Prosiding Seminar Nasional
KARSA NUSANTARA
Kolaborasi Rekayasa dan Sains Nasional
untuk Teknologi, Riset, dan
Kecerdasan Buatan

Fakultas Teknik,
Universitas Wijaya Kusuma Surabaya
ISSN 3090-1154 (Media Online)
Volume 2, Tahun 2025

Prediksi Permintaan Resistor Menggunakan Implementasi Manual Gradient Boost Untuk Optimasi Inventaris

F.X. Wisnu Yudo Untoro*[1]
[1]Program Studi Informatika, Fakultas Teknik, Universitas Wijaya Kusuma Surabaya
Email: wisnuyudo@uwks.ac.id
*Penulis Korespondensi

Abstrak

Penelitian ini bertujuan mengoptimalkan inventaris resistor bulanan di toko komponen elektronik melalui prediksi permintaan yang akurat. Mengatasi tantangan *overstock* dan *understock* yang memicu kerugian investasi, model *Gradient Boosting* diimplementasikan secara manual. Model ini, yang dibangun berdasarkan prinsip dasar *Extreme Gradient Boosting* (XGBoost), dilatih menggunakan data historis permintaan resistor dari tiga bulan sebelumnya. Evaluasi kinerja model selama 20 iterasi pelatihan menunjukkan penurunan konsisten pada metrik *Mean Absolute Error* (MAE) dan *Root Mean Squared Error* (RMSE), mengindikasikan pembelajaran yang efektif. Model berhasil memprediksi permintaan Bulan-4 dan Bulan-5 menggunakan pendekatan "geser waktu". Meskipun sudah menunjukkan kemampuannya dalam memprediksi, penelitian ini dibatasi oleh penggunaan dataset simulasi yang sangat kecil (5 baris), dapat menyebabkan *overfitting* dan membatasi generalisasi. Hasil ini berfungsi sebagai ilustrasi metodologi dan validasi konsep dasar *Gradient Boosting* dalam mendukung keputusan inventaris. Untuk aplikasi nyata, diperlukan dataset lebih besar dan implementasi teroptimasi (misalnya, dengan *library* XGBoost) di masa mendatang.

Kata kunci: *Gradient Boosting* manual, *machine learning*, optimasi inventaris, resistor, *time series forecasting.*

Abstract

This research aims to optimize monthly resistor inventory in electronic component stores through accurate demand prediction. Addressing inventory management challenges like overstocking and understocking, which lead to investment losses, accurate demand prediction becomes crucial. A Gradient Boosting model is manually implemented, built upon the fundamental principles of Extreme Gradient Boosting (XGBoost), and trained using historical resistor demand data from the preceding three months. The model's performance was evaluated over 20 training iterations, showing a consistent decrease in Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) metrics, indicating effective learning. The model successfully predicted demand for Month-4 and Month-5 using a "sliding window" approach. Despite demonstrating predictive capabilities, this research is limited by the use of a very small simulated dataset (5 rows), leading to overfitting and restricted generalization. These results serve as a methodological illustration and validation of the basic Gradient Boosting concept in supporting inventory decisions. For real-world applications, a larger dataset and optimized implementation (e.g., with the XGBoost library) will be necessary in future studies

Keywords*: inventory optimization, machine learning, manual Gradient Boosting, resistor, time series forecasting.*

I. PENDAHULUAN

Manajemen inventaris yang efisien merupakan tulang punggung operasional bagi toko penyedia komponen elektronik, khususnya dalam pengelolaan resistor [1, 2]. Dalam pasar komponen yang cepat berubah dan penuh persaingan, ketersediaan resistor dengan nilai dan toleransi yang tepat pada waktu yang tepat sangat penting untuk menjaga kelangsungan bisnis dan memenuhi permintaan pelanggan yang beragam. Kelebihan stok (*overstock*) resistor, misalnya, dapat mengakibatkan biaya penyimpanan yang membengkak, risiko kerusakan atau kadaluwarsa komponen, serta dana yang mengendap tidak produktif. Sebaliknya, kekurangan stok (*understock*) resistor bisa menyebabkan kehilangan peluang penjualan, ketidakmampuan memenuhi pesanan pelanggan secara langsung, dan bahkan merusak reputasi toko di mata pelanggan [3]. Oleh karena itu, kemampuan untuk memprediksi permintaan resistor dengan akurat menjadi kunci utama dalam mengoptimalkan tingkat inventaris dan menjaga keseimbangan operasional toko, sehingga memastikan perputaran dana investasi yang efisien dan meminimalkan kerugian [4-6].

Resistor, elemen fundamental yang tak terpisahkan dari setiap sirkuit elektronik, menjadi studi kasus yang relevan untuk penelitian ini. Sebagai penjual resistor, toko dihadapkan pada tantangan mengelola ribuan *stock-keeping units* (SKU) yang bervariasi berdasarkan nilai, toleransi, daya, dan karakteristik lainnya. Pola permintaannya dapat menunjukkan ketidakstabilan yang signifikan tergantung

pada nilai resistor, toleransi, harga, dan tren pasar elektronika [7, 8]. Memahami dan memprediksi ketidakstabilan ini secara presisi adalah sangat penting untuk memastikan stok yang stabil dan menghindari disrupsi (gangguan tak terduga yang mengganggu aliran barang atau informasi) dalam rantai pasok toko, sehingga semua kebutuhan pelanggan dapat terpenuhi [9].

Integrasi teknologi *machine learning*, khususnya algoritma *Extreme Gradient Boosting* (XGBoost), menawarkan potensi besar untuk mengatasi keterbatasan metode peramalan konvensional [10]. Kekuatan XGBoost dalam menangani data heterogen dan mengidentifikasi pola kompleks menjadikannya kandidat ideal untuk prediksi permintaan yang lebih akurat [10-13]. Pemilihan XGBoost didasarkan pada kemampuannya yang terbukti dalam menangani data terstruktur dengan akurasi tinggi, kemampuannya untuk mengidentifikasi hubungan non-linear antar variabel, serta fitur regularisasi bawaannya yang efektif mencegah *overfitting* [11, 12, 14, 10]. Hal ini sangat relevan mengingat karakteristik data penjualan komponen yang seringkali kompleks dan menunjukkan ketidakstabilan.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk mengembangkan dan mengevaluasi model prediksi permintaan resistor menggunakan algoritma *Extreme Gradient Boosting* (XGBoost) guna mendukung optimasi inventaris di toko penyedia komponen elektronik. Hasil penelitian ini diharapkan dapat memberikan panduan bagi penjual komponen elektronik dalam mengambil keputusan pengadaan dan pengelolaan stok yang lebih efisien, mengurangi biaya operasional, dan meningkatkan kepuasan pelanggan. Manuskrip ini akan menguraikan latar belakang masalah, metode penelitian yang digunakan, hasil dan pembahasan model, serta kesimpulan dari studi ini.

II. METODOLOGI PENELITIAN

Bagian ini menjelaskan pendekatan, sumber data, teknik pra-pemrosesan data, pengembangan model prediktif, serta metode evaluasi dan strategi prediksi yang digunakan dalam penelitian ini.

2.1. Desain Penelitian

Penelitian ini mengadopsi pendekatan kuantitatif dengan fokus pada pemodelan prediktif. Tujuannya adalah untuk mengembangkan dan mengevaluasi model *machine learning* yang mampu memprediksi permintaan resistor bulanan untuk periode mendatang (Bulan-4 dan Bulan-5). Prediksi ini diharapkan dapat menjadi dasar untuk mendukung optimasi inventaris resistor. Algoritma *Gradient Boosting* dipilih karena kemampuannya dalam menangani hubungan non-linear dan menghasilkan prediksi yang akurat pada data tabular [11-13]. Implementasi manual ini berfungsi sebagai demonstrasi prinsip dasar yang mendasari algoritma teroptimasi seperti XGBoost [10].

2.2. Sumber Data

Dataset yang digunakan dalam penelitian ini merupakan data historis permintaan resistor yang disimulasikan, terdiri dari 5 entri unik (`ID`). Entri ini merepresentasikan jenis-jenis resistor yang umum, seperti seri E12, E24, E48, E96, dan E192, yang masing-masing memiliki nilai resistansi dan toleransi standar [15-18]. Setiap entri mencakup atribut sebagai berikut:

(1) **`ID`**: Pengidentifikasi unik untuk setiap jenis resistor (misalnya, nilai resistor tertentu dalam seri E).

(2) **`Nilai`**: Atribut numerik yang merepresentasikan nilai resistansi resistor (misalnya, dalam Ohm) yang relevan dengan seri E yang dimaksud.

(3) **`Toleransi`**: Atribut numerik yang merepresentasikan tingkat toleransi resistor (misalnya, 0.10, 0.05) sesuai dengan standar seri E.

(4) **`Bulan-3`**: Permintaan aktual resistor tiga bulan sebelum bulan target.

(5) **`Bulan-2`**: Permintaan aktual resistor dua bulan sebelum bulan target.

(6) **`Bulan-1`**: Permintaan aktual resistor satu bulan sebelum bulan target.

(7) **`Permintaan`**: Permintaan aktual resistor pada bulan target (bulan setelah `Bulan-1`), yang berfungsi sebagai variabel dependen atau target yang akan diprediksi.

Untuk memberikan gambaran konkret mengenai susunan data yang digunakan, struktur dataset ini direpresentasikan sebagai berikut:

```
data = {
```

```
    'ID': [1, 2, 3, 4, 5],
    'Nilai': [10, 22, 47, 100, 220],
    'Toleransi': [0.10, 0.05, 0.02, 0.01, 0.10],
    'Bulan-3': [10, 11, 12, 14, 7],
    'Bulan-2': [9, 10, 11, 13, 8],
    'Bulan-1': [8, 9, 13, 12, 6],
    'Permintaan': [9, 10, 14, 12, 7]
}
```

Meskipun dataset ini berukuran kecil, ia digunakan secara spesifik untuk tujuan demonstrasi dan pemahaman mendalam tentang cara kerja algoritma *Gradient Boosting* secara iteratif. Dalam aplikasi dunia nyata, dataset akan mencakup variasi yang lebih luas dari resistor seri E12, E24, E48, E96, dan E192, serta periode historis yang lebih panjang. Dataset yang lebih besar dan representatif sangat diperlukan untuk mencapai akurasi prediksi yang andal dan dapat digeneralisasi

2.3. Pra-pemrosesan Data

Tahap pra-pemrosesan data melibatkan identifikasi fitur dan target, serta persiapan data untuk pelatihan model:

1. **Identifikasi Fitur (Variabel Independen)**: Fitur-fitur yang digunakan untuk melatih model adalah data historis permintaan resistor: `Bulan-1`, `Bulan-2`, dan `Bulan-3`. Meskipun `Nilai` dan `Toleransi` tersedia dalam dataset dan relevan dengan karakteristik resistor (termasuk seri E), mereka tidak secara langsung digunakan sebagai input untuk pohon keputusan dalam implementasi *Gradient Boosting* ini, namun tetap menjadi bagian dari konteks setiap `ID` resistor.
2. **Identifikasi Target (Variabel Dependen)**: Variabel Permintaan ditetapkan sebagai target prediksi, yang merepresentasikan permintaan aktual resistor pada bulan setelah `Bulan-1`.
3. **Pembagian Data**: Karena ukuran dataset yang sangat kecil, seluruh dataset digunakan untuk pelatihan model guna memaksimalkan pembelajaran pola yang ada. Dalam penelitian skala besar, pembagian data menjadi *training*, *validation*, dan *test set* adalah praktik standar.

2.4. Pengembangan Model Prediksi

Model prediksi dibangun menggunakan implementasi manual dari algoritma *Gradient Boosting* Regression. Implementasi ini bertujuan untuk mendemonstrasikan prinsip-prinsip inti dari *Gradient Boosting*, yang menjadi dasar bagi algoritma *Extreme Gradient Boosting* (XGBoost) yang lebih canggih dan teroptimasi. Proses pengembangan model ini mencakup langkah-langkah berikut:

1. **Algoritma Dasar**: *Gradient Boosting* adalah teknik *ensemble* yang membangun model secara sekuensial, di mana setiap model baru mencoba memperbaiki kesalahan yang dibuat oleh model sebelumnya.
2. **Pembelajar Lemah (Weak Learner)**: Setiap model individual dalam *ensemble* ini adalah pohon keputusan (`DecisionTreeRegressor`) dari *library* `scikit-learn`.
3. **Parameter Model**:
   a. **`n_estimators`**: Jumlah pohon keputusan yang dibangun dalam *ensemble* ditetapkan sebanyak 20. Semakin banyak pohon, semakin kompleks model, tetapi juga semakin tinggi risiko *overfitting*.
   b. **`learning_rate`**: Tingkat pembelajaran diatur pada 0.1. Parameter ini mengontrol seberapa besar kontribusi setiap pohon baru terhadap prediksi keseluruhan. Nilai yang lebih kecil membutuhkan lebih banyak pohon tetapi dapat menghasilkan model yang lebih robust [14] (Géron, 2022).
   c. **`max_depth`**: Kedalaman maksimum setiap pohon keputusan dibatasi hingga 3. Pembatasan kedalaman ini membantu mencegah setiap pohon menjadi terlalu kompleks dan mengurangi risiko *overfitting*.
   d. **`random_state`**: Ditetapkan pada 42 untuk memastikan reproduktibilitas hasil pelatihan pohon.
4. **Proses Pelatihan Iteratif**:

a. **Inisialisasi Prediksi Awal**: Prediksi awal (`F_train`) diinisialisasi sebagai rata-rata dari semua nilai target (`Permintaan`) dalam dataset pelatihan. Rumus rata-rata dihitung dengan persamaan matematis berikut:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{1}$$

di mana n adalah jumlah sampel data pelatihan, dan y_i adalah nilai target individual ke i.

b. **Iterasi Boosting**: Untuk setiap dari `n_estimators` iterasi:

1) **Penghitungan Residual**: Residual (kesalahan) dihitung sebagai selisih antara nilai `Permintaan` aktual dan prediksi `F_train` saat ini. Residual ini berfungsi sebagai gradien negatif dari fungsi *loss*.

$$r_i = y_i - \hat{y}_i \tag{2}$$

di mana r_i adalah residual ke i, y_i adalah nilai target individual ke i dan $\hat{y}_i$ adalah nilai prediksi ke i.

2) **Pelatihan Pohon pada Residual**: Sebuah pohon keputusan baru dilatih untuk memprediksi residual ini, bukan target asli. Pohon ini belajar dari kesalahan yang tersisa.

3) **Pembaruan Prediksi**: Prediksi dari pohon yang baru dilatih dikalikan dengan `learning_rate` dan ditambahkan ke `F_train`. Proses ini secara bertahap "mendorong" prediksi model lebih dekat ke nilai aktual.

2.5. Evaluasi Model

Evaluasi model dilakukan untuk mengukur kinerja dan akurasi model pada data pelatihan. Metrik evaluasi yang digunakan adalah:

1. **Mean Absolute Error (MAE)**: Mengukur rata-rata selisih absolut antara nilai prediksi dan nilai aktual. MAE memberikan gambaran langsung tentang seberapa besar rata-rata kesalahan prediksi dalam satuan yang sama dengan variabel target [19, 20]. Penghitungan MAE menggunakan persamaan (3):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{3}$$

di mana n adalah jumlah sampel data, y_i adalah nilai aktual (observasi) ke i, dan $\hat{y}_i$ adalah nilai prediksi ke i.

2. **Root Mean Squared Error (RMSE)**: Mengukur rata-rata akar kuadrat dari selisih kuadrat antara nilai prediksi dan nilai aktual. RMSE lebih sensitif terhadap kesalahan besar (*outlier*) dibandingkan MAE [19, 20]. Penghitungan RMSE menggunakan persamaan (3):

$$RMSE = \sqrt{\frac{i}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \tag{4}$$

di mana n adalah jumlah sampel data, y_i adalah nilai aktual (observasi) ke i, dan $\hat{y}_i$ adalah nilai prediksi ke i.

3. **Visualisasi Evaluasi Iteratif**: Grafik garis digunakan untuk memvisualisasikan perubahan nilai MAE dan RMSE pada setiap iterasi pelatihan. Grafik ini membantu memantau konvergensi model dan mengidentifikasi potensi *overfitting* (jika error pada data validasi mulai meningkat).

4. **Struktur Pohon Keputusan**: Struktur setiap pohon keputusan yang dilatih ditampilkan untuk memberikan wawasan tentang bagaimana model membuat keputusan dan fitur mana yang paling berpengaruh dalam memprediksi residual.

2.6. Strategi Prediksi (Forecasting)

Model yang telah dilatih digunakan untuk memprediksi permintaan pada bulan-bulan mendatang (Bulan-4 dan Bulan-5) menggunakan strategi "geser waktu" (sliding window). Prediksi permintaan yang akurat ini esensial untuk pengambilan keputusan yang tepat dalam optimasi inventaris resistor.

1. **Prediksi Permintaan Bulan-4**:
   Untuk memprediksi `Permintaan` pada Bulan-4, input fitur disiapkan dengan menggeser data historis:
   a. `Bulan-3` (untuk Bulan-4) diambil dari `Bulan-2` asli.
   b. `Bulan-2` (untuk Bulan-4) diambil dari `Bulan-1` asli.
   c. `Bulan-1` (untuk Bulan-4) diambil dari `Permintaan` aktual yang digunakan sebagai target pelatihan.

   Model kemudian menjalankan data input ini melalui semua pohon yang telah dilatih untuk menghasilkan `Prediksi_Permintaan_Bulan_4`.

2. **Prediksi Permintaan Bulan-5**:
   Terdapat dua skenario untuk memprediksi `Permintaan` pada Bulan-5:
   a. **Menggunakan Prediksi Bulan-4**: Jika data aktual Bulan-4 belum tersedia, `Prediksi_Permintaan_Bulan_4` yang dihasilkan model akan digunakan sebagai input historis terbaru (`Bulan-1` yang baru) untuk memprediksi Bulan-5. Logika pergeseran data adalah:
      1) `Bulan-3` (untuk Bulan-5) diambil dari `Bulan-2` yang digunakan untuk prediksi Bulan-4 (yaitu `Bulan-1` asli).
      2) `Bulan-2` (untuk Bulan-5) diambil dari `Bulan-1` yang digunakan untuk prediksi Bulan-4 (yaitu `Permintaan` asli).
      3) `Bulan-1` (untuk Bulan-5) diambil dari `Prediksi_Permintaan_Bulan_4`. Pendekatan ini rentan terhadap akumulasi kesalahan (*error propagation*).

   b. **Menggunakan Data Aktual Bulan-4 (Jika Tersedia)**: Jika data aktual.
      `Permintaan` untuk Bulan-4 sudah tersedia, data aktual tersebut akan digunakan sebagai input historis terbaru (`Bulan-1` yang baru) untuk prediksi Bulan-5. Ini merupakan metode yang lebih disarankan karena mengurangi risiko akumulasi kesalahan dan menghasilkan prediksi yang lebih akurat.

7. Lingkungan Pengembangan

Penelitian ini diimplementasikan menggunakan bahasa pemrograman Python. Pustaka (libraries) utama yang digunakan meliputi:

1. **NumPy**: Untuk operasi numerik dan array.
2. **Pandas**: Untuk manipulasi dan analisis data (DataFrame).
3. **Scikit-learn**: Menyediakan implementasi `DecisionTreeRegressor` dan fungsi `export_text`.
4. **Matplotlib**: Untuk visualisasi grafik evaluasi model.

III. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil dari proses pelatihan model *Gradient Boosting* manual serta prediksi permintaan resistor untuk Bulan-4 dan Bulan-5. Pembahasan mencakup analisis kinerja model dan interpretasi hasil prediksi, serta implikasinya terhadap optimasi inventaris resistor.

3.1. Hasil Pelatihan Model

Pelatihan model *Gradient Boosting* dilakukan selama 20 iterasi, dengan setiap iterasi menambahkan pohon keputusan baru yang berupaya mengurangi residual dari prediksi sebelumnya. **Tabel 1** menunjukkan hasil akhir prediksi model pada data pelatihan permintaan resistor.

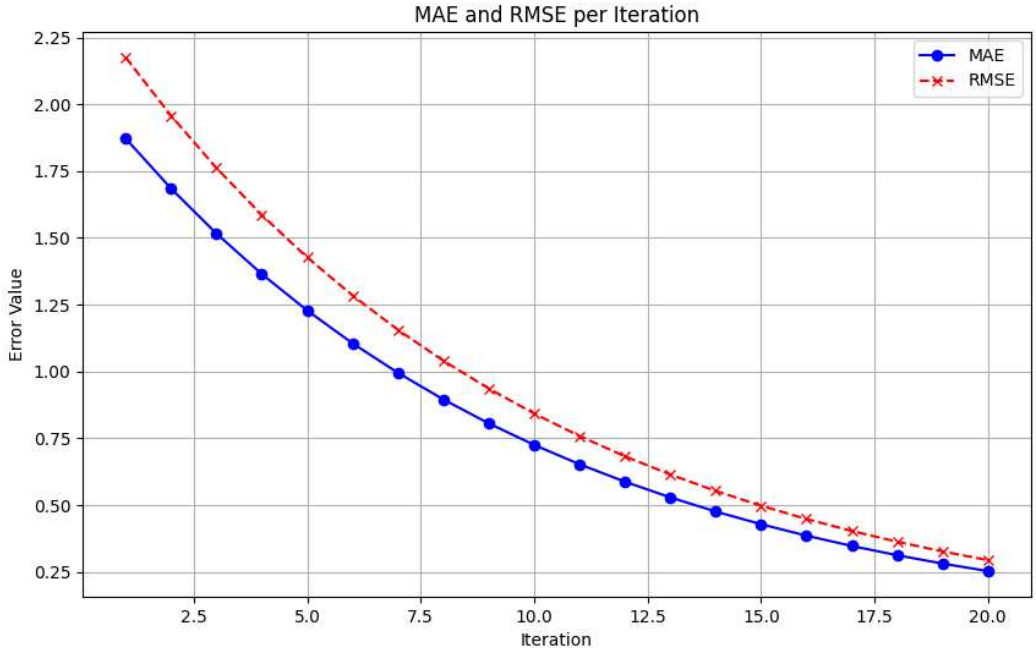
Tabel 1. Hasil Prediksi pada Data Pelatihan

ID	Permintaan	Prediksi_Pelatihan	Residual_Pelatihan
1	9	9.170207	-0.170207
2	10	10.048631	-0.048631
3	14	13.562324	0.437676
4	12	11.805477	0.194523
5	7	7.43361	-0.413361

Pada **Tabel 1** menjelaskan bahwa `Prediksi_Pelatihan` merepresentasikan nilai permintaan resistor yang diprediksi oleh model setelah 20 iterasi pada data yang sama dengan yang digunakan untuk pelatihan. `Residual_Pelatihan` menunjukkan selisih antara nilai permintaan resistor aktual dan prediksi model. Nilai residual yang mendekati nol mengindikasikan bahwa model telah belajar dengan baik untuk menyesuaikan diri dengan pola dalam data pelatihan yang tersedia.

Perkembangan kinerja model selama proses pelatihan dapat diamati melalui nilai MAE dan RMSE per iterasi. Pada Iterasi 1, model dimulai dengan MAE 1.8720 dan RMSE 2.1749. Seiring berjalannya iterasi, kedua metrik error ini menunjukkan penurunan yang konsisten, mencapai MAE 0.2529 dan RMSE 0.2938 pada Iterasi 20. Penurunan ini mengindikasikan bahwa setiap pohon keputusan yang ditambahkan berhasil mengurangi kesalahan prediksi model secara keseluruhan.

Perkembangan kinerja model secara visual juga ditunjukkan pada **Gambar 1**.



Gambar 1. Grafik MAE dan RMSE per Iterasi Pelatihan

Gambar 1 menunjukkan bahwa nilai MAE (garis biru) dan RMSE (garis merah) secara konsisten menurun seiring dengan bertambahnya jumlah iterasi. RMSE secara konsisten lebih tinggi daripada MAE, yang diharapkan karena RMSE memberikan bobot lebih pada kesalahan yang lebih besar (dikarenakan pengkuadratan). Tren penurunan yang stabil ini menunjukkan bahwa algoritma *Gradient Boosting* berhasil mengkonvergensi dan meningkatkan akurasi model pada data pelatihan.

Untuk memahami bagaimana model membuat keputusan dan mengidentifikasi pola dalam data, penting untuk meninjau struktur pohon keputusan yang terbentuk pada setiap iterasi. Sebagai ilustrasi, berikut adalah representasi tekstual dari struktur pohon keputusan yang terbentuk pada Iterasi 1:

```
Iterasi 1 - MAE: 1.8720, RMSE: 2.1749

Struktur Pohon Keputusan:

|--- Bulan-2 <= 10.50
|   |--- Bulan-3 <= 8.50
|   |   |--- value: [-3.40]
```

```
|   |--- Bulan-3 >  8.50
|   |   |--- Bulan-2 <= 9.50
|   |   |   |--- value: [-1.40]
|   |   |--- Bulan-2 >  9.50
|   |   |   |--- value: [-0.40]
|--- Bulan-2 >  10.50
|   |--- Bulan-3 <= 13.00
|   |   |--- value: [3.60]
|   |--- Bulan-3 >  13.00
|   |   |--- value: [1.60]
```

Struktur pohon ini menunjukkan bagaimana model membuat keputusan berdasarkan nilai `Bulan-2` dan `Bulan-3` untuk memprediksi residual. Nilai `value` pada setiap daun pohon merepresentasikan kontribusi residual yang diprediksi oleh pohon tersebut. Seiring dengan iterasi, nilai `value` pada daun pohon akan berubah (mengecil) karena pohon-pohon berikutnya berfokus pada residual yang semakin kecil.

3.2. Hasil Prediksi Permintaan Bulan-4

Setelah model dilatih, model digunakan untuk memprediksi permintaan resistor untuk Bulan-4. Input untuk prediksi ini disiapkan dengan menggeser data historis, di mana nilai `Permintaan` aktual dari dataset pelatihan berfungsi sebagai `Bulan-1` yang baru untuk prediksi Bulan-4. **Tabel 2** menyajikan hasil prediksi permintaan resistor untuk Bulan-4.

Tabel 2. Hasil Prediksi Permintaan Bulan-4

ID	Nilai (Ω)	Toleransi (%)	Bulan-3_for_Bulan_4	Bulan-2_for_Bulan_4	Bulan-1_for_Bulan_4	Prediksi Permintaan_Bulan_4
1	10	10	9	8	9	9.170207
2	22	5	10	9	10	10.048631
3	47	2	11	13	14	13.562324
4	100	1	13	12	12	11.805477
5	220	10	8	6	7	7.413361

Prediksi `Permintaan_Bulan_4` menunjukkan estimasi permintaan **resistor** untuk setiap `ID` berdasarkan pola yang dipelajari dari data historis. Sebagai contoh, untuk `ID=1`, model memprediksi permintaan resistor sebesar 9.170207 untuk Bulan-4. Perlu dicatat bahwa, karena ukuran dataset pelatihan yang sangat terbatas, nilai prediksi ini cenderung sangat mirip dengan prediksi pada data pelatihan, menunjukkan kecenderungan model untuk menghafal pola yang ada.

3.3. Hasil Prediksi Permintaan Bulan-5

Prediksi permintaan resistor untuk Bulan-5 dieksplorasi dalam dua skenario:

3.3.1. Skenario 1: Menggunakan Prediksi Bulan-4

Dalam skenario ini, `Prediksi_Permintaan_Bulan_4` digunakan sebagai input historis terbaru (`Bulan-1` yang baru) untuk memprediksi Bulan-5. **Tabel 3** menunjukkan hasil prediksi di bawah skenario ini.

Tabel 3. Hasil Prediksi Permintaan Bulan-5 (Menggunakan Prediksi Bulan-4)

ID	Nilai (Ω)	Toleransi (%)	Bulan-3_for_Bulan_5	Bulan-2_for_Bulan_5	Bulan-1_for_Bulan_5	Prediksi Permintaan_Bulan_5
1	10	10	8	9	9.170207	9.170207
2	22	5	9	10	10.048631	10.048631
3	47	2	13	14	13.562324	13.562324
4	100	1	12	12	11.805477	11.805477
5	220	10	6	7	7.43361	7.413361

Hasil prediksi ini menunjukkan bagaimana model melanjutkan estimasi permintaan resistor untuk periode yang lebih jauh. Namun, pendekatan ini rentan terhadap akumulasi kesalahan (*error propagation*), di mana kesalahan dari `Prediksi_Permintaan_Bulan_4` dapat memengaruhi akurasi `Prediksi_Permintaan_Bulan_5` [21].

3.3.2. Skenario 2: Menggunakan Data Aktual Bulan-4 (Simulasi)

Untuk menunjukkan dampak penggunaan data aktual, sebuah simulasi data aktual Bulan-4 digunakan sebagai input untuk memprediksi Bulan-5. **Tabel 4** menyajikan hasil prediksi di bawah skenario ini.

Tabel 4. Hasil Prediksi Permintaan Bulan-5 (Menggunakan Data Aktual Bulan-4 Simulasi)

ID	Nilai (Ω)	Toleransi (%)	Bulan-3_for_Bulan_5	Bulan-2_for_Bulan_5	Bulan-1_for_Bulan_5	Prediksi Permintaan_Bulan_5
1	10	10	8	9	9.500000	9.170207
2	22	5	9	10	10.200000	10.048631
3	47	2	13	14	13.800000	13.562324
4	100	1	12	12	12.100000	11.805477
5	220	10	6	7	7.300000	7.413361

Hasil prediksi ini menunjukkan bagaimana model melanjutkan estimasi permintaan resistor untuk periode yang lebih jauh. Namun, pendekatan ini rentan terhadap akumulasi kesalahan (*error propagation*), di mana kesalahan dari `Prediksi_Permintaan_Bulan_4` dapat memengaruhi akurasi `Prediksi_Permintaan_Bulan_5` [21, 22].

3.4. Pembahasan Umum dan Implikasi

Penelitian ini berhasil mendemonstrasikan implementasi manual dari algoritma *Gradient Boosting* untuk tugas prediksi deret waktu. Model menunjukkan kemampuan untuk mempelajari pola dari data historis dan secara iteratif mengurangi kesalahan pada data pelatihan, seperti yang ditunjukkan oleh penurunan MAE dan RMSE yang konsisten. Implementasi manual ini berfungsi sebagai fondasi konseptual untuk memahami cara kerja algoritma *Extreme Gradient Boosting* (XGBoost) yang lebih canggih dan teroptimasi [10-13].

Prediksi permintaan resistor yang dihasilkan dari model ini memiliki implikasi signifikan terhadap optimasi inventaris resistor. Dengan mengetahui estimasi permintaan untuk Bulan-4 dan Bulan-5, perusahaan dapat:

1. **Mengurangi Biaya Penyimpanan**: Meminimalkan *overstocking* (kelebihan stok) resistor, yang mengurangi biaya penyimpanan, risiko kerusakan, atau kadaluarsa.
2. **Meningkatkan Ketersediaan Produk**: Mencegah *understocking* (kekurangan stok) resistor, memastikan ketersediaan produk saat dibutuhkan, dan menghindari kehilangan penjualan serta ketidakpuasan pelanggan.
3. **Mengoptimalkan Keputusan Pembelian dan Produksi**: Memungkinkan manajer inventaris untuk merencanakan pembelian bahan baku dan jadwal produksi resistor secara lebih tepat waktu dan efisien, sehingga mengurangi biaya operasional dan meningkatkan *responsiveness* rantai pasok.

Namun, penting untuk mengakui keterbatasan signifikan dari penelitian ini yang diakibatkan oleh ukuran dataset yang sangat kecil (hanya 5 baris data). Dengan data yang terbatas, model cenderung melakukan *overfitting*, yaitu menghafal pola spesifik dalam data pelatihan daripada mempelajari hubungan yang dapat digeneralisasi [12, 14]. Hal ini terlihat dari kesamaan antara prediksi pada data taipelatihan dan prediksi untuk Bulan-4 dan Bulan-5. Oleh karena itu, hasil yang diperoleh dalam studi ini lebih berfungsi sebagai ilustrasi metodologi dan bukti konsep dasar algoritma *Gradient Boosting* daripada sebagai prediksi yang andal untuk aplikasi dunia nyata.

IV. KESIMPULAN

Penelitian ini berhasil mengembangkan dan mendemonstrasikan implementasi manual algoritma *Gradient Boosting* untuk memprediksi permintaan resistor bulanan. Model ini menunjukkan kemampuan untuk mempelajari pola dari data historis permintaan tiga bulan sebelumnya, dengan kinerja evaluasi yang konsisten menunjukkan penurunan nilai MAE dan RMSE pada setiap iterasi pelatihan. Prediksi permintaan untuk Bulan-4 dan Bulan-5 juga berhasil menggunakan strategi "geser waktu" (sliding window).

Hasil prediksi ini memiliki implikasi positif yang signifikan terhadap optimasi inventaris resistor. Dengan estimasi permintaan yang lebih akurat, toko penyedia komponen elektronik dapat, mengurangi biaya penyimpanan akibat *overstocking*, meningkatkan ketersediaan produk dan menghindari kehilangan penjualan akibat *understocking* dan mengoptimalkan perencanaan pembelian dan produksi.

Meskipun demikian, perlu diakui bahwa penelitian ini memiliki keterbatasan substansial, yaitu penggunaan dataset simulasi yang sangat kecil (hanya 5 baris). Ukuran dataset ini menyebabkan model cenderung mengalami *overfitting*, membatasi kemampuan generalisasinya, dan membuat hasil prediksi lebih berfungsi sebagai ilustrasi metodologi daripada sebagai alat prediksi yang siap untuk aplikasi dunia nyata. Untuk pengembangan di masa mendatang, sangat disarankan untuk menggunakan dataset permintaan resistor yang lebih besar dan bervariasi untuk melatih model, sehingga meningkatkan generalisasi, mengimplementasikan teknik validasi silang dan pembagian data yang lebih robust (*training*, *validation*, *test set*), memanfaatkan *library Gradient Boosting* teroptimasi (misalnya XGBoost, LightGBM) untuk skalabilitas dan kinerja yang lebih baik, dan melakukan *hyperparameter tuning* secara sistematis untuk mencapai akurasi optimal.

REFERENSI

[1] Vaka, D. K., "Integrating Inventory Management and Distribution: a Holistic Supply Chain Strategy," *International Journal of Managing Value and Supply Chains (IJMVSC)*, vol. 15, no. 2, pp. 13, Jun. 2024.

[2] V. Pasupuleti, B. Thuraka, C. S. Kodete, and S. Malisetty, "Enhancing Supply Chain Agility and Sustainability through Machine Learning: Optimization Techniques for Logistics and Inventory Management," Logistics, vol. 8, no. 3, p. 73, 2024, doi: 10.3390/logistics8030073

[3] Mweshi, G. K., "Effects of Overstocking and Stockouts on the Manufacturing Sector," *International Journal of Advances in Engineering and Management (IJAEM)*, vol. 4, no. 9, pp. 1054–1064, Sep. 2022. DOI: 10.35629/5252-040910541064

[4] R. Winurputra dan D. E. Ratnawati, "Peramalan Penjualan Produk Menggunakan Extreme Gradient Boosting (XGBOOST) Dan Kerangka Kerja Crisp-Dm untuk Pengoptimalan Manajemen Persediaan (Studi Kasus: Ub Mart)", *JTIIK*, vol. 12, no. 2, pp. 417–428, Apr. 2025, doi: 10.25126/jtiik.2025129451.

[5] S. B. Dalimunthe, R. . Ginting, and S. Sinulingga, "The Implementation Of Machine Learning In Demand Forecasting : A Review Of Method Used In Demand Forecasting With Machine Learning", *j.sist.teknik.industri.*, vol. 25, no. 1, pp. 41–49, Jan. 2023. DOI: https://doi.org/10.32734/jsti.v25i1.9290

[6] P. Dankorpho, "Sales Forecasting for Retail Business Using XGBoost Algorithm," *Journal of Computer Science and Technology Studies*, vol. 6, no. 2, pp. 136–141, Jun. 2024, DOI: https://doi.org/10.32996/jcsts.2024.6.2.15

[7] B. Auffarth, *Machine Learning for Time-Series with Python: Forecast, predict, and detect anomalies with state-of-the-art machine learning methods*. Packt Publishing Ltd, 2021.

[8] Y. Tadayonrad and A. B. Ndiaye, "A new key performance indicator model for demand forecasting in inventory management considering supply chain reliability and seasonality," *Supply Chain Analytics*, vol. 3, p. 100026, 2023. https://doi.org/10.1016/j.sca.2023.100026

[9] K. Kanike, "Factors disrupting supply chain management in manufacturing industries," *Journal of Supply Chain Management Science*, vol. 4, no. 1-2, pp. 1–24, 2023. https://doi.org/10.18757/jscms.2023.6986

[10] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[11] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," Ann. Stat., vol. 29, no. 5, pp. 1189–1232, 2001.

[12] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. New York, NY, USA: Springer, 2009.

[13] Z. He, D. Lin, T. Lau, and M. Wu, "Gradient Boosting Machine: A Survey," arXiv preprint arXiv:1908.06951, 2019. https://doi.org/10.48550/arXiv.1908.06951

[14] A. Géron, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc., 2022.

[15] Logwell.com, "Standard Resistor Values," Logwell.com. [Online]. Tersedia: https://www.logwell.com/tech/components/resistor_values.html. [Diakses: Jun. 6, 2025].

[16] Asenergi, "Standard Resistor Values E24," Asenergi.com. [Online]. Tersedia: https://asenergi.com/pdf/resistors/standard-resistor-values-e24.pdf. [Diakses: Jun. 6, 2025].

[17] ElectricalVolt. (2025). *Standard Resistor Series Values: E3, E6, E12, E24, E48, E96, E192*. [Online]. Tersedia: https://www.electricalvolt.com/standard-resistor-series-values-e3-e6-e12-e24-e48-e96/#h-resistors-e-series-values-and-tolerance-band. [Diakses: Jun. 6, 2025].

[18] "Resistor Chart: Comprehensive Guide to Resistor Values, E-Series, and Color Codes," *Wevolver*. [Online]. Available: https://www.wevolver.com/article/resistor-chart-comprehensive-guide-to-resistor-values-e-series-and-color-codes [Accessed: Jun. 7, 2025].

[19] A. V. Tatachar, "Comparative Assessment of Regression Models Based On Model Evaluation Metrics," International Research Journal of Engineering and Technology (IRJET), vol. 8, no. 09, pp. 853–860, Sep. 2021.

[20] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," PeerJ Computer Science., vol. 7, p. e623, 2021.

[21] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 3rd ed. Melbourne: OTexts, 2021. [Online]. Available: https://otexts.com/fpp3/

[22] S. Ben Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa, "A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition," *Neurocomputing*, vol. 73, no. 10-12, pp. 3428–3438, 2010. DOI:10.1016/j.eswa.2012.01.039