J. Ris. & Ap. Mat. Vol. 10, No. 1 (2026), pp. 13–28.
Jurnal Riset dan Aplikasi Matematika (JRAM)
e-ISSN: 2581-0154
DOI: doi.org/10.26740/jram.v10n1.p13-28

PERBANDINGAN KINERJA PREDIKSI FREKUENSI KLAIM ASURANSI KENDARAAN MENGGUNAKAN ARIMA VS ALGORITMA *RANDOM FOREST*

Bella Cindy Thalita*, Bintang Abyasa Arya Pradipta, Matthew Christiano, Feby Indriana Yusuf, dan Endang Wahyu Handamari

Program Studi Ilmu Aktuaria, Fakultas Matematika dan IPA, Universitas Brawijaya, Malang

*Penulis korespondensi: bellacindy@student.ub.ac.id

ABSTRAK

Frekuensi klaim asuransi kendaraan di Indonesia terus meningkat seiring dengan pertumbuhan jumlah kendaraan bermotor dan intensitas aktivitas berkendara, yang menimbulkan tantangan dalam manajemen risiko asuransi. Ketidakakuratan dalam memprediksi frekuensi klaim dapat menyebabkan ketidakseimbangan antara penetapan premi dan risiko aktual yang ditanggung perusahaan asuransi. Penelitian ini bertujuan untuk membandingkan kinerja model *Autoregressive Integrated Moving Average* (ARIMA) dan algoritma *Random Forest* dalam memprediksi frekuensi klaim asuransi kendaraan. Data yang digunakan berupa catatan historis klaim kendaraan dari repositori GitHub, yang dianalisis menggunakan pendekatan deret waktu untuk ARIMA dan pendekatan *machine learning* dengan *Random Forest*. Kinerja kedua model dievaluasi menggunakan metrik *Mean Absolute Percentage Error* (MAPE). Hasil penelitian menunjukkan bahwa model ARIMA(1,1,2) memberikan nilai MAPE sebesar 18,38%, sedangkan model *Random Forest* menghasilkan MAPE sebesar 19,87%. Hasil ini mengindikasikan bahwa model ARIMA memiliki kemampuan generalisasi yang lebih baik terhadap data pengujian, sementara *Random Forest* lebih unggul dalam menangani data pelatihan. Penelitian ini diharapkan dapat menjadi referensi bagi perusahaan asuransi dalam memilih pendekatan prediktif yang sesuai dengan karakteristik data klaim untuk mendukung pengambilan keputusan aktuaria secara lebih akurat dan efisien.
Kata kunci: ARIMA, Frekuensi Klaim Asuransi,*Random Forest*, *Machine Learning*, Prediksi

ABSTRACT

The frequency of vehicle insurance claims in Indonesia continues to increase in line with the growth of motor vehicle ownership and driving activity intensity, posing challenges for risk management in the insurance sector. Inaccurate prediction of claim frequency may lead to an imbalance between premium determination and the actual risk borne by insurance companies. This study aims to compare the performance of the Autoregressive Integrated Moving Average (ARIMA) model and the Random Forest algorithm in predicting vehicle insurance claim frequency. The dataset used consists of historical vehicle claim records obtained from a public GitHub repository, analyzed using a time series approach for ARIMA and a machine learning approach for Random Forest. The performance of both models was evaluated using the Mean Absolute Percentage Error (MAPE) metric. The results indicate that the ARIMA(1,1,2) model achieved a MAPE value of 18.38%, while the Random Forest model produced a MAPE of 19.87%. These findings suggest that the ARIMA model demonstrates better generalization on testing data, whereas the Random Forest model performs better on training data. This research provides valuable insights for

2020 Mathematics Subject Classification: 62M10, 62P05.
Diterima: 21-06-2025, direvisi: 20-10-2025, dimuat: 08-04-2026.

insurance companies in selecting predictive approaches that align with the characteristics of claim data to support more accurate and efficient actuarial decision-making.

Keywords: *ARIMA, Insurance Claim Frequency, Machine Learning, Random Forest, Prediction*

1. Pendahuluan

Secara global, frekuensi klaim asuransi kendaraan menunjukkan tren peningkatan yang selaras dengan pertumbuhan jumlah kendaraan bermotor serta intensitas aktivitas berkendara. Di Indonesia, data dari Korps Lalu Lintas Polri [1] mencatat bahwa jumlah kendaraan bermotor telah mencapai 168,8 juta unit, melebihi separuh populasi nasional, disertai dengan 1,15 juta kasus kecelakaan lalu lintas sepanjang tahun 2024. Namun, banyak dari kasus tersebut tidak memperoleh santunan sebagaimana mestinya, yang mencerminkan adanya tantangan signifikan dalam sistem manajemen risiko asuransi kendaraan [2]. Karakteristik data frekuensi klaim yang bersifat stokastik dan sangat volatil membuat proses pemodelan prediksi menjadi kompleks dan tidak dapat dilakukan secara sembarangan.

Model *Autoregressive Integrated Moving Average* (ARIMA) telah lama digunakan dalam berbagai studi untuk menangkap pola temporal pada data klaim [3]. Sementara itu, algoritma *Random Forest* menawarkan fleksibilitas serta kemampuan yang lebih adaptif dalam mengelola struktur data yang heterogen dan kompleks. Metode ini telah diterapkan secara luas dalam berbagai studi prediktif, seperti peramalan *gross domestic product* (GDP), prediksi penerimaan mahasiswa, hingga estimasi *return* aset digital [4].

Ketidakakuratan dalam memprediksi frekuensi klaim memiliki implikasi langsung terhadap pengambilan keputusan strategis di perusahaan asuransi, mulai dari penetapan premi, pengelolaan risiko, hingga pemenuhan modal risiko minimum. Model prediksi yang tidak akurat dapat menyebabkan *overpricing* yang menurunkan daya saing atau *underpricing* yang berisiko menimbulkan defisit pembayaran klaim. Dalam skala industri, akurasi prediksi juga turut memengaruhi stabilitas keuangan dan kepercayaan nasabah, terutama pada portofolio asuransi kendaraan yang sangat sensitif terhadap tren kecelakaan. Fakta bahwa banyak klaim tidak tersantuni memperkuat urgensi perlunya sistem prediksi yang andal. Oleh karena itu, pendekatan pemodelan yang tidak hanya akurat secara statistik tetapi juga tangguh dalam menghadapi dinamika *real-time* di lapangan menjadi kebutuhan utama, sebagaimana telah ditegaskan dalam berbagai studi sebelumnya [5] [6].

Berbagai pendekatan telah diajukan dalam studi terdahulu untuk memodelkan frekuensi klaim. Pendekatan *time series* seperti ARIMA-GARCH dan ARMA-GARCH telah menunjukkan performa yang baik untuk data yang bersifat linier [3]. Di sisi lain, pendekatan *machine learning* seperti *Random Forest* terbukti lebih unggul dalam menghadapi kompleksitas tinggi dan keberadaan *noise* dalam data [7] [8]. Namun demikian, masih terbatas studi yang secara eksplisit membandingkan performa ARIMA dan *Random Forest* dalam konteks prediksi frekuensi klaim asuransi kendaraan secara sistematis. Oleh karena itu, penelitian ini berupaya mengisi celah tersebut dengan membandingkan kinerja kedua pendekatan dalam memodelkan frekuensi klaim kendaraan bermotor di Indonesia.

Kebaruan dari penelitian ini terletak pada penerapan dan pembandingan sistematis antara model deret waktu klasik (ARIMA) dan algoritma *machine learning* (*Random Forest*) terhadap data frekuensi klaim asuransi kendaraan di Indonesia, sebuah konteks yang masih jarang dibahas dalam literatur domestik. Dengan menggunakan data empiris yang bersifat nonlinier dan volatil, penelitian ini tidak hanya menerapkan kedua metode tetapi juga membandingkan performanya secara kuantitatif menggunakan metrik *Mean Absolute Percentage Error* (MAPE). Hasil dari analisis ini memberikan kontribusi penting dalam menilai efektivitas pendekatan *machine le-*

arning dibandingkan metode deret waktu konvensional dalam konteks prediksi risiko aktuaria yang dinamis [9] [10].

2. Tinjauan Pustaka

2.1. Pemodelan ARIMA

2.1.1. Model ARIMA

Model *Autoregressive Integrated Moving Average* (ARIMA) adalah kelas model statistik yang digunakan untuk menganalisis dan memprediksi data deret waktu. Model ini sangat efektif untuk data yang menunjukkan ketidakstasioneran, yang dapat diatasi melalui proses diferensiasi (*differencing*). Model ARIMA(p, d, q) merupakan gabungan dari tiga komponen [9], yaitu: komponen pertama, *Autoregressive (AR)* orde p, yang menunjukkan bahwa nilai observasi saat ini bergantung secara linear pada p nilai observasi sebelumnya; komponen kedua, *Integrated (I)* orde d, yang menunjukkan jumlah proses diferensiasi yang diperlukan untuk membuat data deret waktu menjadi stasioner; dan komponen ketiga, *Moving Average (MA)* orde q, yang menunjukkan bahwa nilai observasi saat ini bergantung pada *error* atau *shock* dari q periode sebelumnya.

Secara matematis, model ARIMA(p, d, q) dapat diekspresikan menggunakan operator *backshift* (B), di mana $Z_t = Y_t - Y_{t-1}$, sebagai berikut:

$$\phi_p(B)(1 - B)^d Z_t = \mu + \theta_q(B)a_t \tag{1}$$

Dalam persamaan tersebut, Z_t merepresentasikan nilai observasi pada waktu t, sedangkan $(1 - B)^d$ adalah operator diferensiasi orde d yang diterapkan untuk mencapai stasioneritas. Polinomial *autoregressive* $\phi_p(B) = (1 - \phi_1 B - \ldots - \phi_p B^p)$ menangkap keterkaitan antara data pada nilai-nilai masa lalunya, sementara polinomial *moving average* $\theta_q(B) = (1 - \theta_1 B - \ldots - \theta_q B^q)$ memodelkan keteracakan pada *error* masa lalu. Notasi μ melambangkan konstanta model, dan a_t adalah *white noise error term* pada waktu t, yang diasumsikan independen dan berdistribusi identik dengan *mean* nol dan varians konstan.

2.1.2. Uji Stasioneritas

Stasioneritas adalah asumsi fundamental dalam analisis deret waktu, di mana suatu proses stokastik dikatakan stasioner jika sifat-sifat statistiknya tidak berubah seiring waktu. Secara visual, data stasioner berfluktuasi di sekitar rata-rata yang konstan dengan varians yang juga konstan. Salah satu metode untuk menguji stasioneritas dalam varians adalah melalui transformasi Box-Cox, di mana data dianggap stasioner jika parameter λ mendekati 1 [11]. Apabila data terbukti tidak stasioner, salah satu teknik untuk menstabilkan varians adalah melalui Transformasi Box-Cox, yang didefinisikan sebagai:

$$T(Z_t) = \frac{Y_t^{\lambda} - 1}{\lambda} \tag{2}$$

di mana λ adalah parameter transformasi yang diestimasi.

Untuk menguji stasioneritas dalam rata-rata, salah satu metode yang paling umum digunakan adalah uji *Augmented Dickey-Fuller* (ADF). Uji ADF bekerja dengan mendeteksi keberadaan akar unit (*unit root*), yang merupakan indikator ketidakstasioneran. Model regresi untuk uji ADF dapat direpresentasikan sebagai berikut:

$$\Delta y_t = \alpha + \rho y_{t-1} + \sum_{i=1}^{p-1} \delta_i \Delta y_{t-i} + \beta t + \varepsilon_t \tag{3}$$

Hipotesis yang diuji adalah:

H_0: $\rho = 1$ (atau $\delta = 0$): terdapat *unit root* (data tidak stasioner).
H_1: $\rho < 1$ (atau $\delta < 0$): tidak ada *unit root* (data stasioner).
Statistik uji-t dihitung untuk koefisien ρ (atau δ), dengan rumus:

$$t = \frac{\delta}{\text{se}(\delta)} \tag{4}$$

Hipotesis nol (H_0) ditolak jika nilai statistik uji-t lebih kecil dari nilai kritis pada tabel Dickey-Fuller, atau jika nilai-p (*p-value*) lebih kecil dari tingkat signifikansi (α) yang ditentukan. Penolakan H_0 menandakan bahwa data bersifat stasioner dalam rata-rata. Apabila data tidak stasioner dalam rata-rata perlu dilakukan *differencing*:

2.1.3. Uji Signifikansi Parameter

Setelah estimasi model awal dilakukan, tahapan kritis selanjutnya adalah menguji signifikansi statistik dari setiap parameter yang diestimasi untuk menentukan pengaruh substantifnya terhadap model [11]. Uji t digunakan untuk mengevaluasi apakah koefisien parameter secara statistik berbeda signifikan dari nol.
Hipotesis yang diuji adalah:
H_0: $\rho_j = 0$ (parameter tidak signifikan secara statistik).
H_1: $\rho_j \neq 0$ (parameter signifikan secara statistik).
Statistik uji dihitung menggunakan formula:

$$t = \frac{\hat{\rho}_j}{se(\hat{\rho}_j)} \tag{5}$$

Dalam persamaan ini, $\hat{\rho}_j$ adalah nilai estimasi untuk parameter ke-j, dan $\text{se}(\hat{\rho}_j)$ adalah standar *error* dari estimasi tersebut. Keputusan untuk menolak atau gagal menolak hipotesis nol (H_0) didasarkan pada perbandingan nilai statistik uji-t dengan nilai kritisnya atau dengan membandingkan nilai-p terhadap tingkat signifikansi (α) yang telah ditentukan. Parameter dianggap signifikan jika H_0 ditolak.

2.1.4. Uji White Noise dengan Statistik Ljung-Box

Salah satu asumsi fundamental dalam pemodelan ARIMA adalah bahwa *residual* (sisa) dari model harus bersifat *white noise*, yaitu tidak memiliki pola autokorelasi. Untuk memverifikasi asumsi ini, digunakan uji Ljung-Box yang menguji hipotesis nol bahwa tidak ada autokorelasi dalam deret *residual* hingga lag tertentu [12].
Statistik uji Ljung-Box (Q) dihitung dengan formula:

$$Q = n(n+2)\sum_{k=1}^{K}\frac{\hat{\rho}_k^2}{n-k} \tag{6}$$

Hipotesis yang diuji adalah:
H_0: $\rho_1 = \rho_2 = \ldots = \rho_K = 0$ (*residual* bersifat independen; tidak ada autokorelasi).
H_1: Terdapat setidaknya satu $\rho_k \neq 0$ (*residual* tidak independen; terdapat autokorelasi).
Statistik Q mengikuti distribusi Chi-kuadrat (χ^2) dengan derajat kebebasan ($K - p$), di mana p adalah jumlah parameter estimasi pada lag ke-p. Hipotesis nol ditolak jika statistik Q melebihi nilai kritis distribusi χ^2 pada tingkat signifikansi (α) tertentu.

2.1.5. *Akaike Information Criteria* (AIC)

Dalam situasi di mana terdapat beberapa model kandidat yang valid secara statistik, diperlukan kriteria untuk memilih model yang paling *parsimonious* (hemat parameter) namun tetap memiliki daya suai (*goodness of fit*) yang baik. Salah satu kriteria yang paling umum digunakan

adalah *Akaike Information Criterion* (AIC), yang menyeimbangkan antara kompleksitas model dan kesesuaiannya dengan data [9]. AIC dihitung dengan formula:

$$AIC = 2m - 2\ln(L) \tag{7}$$

Dalam persamaan ini, m adalah jumlah total parameter yang diestimasi dalam model, dan L adalah nilai maksimum dari fungsi likelihood model. Sesuai dengan prinsip parsimoni, model dengan nilai AIC terkecil dianggap sebagai model terbaik di antara kandidat yang ada.

2.2. *Random Forest*

2.2.1. Rekayasa Fitur (*Feature Engineering*)

Rekayasa fitur (*feature engineering*) adalah tahap krusial dalam siklus pengembangan model *machine learning*. Tujuannya adalah mengekstraksi dan memanipulasi fitur mentah (*raw features*) menjadi representasi data yang lebih informatif dan relevan bagi algoritma [13]. Proses ini mencakup berbagai teknik, seperti identifikasi pola tersembunyi, pembentukan fitur baru dari kombinasi fitur yang sudah ada, atau transformasi fitur agar lebih sesuai dengan asumsi model. Dalam konteks prediksi frekuensi klaim, rekayasa fitur dapat dilakukan dengan mengubah tanggal kejadian menjadi jumlah klaim per tanggal (*claim frequency*), yang merepresentasikan intensitas klaim pada periode tertentu.

2.2.2. Pensklalaan Fitur (*Feature Scaling*)

Penskalaan fitur (*feature scaling*) adalah teknik pra-pemrosesan yang bertujuan untuk menyeragamkan rentang nilai pada fitur-fitur numerik suatu dataset [14]. Prosedur ini esensial karena beberapa algoritma *machine learning* sensitif terhadap perbedaan skala, yang dapat menyebabkan fitur dengan rentang nilai lebih besar mendominasi proses pembelajaran. Metode penskalaan umum seperti standarisasi (Z-score) atau normalisasi (Min-Max), memastikan bahwa setiap fitur memberikan kontribusi yang seimbang terhadap model. Hal ini berpotensi meningkatkan kinerja dan stabilitas model secara keseluruhan.

2.2.3. Seleksi Fitur (*Feature Selection*)

Seleksi fitur (*feature selection*) merupakan proses sistematis untuk mengidentifikasi dan memilih subset fitur yang paling informatif dari himpunan fitur yang tersedia [15]. Tujuan utamanya adalah untuk: (1) menyederhanakan model dengan mengurangi dimensionalitas data, (2) memitigasi risiko *overfitting* dengan mengeliminasi fitur yang tidak relevan atau redundan, serta (3) meningkatkan interpretabilitas model dan efisiensi komputasi. Terdapat beragam metode seleksi fitur, mulai dari teknik statistik sederhana hingga pendekatan kompleks yang mengevaluasi kinerja model berdasarkan berbagai kombinasi subset fitur. Dalam penelitian ini, seleksi fitur dilakukan berdasarkan nilai feature importance yang dihasilkan oleh model *Random Forest*, untuk mengidentifikasi fitur paling berpengaruh terhadap frekuensi klaim.

2.2.4. Model Matematis

Secara matematis, *Random Forest* membangun sebuah ensambel yang terdiri dari B pohon keputusan $\{T_1, T_2, \ldots, T_B\}$ [10]. Setiap pohon T dilatih menggunakan sampel *bootstrap* yang ditarik dari data latih asli. Untuk sebuah data input $x \in \mathbb{R}^p$, setiap pohon menghasilkan prediksinya sendiri, yaitu $\hat{y} = T(x)$. Prediksi akhir dari keseluruhan *forest* ($\hat{y}$) diperoleh dengan mengagregasi hasil dari semua pohon.

Untuk tugas klasifikasi, agregasi dilakukan melalui mekanisme *majority voting*, di mana kelas yang paling sering muncul menjadi prediksi akhir:

$$\hat{y} = \text{mode}(T_1(x), T_2(x), \ldots, T_B(x))$$

Sementara itu, untuk regresi, prediksi akhir adalah rata-rata dari prediksi seluruh pohon:

$$\hat{y} = \frac{1}{B} \sum_{b=1}^{B} T_b(x) \tag{8}$$

Proses pemisahan (*splitting*) pada setiap node bertujuan untuk meminimalkan ketidakmurnian (*impurity*) dari node induk. Hal ini dicapai dengan memilih fitur dan nilai ambang batas yang memberikan penurunan *impurity* terbesar. Pada regresi, *impurity* diukur menggunakan varians dari target sebagaimana ditunjukkan oleh Persamaan (2.1).

$$I(t) = \frac{1}{N_t} \sum_{i \in t} (y_i - \bar{y}_t)^2 \tag{9}$$

di mana $I(t)$ adalah *impurity* pada node t, N_t adalah jumlah sampel pada node t, y_i adalah nilai aktual, dan $\bar{y}_t$ adalah rata-rata nilai target di node t. Node dengan *impurity* paling kecil dianggap paling homogen (murni).

Penurunan *impurity* (ΔI) dihitung dengan:

$$\Delta I = I(t) - \left(\frac{n_L}{n} I(t_L) + \frac{n_R}{n} I(t_R) \right) \tag{10}$$

Dalam persamaan ini, $I(t)$ adalah *impurity* pada node induk, sementara $I(t_L)$ dan $I(t_R)$ adalah *impurity* pada node turunan kiri dan kanan. Notasi n adalah jumlah total sampel pada node induk, dengan n_L dan n_R masing-masing adalah jumlah sampel pada node turunan kiri dan kanan.

2.3. Model Evaluasi

Mean Absolute Percentage Error (MAPE) merupakan salah satu indikator utama untuk mengevaluasi ketepatan hasil peramalan. Menurut penelitian dalam studi komparasi metode peramalan, MAPE memberikan keunggulan dalam menyajikan kesalahan prediksi dalam bentuk persentase yang mudah diinterpretasikan [15].

MAPE mengukur rata-rata kesalahan absolut prediksi relatif terhadap nilai aktual. Rumus dasar MAPE diberikan oleh:

$$\text{MAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \left| \frac{X_t - F_t}{X_t} \right| \tag{11}$$

Dalam persamaan tersebut, n adalah jumlah total periode observasi, X_t adalah nilai aktual pada periode t, dan F_t adalah nilai prediksi pada periode t. Komponen dasar dari perhitungan MAPE adalah *Percentage Error* (PE) untuk setiap titik data. Nilai MAPE yang lebih rendah mengindikasikan tingkat kesalahan prediksi yang lebih kecil, yang berarti model memiliki akurasi yang lebih tinggi.

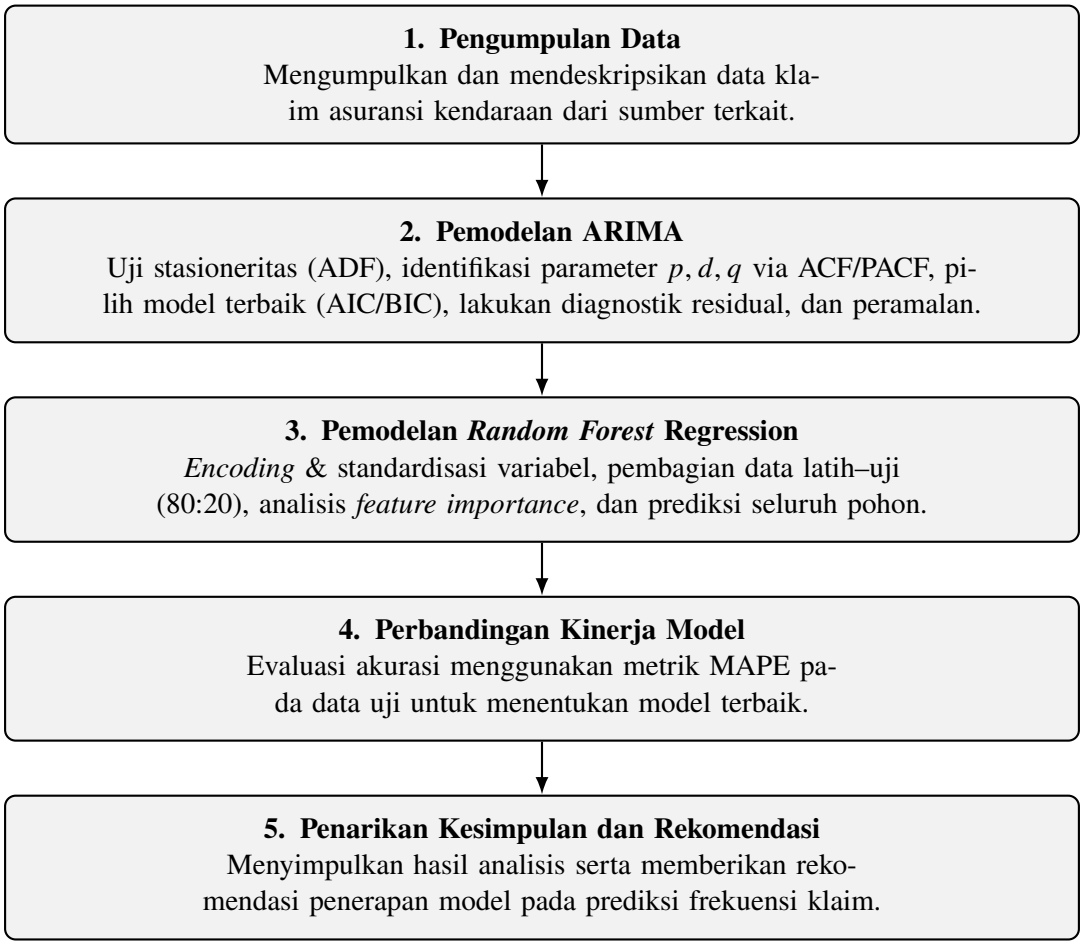
3. Metode Penelitian

Penelitian ini menggunakan data klaim asuransi kendaraan yang diperoleh dari repositori GitHub (https://github.com/RakeshHansrajani/Insurance_Data_Analysis), yang diakses pada 14 Mei 2025, dan bertujuan untuk membandingkan akurasi prediksi frekuensi klaim. menggunakan dua pendekatan yaitu model deret waktu ARIMA dan algoritma *machine learning Random Forest*. Variabel dependen yang digunakan adalah frekuensi klaim (Y), yaitu jumlah klaim asuransi kendaraan yang diajukan oleh nasabah selama periode tertentu. Sementara itu, variabel independen mencakup berbagai fitur pada data klaim kendaraan yang diduga memengaruhi frekuensi

klaim tersebut. Variabel-variabel tersebut antara lain: lama menjadi nasabah (X_1), usia (X_2), jenis kelamin (X_3), tingkat pendidikan (X_4), pekerjaan (X_5), hobi (X_6), dan hubungan tertanggung dengan pemegang polis (X_7). Selain itu, terdapat karakteristik polis yang meliputi nomor polis (X_8), tanggal mulai berlaku polis (X_9), wilayah penerbitan polis (X_{10}), batas tanggungan (X_{11}), nilai tanggungan pribadi (X_{12}), premi tahunan (X_{13}), dan batas maksimum tanggungan tambahan (X_{14}). Fitur lain yang juga digunakan mencakup informasi kejadian klaim, seperti tanggal kejadian (X_{15}), jenis kejadian (X_{16}), jenis tabrakan (X_{17}), tingkat keparahan insiden (X_{18}), pihak berwenang yang dihubungi (X_{19}), lokasi kejadian (X_{20}), serta jumlah kendaraan yang terlibat (X_{21}). Selain itu, terdapat pula variabel terkait dampak kejadian seperti kerusakan properti (X_{22}), cedera tubuh (X_{23}), keberadaan saksi (X_{24}), dan laporan polisi (X_{25}). Nilai klaim yang diajukan juga diperhitungkan, baik total klaim (X_{26}) maupun berdasarkan jenis kerugian, yaitu klaim cedera (X_{27}), klaim properti (X_{28}), dan klaim kendaraan (X_{29}). Karakteristik kendaraan seperti merek (X_{30}), model (X_{31}), dan tahun pembuatan (X_{32}) turut menjadi variabel yang diamati. Terakhir, terdapat variabel pelaporan penipuan (X_{33}) yang menunjukkan apakah suatu klaim terindikasi sebagai klaim palsu atau tidak. Tahapan analisis yang dilakukan dalam penelitian ini adalah sebagai berikut:

1. Pengumpulan Data: Mengumpulkan dan mendeskripsikan data klaim asuransi kendaraan dari sumber terkait.
2. Pemodelan ARIMA
    - Melakukan uji stasioneritas data menggunakan uji *Augmented Dickey-Fuller* (ADF), yang didasarkan pada regresi dalam Persamaan (3) dan statistik uji dalam Persamaan (4).
    - Mengidentifikasi parameter model ARIMA (p, d, q) melalui analisis pola autokorelasi dan autokorelasi parsial, sesuai dengan struktur model ARIMA pada Persamaan (1).
    - Membangun model ARIMA dengan parameter terpilih, dan menuliskan bentuk akhir model sesuai formulasi umum dari Persamaan (1).
    - Melakukan uji diagnostik *residual* untuk memastikan *residual* bersifat white noise, dengan pengujian autokorelasi menggunakan statistik Ljung-Box seperti pada Persamaan (6).
    - Mengevaluasi performa model menggunakan MAPE berdasarkan Persamaan (11).
    - Melakukan proses peramalan jangka pendek berdasarkan model ARIMA yang telah dibangun, serta menyusun interval kepercayaan untuk menggambarkan ketidakpastian prediksi.
3. Pemodelan *Random Forest Regression*
    - Melakukan *encoding* dan standarisasi variabel untuk pemodelan *Random Forest*.
    - Membagi data menjadi data latih dan data uji dengan rasio 80:20.
    - Menghitung *impurity node* menggunakan Persamaan (9) dan menentukan pemisahan terbaik berdasarkan penurunan *impurity* dengan Persamaan (10).
    - Mengidentifikasi 5 variabel paling berpengaruh terhadap frekuensi klaim menggunakan *feature importance*.
    - Membangun model *Random Forest* menggunakan 5 fitur terpilih dan melakukan agregasi hasil prediksi seluruh pohon menggunakan Persamaan (8).
    - Mengevaluasi performa model menggunakan MAPE berdasarkan Persamaan (11).
4. Perbandingan Kinerja Model: Membandingkan performa model ARIMA dan *Random Forest* berdasarkan nilai akurasi prediksi (MAPE) untuk menentukan model terbaik.

5. Penarikan Kesimpulan: Menarik kesimpulan dari hasil analisis dan memberikan rekomendasi berdasarkan performa model terbaik dalam memprediksi frekuensi klaim asuransi kendaraan.



4. Hasil dan Pembahasan

4.1. Pemodelan ARIMA

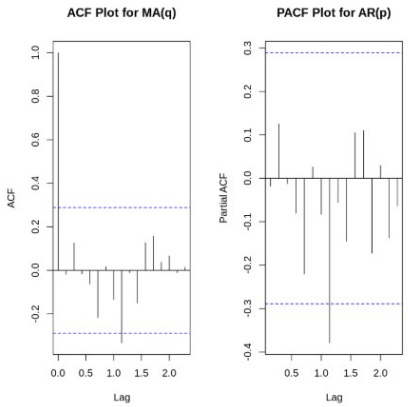
Pada penelitian ini digunakan data historis analisis klaim asuransi kendaraan, mulai tanggal 1 Januari 2015 sampai dengan 1 Maret 2015 dengan menggunakan interval waktu per hari. Data yang digunakan mencakup Tanggal kejadian *(Incident Date)* untuk peramalan menggunakan ARIMA.

4.1.1. Persiapan Data dan Stasioneritas

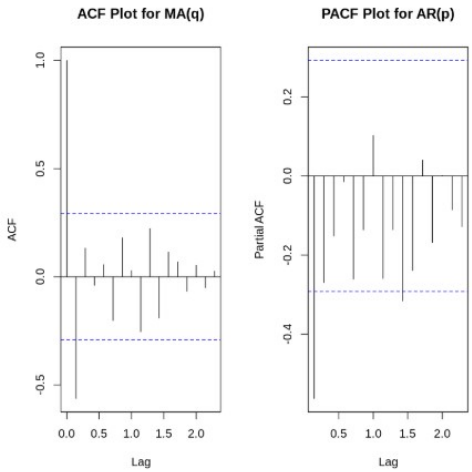
Data dibagi menjadi 46 hari data pelatihan dan 14 hari data pengujian. Pembagian ini bertujuan untuk mengevaluasi kemampuan model dalam menangkap pola musiman mingguan dan memvalidasi prediksi jangka pendek.

Gambar 1 menampilkan plot data pelatihan, menunjukkan fluktuasi harian jumlah klaim yang tinggi, berkisar antara 8 hingga 28 klaim. Meskipun bervariasi, plot ini mengindikasikan pola musiman mingguan yang terlihat jelas, diperkuat oleh garis tren kurva putus-putus berwarna merah. Sementara itu, Gambar 2 memvisualisasikan data pengujian selama 14 hari pasca-pelatihan. Pola fluktuasi masih ada, namun tidak se-volatil data pelatihan. Terlihat kecenderungan penurunan klaim di awal dan akhir periode pengujian, dengan sedikit peningkatan di pertengahan minggu kedua. Garis tren putus-putus berwarna biru pada Gambar 2 membantu mengidentifikasi tren umum pola klaim selama periode ini.

Untuk memastikan akurasi model, stasioneritas data diperiksa menggunakan plot ACF dan PACF pada Gambar 3 serta uji *Augmented Dickey-Fuller* (ADF). Uji ADF awal (p-value



Gambar 1. Plot ACF dan PACF Data Awal



Gambar 2. Plot ACF dan PACF Setelah *Differencing*

= 0,09302) mengindikasikan data tidak stasioner, sehingga dilakukan transformasi *differencing*. Setelah *differencing*, data menjadi stasioner, dikonfirmasi oleh uji ADF (p-value = 0,01) dan plot ACF/PACF yang baru pada Gambar 4.

4.1.2. Pemilihan dan Diagnostik Model

Penentuan kandidat model ARIMA didasarkan pada analisis plot ACF dan PACF. Model-model potensial yang dipertimbangkan antara lain ARIMA(1,1,1), ARIMA(2,1,2), ARIMA(2,1,1), ARIMA(1,1,2), dan ARIMA(0,1,1). Model terbaik kemudian dipilih berdasarkan kriteria *Akaike Information Criterion* (AIC) dan *Bayesian Information Criterion* (BIC), di mana nilai terkecil menunjukkan model yang paling sesuai.

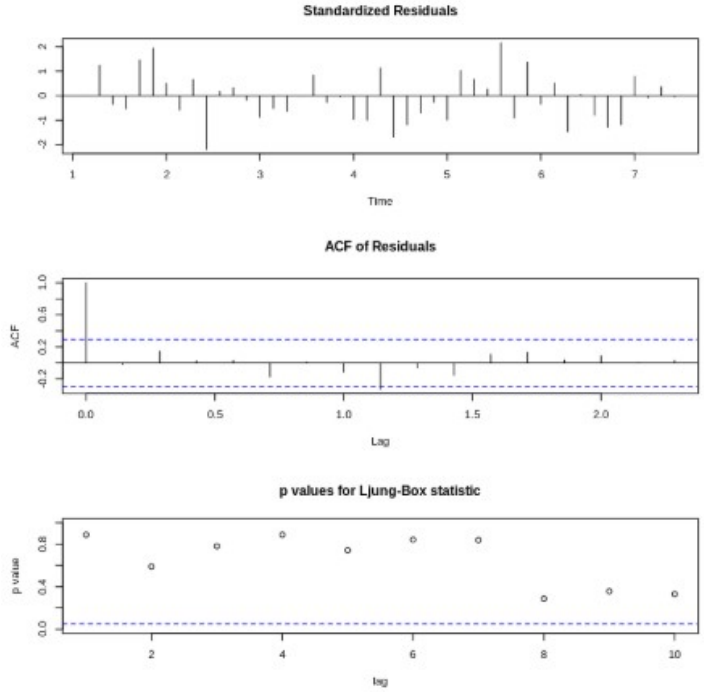
Tabel 1. Nilai AIC dan BIC Model-model ARIMA

Model	AIC	BIC
ARIMA (1,1,1)	289.20	294.55
ARIMA (2,1,2)	285.28	294.20
ARIMA (2,1,1)	288.46	295.60
ARIMA (1,1,2)	**284.74**	**2918.851**
ARIMA (0,1,1)	303.35	3069.231
ARIMA (1,1,4)	286.96	2976.712

Berdasarkan hasil perbandingan nilai AIC dan BIC, model terbaik yang dipilih adalah **ARIMA(1,1,2)** dengan nilai AIC terkecil yaitu **284.7484**. Bentuk model ARIMA(1,1,2) yang diperoleh secara umum dapat dituliskan sebagai:

$$Y_t = Y_{t-1} + \phi_1(Y_{t-1} - Y_{t-2}) + \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2}$$

Di mana Y_t adalah nilai frekuensi klaim pada waktu ke-t, dan ε_t adalah *residual* (*white noise*).

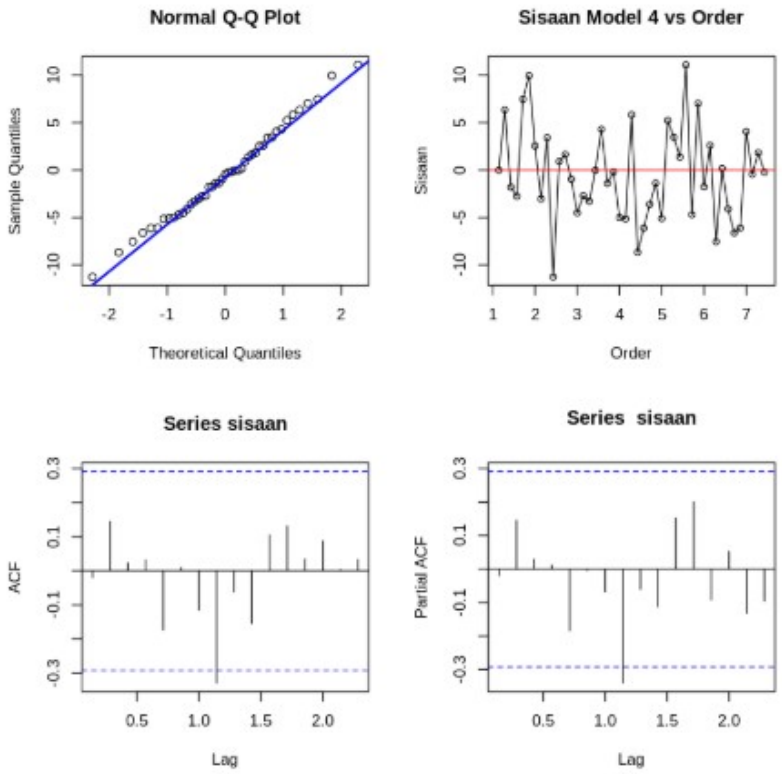


Gambar 3. Hasil Eksplorasi Sisaan

Gambar 5 menunjukkan eksplorasi sisaan model. Plot *residual* terstandarisasi menunjukkan distribusi acak tanpa pola jelas. ACF *residual* tidak menunjukkan autokorelasi signifikan, dan uji Ljung-Box menunjukkan nilai p yang tinggi pada hampir semua *lag*, mengindikasikan sisaan model bersifat *white noise*.

Model terpilih kemudian menjalani serangkaian uji diagnostik *residual* secara ketat (Gambar 5 dan Gambar 6). Hasil analisis menunjukkan bahwa *residual* berfluktuasi di sekitar nol tanpa pola yang jelas, serta tidak menunjukkan autokorelasi signifikan, sebagaimana ditunjukkan oleh uji Ljung-Box yang menghasilkan *p-value* sebesar 0,8872. Nilai *p-value* yang tinggi (di atas 0,05) menunjukkan bahwa tidak terdapat cukup bukti untuk menolak hipotesis nol, yang berarti *residual* tidak memiliki autokorelasi secara signifikan. Selanjutnya, uji Jarque-Bera menghasilka *p-value* sebesar 0,737, yang juga melebihi batas signifikansi 0,05. Hal ini mengindikasikan bahwa *residual* dapat dianggap mengikuti distribusi normal, karena tidak ada cukup bukti untuk menyatakan sebaliknya. Begitu juga, uji-t terhadap rata-rata *residual* memberikan *p-value* sebesar 0,5712, yang menyiratkan bahwa rata-rata *residual* tidak berbeda signifikan dari nol. Artinya, *error* yang dihasilkan model bersifat seimbang dan tidak bias terhadap nilai tertentu.

Secara keseluruhan, *p-value* yang tinggi pada ketiga pengujian ini menunjukkan bahwa asumsi-asumsi dasar dari model *time series* telah terpenuhi, yakni tidak ada autokorelasi, *residual* terdistribusi normal, dan rata-ratanya nol. Akibatnya, model dianggap layak, stabil, dan valid untuk digunakan dalam proses peramalan, karena tidak menunjukkan masalah diagnostik yang berarti.



Gambar 4. Q-Q Plot

4.1.3. Peramalan dan Evaluasi

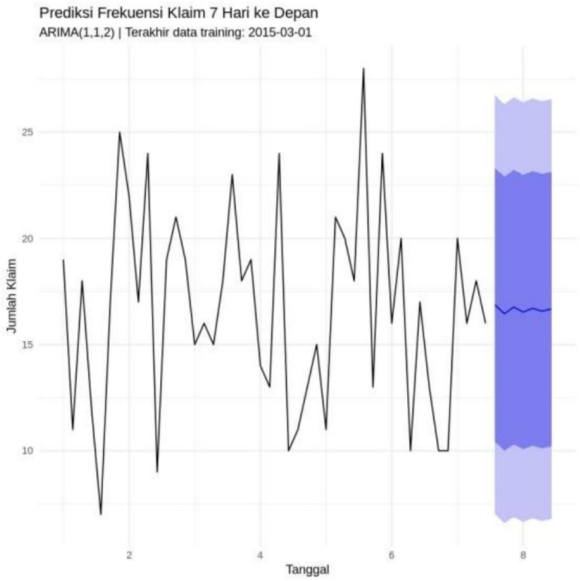
Setelah diperoleh model ARIMA(1,1,2) sebagai model terbaik berdasarkan kriteria AIC serta telah melalui pengujian diagnostik *residual*, langkah selanjutnya adalah melakukan proses *forecasting* terhadap data frekuensi klaim untuk beberapa hari ke depan.

Tabel 2. Hasil Peramalan 7 Hari ke Depan

No	Tanggal	Aktual	Ramalan
1	2015-02-16	16	18.87
2	2015-02-17	26	16.45
3	2015-02-18	25	16.57
4	2015-02-19	10	16.53
5	2015-02-20	14	17.40
6	2015-02-21	19	16.74
7	2015-02-22	17	16.67

Akurasi model diukur menggunakan *Mean Absolute Percentage Error* (MAPE), menghasilkan nilai 18,38%. Ini menunjukkan kinerja prediksi yang cukup baik mengingat variabilitas data klaim. Plot hasil peramalan pada Gambar 7 menunjukkan kemampuan model untuk mengikuti pola fluktuasi data aktual.

Pemodelan yang dilakukan menghasilkan prediksi frekuensi klaim harian dengan nilai yang relatif stabil, yaitu berkisar antara 16 hingga 17 klaim per hari, serta disertai interval kepercayaan 80% dan 95% yang konsisten (antara 7 hingga 27 klaim). Hal ini menunjukkan bahwa model mampu memberikan estimasi yang realistis dan dapat diandalkan dalam konteks peramalan jangka pendek.



Gambar 5. Plot Hasil Peramalan

Tabel 3. Interval Kepercayaan Ramalan

No	Tanggal	Prediksi Klaim	Bawah 80	Atas 80	Atas 95
1	3/2/2015	17	10	23	27
2	3/3/2015	16	10	23	26
3	3/4/2015	17	10	23	27
4	3/5/2015	17	10	23	26
5	3/6/2015	17	10	23	26
6	3/7/2015	17	10	23	26
7	3/8/2015	17	10	23	27

Implikasi praktis dari hasil ini adalah bahwa perusahaan asuransi dapat menggunakan *output* model untuk merencanakan alokasi sumber daya secara lebih efisien, seperti menentukan cadangan klaim harian, mempersiapkan kapasitas pelayanan, dan mengelola risiko operasional. Selain itu, interval kepercayaan yang sempit dan konsisten menunjukkan bahwa model memiliki kestabilan prediktif yang baik, yang penting untuk pengambilan keputusan berbasis data.

4.2. Prediksi *Random Forest*

4.2.1. Persiapan dan Implementasi Data

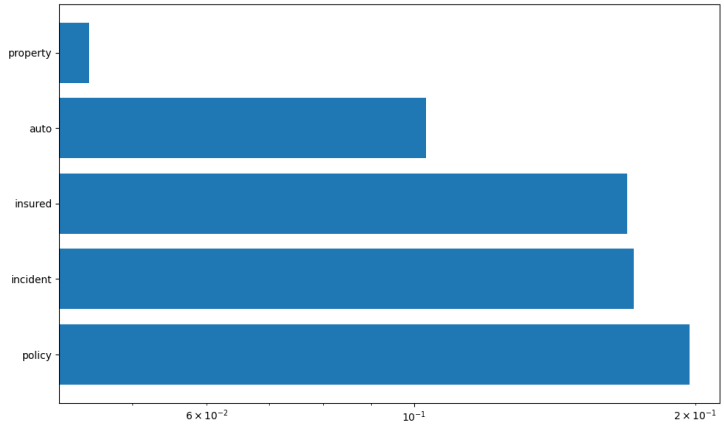
Proses persiapan data untuk model *Random Forest* diawali dengan membagi dataset menjadi 80% data latih dan 20% data uji menggunakan fungsi `train_test_split`. Variabel target `claim_frequency` dihitung berdasarkan frekuensi kemunculan masing-masing nilai pada kolom `incident_date`, yang merepresentasikan jumlah klaim per tanggal insiden. Data kemudian dipisahkan menjadi fitur independen (`X_raw`) dan target dependen (`y_raw`) sebelum memasuki tahap *preprocessing*.

Tahap *preprocessing* mencakup penanganan nilai hilang dengan mengganti nilai kosong pada kolom kategorikal menggunakan modus, dan pada kolom numerik menggunakan rata-rata. Fitur kategorikal ordinal seperti `policy_csl`, `incident_severity`, dan `insured_education_level` dikodekan menggunakan *ordinal encoding*, sementara fitur nominal lainnya dikodekan menggunakan *one-hot encoding* tanpa parameter `drop_first=True`, sehingga seluruh kategori tetap digunakan. Seluruh fitur hasil encoding kemudian diskalakan menggunakan `StandardScaler`

untuk memastikan distribusi data seragam sebelum digunakan dalam pelatihan model *Random Forest*.

4.2.2. Analisis Fitur Penting

Analisis fitur penting pada model *Random Forest* mengungkap bahwa beberapa kelompok fitur seperti `policy`, `incident`, dan `insured` memiliki nilai *importance* yang relatif tinggi dibandingkan fitur lainnya. Ketiga fitur tersebut berada dalam skala logaritmik sekitar 10^{-1}, menunjukkan kontribusi signifikan dalam proses prediksi frekuensi klaim. Sebaliknya, fitur seperti `property` memiliki nilai *importance* yang jauh lebih kecil (sekitar 10^{-2}), yang mengindikasikan pengaruh prediktif yang sangat rendah. Gambar 6 menunjukkan bahwa fitur `policy` merupakan



Gambar 6. Top 5 Fitur (Skala Logaritmik)

fitur dengan *importance* tertinggi, disusul oleh `incident` dan `insured`, yang secara kolektif berkontribusi besar terhadap performa model. Dominasi tiga fitur utama ini mengindikasikan bahwa model sangat bergantung pada atribut-atribut tersebut dalam menghasilkan prediksi yang akurat. Oleh karena itu, penyederhanaan model melalui *feature selection* dapat dipertimbangkan guna meningkatkan efisiensi dan interpretabilitas tanpa mengorbankan akurasi secara signifikan.

Meskipun begitu, fitur dengan kontribusi rendah tidak serta-merta dapat diabaikan. Dalam kondisi tertentu, fitur-fitur tersebut mungkin tetap menyimpan informasi tambahan yang relevan. Menghapusnya tanpa pertimbangan yang matang berisiko mengurangi cakupan informasi yang dimiliki model, sehingga berpotensi menurunkan kinerja saat diterapkan pada data yang lebih bervariasi di luar data pelatihan.

4.2.3. Evaluasi Model dan Hasil Prediksi

Model *Random Forest* diterapkan untuk memprediksi frekuensi klaim asuransi kendaraan menggunakan sejumlah parameter yang telah ditentukan. Tabel 4 berikut menyajikan konfigurasi parameter yang digunakan dalam proses pelatihan model.

Tabel 4. Konfigurasi Parameter *Random Forest*

Parameter	**Nilai**
n_estimators	100
random_state	42
max_depth	None
min_samples_split	2
min_samples_leaf	1
n_jobs	-1

Setelah model dilatih dengan menggunakan lima fitur terpenting, evaluasi dilakukan terhadap hasil prediksi menggunakan data uji. Tabel 5 menunjukkan perbandingan antara nilai aktual dan hasil prediksi model pada beberapa sampel, serta selisih prediksi terhadap nilai sebenarnya. Perbedaan nilai ini memberikan gambaran sejauh mana model mampu merepresentasikan data aktual dengan hanya menggunakan fitur yang terbatas. Dari Tabel 5, terlihat bahwa hasil

Tabel 5. Perbandingan Prediksi vs Aktual (Top 5 Fitur)

Aktual (y_*test*)	**Prediksi**	**Selisih**
10	18.34	-8.34
18	17.75	0.25
15	17.62	-2.62
18	19.49	-1.49
20	15.99	4.01

prediksi model tidak selalu mendekati nilai aktual. Terdapat selisih yang cukup besar pada beberapa sampel, seperti selisih sebesar -8,34 pada nilai aktual 10. Hal ini menunjukkan bahwa meskipun model hanya menggunakan lima fitur terpenting, akurasinya belum sepenuhnya stabil dan masih terdapat deviasi pada kasus tertentu. Berdasarkan perhitungan, nilai *Mean Absolute Percentage Error* (MAPE) yang dihasilkan sebesar 19,87%, yang menunjukkan bahwa rata-rata kesalahan relatif prediksi model berada di bawah 20%. Nilai ini cukup dapat diterima, namun masih menyisakan ruang untuk peningkatan melalui penyesuaian fitur atau model lebih lanjut.

4.3. Perbandingan Performa Model ARIMA dan *Random Forest*

Perbandingan dua model peramalan, ARIMA dan *Random Forest*, dilakukan menggunakan metrik *Mean Absolute Percentage Error* (MAPE).

Tabel 6. Perbandingan Nilai MAPE antara Model *Random Forest* dan ARIMA

Model	***Train*(%)**	***Test*(%)**
Random Forest	8,73	19,87
ARIMA	27,37	18,38

Berdasarkan Tabel 6, hasil perbandingan nilai *Mean Absolute Percentage Error* (MAPE) menunjukkan bahwa model *Random Forest* memiliki nilai kesalahan yang lebih rendah pada data pelatihan, yaitu sebesar 8,73%, dibandingkan model ARIMA yang mencapai 27,37%. Namun, pada data pengujian, model ARIMA menunjukkan performa yang sedikit lebih baik dengan nilai MAPE sebesar 18,38%, sedangkan *Random Forest* memiliki nilai 19,87%. Hal ini menunjukkan bahwa model ARIMA lebih mampu melakukan generalisasi terhadap data baru, sedangkan *Random Forest* cenderung lebih baik dalam menyesuaikan data pelatihan namun berpotensi mengalami *overfitting*.

4.3.1. Implikasi Peramalan

Hasil peramalan frekuensi klaim asuransi kendaraan memiliki implikasi penting terhadap pengambilan keputusan strategis di perusahaan asuransi. Model ARIMA yang mampu memberikan estimasi stabil dan realistis dalam jangka pendek dapat dimanfaatkan untuk memproyeksikan jumlah klaim harian, sehingga membantu perusahaan dalam menentukan cadangan klaim, mengatur kapasitas pelayanan, serta mengoptimalkan alokasi sumber daya. Di sisi lain, model *Random Forest* yang unggul dalam menangkap hubungan nonlinier antar variabel dapat digunakan untuk analisis yang lebih kompleks, seperti identifikasi faktor-faktor penyebab peningkatan klaim dan pengelolaan risiko berbasis data historis yang beragam.

Kombinasi kedua pendekatan ini memberikan pandangan yang komprehensif: ARIMA efektif untuk peramalan jangka pendek dengan pola temporal yang jelas, sedangkan *Random Forest* lebih adaptif untuk data yang kompleks dan heterogen. Dengan demikian, hasil peramalan dari kedua model ini dapat dijadikan dasar dalam penyusunan strategi mitigasi risiko, perencanaan premi yang lebih akurat, serta pengambilan keputusan operasional yang efisien.

5. Kesimpulan

Penelitian ini bertujuan untuk membandingkan akurasi model *Autoregressive Integrated Moving Average* (ARIMA) dan algoritma *Random Forest* dalam memprediksi frekuensi klaim asuransi kendaraan di Indonesia. Berdasarkan hasil analisis, model ARIMA(1,1,2) menghasilkan nilai *Mean Absolute Percentage Error* (MAPE) sebesar 18,38%, sedangkan model *Random Forest* memperoleh MAPE sebesar 19,87%. Hasil ini menunjukkan bahwa model ARIMA memiliki kemampuan generalisasi yang lebih baik terhadap data pengujian, sedangkan *Random Forest* lebih unggul dalam menangani data pelatihan dengan nilai MAPE yang lebih rendah. Secara keseluruhan, ARIMA lebih sesuai digunakan untuk data dengan pola temporal yang linier dan stabil, sedangkan *Random Forest* lebih adaptif terhadap struktur data yang kompleks dan nonlinier. Dengan demikian, kedua metode memiliki keunggulan masing-masing dan dapat saling melengkapi dalam konteks pemodelan frekuensi klaim asuransi kendaraan.

Sebagai saran, penelitian selanjutnya disarankan untuk menambahkan variabel eksternal seperti jenis kendaraan, lokasi kejadian, atau faktor cuaca guna meningkatkan akurasi prediksi. Selain itu, pengujian model lain seperti *Extreme Gradient Boosting* (XGBoost) atau *Long Short-Term Memory* (LSTM) dapat dilakukan untuk memperoleh hasil yang lebih komprehensif dalam analisis prediksi frekuensi klaim pada data asuransi yang bersifat dinamis.

Daftar Pustaka

[1] C. N. A. Indonesia, "Kendaraan bermotor indonesia tembus 168 jutaan, paling banyak di mana?." https://www.cna.id/indonesia/kendaraan-bermotor-indonesia-tembus-168-jutaan-paling-banyak-di-mana-31031, 2025. Accessed: Jun. 21, 2025.

[2] R. Indonesia, "Undang-undang republik indonesia nomor 22 tahun 2009 tentang lalu lintas dan angkutan jalan." https://peraturan.bpk.go.id/Download/27961/UU%20Nomor%2022%20Tahun%202009.pdf, 2009. Accessed: Jun. 21, 2025.

[3] N. S. Maraya and D. Susanti, "Prediction of motor vehicle insurance claims using arma-garch and arima-garch models," *International Journal of Quantitative Research and Modeling*, vol. 5, no. 2, pp. 154–161, 2024.

[4] J. Yoon, "Forecasting of real gdp growth using machine learning models: Gradient boosting and random forest approach," *Comput Econ*, vol. 57, no. 1, pp. 247–265, 2021.

[5] E. F. Harahap, "Pengaruh strategi pemasaran terhadap keputusan pembelian asuransi kendaraan bermotor pada pt asuransi sinarmas cabang garut," *Journal of Knowledge Management*, vol. 12, no. 01, pp. 12–20, 2018.

[6] T. A. J. Putra, D. C. Lesmana, and I. G. P. Purnaba, "Penentuan premi asuransi kendaraan bermotor menggunakan generalized linear models dengan distribusi tweedie," *Jambura Journal of Mathematics*, vol. 3, no. 2, pp. 115–127, 2021.

[7] A. Pratomo, R. F. Umbara, and A. A. Rohmawati, "Prediksi pergerakan harga saham dengan metode random forest menggunakan trend deterministic data preparation (studi

kasus saham perusahaan pt astra international tbk, pt garuda indonesia tbk, dan pt indosat tbk)," *e-Proceeding of Engineering*, vol. 6, no. 1, pp. 2545–2556, 2019.

[8] N. Gbadamosi, D. Bukolj, R. Adcock, and V. Djakovic, "Forecasting bitcoin with technical analysis: A not-so-random forest," *Int J Forecast*, vol. 39, no. 1, pp. 1–17, 2023.

[9] W. W. S. Wei, *Time Series Analysis: Univariate and Multivariate Methods*. Pearson Education, 2nd ed., 2005.

[10] L. Breiman, "Random forests–random features," Technical Report 567, Department of Statistics, University of California, Berkeley, 1999.

[11] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*. Hoboken, New Jersey: John Wiley & Sons, Inc., 2016.

[12] R. H. Shumway and D. S. Stoffer, *Springer Texts in Statistics Time Series Analysis and its Applications With R Examples*. Springer, 5th ed., 2017.

[13] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[14] C. M. Bishop, *Pattern Recognition and Machine Learning*. Information Science and Statistics, New York: Springer, 2006.

[15] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.