Optimasi Prediksi Jumlah Kontainer Aktual di Kapal Menggunakan *Random Forest* dan *XGBoost* dengan *Hyperparameter Tuning*

Endi Permana*) dan Joko Susilo

Program Studi Sistem Informasi, Institut Bisnis dan Informatika Kwik Kian Gie, Jl. Yos Sudarso Kav 87, Sunter Jakarta 14350, Indonesia.
*) Surel korespondensi : endipermana68@gmail.com

Abstract. *The maritime logistics industry plays a crucial role in ensuring the smooth flow of trade; however, it often faces discrepancies between the number of containers booked and the number actually loaded onto ships. These discrepancies can lead to operational inefficiencies, shipment delays, and additional costs for companies. PT XYZ, as a maritime logistics service provider, encounters similar challenges. Therefore, this study aims to analyze the factors causing container discrepancies and to develop a predictive system for estimating the actual number of containers as a decision-support tool. This research adopts data mining, machine learning, and ensemble learning approaches, focusing on the Random Forest Regressor and Extreme Gradient Boosting (XGBoost) algorithms combined through a Voting Regressor. Hyperparameter tuning using GridSearchCV is applied to improve the model's ability to capture complex data patterns. A quantitative approach following the CRISP-DM framework is employed, including data exploration, cleaning, feature selection, modeling, and evaluation. The study utilizes historical container booking data from PT XYZ in 2023, consisting of more than 138,000 records. The results show that the Voting Regressor achieves the best performance with an R^2 value of 0.7874 and an MSE of 1.6282, supported by consistent RMSE and MAE metrics. The model is implemented in a Flask-based web application that enables practical container count prediction through Microsoft Excel file uploads. The implementation of this predictive system has the potential to help PT XYZ reduce loading discrepancies, minimize additional costs, and optimize logistics planning, while also contributing academically to the application of machine learning in the maritime logistics sector.*

Keywords: *container prediction, machine learning, random forest regressor, xgboost regressor, hyperparameter tuning, ensemble learning*

Diterbitkan oleh LPPM Institut Bisnis dan Informatika Kwik Kian Gie. Jl. Yos Sudarso Kav 87, Sunter Jakarta 14350, Indonesia.
DOI : https://doi.org/10.46806/jib.v14i2.1921

1. Pendahuluan

Perdagangan internasional melalui jalur maritim memiliki peran penting dalam pertumbuhan ekonomi Indonesia, karena memungkinkan pertukaran barang, jasa, dan sumber daya dalam skala besar. Efektivitas transportasi laut menjadi faktor utama kelancaran arus perdagangan, sementara perusahaan logistik maritim berperan penting dalam memastikan setiap pengiriman berjalan sesuai rencana. Sebagai negara maritim, Indonesia sangat bergantung pada transportasi laut yang

efisien, di mana kapal pengangkut berfungsi membawa barang dalam jumlah besar dengan waktu pengiriman yang tepat dan kondisi barang tetap aman.

Namun, sektor logistik maritim menghadapi tantangan signifikan, terutama terkait kesesuaian antara jumlah kontainer yang dipesan dan jumlah yang benar-benar dimuat di kapal. Ketidaksesuaian ini dapat menimbulkan masalah operasional seperti alih kapal, batal muat, dan penataan ulang kontainer di kapal (shifting in vessel), yang berpotensi menurunkan efisiensi operasional serta meningkatkan biaya perusahaan. PT. XYZ, sebagai salah satu perusahaan logistik pelayaran dengan volume pengiriman tinggi, sering mengalami permasalahan ini, sehingga dibutuhkan pemahaman lebih mendalam tentang faktor-faktor yang memengaruhi ketidaksesuaian kontainer untuk perbaikan manajemen operasional.

Penelitian ini berfokus pada analisis kesesuaian antara jumlah kontainer yang dipesan dengan jumlah aktual di kapal, pengembangan model prediksi berbasis data historis, dan penilaian kontribusi prediksi terhadap efisiensi operasional. Model prediksi menggunakan algoritma Machine Learning, khususnya Random Forest Regressor dan Extreme Gradient Boosting (XGBoost), yang dikombinasikan dengan hyperparameter tuning untuk meningkatkan akurasi. Pendekatan ini memungkinkan perusahaan memperkirakan jumlah kontainer secara lebih tepat, meminimalkan risiko ketidaksesuaian, dan mendukung perencanaan operasional yang lebih efektif. Implementasi penelitian ini tidak hanya memberikan manfaat praktis bagi perusahaan, tetapi juga berkontribusi secara akademis. Bagi PT. XYZ, prediksi akurat membantu mengoptimalkan kapasitas kapal, mengurangi biaya tambahan, dan meningkatkan ketepatan waktu pengiriman. Bagi penulis, penelitian ini menjadi kesempatan untuk menerapkan teknologi Machine Learning pada permasalahan nyata, memperluas pemahaman terhadap data operasional logistik, serta meningkatkan kemampuan analisis data. Selain itu, hasil penelitian ini dapat menjadi referensi bagi peneliti atau profesional lain yang ingin mengembangkan model prediksi dalam industri logistik maritim.

Manfaat penelitian juga dirasakan oleh pelanggan dan khalayak umum. Dengan prediksi berbasis data, pelanggan dapat menghindari risiko overstock atau overbooking, hanya membayar untuk kontainer yang benar-benar diperlukan, serta menikmati pengiriman yang lebih tepat waktu. Bagi industri logistik secara umum, penelitian ini menjadi panduan penerapan Machine Learning untuk meningkatkan efisiensi operasional, optimasi sumber daya, dan pengambilan keputusan berbasis data. Secara keseluruhan, penelitian ini menghadirkan solusi inovatif yang menggabungkan pendekatan praktis dan ilmiah dalam pengelolaan logistik maritim.

2. Tinjauan Pustaka

2.1. Algoritma

Menurut Meidyan Permata Putri dkk. (2022:6) Algoritma adalah kumpulan instruksi terkait untuk memecahkan masalah. Perintah dapat dikonversi langkah demi langkah

dari awal hingga akhir. Persiapan membutuhkan urutan dan logika agar algoritma yang dihasilkan sesuai dengan yang diharapkan. Ada tiga aspek yang perlu dipertimbangkan saat menulis program: sintaks, semantik, dan kebenaran logika.

2.2. Kontainer

Menurut Ginting (2021), peti kemas atau container adalah kemasan khusus dengan ukuran standar yang dirancang untuk digunakan berulang kali, serta berfungsi untuk menyimpan dan mengangkut berbagai jenis muatan.

2.3. Data

Menurut Laudon dan Laudon (2020:16) data merupakan aliran fakta mentah yang mencerminkan peristiwa yang terjadi dalam organisasi atau lingkungan fisik, sebelum fakta-fakta tersebut diorganisasikan dan disusun menjadi bentuk yang dapat dipahami dan digunakan oleh manusia.

2.4. Basis Data

Menurut Putri Ariatna Alia et al. (2023:1) Basis data adalah pusat pengelolaan informasi dalam bentuk terstruktur di dalam suatu sistem komputer. Dalam dunia digital yang terus berkembang, basis data adalah fondasi dari hampir semua aplikasi dan sistem informasi. Basis data (database) adalah kumpulan data yang terorganisir dan disimpan secara sistematis di dalam suatu komputer. Data-data ini disimpan menggunakan struktur khusus agar mudah diakses, dimodifikasi, dan dikelola. Basis data memiliki peran penting dalam menyimpan informasi yang diperlukan oleh aplikasi perangkat lunak, sistem bisnis, dan organisasi.

2.5. Sistem Informasi

Laudon dan Laudon (2020:16) menyatakan bahwa sistem informasi adalah kumpulan komponen yang saling terkait yang berfungsi mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi untuk mendukung pengambilan keputusan dan pengendalian dalam organisasi. Selain itu, sistem informasi juga dapat membantu manajer dan pekerja dalam menganalisis masalah, memvisualisasikan hal-hal yang kompleks, serta menciptakan produk baru.

2.6. Data Mining

Menurut Muslim et al. (2019:2), *data mining* merupakan proses ekstraksi informasi penting yang bersifat implisit dan belum diketahui, serta berkaitan dengan bidang seperti statistik, *machine learning*, pengenalan pola, algoritma komputasi, teknologi basis data, dan komputasi performa tinggi. Proses utama *data mining* terdiri dari tiga tahap, yaitu pertama, eksplorasi atau pemrosesan awal data, yang mencakup pembersihan, normalisasi, transformasi, penanganan nilai hilang, reduksi dimensi, dan pemilihan subset *fitur*. Kedua, pembangunan dan validasi model, yakni menganalisis berbagai model untuk memilih yang memberikan kinerja terbaik melalui metode seperti klasifikasi, regresi, *klaster*, dan asosiasi. Ketiga, penerapan model pada data baru untuk menghasilkan prediksi yang efektif pada masalah yang dianalisis.

2.7. Machine Learning

Menurut Wolf (2022:6), *Machine Learning* (ML) adalah proses belajar melakukan suatu tugas dari data, di mana pembelajaran ini dilakukan dengan mengenali dan mengekstraksi pola dari data secara tepat, bukan melalui cara ajaib atau misterius seperti yang sering digambarkan dalam fiksi ilmiah.

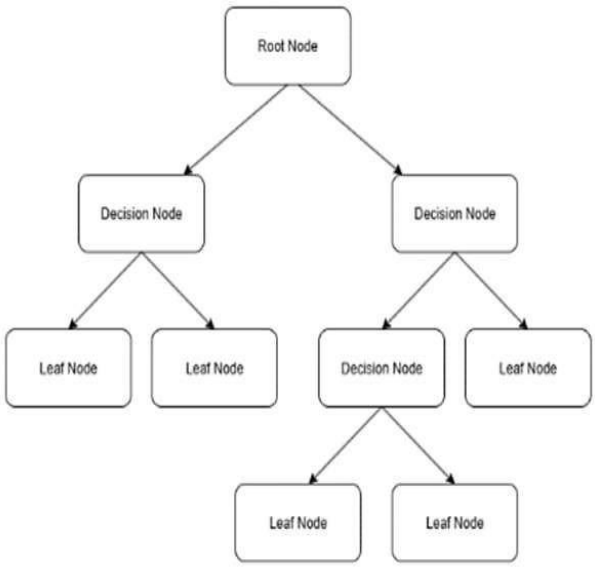
2.8. Bootstrap Aggregating (Bagging)

Menurut Jatmiko et al. (2019), *Bootstrap Aggregating* atau *bagging* adalah metode *ensemble* yang menggabungkan banyak model untuk meningkatkan akurasi prediksi. *Bagging* bekerja dengan melakukan *resampling* acak secara pengembalian dari data asli untuk membentuk himpunan data baru, yang kemudian digunakan untuk membangun sejumlah pohon klasifikasi atau regresi sesuai kebutuhan peneliti.

2.9. Decision Tree

Menurut Sinambela et al. (2023), dalam *data mining* salah satu metode yang umum digunakan adalah sistem klasifikasi, yang berfungsi memetakan data ke kategori tertentu berdasarkan pola dari data historis. Algoritma klasifikasi tidak hanya mampu menangani volume data yang besar dan membuat prediksi kelas, tetapi juga membentuk struktur pengetahuan dari data latih berlabel.

Fokus penelitian ini adalah algoritma pohon keputusan (*decision tree*), yang dapat divisualisasikan secara hierarkis, di mana simpul paling atas disebut *root node* dan simpul di bawahnya disebut *node*, yang masing-masing mewakili pengambilan keputusan melalui pengujian variabel tertentu.



Gambar 1. Proses *Decision Tree*

2.10. Decision Tree Regressor/Regression

Menurut Mahamat et al. (2024), regresi pohon keputusan (*Decision Tree Regression*/DTR) merupakan metode *supervised learning* yang dapat menangani masalah regresi maupun klasifikasi. Metode ini membangun model dalam bentuk

struktur pohon untuk memprediksi variabel target melalui pembelahan data berdasarkan *fitur* tertentu. Kinerja model dievaluasi menggunakan metrik kuantitatif seperti *mean squared error* (MSE), *root mean squared error* (RMSE), dan koefisien determinasi (R^2), yang menunjukkan tingkat kesalahan prediksi dan kemampuan model dalam menjelaskan variasi data.

2.11. Random Forest Regressor/Regression

Menurut Saadah dan Salsabila (2021), *Random Forest Regression* adalah algoritma *Machine Learning* yang termasuk ke dalam *supervised learning*. Algoritma ini bekerja dengan menggabungkan prediksi dari berbagai pohon keputusan (*decision tree*) untuk menghasilkan prediksi yang lebih akurat dan konsisten dibandingkan penggunaan satu pohon keputusan saja.

2.12. Extreme Gradient Boosting (XGBoost)

Menurut Salsabil et al. (2024), XGBoost (*Extreme Gradient Boosting*) adalah metode *Machine Learning* yang efektif untuk klasifikasi dan regresi. Metode ini menggunakan *ensemble learning* dengan membangun model secara bertahap melalui penambahan pohon keputusan secara berurutan, di mana setiap model berfokus memperbaiki kesalahan prediksi model sebelumnya. XGBoost juga mengoptimalkan fungsi objektif, menerapkan *regularisasi* untuk mengendalikan kompleksitas model, dan dapat mengevaluasi pentingnya setiap *fitur* dalam prediksi. Karena kinerjanya yang tinggi dan fleksibilitasnya, XGBoost banyak digunakan dalam berbagai kompetisi data dan aplikasi industri.

2.13. Ensemble Learning

Menurut Sunarko et al. (2023), *ensemble learning* adalah metode dalam pembelajaran mesin yang menggabungkan beberapa model dengan tujuan meningkatkan konsistensi dan akurasi prediksi.

2.14. Hyperparameter

Banerjee et al. (2019) menyatakan bahwa *hyperparameter* adalah parameter yang tetap selama pelatihan model pembelajaran mesin. Parameter ini memengaruhi arsitektur model, seperti jumlah lapisan tersembunyi dan fungsi aktivasi, serta kinerja dan akurasi pelatihan, termasuk *learning rate*, ukuran *batch*, dan jenis *optimizer* yang digunakan.

2.15. Hyperparameter Tuning

Menurut Pradhan dan Kumar (2019:222), proses untuk menemukan nilai optimal pada *hyperparameter* disebut penyetelan *hyperparameter* (*hyperparameter tuning*).

2.16. GridSearchCV

Menurut Siji George dan Sumathi (2020), *GridSearchCV* adalah kelas dalam pustaka Scikit-Learn yang digunakan untuk mencari kombinasi *hyperparameter* terbaik pada suatu model. Metode ini mengevaluasi seluruh kemungkinan kombinasi

nilai *hyperparameter* dan memilih kombinasi yang memberikan performa terbaik berdasarkan metrik evaluasi yang ditentukan.

2.17. Web Framework Flask

Menurut Ningtyas dan Setiyawati (2021), Flask adalah *web framework* Python yang tergolong *micro-framework* karena tidak memerlukan pustaka tambahan atau *database* bawaan. Flask menggunakan *Jinja Template Engine* dan Werkzeug *WSGI Toolkit*, dengan struktur yang terbagi menjadi dokumen statis (seperti CSS, JavaScript, dan gambar) dan *template* dokumen (halaman HTML menggunakan Jinja). Dalam pengembangan aplikasi dengan Flask, *virtual environment* diperlukan untuk mengelola pustaka yang digunakan.

3. Metode

Penelitian ini memusatkan perhatian pada operasional logistik PT. XYZ, salah satu perusahaan besar di sektor pelayaran Indonesia, khususnya cabang Jakarta yang memainkan peran vital dalam arus pengiriman kontainer domestik dan internasional. Perusahaan ini tidak hanya menyediakan layanan pengiriman barang melalui jalur laut, tetapi juga solusi logistik lain seperti pergudangan dan distribusi darat, yang didukung sistem informasi manajemen modern untuk pemantauan *real-time*. Meskipun demikian, cabang Jakarta menghadapi berbagai tantangan, termasuk manajemen arus kontainer yang padat, risiko penundaan pengiriman, serta regulasi dan fluktuasi ekonomi.

Sistem pemesanan kontainer saat ini menghadapi kendala signifikan terkait efisiensi operasional, terutama karena informasi pemesanan yang diterima sering tidak lengkap atau terlambat. Hal ini menyebabkan perbedaan antara jumlah kontainer yang dipesan dengan yang sebenarnya dikirim, munculnya pemesanan berlebihan, dan pengiriman yang tidak sesuai pesanan. Dampaknya meliputi inefisiensi ruang kargo, ketidakpastian dalam perencanaan logistik, dan kerugian finansial akibat kontainer tidak terpakai atau perlu penjadwalan ulang pengiriman.

Alur proses pemesanan kontainer melibatkan empat entitas utama, yaitu pelanggan, sistem, *sales*, dan *admin*. Proses dimulai dengan pemesanan kontainer, baik secara *online* melalui sistem yang menghasilkan *Request Order* (RO) atau secara langsung melalui tim *sales*. Selanjutnya, *admin* mengevaluasi dan menyetujui pesanan, setelah itu pelanggan mengambil kontainer dan membeli segel sebagai bagian dari keamanan pengiriman.

Tahap berikutnya adalah pengisian data *Shipping Instructions* (SI) oleh pelanggan sebagai panduan pengiriman, yang kemudian diperiksa dan divalidasi oleh pihak terkait. Setelah SI disetujui, dokumen resmi diterbitkan dan menjadi bukti bahwa kontainer siap dikirim sesuai instruksi. Dengan memetakan alur pemesanan ini, penelitian bertujuan untuk menganalisis masalah yang muncul dan mengidentifikasi peluang penggunaan teknologi prediktif untuk meningkatkan efisiensi operasional,

mengurangi risiko ketidaksesuaian pengiriman, dan mendukung kualitas layanan di PT. XYZ cabang Jakarta.

3.1. Teknik Pengumpulan Data

Penelitian ini menggunakan data primer dari PT. XYZ untuk periode tahun 2023 dan menerapkan metode kuantitatif untuk memprediksi jumlah kontainer aktual. Dua algoritma *Machine Learning* yang digunakan adalah *Random Forest Regressor* dan *Extreme Gradient Boosting*. Variabel independen mencakup informasi mengenai pelanggan, kapal, komoditas, dan *grade* kontainer, sedangkan variabel dependen adalah jumlah kontainer aktual di kapal.

Untuk meningkatkan akurasi prediksi, digunakan teknik *hyperparameter tuning* pada model yang dibangun.

3.2. Teknik Analisis Data

Penelitian ini menggunakan pendekatan CRISP-DM (*Cross-Industry Standard Process for Data Mining*) sebagai kerangka kerja analisis data. CRISP-DM terdiri dari enam tahap utama, dimulai dengan *business understanding*, yaitu identifikasi tujuan bisnis, yaitu membangun model regresi untuk memprediksi jumlah kontainer aktual berdasarkan data pemesanan dan karakteristik pelanggan. Tahap *data understanding* mencakup pengumpulan, eksplorasi, dan identifikasi masalah pada data seperti *missing value* atau *outlier* melalui statistik deskriptif dan visualisasi.

Selanjutnya, pada tahap *data preparation*, dilakukan prapemrosesan data agar siap digunakan dalam pemodelan. Langkah-langkahnya meliputi imputasi data hilang, seleksi *fitur* relevan, konversi variabel kategorikal menggunakan *one-hot encoding*, serta penskalaan *fitur* numerik dengan *StandardScaler* agar distribusi data memiliki rata-rata 0 dan standar deviasi 1. Data kemudian dibagi menjadi data latih dan data uji dengan rasio 70:30 untuk memastikan evaluasi model yang akurat.

Pada tahap *modeling*, dibangun model prediksi menggunakan *Random Forest Regressor* dan *XGBoost Regressor*. Proses *hyperparameter tuning* dilakukan dengan *GridSearchCV* untuk menemukan kombinasi parameter terbaik, seperti jumlah pohon (*n_estimators*) dan kedalaman maksimum pohon (*max_depth*), dengan *5-fold cross-validation* untuk memastikan generalisasi model. Setelah parameter optimal ditemukan, model dilatih kembali pada data latih untuk menghasilkan prediksi yang optimal.

Tahap *evaluation* menggunakan metrik kuantitatif seperti *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE), dan koefisien determinasi (R^2) untuk mengukur kinerja model. Metrik ini memberikan gambaran mengenai kesalahan prediksi dan kemampuan model dalam menjelaskan variasi data, sehingga memungkinkan penilaian performa secara objektif.

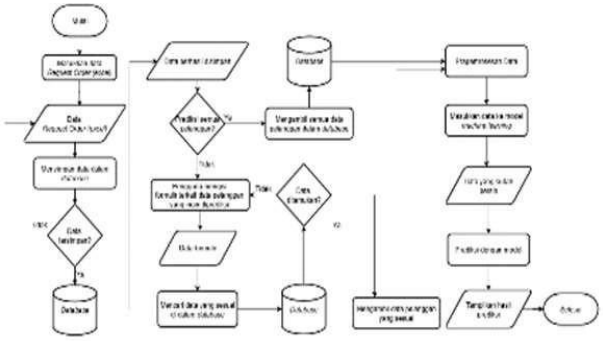
Tahap terakhir adalah *deployment*, di mana model yang telah dilatih dan dievaluasi diimplementasikan dalam bentuk aplikasi sederhana berbasis Python dan Flask. Sistem ini memungkinkan pengguna memasukkan data dan memperoleh prediksi

jumlah kontainer secara langsung, sehingga penelitian tidak hanya bersifat teoritis tetapi juga memiliki manfaat praktis untuk operasional perusahaan.

4. Hasil dan Pembahasan

4.1. Sistem yang Diusulkan

Sistem prediksi jumlah kontainer aktual yang diusulkan dalam penelitian ini dirancang sebagai *support system* untuk membantu pengambilan keputusan yang lebih cepat dan akurat dalam proses pengiriman kontainer, tanpa menggantikan sistem operasional yang sudah ada. Sistem ini memanfaatkan data historis pengiriman kontainer untuk memprediksi jumlah kontainer yang akan dipesan dan dikirim ke kapal. Dengan adanya prediksi ini, tim operasional maupun tim penjualan dapat mengantisipasi kebutuhan logistik, meminimalkan risiko keterlambatan pengiriman, serta menghindari kekurangan kapasitas, sehingga memberikan nilai tambah berupa informasi prediktif yang relevan.



Gambar 2. Diagram Alur Sistem Prediksi Aktual Kontainer

Alur kerja sistem prediksi jumlah kontainer aktual yang diusulkan dalam penelitian ini digambarkan pada Gambar 2. Proses dimulai saat pengguna mengunggah data *Request Order* dalam format Microsoft Excel, yang kemudian disimpan ke dalam *database*. Pengguna dapat memilih prediksi untuk semua pelanggan atau pelanggan tertentu. Untuk pelanggan tertentu, sistem mencari data yang sesuai, lalu melakukan prapemrosesan seperti pemilihan *fitur*, *encoding* variabel kategorikal, dan penskalaan data agar sesuai dengan input model *machine learning*. Selanjutnya, data diprediksi menggunakan model yang telah dilatih, dan hasilnya ditampilkan kepada pengguna dalam bentuk yang mudah dipahami, menyelesaikan alur kerja sistem.

4.2. Rancangan Basis Data

Penelitian ini menggunakan MySQL sebagai *Database Management System* (DBMS) karena kestabilannya, kemudahan penggunaan, dan kemampuannya dalam menangani kueri berskala besar. Basis data untuk sistem prediksi jumlah kontainer dirancang dengan pendekatan sederhana menggunakan satu tabel utama sebagai penyimpanan data. Tabel ini menyimpan data dari dokumen Microsoft Excel yang diunggah pengguna, berisi informasi terkait permintaan prediksi jumlah kontainer. Setelah data tersimpan, sistem akan mengirimkan entri tersebut ke model *machine*

learning untuk melakukan prediksi. Setiap baris dalam tabel merepresentasikan permintaan pengguna, dengan atribut utama seperti yang ditunjukkan pada Tabel 1.

Tabel 1. Atribut Entitas *Request Order* (RO)

Nama Kolom	Tipe Data	Deskripsi
ID	Integer (11)	*Primary key* yang digunakan untuk mengidentifikasi setiap *record* secara unik.
MONTH	Integer (2)	Bulan terkait dengan data ini, disimpan sebagai angka (1-12).
MASKED_NAME	Varchar (10)	Nama pelanggan yang telah disamarkan atau diubah untuk menjaga privasi.
CUSTOMER_ SEGMENTATION	Varchar (25)	Kategori atau segmentasi pelanggan berdasarkan bisnis atau industri mereka.
VESSELID	Varchar (25)	ID kapal yang digunakan untuk pengiriman atau transportasi barang.
POD	Varchar (10)	Pelabuhan tujuan tempat barang akan dibongkar.
CONTAINERTYPE	Integer (2)	Jenis kontainer yang digunakan, direpresentasikan dalam bentuk kode numerik.
COMM_GRADE_RO	Varchar (25)	Kategori dan *grade* dari komoditas yang sedang dikirim.
QTY_RO	Integer (11)	Jumlah kontainer yang dipesan.
USER_ID	Varchar (255)	ID pengguna yang terkait dengan transaksi atau data ini.

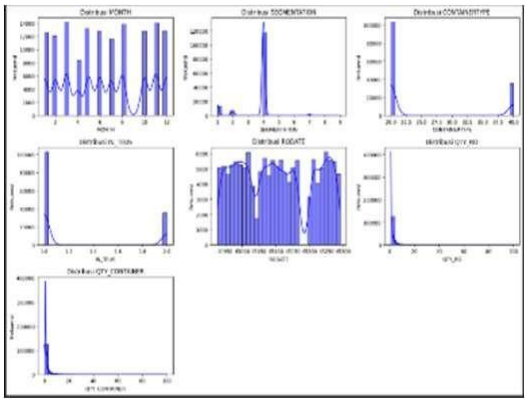
4.3. Deskripsi Dataset

Dataset yang digunakan berasal dari data historis permintaan pesanan (*Request Order*/RO) PT. XYZ pada tahun 2023, yang terdiri dari 138.215 baris dan 17 kolom. Dataset ini memuat berbagai informasi terkait pemesanan kontainer, memberikan gambaran pola permintaan sepanjang periode tersebut. Ukuran dan kompleksitas dataset memungkinkan analisis mendalam terhadap hubungan antar *fitur*, sehingga mendukung pembangunan model *machine learning* yang mampu menghasilkan prediksi jumlah kontainer yang akurat.

4.4. Exploratory Data Analysis (EDA)

4.4.1. Visualisasi Data

Visualisasi data dilakukan untuk memberikan gambaran mengenai pola dan distribusi *fitur* dalam dataset yang akan dianalisis.



Gambar 3. Visualisasi Distribusi Data untuk Fitur Numerik

Gambar 3 menampilkan distribusi beberapa *fitur* numerik, yang digunakan untuk memahami sebaran data dan mendeteksi potensi ketidakseimbangan maupun *outlier*. Distribusi *fitur* MONTH menunjukkan penyebaran data yang relatif merata sepanjang tahun. *Fitur* SEGMENTATION memperlihatkan dominasi kategori tertentu, sedangkan CONTAINERTYPE menunjukkan beberapa jenis kontainer yang lebih sering digunakan. Distribusi IN_TEUS menyoroti bahwa mayoritas data terkonsentrasi pada nilai rendah dengan beberapa nilai ekstrem. Sementara itu, QTY_CONTAINER dan QTY_RO menunjukkan bahwa sebagian besar data memiliki jumlah kecil, meskipun terdapat beberapa nilai tinggi yang dapat dikategorikan sebagai *outlier*.

4.4.2. Statistik Deskriptif

Analisis statistik deskriptif dilakukan untuk memahami karakteristik dataset sebelum dilakukan pemodelan. Statistik ini merangkum informasi dasar mengenai distribusi data, baik untuk *fitur* numerik maupun kategorikal.

a. Fitur Numerik

Fitur numerik dianalisis menggunakan ukuran pemusatan dan penyebaran data, seperti rata-rata (*mean*), median, standar deviasi, nilai minimum, maksimum, dan kuartil.

- MONTH memiliki rata-rata 6,39 dengan standar deviasi 3,55, mencakup seluruh bulan dari 1 hingga 12.
- SEGMENTATION memiliki rata-rata 3,66 dan standar deviasi 1,00, menunjukkan variasi segmentasi pelanggan antara 1 hingga 9.
- CONTAINERTYPE memiliki rata-rata 25,12 dan standar deviasi 8,73, dengan nilai minimum 20 dan maksimum 40, menandakan beberapa jenis kontainer berbeda berdasarkan kode numerik.
- IN_TEUS menunjukkan mayoritas pengiriman menggunakan satuan TEUs kecil, dengan rata-rata 1,26 dan standar deviasi 0,44.
- QTY_RO memiliki rata-rata 2,07 dan standar deviasi 3,01, dengan nilai maksimum 100, menandakan variasi jumlah pesanan yang signifikan.
- QTY_CONTAINER memiliki rata-rata 1,63 dan standar deviasi 2,83, dengan nilai maksimum 100, menunjukkan beberapa pengiriman kontainer yang sangat besar meskipun mayoritas data berkisar di median 1.

b. Fitur Kategorikal

Fitur kategorikal dianalisis untuk memahami distribusi kategori dalam dataset.

- MASKED_NAME (identitas pelanggan yang disamarkan) memiliki 413 nilai unik, dengan ID_87 paling sering muncul sebanyak 6.517 kali, menunjukkan dominasi beberapa pelanggan tertentu.

- CUSTOMER_SEGMENTATION memiliki 6 kategori, dengan FORWARDER paling sering muncul (117.725 kali), menunjukkan mayoritas pelanggan merupakan perusahaan ekspedisi.
- VESSELID (identitas kapal) memiliki 50 nilai unik, di mana kapal RAH paling sering muncul (7.160 kali).
- POD (*Port of Discharge*) memiliki 37 nilai unik, dengan IDPNK sebagai pelabuhan tujuan dominan (15.546 kali).
- ABBREVIATION memiliki 9 kode singkatan, DC paling sering muncul (99.957 kali), kemungkinan merepresentasikan tipe layanan tertentu.
- OWNER memiliki 2 nilai unik, dengan COC (*Carrier Owned Container*) paling banyak (128.211 kali), menandakan sebagian besar kontainer dimiliki oleh pihak pelayaran.
- COMM_GRADE_RO (jenis komoditas) memiliki 123 nilai unik, dengan kategori MAKANAN paling sering muncul (34.841 kali), menunjukkan makanan sebagai muatan paling umum.

4.5. Prapemrosesan Data

Pada tahap prapemrosesan data, penulis menggunakan Python dengan pustaka populer seperti *pandas*, *numpy*, dan *scikit-learn* untuk membersihkan dan menyiapkan data sebelum digunakan dalam model *machine learning*.

4.5.1. Imputasi Data Hilang

Langkah pertama adalah menangani data hilang. Penulis memeriksa setiap kolom pada dataset menggunakan kode df.isnull().sum(). Hasil pengecekan menunjukkan bahwa seluruh kolom sudah lengkap dan tidak terdapat nilai hilang, sehingga tidak diperlukan langkah imputasi atau penghapusan data.

4.5.2. Pemilihan Fitur

Fitur yang relevan untuk prediksi dipilih sebagai variabel independen (X) dan target sebagai variabel dependen (y). Setiap *fitur* memiliki peranan penting, antara lain:

- MASKED_NAME: identifikasi pelanggan.
- CONTAINERTYPE: jenis kontainer yang mempengaruhi kapasitas pengiriman.
- CUSTOMER_SEGMENTATION: segmentasi pelanggan yang mempengaruhi pola permintaan.
- COMM_GRADE_RO dan POD: memberikan informasi terkait *grade* kontainer dan tujuan pengiriman.
- VESSELID: kapal yang digunakan dalam pengiriman.
- QTY_RO: jumlah pesanan yang diterima.

4.5.3. Pembersihan dan Transformasi Data

Kolom COMM_GRADE_RO dibersihkan dari angka menggunakan ekspresi reguler agar hanya tersisa teks. Selanjutnya, semua *fitur* kategorikal diubah menjadi representasi numerik menggunakan *one-hot encoding* dengan kode pd.get_dummies(). Variabel numerik diskalakan menggunakan *StandardScaler* agar semua *fitur* berada pada skala yang sama, mendukung kinerja model *machine learning*.

4.5.4. Pembagian Data (Train-Test Split)

Dataset dibagi menjadi data latih dan data uji menggunakan fungsi train_test_split dari *scikit-learn* dengan rasio 60% untuk pelatihan dan 40% untuk pengujian (test_size=0.4), serta random_state=42 untuk memastikan reproduktibilitas. Hasil pembagian ini menghasilkan empat subset: X_train, X_test, y_train, dan y_test, yang digunakan untuk melatih dan mengevaluasi model prediksi.

4.6. Pembangunan Model

Tahapan pembangunan model diawali dengan pencarian kombinasi parameter terbaik (*hyperparameter tuning*) untuk meningkatkan performa prediksi. Penelitian ini menggunakan dua model regresi *ensemble*, yaitu *Random Forest Regressor* dan *XGBoost Regressor*.

Untuk memastikan model bekerja secara optimal, dilakukan *GridSearchCV*, yaitu metode yang menguji berbagai kombinasi parameter secara sistematis dengan validasi silang (*cross-validation*). *GridSearchCV* tidak hanya membantu menemukan kombinasi parameter terbaik, tetapi juga mengurangi risiko *overfitting* dengan memastikan model memiliki performa yang konsisten pada berbagai subset data pelatihan.

Parameter utama yang diuji meliputi jumlah pohon dalam model *ensemble* (n_estimators) dan kedalaman maksimum dari setiap pohon keputusan (max_depth). *Random Forest* diuji dengan beberapa pilihan jumlah pohon dan kedalaman, sedangkan XGBoost menggunakan kombinasi parameter yang lebih ringan karena model ini lebih sensitif terhadap *overfitting* bila kedalaman pohon terlalu besar. Hasil *tuning* menunjukkan parameter terbaik sebagai berikut:

Tabel 2. Hasil *Hyperparameter Tuning*

Models	**Best Score**	**Best Parameters**
Random Forest	0.812234	{'max_depth':10, 'n_estimators': 30}
XGBoost	0.810942	{'max_depth':3, 'n_estimators': 30}

Tabel 2 memperlihatkan hasil *hyperparameter tuning* untuk dua model regresi yang digunakan, yaitu *Random Forest Regressor* dan *XGBoost Regressor*, dengan teknik *GridSearchCV* untuk menemukan kombinasi kedalaman maksimum pohon (max_depth) dan jumlah pohon dalam *ensemble* (n_estimators) yang memberikan performa terbaik berdasarkan *cross-validation score*.

Hasil *tuning* menunjukkan bahwa *Random Forest* mencapai skor tertinggi sebesar 0,812234 dengan max_depth 10 dan n_estimators 30, menunjukkan model mampu

menangkap kompleksitas data secara optimal dengan kedalaman pohon sedang dan jumlah *estimator* yang efisien, sedangkan XGBoost memperoleh skor 0,810942 dengan max_depth 3 dan n_estimators 30, yang menandakan model bekerja optimal dengan pohon lebih dangkal sehingga lebih tahan terhadap *overfitting* dan tetap menjaga generalisasi.

Secara keseluruhan, kedua model menunjukkan performa yang kompetitif dengan selisih skor sangat tipis, sehingga kedua model digunakan sebagai pendekatan *ensemble* untuk meningkatkan akurasi prediksi, memberikan perspektif yang lebih komprehensif terhadap data, dan memperkuat potensi pemodelan yang berbeda.

Setelah diperoleh kombinasi parameter terbaik melalui proses *hyperparameter tuning*, tahap selanjutnya adalah melatih model menggunakan parameter tersebut pada dua algoritma regresi yang digunakan, yaitu *Random Forest Regressor* dengan max_depth 10 dan n_estimators 30, serta *XGBoost Regressor* dengan max_depth 3 dan n_estimators 30, menggunakan data pelatihan yang telah melalui prapemrosesan.

Setelah kedua model dilatih secara individual, keduanya digabungkan ke dalam sebuah model *ensemble* menggunakan pendekatan *Voting Regressor*, di mana prediksi akhir diperoleh dari rata-rata prediksi masing-masing model konstituen. Model *ensemble* ini, bernama ensemble_voting_model, terdiri dari 'rf' untuk *Random Forest* dan 'xgb' untuk *XGBoost*, dan dilatih pada data yang sama sehingga diharapkan meningkatkan akurasi dan stabilitas prediksi dengan memanfaatkan kekuatan kombinasi kedua model.

4.7. Teknik Evaluasi Model

Dalam penelitian ini, kinerja model prediksi dievaluasi menggunakan metrik regresi yang umum, yakni *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE), dan Koefisien Determinasi (R^2), dengan hasil pengujian dari model yang telah disajikan pada tabel di bawah ini.

Tabel 3. Metrik Evaluasi Model

Model	**MSE**	**RMSE**	**MAE**	**R^2**
ensemble_voting_model	1.6282	1.276	0.5987	0.7874

Berdasarkan Tabel 3, model menghasilkan nilai MSE sebesar 1,6282, yang menggambarkan rata-rata kuadrat kesalahan prediksi terhadap nilai aktual. Nilai RMSE sebesar 1,2760 menunjukkan bahwa rata-rata deviasi prediksi dari nilai sebenarnya berada sekitar 1,276 satuan, sehingga lebih mudah diinterpretasikan. Sementara itu, MAE sebesar 0,5987 mengindikasikan bahwa secara rata-rata selisih absolut antara nilai prediksi dan nilai aktual kurang dari 1 satuan, menunjukkan tingkat kesalahan yang relatif kecil. Nilai R^2 sebesar 0,7874 menunjukkan bahwa

sekitar 78,74% variasi dalam data target dapat dijelaskan oleh model, sedangkan sekitar 21,26% variasi tidak dapat dijelaskan.

Secara keseluruhan, hasil evaluasi ini menunjukkan bahwa *ensemble voting model* yang digunakan dalam penelitian ini mampu memberikan performa prediksi yang cukup akurat dan andal.

4.8. Perancangan Desain Graphical User Interface (GUI)

Dalam penelitian ini, penulis merancang desain antarmuka pengguna (*Graphical User Interface*/GUI) menggunakan Figma dengan tujuan menghadirkan tampilan yang sederhana, responsif, dan mudah digunakan. Desain antarmuka ini dibuat agar pengguna dapat mengoperasikan sistem prediksi tanpa memerlukan keahlian teknis khusus, sehingga sistem dapat diakses secara lebih luas oleh berbagai pihak. Berikut ditampilkan rancangan desain GUI yang telah dibuat:

Gambar 4. Desain GUI (Halaman Utama)

Gambar 5. Desain GUI (Unggah Dokumen Microsoft Excel)

Gambar 6. Desain GUI (Prediksi Semua Pelanggan)

Gambar 7. Desain GUI (Prediksi per Pelanggan)

4.9. Implementasi Graphical User Interface (GUI)

Dalam penelitian ini, penulis menggunakan teknologi berbasis web, yaitu HTML untuk membangun struktur halaman, CSS untuk mengatur tampilan dan tata letak, serta JavaScript untuk menangani interaktivitas sistem. Antarmuka pengguna ini dirancang agar dapat mendukung proses unggah data, prapemrosesan, hingga penyajian hasil prediksi secara interaktif. Adapun beberapa komponen utama dari GUI yang dibangun adalah sebagai berikut:

4.9.1. Halaman Awal

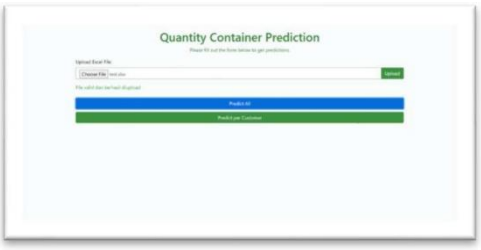
Halaman ini merupakan tampilan pertama yang muncul saat pengguna mengakses sistem. Pada halaman ini disediakan beberapa tombol navigasi utama yang

memudahkan pengguna untuk berpindah ke *fitur-fitur* penting, seperti unggah dokumen, melakukan prediksi untuk seluruh pelanggan, maupun melakukan prediksi khusus per pelanggan. Desain halaman awal dibuat sederhana dan intuitif agar pengguna dapat langsung memahami alur penggunaan sistem tanpa kesulitan. Tampilan halaman awal dapat dilihat pada Gambar 8.

Gambar 8. Halaman Awal

4.9.2. Unggah Dokumen Microsoft Excel

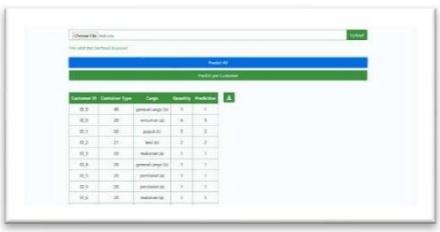
Pada bagian ini, pengguna dapat mengunggah dokumen Microsoft Excel yang berisi data pelanggan. Data dari dokumen tersebut kemudian disimpan secara otomatis ke dalam *database* MySQL untuk selanjutnya digunakan sebagai input dalam proses prediksi. Setelah dokumen berhasil diunggah, sistem akan menampilkan notifikasi yang menandakan bahwa data telah tersimpan dengan sukses. Tampilan *fitur* unggah dokumen dapat dilihat pada Gambar 9.



Gambar 9. Unggah Dokumen Microsoft Excel

4.9.3. Prediksi Semua Pelanggan

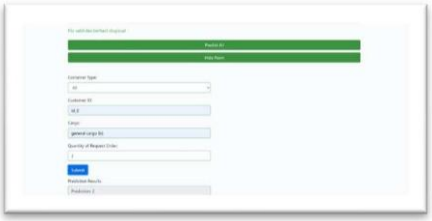
Fitur ini menampilkan hasil prediksi untuk seluruh data pelanggan yang telah diunggah sebelumnya. Hasil prediksi ditampilkan dalam bentuk tabel yang berisi informasi input serta output hasil prediksi dari model. Dengan tampilan tabel ini, pengguna dapat dengan mudah membandingkan data aktual dengan hasil prediksi yang dihasilkan oleh sistem. Ilustrasi tampilan *fitur* ini ditunjukkan pada Gambar 10.



Gambar 10. Prediksi Semua Pelanggan

4.9.4. Prediksi per Pelanggan

Pada *fitur* ini, pengguna dapat memasukkan data pelanggan secara manual melalui *form* input yang telah disediakan. Data yang dimasukkan akan langsung diproses oleh model, dan hasil prediksi akan ditampilkan secara otomatis setelah pengguna menekan tombol *submit*. *Fitur* ini dirancang untuk memudahkan pengecekan prediksi secara individual, sehingga pengguna dapat memperoleh informasi prediksi yang lebih spesifik untuk pelanggan tertentu. Tampilan *fitur* ini ditunjukkan pada Gambar 11.



Gambar 11. Prediksi per Pelanggan

4.10. Spesifikasi Sistem Prediksi Kontainer

Sistem yang dibangun memungkinkan pengguna menjalankan prediksi menggunakan model *machine learning* yang telah dilatih, yaitu RandomForestRegressor dan XGBRegressor. Kedua model tersebut dikonfigurasi dengan jumlah *estimators* sebanyak 30 pohon serta kedalaman maksimal yang relatif kecil, sehingga tetap efisien dan dapat dijalankan pada perangkat dengan spesifikasi standar. Detail kebutuhan sistem ditunjukkan pada Tabel 4.

Tabel 4. Spesifikasi Minimum Perangkat

Komponen	**Minimum**	**Disarankan**
RAM	4 GB	8 GB
CPU	Intel i3 / Ryzen 3 (Dual-core)	Intel i5 / Ryzen 5 (Quad-core)
Penyimpanan	SSD 128 GB	SSD 256 GB
GPU	Tidak diperlukan	Tidak diperlukan
OS	Windows 10/ Linux/ macOS	Windows 10/ Linux/ macOS

5. Kesimpulan

Berdasarkan penelitian yang dilakukan, dapat disimpulkan bahwa analisis data historis pemesanan tahun 2023 pada PT. XYZ mengungkap adanya ketidaksesuaian signifikan antara jumlah kontainer yang dipesan dengan jumlah yang benar-benar dimuat di kapal. Ketidaksesuaian ini memicu berbagai permasalahan operasional seperti alih kapal, batal muat, dan *shifting* kontainer, yang akhirnya menurunkan efisiensi operasional dan meningkatkan biaya perusahaan. Penelitian berhasil mengembangkan sistem prediksi jumlah kontainer aktual menggunakan algoritma *Random Forest Regressor* dan *XGBoost Regressor* yang dikombinasikan dalam model *Voting Regressor*. Proses *hyperparameter tuning* dengan *GridSearchCV* terbukti mampu meningkatkan performa model. Hasil evaluasi menunjukkan model *Voting*

Regressor memiliki kinerja terbaik dengan nilai R^2 sebesar 0,7874, MSE sebesar 1,6282, serta nilai RMSE dan MAE yang relatif rendah, membuktikan kemampuan *machine learning* dalam menghasilkan prediksi yang lebih akurat dan stabil dibandingkan perkiraan manual. Implementasi model prediksi ke dalam aplikasi web berbasis Flask telah memberikan solusi praktis yang dapat langsung digunakan. Sistem ini memungkinkan pengguna mengunggah data dan memperoleh prediksi jumlah kontainer aktual secara otomatis. Informasi prediktif ini diharapkan dapat membantu PT. XYZ dalam merencanakan alokasi kontainer, mengantisipasi *mismatch*, dan mengoptimalkan kapasitas kapal, sehingga meningkatkan efisiensi logistik dan menekan biaya tambahan.

Untuk pengembangan penelitian lebih lanjut, disarankan agar dilakukan penambahan *fitur* kontinu, seperti rata-rata historis pengiriman, volume per rute, atau tren permintaan musiman, agar model dapat mengenali pola yang lebih kompleks dan meningkatkan akurasi prediksi. Selain itu, untuk meningkatkan manfaat praktis, sistem prediksi sebaiknya diintegrasikan langsung dengan aplikasi atau sistem logistik internal PT. XYZ. Integrasi ini akan memungkinkan penggunaan hasil prediksi secara otomatis dan *real-time* oleh tim operasional, tanpa memerlukan proses input manual.

Daftar Pustaka

Alia, P. A., Prayogo, J. S., Kartiko, E. Y., Prasetyo, D., Khairunusi, Y., Na'am, J., Wijaya, A., Setyadi, A. T., Remawati, D., Mair, Z. R., Febriana, R. W., Setyadinsa, R., Maspupah, A., & Cahyono, W. A. (2023). *Sistem Basis Data*. PT Penamuda Media.

Banerjee, P., Kumar, B., Singh, A., Kumar, R., & Kumar, R. (2019). Comparative performance analysis of optimized round robin scheduling (ORR) using dynamic time quantum with round robin scheduling using static time quantum in Real Time System. *International Journal of Engineering and Computer Science*, *8*(12), 24890–24893. https://doi.org/10.18535/ijecs/v8i12.4399

Fauziyah, S., & Sugiarti, Y. (2022). Literature Review: Analisis Metode Perancangan Sistem Informasi Akademik Berbasis Web. *Jurnal Ilmiah Ilmu Komputer*, *8*(2), 87–93. https://doi.org/10.35329/jiik.v8i2.229

Ginting, D. (2021). Penanganan Pengangkutan Barang Melalui Container Pada Pt. Elang Sriwijaya Perkasa Palembang. *Agriprimatech*, *5*(1), 23–30. https://doi.org/10.34012/agriprimatech.v5i1.2074

Jatmiko, Y. A., Padmadisastra, S., & Chadidjah, A. (2019). Analisis Perbandingan Kinerja Cart Konvensional, Bagging Dan Random Forest Pada Klasifikasi Objek: Hasil Dari Dua Simulasi. *Media Statistika*, *12*(1), 1–12. https://doi.org/10.14710/medstat.12.1.1-12

Laudon, K. C., & Laudon, J. P. (2020). *Management Information Systems: Managing the Digital Firm* (16th ed.). Pearson Education Limited.

Mahamat, A. A., Boukar, M. M., Leklou, N., Celino, A., Obianyo, I. I., Bih, N. L., Stanislas, T. T., & Savastanos, H. (2024). Decision Tree Regression vs. Gradient Boosting Regressor Models for the Prediction of Hygroscopic Properties of Borassus Fruit Fiber. *Applied Sciences (Switzerland)*, *14*(17). https://doi.org/10.3390/app14177540

Mallach, E. G. (2020). *Information Systems: What Every Business Student Needs to Know* (2nd ed.). Routledge.

Muslim, M. A., Prasetiyo, B., Mawarni, E. L. H., Herowati, A. J., Mirqotussa'adah, S., Rukmana, S. H., & Nurzahputra, A. (2019). *Data Mining: Algoritma C4.5, Disertai Contoh Kasus dan Penerapannya dengan Program Computer*.

Ningtyas, D. F., & Setiyawati, N. (2021). Implementasi Flask Framework pada Pembangunan Aplikasi Purchasing Approval Request. *Jurnal Janitra Informatika Dan Sistem Informasi*, *1*(1), 19–34. https://doi.org/10.25008/janitra.v1i1.120

Pradhan, M., & Kumar, U. D. (2019). *Machine Learning Using Python*. Wiley.

Putra, A. I., & Santika, R. R. (2020). Implementasi Machine Learning dalam Penentuan Rekomendasi Musik dengan Metode Content-Based Filtering. *Edumatic : Jurnal Pendidikan Informatika*, *4*(1), 121–130. https://doi.org/10.29408/edumatic.v4i1.2162

Putri Primawanti, E., & Ali, H. (2022). Pengaruh Teknologi Informasi, Sistem Informasi Berbasis Web Dan Knowledge Management Terhadap Kinerja Karyawan (Literature Review Executive Support Sistem (Ess) for Business). *Jurnal Ekonomi Manajemen Sistem Informasi*, *3*(3), 267–285. https://doi.org/10.31933/jemsi.v3i3.818

Putri, M. P., Barovih, G., Azdy, R. A., Yuniansyah, Saputra, A., Sriyeni, Y., Rini, A., & Admojo, F. T. (2022). *Algoritma dan Struktur Data*. Widina Bhakti Persada.

Putri, R. A. (2022). *Buku Ajar Basis Data* (Edisi Kedua). Media Sains Indonesia.

Rayadin, M. A., Musaruddin, M., Saputra, R. A., & Isnawaty, I. (2024). Implementasi Ensemble Learning Metode XGBoost dan Random Forest untuk Prediksi Waktu Penggantian Baterai Aki. *BIOS: Jurnal Teknologi Informasi Dan Rekayasa Komputer*, *5*(2), 111–119.

Raharjo, B. (2021). *Sistem Manajemen Database*. Yayasan Prima Agus Teknik.

Saadah, S., & Salsabila, H. (2021). Prediksi Harga Bitcoin Menggunakan Metode Random Forest. *Jurnal Komputer Terapan*, *7*(1), 24–32. https://doi.org/10.35143/jkt.v7i1.4618

Salsabil, M., Lutvi, N., & Eviyanti, A. (2024). Implementasi Data Mining Dalam Melakukan Prediksi Penyakit Diabetes Menggunakan Metode Random Forest Dan Xgboost. *Jurnal Ilmiah Komputasi*, *23*(1), 51–58. https://doi.org/10.32409/jikstik.23.1.3507

Saputra, D. B., Atina, V., & Nastiti, F. E. (2024). Penerapan Model Crisp-Dm Pada Prediksi Nasabah Kredit. *Idealis: Indonesia Journal Information System*, *7*, 240–247.

Siji George, C. G., & Sumathi, B. (2020). Grid search tuning of hyperparameters in random forest classifier for customer feedback sentiment prediction. *International Journal of Advanced Computer Science and Applications*, *11*(9), 173–178. https://doi.org/10.14569/IJACSA.2020.0110920

Sinambela, D. P., Naparin, H., Zulfadhilah, M., & Hidayah, N. (2023). Implementasi Algoritma Decision Tree dan Random Forest dalam Prediksi Perdarahan Pascasalin. *Jurnal Informasi Dan Teknologi*, *5*(3), 58–64. https://doi.org/10.60083/jidt.v5i3.393

Stair, R. M., Reynolds, G. W., Bryant, J., Frydenberg, M., Greenberg, H., & Schell, G. (2021). *Principles of Information Systems*. Cengage Learning.

Sunarko, B., Hasanah, U., Hidayat, S., Muhammad, N., Ardiansyah, M. I., Ananda, B. P., Hakiki, M. K., & Baroroh, L. T. (2023). Penerapan Stacking Ensemble Learning untuk Klasifikasi Efek Kesehatan Akibat Pencemaran Udara. *Edu Komputika Journal*, *10*(1), 55–63. https://doi.org/10.15294/edukomputika.v10i1.72080

Syahputri, K., Irwan, M., & Nasution, P. (2023). Peran Database Dalam Sistem Informasi Manajemen. *Jurnal Akuntansi Keuangan Dan Bisnis*, *1*(2), 54–58. https://jurnal.ittc.web.id/index.php/jakbs/article/view/36

Ucha Putri, S., Irawan, E., Rizky, F., Tunas Bangsa, S., -Indonesia Jln Sudirman Blok No, P. A., & Utara, S. (2021). Implementasi Data Mining Untuk Prediksi Penyakit Diabetes Dengan Algoritma C4.5. *Jurnal*, *2*(1), 39–46.

Umam, K. (2021). *Algoritma dan Pemrograman Komputer dengan Python*. Duta Media Publishing.

Umar, R., Hadi, A., Widiandana, P., & Anwar, F. (2019). Perancangan Database Point of Sales Apotek Dengan Menerapkan Model Data Relasional. *Query: Journal of Information Systems*, 33–41.

Zahra, R. A. (2016). *Penerapan Algoritma Random Forest Dengan Hyperparameter Tuning Untuk Memprediksi Harga Sewa Kost Di Kota Bandar Lampung*. Skripsi.