Sottocornola, Andrea Janes, Fabio Stella, Markus Zanker [28].
Sumber Jurnal	*Tourism and Hospitality Management.* (Scopus Q3)
Tahun	2023
Permasalahan	Memprediksi apakah permintaan penawaran (*request for quotation*) akan berubah menjadi pemesanan aktual menggunakan teknik *Machine Learning* untuk mengoptimalkan prioritas manajemen korespondensi hotel.
Framework	*Random Forest*, *Extra Tree*, *Gaussian Naive Bayes*, *Multi-layer Perceptron* (MLP), *Support Vector Classifier* (SVC), 10-fold *Cross-Validation*, *Class Imbalance Subsampling*, *Feature Selection* dengan *Entropy-based Scoring.*
Temuan	*Random Forest* mencapai *F1-score* tertinggi (35%), *precision* 48%, *recall* 27%. MLP mencapai *precision* tertinggi (66%) dan *accuracy* 92%. *Naive Bayes* mencapai *recall* tertinggi (65%). *Feature* terpenting: *CR_RequestedDaysBeforeArrival* (0.31), *CG_CountryCode* (0.20), *CR_Duration* (0.12).
Pembahasan	Algoritma *Random Forest* dan *Extra Tree*) menunjukkan performa superior dalam prediksi *booking* hotel. *Class imbalance subsampling* dengan ratio 1:3 menghasilkan *trade-off* optimal *precision-recall*. MLP dan *Random Forest* paling menjanjikan dengan *F1-score* yaitu 42% setelah optimasi.
Relevansi	Memberikan validasi empiris tentang efektivitas *Random Forest* dalam prediksi hotel *booking* dan metodologi *handling class imbalance*. *Framework feature selection* dan *comparative analysis multiple algorithms* sangat relevan untuk penelitian komparasi model.
Referensi ke-4	
Judul	*Development of Machine Learning Model to Predict Hotel Room Reservation Cancellations*

Nama Penulis	Eka Rahmawati, Galih Setiawan Nurohim, Candra Agustina, Denny Irawan, Zainal Muttaqin [29].
Sumber Jurnal	Jurnal Teknologi Informasi Dan Terapan (J-TIT) (SINTA 2)
Tahun	2024
Permasalahan	Prediksi pembatalan reservasi hotel menggunakan *multiple Machine Learning algorithms* pada *dataset* Hotel Borobudur.
Framework	*Random Forest*, *Logistic Regression*, *Bayesian Networks, Cross-validation*.
Temuan	*Random Forest*: Akurasi 86.36%, Precision 88.06%, *Recall* 93.65%, *F1-score* 90.77%. *Random Forest outperforms Logistic Regression* dan *Bayesian Networks*.
Pembahasan	*Random Forest* sangat *robust* untuk *non-linear relationships* dan *imbalanced* hotel data. *Bayesian Networks underperform* pada dataset kompleks.
Relevansi	Konfirmasi superioritas *Random Forest* pada prediksi hotel-*related tasks* dalam konteks Indonesia, serta memberikan *benchmark* performa untuk evaluasi komparatif.
Reverensi ke-5	
Judul	*Predicting Hotel Booking Cancellations Using Machine Learning for Revenue Optimization*
Nama Penulis	Andy Hermawan, Aji Saputra, Nabila Lailinajma, Reska Julianti, Timothy Hartanto, Troy Kornelius Daniel [30].
Sumber Jurnal	*Router*: Jurnal Teknik Informatika dan Terapan (SINTA 3).
Tahun	2025
Permasalahan	Optimasi model *Machine Learning* untuk prediksi dan mitigasi pembatalan *booking* hotel dengan fokus *revenue optimization*.
Framework	*XGBoost*, *SMOTE resampling*, *Threshold tuning*, *SHAP interpretation*.
Temuan	*F2-score* 0.79 (*XGBoost* terbaik). Finansial: penghematan $286,800. *Key features*: *deposit type*, *special requests*, dan *market segment*.
Pembahasan	*XGBoost* dengan *hyperparameter tuning* dan *threshold adjustment* memberikan ROI tertinggi dalam hotel *revenue management.*
Relevansi	Demonstrasi aplikasi praktis *Gradient Boosting* dalam konteks bisnis hotel Indonesia, serta metodologi *threshold optimization* yang dapat diadaptasi.
Referensi ke-6	
Judul	*Performance Comparison of Random Forest (RF) and Classification and Regression Trees (CART) for Hotel Star Rating Prediction*
Nama Penulis	Annisaa Utami, Dimas Fanny Hebrasianto Permadi, Yesy Diah Rosita, Jumanto Unjung [31].
Sumber Jurnal	SJI: *Scientific Journals of Informatics* (SINTA 3)
Tahun	2024

Permasalahan	Evaluasi komparatif *Random Forest* vs *CART* untuk klasifikasi rating hotel berbasis *customer reviews*.
Framework	*Random Forest*, *CART*, *Hyperparameter Tuning*, *Cross-validation*.
Temuan	*Random Forest*: Akurasi >99%, *superior robustness*. *CART*: *prone to overfitting* pada dataset besar.
Pembahasan	*Random Forest* konsisten unggul dibanding *CART* pada berbagai ukuran dataset hotel. *Ensemble method* lebih stabil.
Relevansi	Validasi keunggulan *Random Forest* dalam *domain* hotel *classification tasks* dan pentingnya analisis yang komparatif antara *ensemble* vs *single tree algorithms*.
Referensi ke-7	
Judul	*Personalized Restaurant Recommendations: A Hybrid Filtering Approach for Mobile Applications*
Nama Penulis	Christopher Matthew Marvelio, Alexander Waworuntu [32].
Sumber Jurnal	*International Journal of New Media Technology* UMN (IJMT UMN)
Tahun	2025
Permasalahan	Pengembangan sistem rekomendasi restaurant menggunakan *hybrid filtering* (*content-based* dan *collaborative filtering*).
Framework	*Hybrid Filtering*, *Content-Based Filtering*, *Collaborative Filtering*, *React Native*, *Flask*.
Temuan	*EUCS satisfaction* 93.9%. *The hybrid approach outperforms single filtering methods significantly*.
Pembahasan	Kombinasi *multiple filtering techniques* menghasilkan rekomendasi lebih akurat dan personal dibanding *single approach*.
Relevansi	Memberikan *insight* tentang *hybrid Machine Learning approaches* dan metodologi evaluasi *user satisfaction* yang dapat diadaptasi untuk penelitian prediksi hotel LoS.
Referensi ke-8	
Judul	*Adverse Media Classification: A New Era of Risk Management with XGBoost and Gradient Boosting Algorithms*
Nama Penulis	Reza Juliandri, Monika Evelin Johan, Jansen Wiratama, Samuel Ady Sanjaya [33].
Sumber Jurnal	2024 5th *International Conference on Big Data Analytics and Practices* (IBDAP) – IEEE Xplore
Tahun	2024
Permasalahan	Klasifikasi berita negatif (*adverse media*) untuk mendukung sistem *customer due diligence* dalam industri *fintech*, mengurangi risiko *fraud* dan *money laundering.*
Framework	*XGBoost*, *Gradient Boosting, Back Translation, Grid Search Hyperparameter Tuning,* 10-*fold Cross-Validation*, *Class Imbalance Subsampling*.

Temuan	*Gradient Boosting* (753 *records*) mencapai akurasi 82.31% pada test data dan 84.93% pada *validation data*, *XGBoost* (1,281 *records*) mencapai akurasi 78.54% pada test data dan 82.52% pada *validation data*.
Pembahasan	*Back translation* efektif menangani *class imbalance, Grid Search* yang membantu optimasi pada *parameter*, dan *ensemble methods* unggul dalam klasifikasi *news headline* berbasis teks.
Relevansi	Metodologi *comparative analysis* dan teknik *preprocessing* teks (*back translation*, *TF-IDF/CountVectorizer*) dapat diadaptasi untuk prediksi lama menginap tamu hotel menggunakan *Random Forest* dan *Gradient Boosting*.

Seluruh penelitian terdahulu yang dirangkum dalam Tabel 2.1 memiliki area kajian yang sejenis, yaitu penerapan algoritma *Machine Learning* pada industri perhotelan dan sektor terkait, dengan fokus utama pada prediksi perilaku pelanggan, okupansi, hingga pembatalan reservasi. Berdasarkan hasil dari delapan jurnal tersebut, penelitian pertama dan kedua [26] [27], menjadi acuan utama dalam penelitian ini karena keduanya membahas penerapan *Random Forest* dan *XGBoost* dalam konteks prediksi hotel. Pada penelitian pertama, algoritma *Random Forest* berhasil mencapai nilai R^2 sebesar 0,85 dan MAE 1,06, menunjukkan performa tinggi dalam memprediksi *length of stay* (LoS) berdasarkan fitur seperti *country* dan *guest type* [26]. Sementara pada penelitian keduanya, model *XGBoost* memperoleh akurasi prediksi sebesar 85%, membuktikan kemampuannya dalam menangani data hotel berskala besar.

Penelitian lain turut memberikan kontribusi penting terhadap pengembangan model prediktif. Penelitian yang membandingkan berbagai algoritma seperti *Random Forest*, *MLP*, dan *Naive Bayes*, dengan hasil *F1-score* 42% setelah optimasi, menunjukkan kinerja lebih baik dalam prediksi *hotel booking conversion* [28]. Selanjutnya penelitian lain juga memperkuat temuan tersebut dengan menunjukkan bahwa *Random Forest* mencapai akurasi 86,36%, *precision* 88,06%, dan *recall* 93,65%, mengungguli *Logistic Regression dan Bayesian Networks* [29]. Penelitian lain juga membuktikan superioritas *XGBoost* dengan hasil *F2-score* 0,79 serta dampak finansial berupa penghematan sebesar $286.800, menegaskan potensi algoritma ini dalam optimasi pendapatan hotel melalui prediksi pembatalan.

Selain itu, penelitian yang menunjukkan bahwa *Random Forest* memiliki akurasi di atas 99% pada klasifikasi *rating* hotel, mengungguli metode *CART* yang rentan terhadap *overfitting* [31]. Pendekatan serupa juga tampak pada penelitian terdahulu yang ke tujuh yang menggunakan metode *hybrid filtering* dan menghasilkan tingkat kepuasan pengguna (EUCS) sebesar 93,9%, membuktikan efektivitas kombinasi pendekatan *Machine Learning* dalam sistem rekomendasi restoran [32]. Adapun penelitian terakhir menjelaskan terkait performa *Gradient Boosting* dan *XGBoost* dalam klasifikasi teks, dengan akurasi masing-masing 84,93% dan 82,52%, serta menekankan pentingnya teknik *feature engineering* seperti *back translation* untuk meningkatkan kualitas model.

Berdasarkan hasil dari seluruh penelitian terdahulu, dapat disimpulkan bahwa algoritma *Random Forest* dan *Gradient Boosting* memiliki keunggulan signifikan dalam hal akurasi, stabilitas, dan kemampuan menangani data kompleks maupun tidak seimbang. Oleh karena itu, dalam penelitian ini, kedua algoritma tersebut diadopsi untuk membangun model prediksi *length of stay* (LoS) tamu hotel. Analisis dilakukan untuk membandingkan performa keduanya dengan menggunakan metrik regresi standar yaitu RMSE, MAE, dan R^2, tanpa menerapkan metrik klasifikasi.

Berbeda dengan penelitian terdahulu yang banyak berfokus pada prediksi pembatalan reservasi, konversi pemesanan, atau rekomendasi layanan, penelitian ini memiliki ruang kajian yang lebih spesifik. Penelitian ini berfokus pada pembangunan model prediksi lama menginap tamu (*length of stay*) dengan memanfaatkan data transaksi nyata dari operasional hotel.

2.2 Tinjauan Teori

2.2.1 Machine Learning

Machine Learning adalah bidang ilmu komputer yang berfokus pada pengembangan algoritma dan teknik yang memungkinkan sistem untuk belajar dari data dan meningkatkan kinerjanya seiring bertambahnya pengalaman tanpa pemrograman eksplisit [34]. Konsep ini dipopulerkan oleh Tom M. Mitchell dengan menegaskan bahwa pembelajaran berorientasi pada peningkatan

performa model yang diukur dengan metrik tertentu ketika diberikan lebih banyak data [35]. Algoritma pembelajaran mesin pada dasarnya dapat diklasifikasikan ke dalam tiga kategori utama, yakni *supervised learning*, *unsupervised learning*, dan *reinforcement learning*. Pada *supervised learning*, proses pelatihan dilakukan dengan memanfaatkan data yang telah dilabeli untuk membimbing model dalam melakukan prediksi. Sebaliknya, *unsupervised learning* berfokus pada penggalian pola atau struktur yang tersembunyi tanpa adanya bantuan label pada data. Adapun *reinforcement learning* menekankan pada interaksi agen dengan lingkungannya, di mana agen tersebut memperoleh pengalaman melalui umpan balik berupa penghargaan (*reward*) atau hukuman (*punishment*) guna mencapai tujuan tertentu secara optimal.

Paradigma *Machine Learning* terbagi menjadi *supervised learning* untuk tugas klasifikasi dan regresi dengan data berlabel, *unsupervised learning* untuk eksplorasi struktur data tanpa label, serta *reinforcement learning* untuk pengambilan keputusan berbasis *reward* [36]. Dalam penelitian ini, pendekatan *supervised learning* dipilih karena tersedia data historis lengkap beserta target *variable* (lama menginap). Data awal divalidasi, kemudian dilakukan *preprocessing* seperti penanganan *missing values*, *encoding categorical variables*, dan *feature scaling* sebelum dimasukkan ke dalam model regresi.

2.2.2 Feature Engineering

Feature engineering adalah proses transformasi data mentah menjadi fitur yang lebih informatif dan representatif untuk meningkatkan performa model *Machine Learning* [37]. Proses ini meliputi seleksi fitur, ekstraksi fitur, transformasi, dan pembuatan fitur baru yang dapat menangkap pola penting dalam data. *Feature engineering* sangat krusial karena kualitas fitur secara langsung mempengaruhi kemampuan model dalam belajar dan melakukan prediksi.

Penelitian terkini menekankan pentingnya automasi *feature engineering* dengan metode seperti *feature selection* berbasis statistik dan algoritma pembelajaran untuk mengurangi beban manual dan meningkatkan

efisiensi pengembangan model [38]. Dalam praktiknya, *feature engineering* dapat melibatkan teknik seperti *encoding* variabel kategorikal, normalisasi, pembuatan interaksi antar fitur, serta penggunaan *domain knowledge* untuk menambah fitur yang relevan.

2.2.3 Evaluasi Regresi

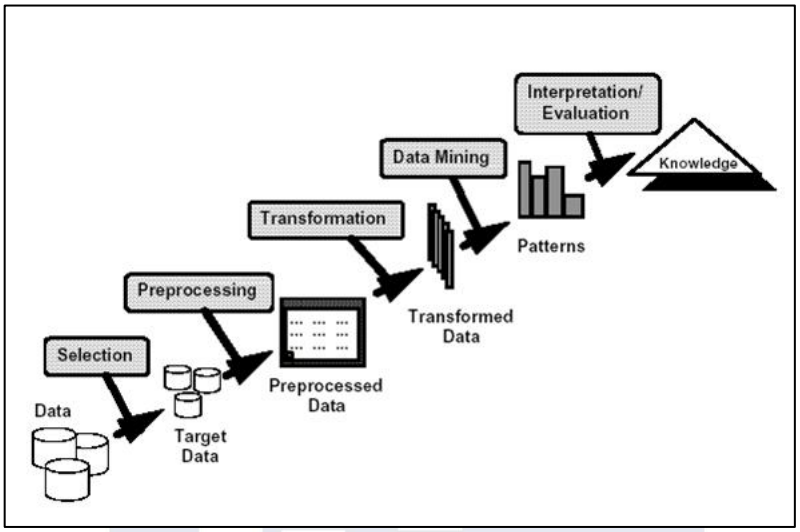
Evaluasi regresi adalah proses pengukuran kinerja model regresi dalam memprediksi nilai kontinu. Metrik evaluasi yang umum digunakan meliputi *Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE), *Mean Absolute Percent Error* (MAPE), dan *Coefficient of Determination* (R^2). RMSE mengukur akar kuadrat rata-rata dari selisih kuadrat antara nilai prediksi dan aktual, memberikan penalti lebih besar pada *error* besar. MAE menghitung rata-rata nilai mutlak selisih, memberikan gambaran *error* rata-rata yang lebih stabil terhadap *outlier*. R^2 mengukur proporsi variansi data yang dapat dijelaskan oleh model, memberikan ukuran kecocokan model secara keseluruhan [39]. Pemilihan metrik evaluasi harus disesuaikan dengan tujuan dan karakteristik data. Kombinasi ketiga metrik ini memberikan gambaran komprehensif tentang akurasi, *robustness*, dan kemampuan generalisasi model. Selain itu, teknik validasi silang (*cross-validation*) dan uji statistik seperti *paired t-test* digunakan untuk memastikan keandalan dan signifikansi perbedaan performa antar model.

2.3 Framework dan Algoritma yang digunakan

2.3.1 Knowledge Discovery in Database (KDD)

Knowledge Discovery in Database (KDD) adalah proses yang digunakan untuk menemukan pola, informasi, dan pengetahuan yang bermanfaat dari kumpulan data besar. Tujuan utama KDD adalah mentransformasikan data mentah menjadi pengetahuan yang dapat digunakan untuk pengambilan Keputusan [40]. Proses ini terdiri dari beberapa tahapan yang sistematis, dimulai dari pemilihan data, *preprocessing*, transformasi, penambangan data (*data mining*), hingga interpretasi dan evaluasi hasil. KDD sangat penting dalam konteks analisis data besar dan *Machine Learning* karena

menyediakan kerangka kerja yang terstruktur guna mengolah dan mengekstrak informasi penting dari data yang tersebar dan kompleks.



Gambar 2. 1 Tahapan Proses Knowledge Discovery in Database (KDD)

Sumber: [41]

Gambar 2.1 menunjukkan tahapan-tahapan dalam metodologi *Knowledge Discovery in Database* (KDD) yang menggambarkan alur transformasi data dari bentuk awal hingga menghasilkan pengetahuan. Kerangka kerja KDD banyak digunakan dalam penelitian berbasis data *mining* dan *Machine Learning* karena kemampuannya dalam menyediakan struktur analisis yang jelas, sistematis, dan berorientasi pada eksplorasi pola dari data berskala besar. Dalam penelitian ini, *framework* KDD dipilih karena sesuai dengan karakteristik data transaksi hotel yang bersifat kompleks dan memiliki banyak variabel operasional. Penelitian diawali dengan tahap *business understanding* untuk memahami permasalahan operasional Hotel XYZ, proses pengolahan data selanjutnya mengikuti tahapan inti KDD sebagai kerangka teknis analisis data. Pada tahap akhir, hasil evaluasi difokuskan pada *business recommendation*, yaitu penerjemahan hasil analisis dan prediksi menjadi rekomendasi keputusan operasional yang relevan bagi manajemen hotel. Penerapan KDD dalam penelitian ini juga bertujuan untuk menghasilkan pengetahuan yang dapat mendukung pengambilan keputusan bisnis secara

terukur dan berbasis data, tanpa memasuki tahap implementasi sistem secara langsung. Berikut ini merupakan tahapan-tahapan dalam KDD:

1) *Selection*

Tahap *selection* merupakan proses awal dalam KDD yang berfokus pada pemilihan *subset* data yang relevan dari keseluruhan *dataset* yang tersedia. Pada tahap ini, data yang dianggap penting dan sesuai dengan tujuan analisis dipilih untuk diproses lebih lanjut. Pemilihan data ini sangat krusial karena akan menentukan kualitas dan fokus dari proses penemuan pengetahuan selanjutnya. Data yang tidak relevan atau berlebihan dapat diabaikan agar proses analisis menjadi lebih efisien dan hasilnya lebih akurat.

2) *Pre-processing*

Selanjutnya terdapat tahapan *preprocessing*, yang dimana tahap ini adalah tahap diamana data di lakukan persiapan dengan eksplorasi data, penegcekan tipe data dan permbersiahan data yang bertujuan menghilangkan berbagai masalah seperti *noise*, nilai yang hilang, duplikasi, dan inkonsistensi dalam data. Proses ini memastikan bahwa data yang digunakan dalam analisis sudah dalam kondisi yang bersih dan konsisten sehingga tidak mengganggu hasil pemodelan. Selain itu, *preprocessing* juga dapat meliputi pengisian nilai yang hilang dan penghapusan data yang tidak valid, sehingga meningkatkan kualitas data secara keseluruhan.

3) *Transformation*

Pada tahap ini, penelitian ini melakukan data dinormalisasi untuk menyamakan skala variabel numerik, dilakukan rekayasa/pemilihan fitur, dan integrasi data sehingga dataset siap untuk pemodelan. Transformasi ini bertujuan mencegah dominasi fitur berkisar besar dan memperkenalkan variabel turunan yang menangkap dinamika hotel lebih baik.

4) *Data Mining*

Tahap ini dilakukannya pemodelan dengan membagi data menjadi set latih dan uji lalu menerapkan dua algoritma: *Random Forest* dan *Gradient Boosting*, dengan konfigurasi *hyperparameter* terkait yang

dikontrol di *Python*. Tujuan tahap ini adalah membangun model regresi untuk memprediksi LoS dan membandingkan performa kedua pendekatan model *Machine Learning*.

5) *Evaluation*

Pada tahap akhir, model dievaluasi menggunakan metrik regresi RMSE, MAE, MAPE, dan R^2 untuk menilai akurasi dan kemampuan penjelasan varians, selain metrik numerik, dilakukan analisis *feature importance* untuk mengidentifikasi variabel paling berpengaruh terhadap lama menginap tamu, lalu hasil teknis diinterpretasikan menjadi rekomendasi bisnis.

2.3.2 Algoritma Random Forest

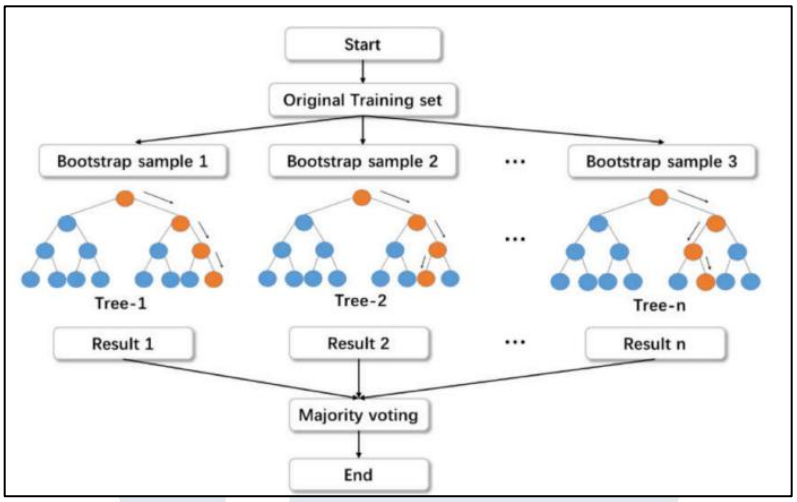
Random Forest adalah salah satu metode yang terdiri dari kumpulan *decision trees* yang dibangun secara acak (*bootstrap sampling*) dan independen, kemudian hasilnya digabungkan (*majority voting)* untuk klasifikasi atau *averaging* untuk regresi) untuk menghasilkan prediksi akhir yang lebih akurat dan stabil. Algoritma ini dikenal karena kemampuannya dalam menangani data dengan dimensi tinggi dan kompleksitas *non-linear*, serta ketahanannya terhadap *overfitting* [42]. Secara umum, *Random Forest* terdiri dari dua elemen utama: pembentukan pohon keputusan secara acak dan penggabungan hasilnya menjadi *output* akhir. Formulasi matematis dari *Random Forest* melibatkan proses *bootstrap sampling* yang menghasilkan *dataset subset* acak D_1, D_2, ..., D_n, masing-masing digunakan untuk membangun pohon keputusan independen. Prediksi akhir γ pada model regresi dihitung sebagai rata-rata prediksi dari *n* pohon keputusan:

$$\Upsilon = \frac{1}{n}\sum_{i=1}^{n} fi(x) \qquad (2.1)$$

Berdasarkan rumus *Random Forest* di (2.1) menjelaskan bahwa hasil prediksi akhir diperoleh dari rata-rata semua prediksi yang dihasilkan oleh setiap pohon keputusan. Setiap pohon dibangun dari *subset* data acak yang berbeda melalui proses *bootstrap*. Setelah semua pohon menghasilkan

prediksinya masing-masing, seluruh nilai prediksi dijumlahkan lalu dibagi dengan jumlah pohon untuk menghasilkan nilai akhir yang lebih stabil dan akurat dibandingkan prediksi dari satu pohon saja.

Berikut ini merupakan struktur pelatihan algoritma *Random Forest*:



Gambar 2. 2 Struktur Algoritma *Random Forest*

Sumber: [43]

Gambar 2.2 di atas memperlihatkan proses pelatihan *Random Forest* yang dimulai dengan *bootstrap sampling* data, membangun pohon keputusan pada setiap sampel, dan menggabungkan hasil prediksi dari seluruh pohon dengan *majority voting*. Struktur ini memungkinkan *Random Forest* mengurangi varians model sekaligus mempertahankan bias rendah karena kombinasi pohon yang bervariasi. Dalam aplikasi seperti prediksi lama menginap tamu hotel, *Random Forest* dapat menangkap pola yang kompleks dan beragam dengan stabilitas yang baik meski data berisik atau tidak seimbang.

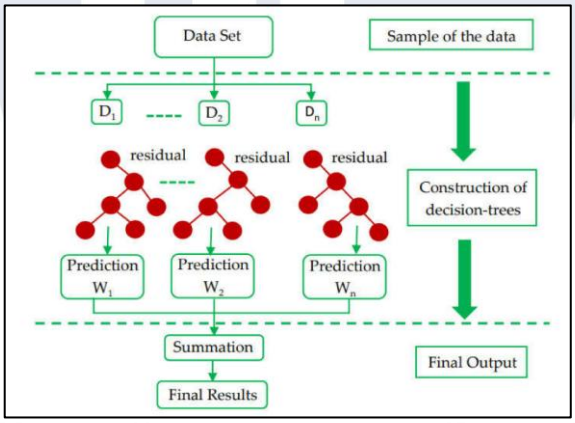
2.3.2 Algoritma Gradient Boosting

Gradient Boosting adalah teknik membangun model secara berurutan. Pada setiap iterasi, model baru ditujukan untuk memperbaiki kesalahan (*residual*) dari model sebelumnya dengan menggunakan pendekatan penurunan gradien pada fungsi *loss*. Metode ini menghasilkan model kombinasi yang kuat

dari sejumlah *weak learner* seperti pohon keputusan yang sederhana [44]. Rumus umum *Gradient Boosting* adalah:

$$\boldsymbol{F_m\ (x) = \ F_{m-1}(x) + \gamma_m h_m\ (x)} \qquad (2.2)$$

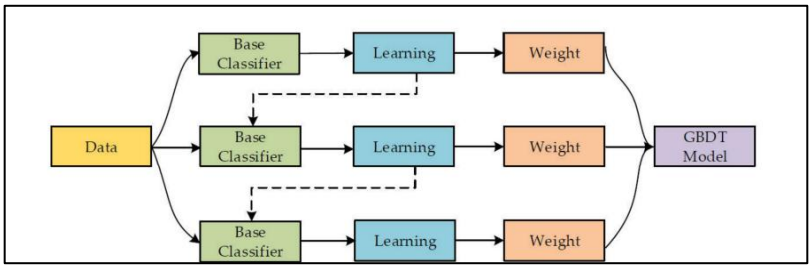
Rumus *Gradient Boosting* menunjukkan bahwa model pada iterasi ke-m merupakan hasil penambahan antara model sebelumnya dengan kontribusi model baru yang dikalikan oleh nilai *learning rate*. Model sebelumnya berfungsi sebagai dasar prediksi, sementara model baru (*weak learner*) berperan untuk memperbaiki kesalahan prediksi yang masih ada. *Learning rate* mengatur seberapa besar pengaruh model baru terhadap model akhir agar proses pembelajaran tetap stabil.



Gambar 2. 3 Struktur Algoritma *Gradient Boosting*

Sumber: [45]

Berdasarkan gambar 2.3 diatas, menggambarkan proses *algoritma Gradient Boosting* yang pembangunan model dilakukan secara berurutan berdasarkan *residual error*. Dataset awal dibagi menjadi beberapa *subset data* (D_1, D_2, ..., D_n), di mana setiap *subset* digunakan untuk membangun *decision tree* yang fokus pada memprediksi residual dari model sebelumnya. Setiap pohon menghasilkan prediksi (W_1, W_2, ..., W_n) yang kemudian digabungkan melalui proses *summation* untuk menghasilkan prediksi akhir (*final results*) [45]. Proses ini memungkinkan algoritma untuk secara progresif memperbaiki kesalahan prediksi dari model sebelumnya, sehingga menghasilkan model yang lebih akurat dan *robust* dalam menangkap pola kompleks dalam data.



Gambar 2. 4 Proses Training *Gradient Boosting*

Sumber: [46]

Selanjutnya, berdasarkan gambar 2.4 diatas mengilustrasikan terkait alur proses *training* pada algoritma *Gradient Boosting*, yang menunjukkan bagaimana data *input* diproses melalui serangkaian *base classifier* secara berurutan. Setiap *base classifier* menjalani tahap *learning* dan menghasilkan *weight* (bobot) yang menentukan kontribusinya terhadap model gabungan akhir. Garis putus-putus menunjukkan adanya *feedback mechanism* di mana *output* dari setiap *classifier* mempengaruhi *training classifier* berikutnya, memungkinkan setiap model baru untuk fokus pada kesalahan yang dibuat oleh model sebelumnya. Proses ini berlanjut secara iteratif hingga mencapai konvergensi atau kriteria penghentian yang ditetapkan, menghasilkan GBDT (*Gradient Boosting Decision Tree*) Model yang merupakan agregasi dari seluruh *base classifier* dengan pembobotan yang optimal untuk meminimalkan fungsi *loss* keseluruhan.

2.3.3 Evaluasi Model Prediksi

Evaluasi model prediksi dalam *Machine Learning* adalah langkah penting untuk mengetahui seberapa baik sebuah model dapat memperkirakan hasil pada data yang belum pernah ditemui sebelumnya. Pada prinsipnya, proses evaluasi ini menggunakan berbagai metrik, seperti *Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE), *Mean Absolute Percent Error* (MAPE) dan *Coefficient of Determination* (R^2), yang semuanya memberikan sudut pandang berbeda mengenai kinerja model [47]. RMSE akan sangat menonjolkan *error* yang nilainya besar, MAE lebih menggambarkan rata-rata deviasi model terhadap data asli secara umum, dan R^2 memberikan gambaran umum tentang seberapa besar variasi data yang berhasil dijelaskan oleh model

[48]. Pada penelitian ini, proses evaluasi model prediksi akan menggunakan ketiga metrik tersebut agar hasil yang didapat benar-benar objektif, valid, dan mampu memberi gambaran menyeluruh tentang keandalan algoritma *Random Forest* maupun *Gradient Boosting*. Untuk memastikan evaluasi dilakukan secara objektif, sejumlah rumus statistik dipakai agar hasil pengujian model dapat dipertanggungjawabkan. Berikut ini merupakan ketiga rumus berdasarkan RMSE, MAE, MAPE, dan R^2:

1) Rumus RMSE (*Root Mean Squared Error*):

$$\boldsymbol{RMSE} = \sqrt{\frac{\mathbf{1}}{\boldsymbol{N}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}}(\boldsymbol{yi} - \boldsymbol{\breve{y}i})^{\mathbf{2}}} \tag{2.3}$$

Rumus ini menghitung akar dari rata-rata kuadrat *error* antara nilai aktual (yi) dan prediksi model (ŷi), dengan *n* sebagai jumlah data, sehingga sangat peka terhadap *error* besar. Dengan tujuan rumus ini untuk menghitung akar dari rata-rata kuadrat *error* antara nilai aktual dan prediksi model, sehingga sangat peka terhadap *error* besar.

2) Rumus MAE (*Mean Absolute Error*):

$$\boldsymbol{MAE} = \frac{\mathbf{1}}{\boldsymbol{N}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}}(\boldsymbol{yi} - \boldsymbol{\breve{y}\,i}) \tag{2.4}$$

MAE menghitung nilai rata-rata dari selisih *absolut* antara nilai aktual (yi) dan nilai prediksi ($\breve{y}i$), serta n sebagai jumlah seluruh data, sehingga memberikan pengukuran *error* rata-rata yang stabil dan mudah diinterpretasikan. Dengan tujuan rumus MAE untuk menghitung rata-rata dari selisih *absolut* antara nilai aktual dan hasil prediksi.

3) Rumus MAPE (*Mean Absolute Percentage Error*):

$$\boldsymbol{MAPE} = \sum_{\boldsymbol{t=1}}^{\boldsymbol{n}}\left|\frac{\boldsymbol{yi} - \boldsymbol{\widehat{yi}}}{\boldsymbol{\widehat{yi}}}\right| \times \mathbf{100}\% \tag{2.5}$$

MAPE menghitung rata-rata persentase kesalahan *absolut* antara nilai aktual (y_i) dan nilai prediksi ($\hat{y}_i$), dengan n sebagai jumlah data. Rumus ini menampilkan seberapa besar kesalahan model dalam bentuk persentase terhadap nilai aktual. Tujuan dari rumus MAPE adalah untuk mengukur model dalam satuan persen.

4) Rumus R^2 (Koefisien Determinasi):

$$\boldsymbol{R = \frac{Covar\,(x,y)}{\sqrt{Var(x)Var(y)}}} \tag{2.6}$$

Rumus R^2 (Koefisien Determinasi) menunjukkan hubungan antara nilai prediksi (x) dan nilai aktual (y). Nilai kovarians antara keduanya, yaitu covar(x,y), dibandingkan dengan hasil akar dari perkalian varians masing-masing variabel, √Var(x)Var(y), untuk menghasilkan koefisien korelasi *Pearson* (R). Nilai R kemudian dikuadratkan menjadi R^2 untuk menunjukkan seberapa besar variasi nilai. Semakin tinggi nilai R^2, semakin baik model dalam pada hubungan antara variabel.

Berikut merupakan tabel kriteria evaluasi metrik regresi yang akan digunakan untuk menilai hasil model.

Tabel 2. 2 Kriteria Evaluasi Metrik Regresi

Metrik Evaluasi	Rentang Nilai	Kategori
RMSE	Semakin kecil semakin baik	Akurasi tinggi jika nilai mendekati 0
MAE	Semakin kecil semakin baik	*Error* rata-rata kecil menunjukkan prediksi stabil
MAPE	< 10%	Akurasi sangat baik
	10% – 20%	Akurasi baik
	20% – 50%	Akurasi moderat
	> 50%	Akurasi rendah / tidak dapat diandalkan
R^2	0.90 – 1.00	*Excellent fit*
	0.80 – 0.90	*Good fit*
	0.70 – 0.80	*Fair fit*
	0.50 – 0.70	*Poor fit*
	< 0.50	*Failure fit*

Tabel 2.2 menjelaskan kriteria penilaian untuk metrik regresi yang digunakan dalam evaluasi model. RMSE dan MAE menunjukkan tingkat kesalahan *absolut* antara nilai aktual dan prediksi, di mana nilai yang semakin kecil menandakan model bekerja dengan baik. MAPE memberikan gambaran persentase kesalahan, sehingga memudahkan interpretasi prediksi dalam konteks operasional; nilai MAPE di bawah 10% umumnya dianggap sangat baik. Sementara itu, R^2 digunakan untuk menilai seberapa besar variansi data yang dapat dijelaskan oleh model, dengan nilai mendekati 1 menunjukkan kecocokan model yang sangat baik.

2.4 Tools yang digunakan

2.4.1 Python

Python merupakan bahasa pemrograman tingkat tinggi yang berorientasi objek dan bersifat sumber terbuka, dikenal karena kemudahannya untuk dipahami serta fleksibilitasnya dalam berbagai bidang pengembangan [49]. Dengan manajemen memori otomatis dan sintaks yang sederhana, *python* memungkinkan pengembang untuk membangun berbagai jenis proyek mulai dari pengembangan situs, pengelolaan data, hingga pembuatan permainan. Selain itu, *python* juga memiliki ekosistem *library* yang kuat untuk berbagai bidang komputasi modern seperti *Big Data*, *Deep Learning*, *Data Mining*, *Machine Learning*, *Deep Visualization*, dan *Data Science* [50]. Keunggulan inilah yang menjadikan *python* sebagai salah satu bahasa pemrograman utama dalam pengembangan kecerdasan buatan dan aplikasi berbasis data.

2.4.1 Jupyter Notebook

Jupyter Notebook merupakan sebuah aplikasi berbasis *web* bersifat *open source* yang dirancang untuk memfasilitasi pembuatan serta penyebaran dokumen interaktif yang mengintegrasikan kode pemrograman, visualisasi data, perhitungan matematis, dan penjelasan naratif dalam satu kesatuan [51]. Aplikasi ini banyak dimanfaatkan dalam berbagai kegiatan analisis data seperti pembersihan dan transformasi data, simulasi numerik, pemodelan statistik, serta pengembangan model pembelajaran mesin [52]. Dan *Jupyter Notebook*

mendukung penggunaan berbagai pustaka dasar *python* seperti *numpy*, *pandas*, *matplotlib*, dan *seaborn* yang berfungsi untuk pengolahan data, analisis statistik, serta visualisasi informasi secara efisien.

2.4.1 MariaDB

MariaDB merupakan sistem manajemen basis data relasional bersifat *open-source* yang dikembangkan sebagai turunan (*fork*) dari *MySQL* dengan tujuan mempertahankan kompatibilitas sekaligus meningkatkan kinerja dan keandalannya [53]. Sistem ini mendukung berbagai *storage engine* seperti *Aria*, *InnoDB*, dan *ColumnStore* yang memungkinkan pengelolaan data berskala besar dengan efisien. *MariaDB* juga dilengkapi fitur *replication*, *clustering*, serta optimasi *query* yang mendukung kebutuhan analisis dan pengolahan data secara cepat dan stabil [54]. Keunggulan-keunggulan ini menjadikan *MariaDB* sebagai salah satu pilihan utama dalam pengembangan aplikasi riset dan sistem informasi modern.