

**STUDI PERBANDINGAN DATA TIME SERIES HARGA  
EMAS MENGGUNAKAN METODE XGBOOST, STL, DAN  
FACEBOOK PROPHET**

**RIZAL MUJAHIDDAN**



**DEPARTEMEN ILMU KOMPUTER  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
INSTITUT PERTANIAN BOGOR  
BOGOR  
2023**

## **PERNYATAAN MENGENAI SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA**

Dengan ini saya menyatakan bahwa skripsi dengan judul “STUDI PERBANDINGAN DATA TIME SERIES HARGA EMAS MENGGUNAKAN METODE XGBOOST, STL, DAN FACEBOOK PROPHET” adalah karya saya dengan arahan dari dosen pembimbing dan belum diajukan dalam bentuk apapun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir laporan akhir ini.

Dengan ini saya melimpahkan hak cipta dari karya tulis saya kepada Institut Pertanian Bogor.

Bogor, November 2023

Rizal Mujahiddan  
G64190069

## ABSTRAK

RIZAL MUJAHIDDAN. STUDI PERBANDINGAN DATA TIME SERIES HARGA EMAS MENGGUNAKAN METODE XGBOOST, STL, DAN FACEBOOK PROPHET. Dibimbing oleh ENDANG PURNAMA GIRI dan MEDRIA KUSUMA DEWI HARDHIENATA

Harga emas merupakan aspek penting bagi para pelaku pasar karena nilainya yang stabil dan cenderung naik. Penelitian ini membandingkan tiga model analisis data time series harga emas yakni *Extreme Gradient Boosting* (XGBoost), *Seasonal-Trend Decomposition Procedure Based on LOESS* (STL), dan Facebook Prophet. Dalam penelitian ini, ketiga model tersebut dievaluasi dengan mempertimbangkan faktor eksternal seperti inflasi dan nilai tukar mata uang. Data harga emas dan faktor eksternal dikumpulkan dari Januari 2010 hingga April 2023. Hasil penelitian menunjukkan bahwa pada pendekatan *univariate*, model Facebook Prophet memiliki *Symmetrical Mean Absolute Percentage Error* (SMAPE) sebesar 6,28% dan *Mean Absolute Scaled Error* (MASE) sebesar 6,0458. Pada pendekatan *multivariate*, XGBoost memiliki SMAPE sebesar 12,63% dan MASE sebesar 11,815. Dari hasil penelitian, dapat disimpulkan bahwa Facebook Prophet dengan pendekatan *univariate* lebih akurat dibandingkan pendekatan XGBoost dan STL. Sementara itu, XGBoost dengan pendekatan *multivariate* lebih akurat daripada Facebook Prophet dalam kasus yang diujikan.

Kata kunci: Emas, Facebook Prophet, Keuangan, Peramalan,, STL, XGBoost

## ABSTRACT

RIZAL MUJAHIDDAN. COMPARATIVE STUDY OF GOLD PRICE TIME SERIES DATA USING XGBOOST, STL, AND FACEBOOK PROPHET METHODS. Supervised by ENDANG PURNAMA GIRI and MEDRIA KUSUMA DEWI HARDHIENATA

Gold price is an important aspect for market participants because its value is stable and tends to increase. This study compares three gold price time series data analysis models namely *Extreme Gradient Boosting* (XGBoost), *Seasonal-Trend Decomposition Procedure Based on LOESS* (STL), and Facebook Prophet. In this study, the three models are evaluated by considering external factors such as inflation and currency exchange rates. Gold price data and external factors were collected from January 2010 to April 2023. The results show that in the *univariate* approach, the Facebook Prophet model has a *Symmetrical Mean Absolute Percentage Error* (SMAPE) of 6.28% and a *Mean Absolute Scaled Error* (MASE) of 6.0458. In the *multivariate* approach, XGBoost has a SMAPE of 12.63% and a MASE of 11.815. From the research results, it can be concluded that Facebook Prophet with the *univariate* approach is more accurate than the XGBoost and STL approaches. Meanwhile, XGBoost with a *multivariate* approach is more accurate than Facebook Prophet in the case tested.

*Keywords:* Gold price, Facebook Prophet, Finance, Forecasting, STL, XGBoost

**STUDI PERBANDINGAN DATA TIME SERIES HARGA  
EMAS MENGGUNAKAN METODE XGBOOST, STL, DAN  
FACEBOOK PROPHET**

**RIZAL MUJAHIDDAN**

Skripsi  
sebagai salah satu syarat untuk memperoleh gelar  
Sarjana pada  
Program Studi Ilmu Komputer

**DEPARTEMEN ILMU KOMPUTER  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
INSTITUT PERTANIAN BOGOR  
BOGOR  
2023**

Tim Penguji pada Ujian Skripsi:

1. Dr. Karlisa Priandana S.T., M.Eng

Judul : Studi Perbandingan Data Time Series Harga Emas  
Menggunakan Metode XGBoost, STL, dan Facebook  
Prophet  
Nama : Rizal Mujahiddan  
NIM : G64190069

Disetujui oleh

Pembimbing 1:

Endang Purnama Giri S.Kom., M.Kom.

\_\_\_\_\_

Pembimbing 2:

Medria Kusuma Dewi Hardhienata S.Komp., Ph.D.

\_\_\_\_\_

Diketahui oleh

Ketua Departemen Ilmu Komputer:

Dr. Sony Hartono Wijaya S.Kom., M.Kom.

198108092008121002

\_\_\_\_\_

Tanggal Ujian:  
3 November 2023

Tanggal Lulus:

## **PRAKATA**

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas segala karunia-Nya sehingga karya ilmiah ini berhasil diselesaikan. Judul yang dipilih dalam penelitian yang dilaksanakan sejak bulan Januari 2023 sampai bulan Juli 2023 adalah Studi Perbandingan Analisis Data Time Series Harga Emas Menggunakan XGBoost, STL, dan Facebook Prophet.

Terima kasih penulis ucapkan kepada:

1. Keluarga penulis yang senantiasa mendukung dalam pelaksanaan penelitian, terdiri dari kedua orang tua, yaitu bapak Cecep Gunawan dan ibu Minah Mintarsih serta kedua saudara perempuan penulis yaitu Sabriyani Chairunissa dan Zackyah Miftahul Zannah.
2. Bapak Endang Purnama Giri S.Kom., M.Kom. selaku dosen pembimbing pertama dan ibu Medria Kusuma Dewi Hardhienata S.Komp., Ph.D. selaku dosen pembimbing kedua atas bimbingan dan arahnya selama berjalannya penelitian.

Bogor, November 2023

Rizal Mujahiddan

## DAFTAR ISI

<b>DAFTAR ISI</b>	viii
<b>DAFTAR GAMBAR</b>	ix
<b>DAFTAR TABEL</b>	ix
<b>I PENDAHULUAN</b>	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	3
1.5 Ruang Lingkup	3
<b>II TINJAUAN PUSTAKA</b>	4
2.1 Harga Emas	4
2.2 Model Peramalan pada Univariate dan Multivariate	4
2.3 Model Facebook Prophet	5
2.4 Seasonal-Trend Decomposition Procedure Based on LOESS (STL)	9
2.5 Extreme Gradient Boosting (XGBoost)	11
2.6 Perbandingan Model peramalan harga emas	14
2.7 Granger Causality	14
2.8 Symmetric Mean Absolute Percentage Error (SMAPE)	16
<b>III METODE PENELITIAN</b>	19
3.1 Alat dan Bahan	19
3.2 Tahapan Penelitian	19
3.2.1 Studi literatur	20
3.2.2 Pengambilan Data	20
3.2.3 Analisis Eksplorasi Data	20
3.2.4 Pengolahan data	22
3.2.5 Melatih data dengan 3 model	22
3.2.6 Tuning parameter	29
3.2.7 Evaluasi Model	31
3.2.8 Peramalan Data	31
3.3 Deskripsi Data	31
3.4 Skenario Eksperimen	38
3.5 Lingkungan Penelitian	40
<b>IV HASIL DAN PEMBAHASAN</b>	42
4.1 Analisis Statistik Data	42
4.2 Perbandingan Model dan Hasil Peramalan	46
<b>V KESIMPULAN DAN SARAN</b>	50
5.1 Kesimpulan	50
5.2 Saran	50
<b>DAFTAR PUSTAKA</b>	51



## DAFTAR GAMBAR

1	Contoh grafik komponen prophet (Taylor dan Letham 2017)	8
2	Alur Penelitian	19
3	Grafik harga emas berdasarkan waktu	32
4	Grafik kurs jual berdasarkan waktu	33
5	Grafik kurs beli berdasarkan waktu	33
6	Grafik suku bunga BI berdasarkan waktu	34
7	Grafik inflasi di Indonesia berdasarkan waktu	34
8	Grafik harga emas berdasarkan waktu setelah diimputasi	35
9	Grafik kurs jual berdasarkan waktu setelah diimputasi	35
10	Grafik kurs beli berdasarkan waktu setelah diimputasi	36
11	Grafik data inflasi berdasarkan waktu setelah diimputasi	36
12	Grafik data inflasi berdasarkan waktu setelah diimputasi	37
13	Grafik normalisasi master data	37
14	Ilustrasi Metode <i>Recursive Tabular Regression Forecaster</i> pada <i>library sktime</i> (Löning <i>et al.</i> 2019)	38
15	<i>Expanding Window</i> dari referensi Kingphai dan Moshfeghi (2023)	49
16	Persebaran data harga emas	42
17	Persebaran kurs jual dollar	43
18	Persebaran kurs beli dollar	43
19	Persebaran data inflasi	43
20	Persebaran data suku bunga BI	44
21	<i>Heatmap p-value Granger Causality</i>	44
22	Grafik autocorrelation	45
23	Partial autocorrelation	45
24	Grafik Peramalan Harga Emas Semua Model	48

## DAFTAR TABEL

1	Penelitian Harga Emas Sebelumnya	14
2	Parameter <i>default</i> Facebook Prophet	22
3	Parameter <i>default</i> XGBoost	23
4	Parameter <i>default</i> STL	23
5	<i>Tuning</i> parameter Facebook Prophet	30
6	<i>Tuning</i> parameter XGBoost	30
7	<i>Tuning</i> parameter STL	31
8	Master data	32
9	Skenario Eksperimen	40
10	Statistik Master Data	42
11	Kinerja Model Facebook Prophet pada Data <i>univariate</i> dengan Metode <i>cross validation (expanding window)</i>	46
12	Kinerja Model Facebook Prophet pada Data <i>multivariate</i> dengan Metode <i>cross validation (expanding window)</i>	46

13	Kinerja Model XGBoost pada Data <i>univariate</i> dengan Metode <i>cross validation (expanding window)</i>	46
14	Kinerja Model XGBoost pada Data <i>univariate</i> dengan Metode <i>cross validation (expanding window)</i>	47
15	Kinerja Model STL pada Data <i>univariate</i> dengan Metode <i>cross validation (expanding window)</i>	47
16	Kinerja seluruh model	47
17	Hasil Peramalan Seluruh Model	48

# I PENDAHULUAN

## 1.1 Latar Belakang

Emas telah memegang peran penting dalam budaya selama sejarah, melambangkan keindahan, kemurnian, kekuatan, dan prestasi. Hingga saat ini, emas masih menjadi simbol yang signifikan dalam masyarakat, digunakan sebagai uang, penghargaan, cincin pernikahan, medali Olimpiade, dan barang-barang keagamaan seperti salib. Kelangkaannya dan nilai yang tinggi membuatnya menjadi alat tukar yang dicari dan simpanan nilai yang berharga. Meskipun sulit diramal karena dipengaruhi oleh banyak faktor, peramalan harga emas menjadi penting bagi para investor karena mampu mengurangi inflasi dan membantu kebijakan moneter. Sebagai aset berharga, emas memainkan peran krusial dalam sistem keuangan dan stabilitas sosial masyarakat (Das *et al.* 2022).

Meramalkan pergerakan harga emas adalah isu penting bagi banyak pihak yang berkepentingan di pasar global. Baik itu investor, perusahaan penambangan, maupun entitas lain yang menggunakan emas sebagai barometer andal untuk menilai performa ekonomi dunia di masa mendatang. Dengan meramal pergerakan harga emas, diharapkan mereka dapat mengambil langkah yang benar dalam hal investasi, produksi, dan konsumsi, serta melakukan *hedging* terhadap kemungkinan risiko dan ketidakpastian (Jabeur *et al.* 2021).

Terdapat beberapa penelitian pendahulu yang telah dilakukan untuk melakukan peramalan harga emas. Penelitian Makala dan Li (2021) telah menerapkan model *Autoregressive Integrated Moving Average* (ARIMA) untuk meramalkan harga emas dengan akurasi sebesar 99,7%. Penelitian Makala dan Li (2021) menggunakan data harga emas dunia dari situs *goldhub* (WGC 2023). Di sisi lain, penelitian Chai *et al.* (2021) telah menyelidiki hubungan dinamis antara harga emas, minyak mentah, indeks dollar AS, dan indeks volatilitas. Dalam penelitian tersebut digunakan model STL-ETS untuk mengakomodasi tren fluktuasi pengembalian harga emas secara akurat dan memberikan analisis mendalam tentang korelasi emas dengan indikator ekonomi penting lainnya. Penelitian Chai *et al.* (2021) menggunakan data harga emas dunia dari situs *goldhub* (WGC 2023). Penelitian Jabeur *et al.* (2022) memberikan perbandingan antara enam model pembelajaran mesin untuk meramalkan pergerakan harga emas. Penelitian tersebut mengusulkan pendekatan inovatif menggunakan algoritma XGBoost dan nilai interaksi SHAP untuk menginterpretasikan ramalan dengan akurat. Penelitian ini memberikan wawasan penting tentang bagaimana model pembelajaran mesin dapat diterapkan untuk peramalan harga emas. Penelitian tersebut menggunakan data harga emas Amerika Serikat dari situs *goldhub* (WGC 2023).

Penelitian lain oleh Kishann dan RamaParvathy (2022) berhasil mengembangkan model Facebook Prophet yang mencapai akurasi sebesar 97,2% dalam meramalkan harga emas. Keunggulan penelitian ini adalah tingginya akurasi ramalan yang telah dicapai. Penelitian tersebut menggunakan data dari Negara

India. Berdasarkan penelitian Makridakis *et al.* (2019) dan Ouyang *et al.* (2021) menunjukkan bahwa model statistik dan model *machine learning* (model tidak termasuk *neural network*) lebih efektif dibandingkan dengan model berbasis *neural network* untuk peramalan. Oleh karena itu, model berbasis *neural network* tidak dianjurkan dalam konteks peramalan harga emas.

Berdasarkan hasil penelitian Kishann dan RamaParvathy (2022), model Facebook Prophet memiliki tingkat akurasi yang tinggi sebesar 97,2% untuk meramal harga emas. Berdasarkan hasil penelitian Jabeur *et al.* (2022), model XGBoost memiliki tingkat akurasi yang tinggi yakni sebesar 99,4% untuk meramal harga emas. Oleh karena itu, kedua model ini dianggap sebagai pilihan yang baik untuk digunakan dalam penelitian ini. Selanjutnya, penggunaan model STL dikombinasikan dengan model statistik lain telah terbukti meningkatkan akurasi sesuai dengan penelitian Ouyang *et al.* (2021).

Oleh karena itu, model STL juga dipertimbangkan sebagai pilihan untuk digunakan dalam penelitian ini. Dengan mempertimbangkan temuan dari penelitian-penelitian sebelumnya, penelitian ini akan menggunakan model Facebook Prophet, XGBoost dan STL sebagai model utama untuk meramalkan harga emas.

ARIMA tidak digunakan karena memiliki kelemahan utama yang terletak pada keterbatasannya dalam menangani data keuangan yang cenderung penuh dengan *irregularitas* dan *anomali* yang dapat mengakibatkan kesulitan dalam menghasilkan prediksi yang akurat dan stabil (Syrris dan Shenai 2021). Penelitian Risse (2019) menyatakan bahwa model *multivariate* lebih baik dibandingkan dengan model *univariate* dalam segi akurasi. Penelitian tersebut menggunakan model DWT-SVR. Model ini tidak digunakan karena penerapannya lebih kompleks.

Perbedaan utama penelitian ini dengan penelitian-penelitian sebelumnya, yakni penggunaan dataset harga emas di Indonesia yang dikumpulkan secara mandiri dari situs *logammulia.com*. Selain itu, pada penelitian ini dilakukan juga analisis perbandingan model *univariate* dan *multivariate* pada 3 model regresi XGBoost, STL dan Facebook Prophet. Untuk memperlihatkan pengaruh eksternal terhadap harga emas, penelitian ini juga menggunakan uji hipotesis *Granger Causality*. Selanjutnya dari analisis uji kinerja model, metrik yang digunakan adalah *mean absolute scaled error* (MASE) dan *symmetric mean absolute percentage error* (SMAPE).

## **1.2 Rumusan Masalah**

Permasalahan dalam penelitian ini adalah minimnya penelitian yang ada saat ini yang berfokus pada perbandingan metode XGBoost, STL, dan Facebook Prophet untuk meramal harga emas di Indonesia. Selain itu, belum banyak penelitian pendahulu yang menganalisis penggunaan pendekatan *univariate* dan

*multivariate* dalam meramalkan data emas di Indonesia. Berdasarkan hal tersebut, perumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Menentukan model terbaik antara model XGBoost, STL, dan Facebook Prophet untuk meramalkan data harga emas di Indonesia.
2. Menentukan pendekatan terbaik *multivariate* atau *univariate* untuk meramal harga emas di Indonesia.

### **1.3 Tujuan Penelitian**

Penelitian ini bertujuan membandingkan kinerja tiga metode analisis data time series yaitu XGBoost, STL, dan Facebook Prophet dalam meramal harga emas, serta melakukan perbandingan antara model *univariate* dan *multivariate* pada ketiga model tersebut.

### **1.4 Manfaat Penelitian**

Penelitian ini diharapkan dapat digunakan oleh investor dan pelaku pasar lainnya untuk membuat keputusan investasi yang lebih baik. Selain itu, diharapkan hasil perbandingan ketiga model ini dalam penelitian ini dapat menjadi dasar bagi penelitian selanjutnya untuk mengembangkan model yang lebih baik di masa datang.

### **1.5 Ruang Lingkup**

Ruang lingkup digunakan untuk penelitian adalah sebagai berikut:

1. Data harga Emas diperoleh dari situs logam mulia (Logam Mulia, 2022).
2. Data yang digunakan dari 4 Januari 2010 hingga 30 April 2023.
3. Metrik yang digunakan adalah *mean absolute scaled error* (MASE) dan *Symmetrical mean absolute percentage error* (SMAPE).

## II TINJAUAN PUSTAKA

### 2.1 Harga Emas

Harga emas merupakan salah satu indikator penting dalam pasar finansial. Harga emas cenderung naik ketika pasar saham menurun, sehingga digunakan sebagai *safe haven* bagi investor. Grafik dari harga emas menunjukkan pergerakan harga emas dalam jangka waktu tertentu yang dapat membantu investor dalam mengambil keputusan investasi.

Faktor-faktor yang mempengaruhi harga emas meliputi faktor ekonomi, politik, dan geopolitik. Faktor ekonomi seperti tingkat inflasi, suku bunga, dan kondisi perekonomian global memiliki dampak yang signifikan pada harga emas. Faktor politik dan geopolitik seperti perang, kerusuhan, dan perubahan kebijakan pemerintah juga dapat mempengaruhi harga emas.

Menurut studi yang dilakukan oleh Anandasayanan *et al.* (2019) dan Aleemi *et al.* (2016), faktor ekonomi seperti tingkat inflasi memiliki pengaruh yang signifikan terhadap harga emas. Pada studi tersebut, data harga emas dan faktor ekonomi dianalisis menggunakan metode analisis korelasi dan hasilnya menunjukkan bahwa tingkat inflasi dan suku bunga memiliki korelasi negatif dengan harga emas.

Menurut Anandasayanan *et al.* (2019), pengujian hubungan antara inflasi dan harga emas dengan menggunakan data bulanan dari Januari 1999 hingga Desember 2018. Hasil pengujian tersebut menunjukkan adanya hubungan yang kuat dan positif antara inflasi dan harga emas. Analisis regresi menunjukkan bahwa inflasi berdampak signifikan terhadap harga emas dan uji *Granger causality* menemukan bahwa ada hubungan searah antara harga emas dan inflasi. Sehingga saat masa periode inflasi yang tinggi, lebih disarankan untuk membeli emas dari pada berinvestasi pada pasar saham.

### 2.2 Model Peramalan pada Univariate dan Multivariate

Data deret waktu merupakan salah satu jenis data yang sangat umum digunakan untuk mengambil informasi dan merekam berbagai aktivitas di berbagai bidang. Setiap ranah yang melibatkan pengukuran waktu melalui sensor, sensus, atau catatan transaksi, pencatatan urutan pengamatan yang terindeks dengan penanda waktu memberikan wawasan tentang perkembangan suatu kuantitas yang dapat diukur di masa lalu. Data deret waktu dapat digunakan untuk membuat visualisasi data, menemukan pola dan korelasi berulang, mengklasifikasikan dan mengelompokkan data, mendeteksi pencilan, mensegmentasikan data, meramalkan dan mensimulasikan data (Dama dan Sinoquet 2021).

Peramalan adalah proses memprediksi nilai masa depan berdasarkan data masa lalu. Ada dua jenis utama peramalan yakni *univariate* dan *multivariate*. Peramalan *univariate* hanya menggunakan satu variabel dari masa lalu untuk meramal variabel yang sama di masa depan. Peramalan *multivariate* menggunakan lebih dari satu variabel dari masa lalu untuk meramal satu atau beberapa variabel di

masa depan. Metode yang berbeda dapat menangani berbagai jenis data dan hubungan. Beberapa metode membutuhkan formula yang jelas, sementara yang lain dapat menggunakan simulasi atau *neural network*. Teori-teori dari bidang lain dapat membantu untuk memahami hubungan-hubungan ini. Peramalan juga dapat memprediksi beberapa variabel secara bersamaan (Petropoulos *et al.* 2022).

Peramalan *univariate* dan *multivariate* memiliki perbedaan signifikan dalam hal struktur data, pra-proses, dan hasil *output*. Peramalan *univariate* lebih sederhana dengan menggunakan data indeks waktu dan satu target yang akan diramalkan, sedangkan peramalan *multivariate* melibatkan fitur-fitur yang berhubungan dengan indeks waktu. Pra-proses pada peramalan *multivariate* lebih kompleks karena memerlukan pemilihan fitur yang relevan, sementara peramalan *univariate* lebih sederhana dalam hal ini. *Output* dari peramalan *multivariate* mencakup ramalan untuk satu atau lebih target, sedangkan peramalan *univariate* hanya fokus pada ramalan untuk satu target berdasarkan data historis.

Peramalan deret waktu penting dalam berbagai *domain* seperti keuangan, bisnis, pengendalian proses kualitas industri, transaksi digital, distribusi tenaga listrik, meteorologi, biologi dan kedokteran. Ini membantu menganalisis masa lalu dan meramalkan masa depan. Misalnya, dapat digunakan untuk merencanakan kebijakan oleh berbagai organisasi di bidang keuangan dan bisnis, mendeteksi situasi abnormal atau penipuan dalam pengendalian proses kualitas industri dan transaksi digital, mengontrol beban melalui pemantauan canggih dan mengatasi gangguan dalam aliran daya listrik di distribusi tenaga listrik, membimbing pengambilan keputusan yang terinformasi untuk pertanian, navigasi udara dan laut di meteorologi, memperoleh pengetahuan tentang aktivitas gen yang diberikan status terpengaruh / tidak terpengaruh di biologi, meramalkan penyebaran penyakit, memperkirakan tingkat kematian atau menilai risiko yang bergantung pada waktu di kedokteran (Dama dan Sinoquet 2021).

### 2.3 Model Facebook Prophet

Facebook Prophet merupakan sebuah metode yang digunakan untuk meramal data rangkaian waktu dengan menggunakan model aditif, dimana *trend non-linear* ditentukan berdasarkan musiman tahunan, mingguan, dan harian, serta efek-efek liburan (Taylor dan Letham 2017). Namun, pada model ini bersifat kompleks (Korstanje 2021).

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \quad (1)$$

$t$  = Waktu

$g(t)$  = Komponen tren

$s(t)$  = Komponen Seasonal

$h(t)$  = Komponen *Holiday*

$\epsilon_t$  = Komponen residual

$y(t)$  = Data deret waktu

Persamaan (1) merupakan dasar dari model Facebook Prophet yang telah dikembangkan oleh penelitian Harvey dan Peters (1990). Facebook Prophet menggunakan kurva pertumbuhan linear atau logistik untuk memodelkan *trend*. Model Facebook Prophet untuk peramalan pertumbuhan menggunakan model pertumbuhan logistik untuk menggambarkan pertumbuhan data yang *nonlinier* dan mencapai titik jenuh (*carrying capacity*) seperti pertumbuhan populasi di ekosistem alami. Persamaan yang digunakan logistik adalah sebagai berikut:

$$g(t) = \frac{C}{1+e^{(-k(t-m))}} \quad (2)$$

$t$  = Waktu

$g(t)$  = Komponen tren

$C$  = *Carrying Capacity*

$k$  = Laju pertumbuhan

$m$  = Parameter *offset*

Persamaan (2) menunjukkan bahwa ada 2 hal yang perlu dipertimbangkan. Pertama, bahwa *carrying capacity* tidak konstan dan berubah berdasarkan waktu. Kedua, laju pertumbuhan tidak konstan. Dalam model Facebook Prophet, ada beberapa titik perubahan (*change points*)  $S$ . Pada daerah yang berbatasan titik perubahan, daerah satu dengan yang lain memiliki nilai laju pertumbuhan yang berbeda-beda yang dapat diartikan seperti vektor laju  $\delta$ . Laju perubahan pada titik tertentu dengan nilai awalnya  $k$ , maka  $k + \sum_{j:t>s_j} \delta_j$ . Dengan vektor  $a(t) \in \{0,1\}^S$ , maka dapat didefinisikan seperti:

$$a_j(t) = 1 - [t < s_j] \quad (3)$$

Persamaan (3) menunjukkan bahwa vektor  $a(t)$  merujuk pada waktu  $t$  sudah melewati titik perubahan pada  $s_j$ . Laju pada waktu  $t$  adalah  $k + a(t)^T \delta$ . Ketika mengubah parameter laju  $k$ , perlu juga menyesuaikan parameter offset  $m$  agar terjadi koneksi yang mulus antara ujung segmen, dan penyesuaian yang tepat untuk titik perubahan  $j$  dapat dihitung dengan mudah seperti pada Persamaan (4). Persamaan (4) adalah sebagai berikut:

$$\gamma_j = (s_j - m - \sum_{l<j} \gamma_l) \left( 1 - \frac{k + \sum_{l<j} \delta_l}{k + \sum_{l \leq j} \delta_l} \right) \quad (4)$$

Persamaan (4) memiliki beberapa variabel yang sama untuk variable seperti  $m, j, l, k$  dan  $\delta$ . Parameter  $\gamma$  digunakan untuk menghaluskan komponen tren karena model Facebook Prophet memiliki banyak titik perubahan yang menyebabkan berbagai perbedaan dalam laju pertumbuhan. Berdasarkan dari Persamaan (1), (2), (3) dan (4), maka menghasilkan fungsi tren seperti berikut:

$$g(t) = \frac{C(t)}{1 + \exp(-(k + a(t)^T \delta)(t - (m + (a(t)^T \gamma))))} \quad (5)$$

Persamaan (5) merupakan pengembangan dari Persamaan (1) dengan syarat *carrying capacity* tidak konstan dan laju pertumbuhan  $k$  berubah berdasarkan



waktu. Persamaan (5) menunjukkan parameter  $\gamma$  dihitung untuk menyesuaikan antar titik perubahan.

Untuk masalah peramalan yang tidak menunjukkan pertumbuhan yang jenuh, model pertumbuhan dengan tingkat konstan berpotongan memberikan model yang lebih sederhana dan seringkali bermanfaat. Model *trend* ini menggunakan fungsi seperti berikut:

$$g(t) = (k + a(t)^T * \delta)t + (m + a(t)^T \gamma) \quad (6)$$

Persamaan (6) menunjukkan bahwa  $k$  sebagai tingkat pertumbuhan,  $\delta$  sebagai penyesuaian tingkat,  $m$  sebagai parameter offset, dan  $\gamma_j$  diatur sebagai  $-\mathcal{S}_j \delta_j$  agar fungsi tetap berkelanjutan. Penerapan model ini memungkinkan peramalan yang lebih efisien dan hasil yang relevan dengan data yang ada, karena tingkat pertumbuhan dan penyesuaian parameter telah diatur dengan tepat untuk memastikan kelancaran dan konsistensi dalam peramalan data.

$$S(t) = \sum_{k=1}^K (a_k \cos(\frac{2\pi kt}{P}) + b_k \sin(\frac{2\pi kt}{P})) \quad (7)$$

$$\beta = [a_1, b_1, \dots, a_n, b_n] \quad (8)$$

$$X(t) = \left[ \cos(\frac{2\pi(1)t}{P}), \dots, \sin(\frac{2\pi(N)t}{P}) \right] \quad (9)$$

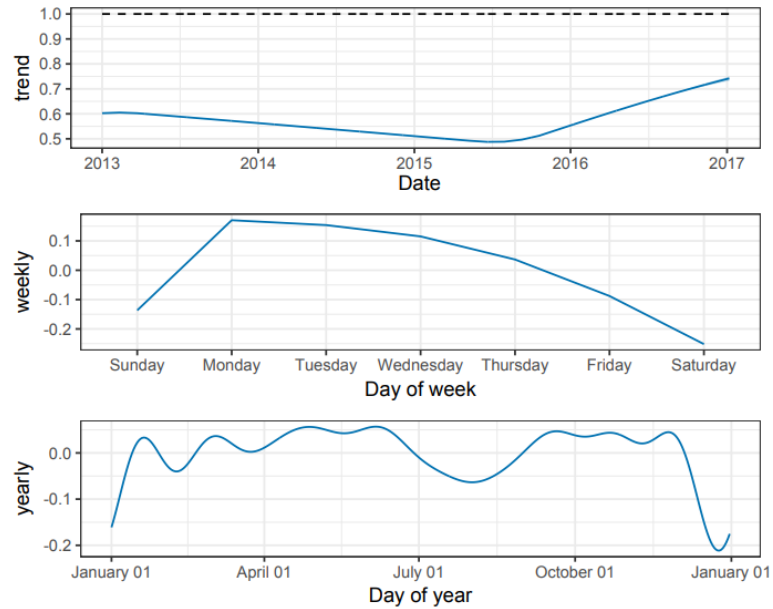
$$S(t) = X(t)\beta \quad (10)$$

Persamaan (7) menunjukkan bahwa arti simbol dari  $X(t)$  mewakili komponen seasonal pada waktu  $t$ ,  $K$  adalah jumlah komponen Fourier yang diperlukan untuk menangkap musiman dengan akurat,  $P$  adalah periode musiman, serta  $a_k$  dan  $b_k$  adalah koefisien dari deret Fourier. Persamaan (8), (9), (10), merupakan bentuk matriks dari Persamaan (7). Nilai  $\beta$  ini dideklarasikan sebagai vektor *weight of seasonality*. *Weight of seasonality* merupakan nilai bobot pada setiap komponen Fourier. Selain *seasonal*, Facebook Prophet juga memasukkan dampak liburan dan acara khusus pada deret waktu. Rumus untuk efek liburan pada waktu  $t$  adalah:

$$Z(t) = [1(t \in D_1), \dots, 1(t \in D_L)] \quad (11)$$

$$h(t) = Z(t)\kappa \quad (12)$$

Persamaan (11) menunjukkan bahwa arti simbol dari  $Z(t)$  mewakili matriks komponen *holiday*.  $D_i$  melambangkan tanggal liburan.  $t$  adalah waktu. Pada Persamaan (12),  $\kappa$  melambangkan vektor *Weight of Holiday*. *Weight of Holiday* adalah nilai bobot yang mempengaruhi pada *Holiday* tertentu dalam meramalkan data *time series*.



Gambar 1 Contoh grafik komponen prophet (Taylor dan Letham 2017)

Gambar 1 menunjukkan bahwa data akan dipecah menjadi 3 komponen. Komponen tersebut yakni, *trend*, *weekly*, dan *yearly*. Pada bagian *yearly*, berpola gelombang sebanyak 3 kali dengan gelombang berbeda. Grafik ini merupakan penerapan dari *fourier* pada bagian komponen *seasonal*. Grafik ini juga menunjukkan bahwa Facebook Prophet bisa memiliki banyak seasonal pada saat bersamaan.

$$normal(\frac{c}{(1 + e^{-(k+A*\delta)*(t-(m+A*\gamma))})}) + X * \beta, \epsilon) \quad (13)$$

$$normal((k + A * \delta) * t + (m + A * \gamma) + X * \beta, \sigma) \quad (14)$$

Facebook Prophet menggunakan *backend* Stan. Facebook Prophet dilatih dengan menggunakan algoritma Stan's L-BFGS. Pelatihan model Facebook Prophet mengacu pada mendapatkan nilai *likelihood*. Facebook Prophet memiliki cara kerja seperti berikut. Pertama tentukan parameter. Parameter yang ditentukan adalah  $\epsilon$  (Error),  $\delta$  (vektor penyesuaian titik perubahan),  $\beta$  (Komponen Seasonal),  $k$  (*growth rate*) dan  $m$  (*offset parameter*). Kedua, tentukan apakah dengan menggunakan *trend* logistik atau linear. Ketiga, jika menggunakan *trend* logistik, maka rumus *likelihood* yang digunakan adalah Persamaan (13). Jika menggunakan *trend* linear, maka rumus *likelihood* yang digunakan adalah Persamaan (14).

Model Facebook Prophet memiliki beberapa kelemahan yang perlu dipertimbangkan yaitu Facebook Prophet memiliki keterbatasan dalam pemahaman yang tepat tentang bagaimana dan apa yang dipelajari oleh model tersebut. Hal ini disebabkan oleh Facebook Prophet yang secara otomatis melakukan sebagian besar pekerjaan untuk pengguna tanpa memerlukan pemahaman teori yang mendalam. Facebook Prophet memiliki jumlah parameter yang terbatas yang dapat diatur dan

disesuaikan, yang dapat menjadi kelebihan jika model berfungsi dengan baik, tetapi juga dapat menjadi kelemahan jika model tidak memberikan tingkat akurasi yang diinginkan. Terakhir, meskipun Facebook Prophet dapat memberikan hasil yang baik dengan pengaturan parameter *default*, namun ketika digunakan untuk ramalan yang lebih kompleks, kemampuan model ini dapat menjadi terbatas. Hal ini disebabkan oleh sifat modular dan sederhana dari model regresi yang digunakan oleh Facebook Prophet. Oleh karena itu, walaupun Facebook Prophet menawarkan kemudahan pada penggunaannya, pengguna juga harus mempertimbangkan kelemahan-kelemahan ini sebelum memilih menggunakan model Facebook Prophet untuk masalah peramalan (Korstanje 2021).

#### 2.4 Seasonal-Trend Decomposition Procedure Based on LOESS (STL)

STL adalah proses penguraian time series menjadi tiga komponen utama, yaitu *trend*, *Seasonal*, dan komponen Residu. Rumus STL adalah sebagai berikut:

$$Y_v = T_v + S_v + R_v \quad (15)$$

$Y_v$  = Data Time Series

$T_v$  = Komponen Trend

$S_v$  = Komponen Seasonal

$R_v$  = Komponen Residu

Persamaan (15) bahwa perlu dicatat bahwa STL (*Seasonal-Trend decomposition* menggunakan LOESS) hanya terdiri dari 3 komponen, sedangkan model Facebook Prophet memiliki 4 komponen. STL menggunakan metode *smoothing* yakni LOESS (*locally estimated scatterplot smoothing*). LOESS (*locally estimated scatter plot smoothing*) adalah metode non-parametrik untuk melakukan *smoothing* pada data (Cleveland *et al.* 1990). LOESS menggunakan regresi dengan nilai bobot tertentu ( $W$ ) dengan syarat seperti berikut:

1.  $W(x) > 0$  untuk  $|x| < 1$ .
2.  $W(-x) = W(x)$ .
3.  $W(x)$  merupakan fungsi tidak naik untuk  $x \geq 0$ .
4.  $W(x) = 0$  untuk  $|x| \geq 1$ .

Dengan syarat ini, maka fungsi  $W$  akan bervariasi seperti fungsi *tricube* dan *gaussian*. Namun, dalam banyak kasus, fungsi  $W$  yang sering digunakan adalah *tricube*. Fungsi *tricube* memiliki bentuk sebagai berikut:

$$W(x) = \{(1 - |x|^3)^3, |x| < 1; 0, |x| \geq 1\} \quad (16)$$

Persamaan (16) digunakan untuk fungsi kernel *tricube* yang memiliki nilai dari nol hingga satu. Fungsi ini meningkatkan kinerja regresi berbobot lokal. Fungsi-fungsi dengan kontak yang halus serupa diharapkan juga memberikan hasil yang baik. Namun, fungsi bobot *uniform*, dengan *discontinuity* pada nilai satu, menunjukkan kinerja yang kurang baik. *Discontinuity* pada fungsi *uniform* menyebabkan *local roughness* dan aproksimasi distribusi yang kurang memuaskan (Cleveland dan Devlin 1988).

Algoritma LOESS memiliki perbedaan dengan regresi linear dalam hal metode yang digunakan. Pada algoritma LOESS menggunakan metode *weighted least square*, sedangkan regresi linear menggunakan *ordinary least square*. Untuk mekanisme LOESS adalah sebagai berikut:

1. Tentukan parameter  $d$  (*degree*), fungsi  $W$  (*Weight*), banyak iterasi ( $t$ ), parameter untuk menentukan smoothing ( $f$ ) dan parameter untuk menentukan ukuran *window* ( $span$ )
2. Urutkan variabel independen
3. Pada setiap titiknya hitunglah seperti :
  - a. buatlah jarak di setiap titiknya
  - b. buatlah window berdasarkan parameter  $span$  untuk tetangga terdekat dari titik tersebut
  - c. menskala dengan jarak tersebut berdasarkan jarak maksimum.
  - d. terapkan fungsi  $W$  pada jarak tersebut
  - e. hitunglah regresi dengan metode *weighted least square* dan dapatkan persamaan regresi
  - f. hitunglah  $y$  duga dari persamaan tersebut untuk titik tersebut
4. Hubungkan titik duga tersebut
5. Lakukanlah iterasi ulang dengan mengubah parameter tersebut untuk meminimalisir *error* yang terjadi.

Metode ini berdasarkan dari penelitian Cleveland (1979). Metode ini memperkirakan nilai rata-rata dari variabel dependen pada setiap titik dengan menggunakan regresi lokal pada titik tersebut. Regresi lokal ini dilakukan dengan menggunakan polinomial rendah derajat yang disesuaikan dengan data di sekitar titik tersebut (Cleveland *et al.* 1990). Mekanisme STL adalah sebagai berikut :

1. *Inner Loop*
  - a. *Detrending*.
  - b. *Cycle-subseries Smoothing*.
  - c. *Low-Pass Filtering of smoothed Cycle-Subseries*, Langkah ini dilakukan *filtering* dengan *moving average*.
  - d. *Detrending of Smoothed Cycle-Subseries*, Langkah ini dilakukan *filtering* dengan *moving average* dan untuk mendapatkan komponen seasonal.
  - e. *Deseasonalizing*
  - f. *Trend Smoothing*
2. *Outer Loop*
  - a. Mencari nilai Residu yang sudah dilakukan dari *Inner Loop*

*Detrending* dilakukan dengan rumus  $Y - T$  dimana  $Y$  merupakan nilai aktual dan  $T$  merupakan komponen *trend*. *Cycle-subseries Smoothing* dilakukan pada metode *smoothing* dengan menggunakan teknik LOESS. *Low-Pass Filtering of Smoothed Cycle Subseries* melakukan beberapa *moving average*. *Detrending of*

*Smoothed Cycle-Subseries* melakukan *detrending* dengan cara  $S = C - L$  dimana  $L$  adalah hasil *Low-Pass Filtering of smoothed Cycle-Subseries*,  $C$  adalah hasil *Cycle-subseries Smoothing* dan  $S$  merupakan komponen *seasonal*. Kemudian dilakukan *Deseasonalization* dengan rumus  $Y - S$  (Cleveland *et al.* 1990).

Model STL memiliki beberapa keunggulan. Pertama, ia mampu menangani berbagai jenis pola musiman. Kedua, model ini mengizinkan perubahan komponen musiman seiring waktu dengan kontrol yang dapat diatur oleh pengguna. Selain itu, model STL dapat diadaptasi tingkat *smooth* dari siklus *trend* sesuai preferensi. Keunggulan lainnya adalah tahan terhadap data *outlier*. Selain itu, model STL memiliki beberapa kelemahan. Salah satunya adalah ketidakmampuan model untuk secara otomatis menangani variasi yang berkaitan dengan peristiwa tertentu. Model ini juga berfokus hanya pada dekomposisi aditif yang dapat mempengaruhi interpretasi terhadap komponen residu (Cleveland *et al.* 1990).

## 2.5 Extreme Gradient Boosting (XGBoost)

XGBoost merupakan suatu algoritma turunan dari *Gradient Boosting Tree* yang sudah dimodifikasi sehingga bisa menangani data yang sangat besar dan tentunya memiliki banyak fitur, menangani banyak data yang hilang, dan menangani banyak jenis variabel seperti numerik dan kategorikal (Shi *et al.* 2018). *Gradient Boosting Tree* merupakan salah satu metode *ensemble* dengan algoritma *gradient boosting* yang digunakan untuk memecahkan masalah machine learning seperti regresi, dan klasifikasi (Shi *et al.* 2018).

*Gradient Tree Boosting* adalah sebuah teknik *machine learning* yang kuat dan populer yang dapat digunakan untuk membangun model prediksi lebih akurat. Teknik ini menggabungkan pohon keputusan secara iteratif menggunakan metode boosting. Pohon keputusan ini bertindak sebagai *weak learners* untuk membentuk model yang lebih kuat. Selama proses training, model berusaha mengoptimalkan fungsi *loss*-nya yang berfungsi untuk mengukur sejauh mana prediksi model berbeda dari nilai sebenarnya pada data pelatihan. Fungsi *loss* mencerminkan perbedaan antara prediksi model dengan nilai aktual dengan tujuan untuk mencari kombinasi parameter dan bobot yang mengarahkan model ke arah yang lebih akurat. Fungsi loss adalah sebagai berikut:

$$L = \sum_{i=1}^n l(y_i, \hat{y}_i + f(x_i)) + \Omega(f) \quad (17)$$

$$\Omega(f) = \gamma T + \frac{\lambda \|w\|^2}{2} \quad (18)$$

$L$  = Total Loss

$l$  = Loss

$y_i$  = Nilai sebenarnya

$\hat{y}_i$  = Nilai prediksi

$f(x_i)$  = Nilai prediksi pada satu pohon

$\Omega(f)$  = Fungsi penalti

$\gamma$  = Parameter *regularization* seluruh model

$\lambda$  = Parameter *regularization* pada satu pohon  
 $w$  = *Weight* pada satu pohon  
 $T$  = Banyaknya daun pada satu pohon  
 $i$  = Indeks

Persamaan (17) menunjukkan bahwa *loss* bergantung dengan nilai praduga dengan nilai asli. Persamaan (18) berfungsi sebagai *regularized* agar dapat menghindarkan *overfitting*. Persamaan tersebut bergantung pada pohon satu dengan yang lain.

$$L = \sum_{i=1}^n [g_i f(x_i) + h_i f^2(x_i)] + \Omega(f) \quad (19)$$

$$g_i = \delta_y l(y_i, \hat{y}_i) \quad (20)$$

$$h_i = \delta_y^2 l(y_i, \hat{y}_i) \quad (21)$$

$g_i$  = *Gradient function* pada XGBoost (*first partial derivative of loss function*)  
 $h_i$  = *Hessian function* pada XGBoost (*second partial derivative of loss function*)  
 $\delta_y$  = operator *partial derivative*

Ekspansi Taylor adalah metode yang sering digunakan dalam bidang matematika, *machine learning*, dan *engineering*. Metode ini digunakan untuk menyederhanakan persamaan yang kompleks menjadi persamaan yang lebih sederhana dengan menggunakan operator *derivative* (He 2019). Dalam penelitian ini, ekspansi Taylor digunakan untuk menyederhanakan Persamaan (17) menjadi Persamaan (19). Persamaan (20) dan Persamaan (21) juga didapatkan dari ekspansi Taylor.

$$\begin{aligned}
 L^{(t)} &= \sum_{i=1}^n [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \\
 &= \sum_{j=1}^T \left[ \left( \sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left( \sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma T \quad (22)
 \end{aligned}$$

$$w_j = - \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \quad (23)$$

$j$  = indeks daun  
 $I_j = \{i \mid q(x_i) = j\}$ , *record set* yang melewati daun  $j$ .  
 $t$  = iterasi ke- $t$

Persamaan (22) merupakan pengembangan Persamaan (19) dengan perhitungan antar daun ( $j$ ). Persamaan (22) diminimalkan agar mendapatkan model optimum. Berdasarkan hal tersebut, Persamaan (23) diturunkan dari Persamaan (22). Pada Persamaan (23), *Gradient function* dan *hessian function* digunakan untuk menghitung *weight* daun.

$$L = - \frac{1}{2} \sum_{j=1}^T \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \quad (24)$$

Persamaan (24) merupakan persamaan akhir yang dapat digunakan untuk mengoptimalkan model. Persamaan (24) merupakan nilai evaluasi seperti *decision tree*. XGBoost adalah pustaka yang dirancang untuk melatih model pembelajaran mesin dengan efisien. XGBoost merupakan singkatan dari eXtreme Gradient Boosting. Pustaka ini menggunakan beberapa fitur dan algoritma untuk mempercepat proses latih seperti *Approximate Algorithm*, *Sparsity-aware Split Finding*, *Parallel Learning*, *Cache-aware Access*, dan *Blocks for Out-of-core Computation* (Chen dan Guestrin 2016).

*Approximate Algorithm* pada XGBoost digunakan untuk mendukung *gradient tree boosting* dalam dua pengaturan yaitu data yang sangat besar dan data yang sangat jarang. Algoritma ini mempercepat proses perhitungan *gain* untuk setiap split yang mungkin pada setiap fitur dengan menghitung nilai *gain* secara aproksimasi. Hal ini dilakukan dengan mengambil sampel dari data latih dan menghitung nilai *gain* pada sampel tersebut. Dalam hal ini, XGBoost menggunakan algoritma subsampling untuk mengambil sampel dari data latih (Chen dan Guestrin 2016).

*Sparsity-aware Split Finding* pada XGBoost digunakan untuk menangani sparsitas pada data seperti keberadaan data yang hilang, entri nol padat, dan nilai yang di-*encode* satu-per-satu. Algoritma ini membuat XGBoost menyadari sparsitas tersebut dengan mendefinisikan arah default untuk mereka. Dalam hal ini, XGBoost mencoba kedua arah dalam split dan menemukan arah *default* dengan menghitung *gain* (Chen dan Guestrin 2016).

*Parallel Learning* pada XGBoost digunakan untuk mempercepat proses pelatihan model dengan membagi data menjadi beberapa bagian dan melatih model pada setiap bagian secara independen. Hal ini dilakukan dengan menggunakan beberapa *thread* untuk mempercepat proses pelatihan (Chen dan Guestrin 2016).

*Cache-aware Access* pada XGBoost digunakan untuk mempercepat proses perhitungan skor kesamaan dan nilai output dengan menggunakan memori *cache* pada sistem. Hal ini dilakukan dengan mengambil data dari memori *cache* dan menghitung skor kesamaan dan nilai *output* pada data tersebut (Chen dan Guestrin 2016).

*Blocks for Out-of-core Computation* dalam XGBoost adalah fitur yang memungkinkan pelatihan model dengan dataset besar yang melebihi kapasitas memori. Fitur ini membagi data pelatihan menjadi blok-blok yang lebih kecil untuk penggunaan memori yang lebih efisien. Selain itu, XGBoost juga menggunakan kompresi blok untuk mengurangi ukuran blok dalam memori dengan algoritma kompresi, serta pemecahan blok untuk memproses blok secara paralel. Dengan kombinasi fitur ini, XGBoost mampu melatih model pada dataset besar, mengatasi kendala memori, dan meningkatkan skalabilitas pelatihan model *machine learning* (Chen dan Guestrin 2016).

## 2.6 Perbandingan Model peramalan harga emas

Faktor ekonomi seperti tingkat inflasi, suku bunga, dan kondisi perekonomian global memiliki dampak yang signifikan pada harga emas (Anandasayanan *et al.* 2019). Tingkat inflasi dan suku bunga memiliki korelasi negatif dengan harga emas (Anandasayanan *et al.* 2019). Spekulasi pada tingkat kepemilikan emas pada masa depan dianggap penting (Farhat dan Ghalayini 2020). Metrik yang digunakan sebagai tolak ukur pada model peramalan adalah *R-Square*. Perbandingan model peramalan bisa dilihat pada Tabel 1.

Tabel 1 Penelitian Harga Emas Sebelumnya

No	Nama Model	Dataset	Sumber data	Negara	Referensi	<i>R-square</i>
1.	XGBoost	Jan 1986 - Des 2019	<i>Goldhub</i>	Dunia	Jabeur <i>et al.</i> 2022	99,4%
2.	MLP with Whale Optimization	Sept 1987 - Agu 2017	<i>World Bank</i>	Dunia	Alameer <i>et al.</i> 2019	99%
3.	DWT-SVR	Jan 1992 - December 2016	FRED	Dunia	Risse 2019	60%
4.	LSTM with Pandemic	30 Jul 2015 - 22 Jan 2020	<i>Yahoo Finance</i>	Dunia	Khani <i>et al.</i> 2022	85,759%
5.	STL	3 Jan 2006 - 31 Jan 2019	<i>Goldhub</i>	Dunia	Chai <i>et al.</i> 2021	0,003%
6.	Facebook Prophet	2009 - 2020	<i>Goldhub</i>	Dunia	Kishann dan RamaParvathy 2022	95,725%
7.	STL, XGBoost, Facebook Prophet	04 Januari 2010 - 31 April 2023	Bank Indonesia dan Logam Mulia	Indonesia	Penelitian ini	-

Tabel 1 menunjukkan penelitian harga emas dengan metrik *R-Square*. Berdasarkan hasil penelitian dari Tabel 1, model *XGBoost* merupakan model terbaik yang dapat digunakan karena memiliki akurasi yang tinggi dan rentang data yang cukup besar.

## 2.7 Granger Causality

Granger causality adalah sebuah metode statistik yang digunakan untuk menentukan apakah suatu variabel dapat digunakan untuk memprediksi variabel lainnya pada suatu deret waktu. Metode ini ditemukan oleh Clive Granger pada tahun 1969 dan telah banyak digunakan dalam berbagai bidang seperti ekonomi,



keuangan, genetika, dan neurosains. Granger causality memungkinkan kita untuk menentukan apakah suatu variabel memiliki pengaruh kausal pada variabel lainnya. Dalam Granger causality, pengaruh kausal diukur dengan melihat seberapa baik variabel yang menjadi prediktor dapat memprediksi variabel target. Jika prediktor tersebut dapat memprediksi target dengan lebih baik daripada model yang hanya menggunakan data target saja, maka kita dapat menyimpulkan bahwa prediktor tersebut memiliki pengaruh kausal pada target (Shojaie dan Fox 2021).

Granger mendefinisikan *kausalitas* seperti nilai masa lalu dari deret waktu  $y_t$  memprediksi nilai masa depan dari nilai deret  $x_t$ . Granger mengatakan  $y$  itu 'kausal' untuk  $x$  jika  $var[x_t - P(x_t | H_{<t})] < var[x_t - P(x_t | H_{<t} \setminus y_t)]$ ,  $H_{<t}$  artinya nilai jaman dulu yang relevan dengan informasi hingga waktu  $t - 1$ .  $P(x_t | H_{<t})$  adalah nilai prediksi optimal  $x_t$  dengan diberikan nilai  $H_t$  (Shojaie dan Fox 2021).

$$A^0 x_t = \sum_{k=1}^d A^k x_{t-k} + e_t \quad (25)$$

$A$  = Matriks

$d$  = Banyak Lag yang digunakan

$t$  = Waktu

$e_t$  = error pada waktu  $t$

$x_t$  = data time series pada waktu  $t$

Pada penelitian Granger (1969) awalnya *Granger causality* bisa diidentifikasi dengan menggunakan model linear yang unik. Penelitian Granger menyadari bahwa bahwa model linear unik ini tidak mendapatkan matriks identitas. Hal tersebut, *Granger causality* menggunakan model *Vector Autoregressive* (VAR) seperti pada Persamaan (25). Pada Persamaan (25), khusus pada matriks  $A^0$  berbentuk diagonal.

*Granger causality* menggunakan dua model VAR seperti pada Persamaan (26) dan Persamaan (27). Hal ini disebabkan metode ini terbatas pada model *bivariate* dan menghiraukan dampak dari variabel lainnya. *Granger causality* ditunjukkan dengan adanya nilai yang tidak nol dalam koefisien autoregressive pada Persamaan (26) dan Persamaan (27).

$$A^0_k x_t = \sum_{k=1}^d A^k_{xx} x_{t-k} + \sum_{k=1}^d A^k_{xy} y_{t-k} + e_{t,x} \quad (26)$$

$$A^0_k y_t = \sum_{k=1}^d A^k_{yy} y_{t-k} + \sum_{k=1}^d A^k_{yx} x_{t-k} + e_{t,y} \quad (27)$$

*Granger causality test* dilakukan dengan menghitung nilai *F-test* dengan model yang lengkap independen variabelnya (*full model*) serta dengan model yang telah dikurangi salah satu independen variabelnya (*reduced model*). *Full model* dapat ditunjukkan dengan Persamaan (26). Akan tetapi, *reduced model* dapat ditunjukkan Persamaan (26) dengan dihilangkan komponen  $y$ . Nilai *F-test* dapat dihitung dengan menggunakan Persamaan (28) yang berasal dari penelitian Shojaie dan Fox (2021).

$$F = \frac{(RSS_{red} - RSS_{full})/(r-s)}{RSS_{full}/(T-r)} \quad (28)$$

$RSS_{Full}$  = Residual Sum Squared (Full Model)  
 $RSS_{Red}$  = Residual Sum Squared (Reduced Model)  
 $T, r, s$  = degree of freedom

$$RSS = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (29)$$

$y_i$  = Nilai Sebenarnya  
 $\hat{y}_i$  = Nilai Prediksi  
 $n$  = Banyak data

Pada Persamaan (28) dan Persamaan (29), *Residual Sum Square* merupakan salah satu komponen yang digunakan untuk uji *analysis of variance* (ANOVA) (Aydin 2008). Pada persamaan (28),  $\alpha$  merupakan tingkat signifikansi. Berdasarkan dari penelitian Cowles dan Davis (1982), nilai  $\alpha$  yang baik digunakan adalah 0,05.  $\alpha$  digunakan untuk uji F-Test dengan cara membandingkan nilai  $\alpha$  dengan *P-value*. *P-value* bisa didapatkan dari nilai F-test dengan tabel F-test. Nilai T adalah *total seluruh degree of freedom* pada model tersebut. Nilai *r* adalah *degree of freedom* bagian regresi pada *reduced model*. Nilai *s* adalah *degree of freedom* bagian regresi pada *full model*. Berdasarkan Persamaan (28), *y* mengalami *Granger causality* untuk *x* jika nilai *p-value* yang berasal dari F-test melebihi (1- $\alpha$ )% kuartil (Shojaie dan Fox 2021).

## 2.8 Symmetric Mean Absolute Percentage Error (SMAPE)

*Symmetric Mean Absolute Percentage Error* (SMAPE) adalah ukuran statistik yang digunakan untuk mengevaluasi akurasi prediksi dalam analisis deret waktu. Ini adalah variasi dari Mean Absolute Percentage Error (MAPE) yang simetris di sekitar nol. SMAPE dihitung sebagai berikut:

$$SMAPE = \frac{100}{n} \sum_{t=1}^n \frac{|F_t - A_t|}{(|A_t| + |F_t|)/2} \quad (26)$$

$t$  = Index  
 $A$  = Actual Value  
 $F$  = Forecast Value  
 $n$  = Banyak data

Persamaan (26) menjelaskan tentang rumus SMAPE. SMAPE memiliki rentang nilai antara 0 dan 2, dimana nilai 0 menunjukkan akurasi sempurna, sedangkan nilai 2 menunjukkan bahwa ramalan tersebut salah (Armstrong 1985).

SMAPE memiliki potensi untuk menghasilkan kesalahan simetris bahkan dalam rentang peramalan yang asimetris. Salah satu keunggulan dari SMAPE adalah tidak mengalami masalah seperti MAPE yang dapat menghasilkan nilai yang sangat besar atau tak terhingga. Selain itu, karena adanya batas kesalahan yang telah ditentukan. SMAPE lebih tahan terhadap nilai-nilai pencilaan dengan memberikan

penekanan yang lebih sedikit pada nilai-nilai ekstrim dibandingkan dengan metrik lain yang tidak memiliki batasan kesalahan (Chen 2017).

Nilai SMAPE memiliki interpretasi yang sama dengan MAPE. Nilai SMAPE dikatakan *highly accurate forecasting* jika memiliki nilai kurang dari 10%. Nilai SMAPE dikatakan *good forecasting* jika memiliki nilai dari 10% hingga 20%. Nilai SMAPE dikatakan *reasonable forecasting* jika memiliki nilai 20% hingga 50%. Nilai SMAPE dikatakan *inaccurate forecasting* jika memiliki nilai lebih dari 50% (Hu *et al.* 2022).

## 2.9 Mean Absolute Scaled Error (MASE)

*Mean Absolute Scaled Error* (MASE) mengatasi keterbatasan kesalahan relatif tradisional dalam menilai akurasi ramalan. Berbeda dengan kesalahan relatif dengan rata-rata yang tidak terdefinisi dan variasi tak terhingga. MASE menskalakan kesalahan menggunakan Mean Absolute Error (MAE) dalam sampel dari metode *naive forecast*. MASE dapat dihitung dengan:

$$MASE = \frac{\text{mean}(A_t - F_t)}{\frac{1}{n-1} \sum_{t=2}^n |A_t - A_{t-1}|} \quad (27)$$

$t = \text{Index}$

$A = \text{Actual Value}$

$F = \text{Forecast Value}$

$n = \text{Banyak data}$

Persamaan (27) menunjukkan mengenai rumus MASE. menunjukkan bahwa nilai MASE berkisar 0 hingga tak terbatas. MASE (*Mean Absolute Scaled Error*) membandingkan akurasi ramalan dengan akurasi *naive forecast*. Jika MASE kurang dari satu, itu berarti bahwa ramalan lebih baik daripada *naive forecast* (Hyndman 2006). MASE sering digunakan karena ketahanannya terhadap pencilan, interpretasi yang lebih baik dibandingkan RMSSE, dan variabilitas yang lebih rendah dalam sampel kecil dibandingkan MdASE (Hyndman dan Koehler 2006).

## 2.10 Library Pycaret

*Pycaret* merupakan *library python open-source* yang digunakan untuk sistem otomatis pada pembuatan *machine learning* dengan *syntax code* yang sederhana. Pada *Pycaret*, dilengkapi berbagai fungsi pendukung *machine learning* mulai dari tahapan pra-proses hingga *deployment*. *Pycaret* ini menggunakan beberapa *library* lainnya seperti *library Scikit-learn*, *Prophet*, *XGBoost*, *LightGBM*, *CatBoost*, *Optuna*, *Hyperopt*, *Ray*, *Sktime* dan lain-lain (Ali 2020).

*Pycaret* memiliki lima modul utama yang meliputi modul klasifikasi, regresi, analisis *time series*, *clustering* dan deteksi anomali. Pada penelitian ini, modul yang digunakan adalah *time series* karena untuk meramalkan data yang berbentuk *time series*. Modul ini menggunakan *library sktime* sebagai *library* utama pada model *machine learning* yang secara khusus digunakan dalam menangani dan meramalkan data *time series* (Ali 2020).

Model yang digunakan untuk meramal data pada penelitian ini adalah STL, Facebook Prophet, dan XGBoost yang telah dimodifikasi untuk peramalan data (*XGBoost\_cds\_dt*). STL yang digunakan merupakan model yang berasal dari *library sktime* dan telah diimplementasi dari *library statsmodel* dengan model STL. Facebook Prophet yang digunakan merupakan model yang berasal dari *library sktime* lalu telah diimplementasi dari *library Prophet* dengan model FBProphet. XGBoost yang digunakan merupakan model yang berasal dari *library sktime* yang sebelumnya dilakukan transformasi dengan *deseasonalisasi* dan *detrending*. Kemudian dilakukan peramalan dengan menggunakan model XGBoost pada data *time series* (Ali 2020).

*Sktime* merupakan *library* yang memiliki *interface* seperti *scikit-learn* bertujuan untuk membuat machine learning dengan data *time series*. *Sktime* memiliki desain API seperti *time series classification*, *classical forecasting*, *transformers*, *composition*, dan *benchmarking* (Löning et al. 2019).

### III METODE PENELITIAN

#### 3.1 Alat dan Bahan

Alat dan bahan yang digunakan untuk penelitian adalah sebagai berikut:

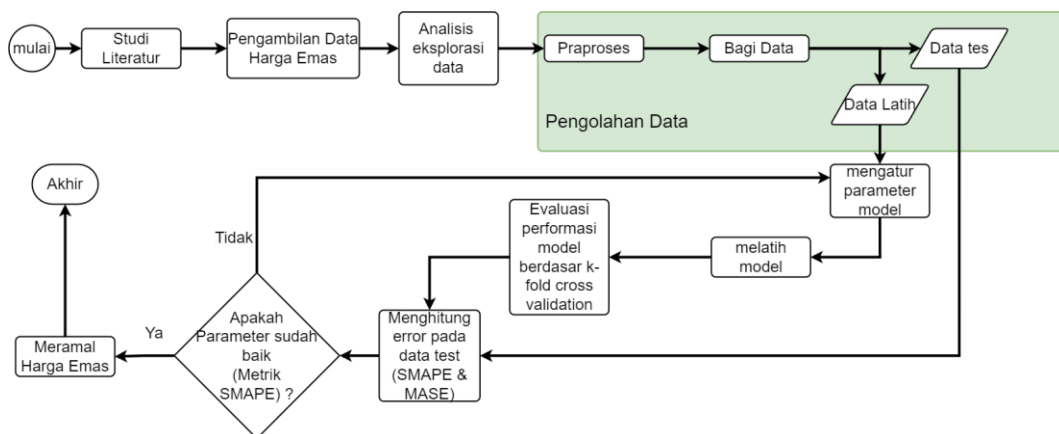
1. Python 3.9.16
2. Scikit learn 1.2.2
3. Pandas 1.4.4
4. Seaborn 0.11.2
5. Matplotlib 3.5.3
6. Pycaret 3.0.4

Penelitian ini menggunakan Python sebagai bahasa pemrograman yang populer dan fleksibel. Pustaka Scikit-learn dipakai untuk mengukur kinerja model dengan berbagai metrik evaluasi. Pustaka Pandas digunakan untuk mengolah data tabular dengan mudah. Pustaka Seaborn dan Matplotlib dipilih untuk menampilkan data secara grafis dengan fungsi visualisasi data yang kaya dan menarik. Pustaka Pycaret diandalkan untuk melaksanakan proses *machine learning* secara efisien dan mudah dengan fitur-fitur otomatisasi dan optimisasi *machine learning*.

#### 3.2 Tahapan Penelitian

Tahapan penelitian adalah sebagai berikut:

1. Studi literatur
2. Pengambilan Data Harga emas
3. Analisis Eksplorasi data
4. Pengolahan data
  - a. Praproses data
  - b. Pembagian dataset
5. Melatih data dengan 3 model
6. *Tuning hyperparameter*
7. Evaluasi model
8. Peramalan model



Gambar 2 Alur Penelitian

Pada Gambar 2 menampilkan alur penelitian. Alur penelitian meliputi studi literatur hingga meramal harga emas. Alur penelitian berdasarkan referensi dari

penelitian Nawaz *et al.* (2021). Penjelasan mengenai Alur penelitian akan diterangkan pada sub bab selanjutnya.

### 3.2.1 Studi literatur

Studi literatur yang dilakukan adalah dengan mengkaji penelitian sebelumnya yaitu artikel yang berhubungan dengan model Facebook Prophet (Taylor dan Letham 2017), model *XGBoost* (Jabeur *et al.* 2022) dan model *STL* (Chai *et al.* 2021) untuk meramal harga emas dengan faktor eksternal. Beberapa penelitian mengenai penerapan model *machine learning* untuk peramalan harga emas serta mencari korelasi pada faktor eksternal yang mempengaruhi harga emas seperti penelitian Khani *et al.* (2022).

### 3.2.2 Pengambilan Data

Data yang diambil pada penelitian ini berupa data harga emas, suku bunga, inflasi dan kurs mata uang dollar USA. Pengambilan data harga emas diperoleh dengan memanfaatkan *Highcharts* API melalui *console* pada *browser* untuk mengambil data. Data suku bunga, inflasi, dan kurs mata uang diperoleh dari website resmi Bank Indonesia (BI).

### 3.2.3 Analisis Eksplorasi Data

Analisis eksplorasi data dilakukan untuk mendapatkan wawasan yang lebih mendalam mengenai dataset yang digunakan. Tujuan utamanya adalah untuk melihat persebaran data secara menyeluruh dan mengeksplorasi hubungan antara variabel-variabel yang relevan. Salah satu aspek penting dari analisis ini adalah visualisasi data berdasarkan tanggal yang memungkinkan kita untuk melihat *trend* dan pola yang berkaitan dengan waktu. Pada penelitian ini, digunakan fungsi *histplot* untuk melihat persebaran data dengan menggunakan *library seaborn*. Parameter dari fungsi *histplot* sebagai berikut:

1. *data*
2. *x*
3. *y*
4. *hue*
5. *weights*
6. *stat*
7. *bins*
8. *binrange*
9. *discrete*
10. *cumulative*
11. *binwidth*

Untuk melakukan input data, menggunakan Parameter data. Nilai *default* dari Parameter *data* adalah *None*. Parameter *x* dan *y* digunakan untuk menentukan posisi *axis* x dan y. Nilai *default* dari kedua parameter ini adalah *None*. Sedangkan

untuk melakukan fitur *mapping* dengan warna menggunakan Parameter *hue*. Nilai *default* dari Parameter *hue* adalah None. Parameter *weights* digunakan untuk menentukan *weight* pada masing - masing *bin*. Nilai *default* dari Parameter *weights* adalah None. Parameter *stat* digunakan untuk mengagregat setiap *bin*. Nilai *default* dari Parameter *stat* adalah ‘count’. Parameter *bins* digunakan untuk membagi *bins* secara umum. Nilai *default* dari Parameter *bins* adalah ‘auto’. Parameter *binrange* digunakan untuk menentukan nilai terendah dan tertinggi pada *bin edges*. Nilai *default* dari Parameter *binrange* adalah None. Parameter *discrete* digunakan untuk membuat *binning* dengan nilai *range bin* bertipe *integer*. Nilai *default* dari Parameter *discrete* adalah True. Parameter *cumulative* digunakan untuk membuat nilai kumulatif. Nilai *default* dari Parameter *cumulative* adalah False. Parameter *binwidth* digunakan untuk menentukan lebar setiap *bin*. Nilai *default* dari Parameter *binwidth* adalah None. Pada Parameter *binwidth* dapat ditentukan dengan metode *freedman diaconis estimator*. Rumus *freedman diaconis estimator* dapat dihitung dengan Persamaan (28).

$$h = 2 * \frac{IQR}{\sqrt[3]{n}} \quad (28)$$

$h$  = Binwidth

$IQR$  = Interquartile Range

$n$  = Banyak data

Hasil histogram dapat memvisualisasikan sebaran data yang dapat membantu menentukan langkah-langkah pra-proses yang tepat. Visualisasi yang dapat melihat persebaran data lebih presisi dari histogram adalah *kdeplot*. *Kdeplot* dapat diplot dengan menggunakan library *seaborn*. *Kdeplot* dari kata KDE dan *plot*. KDE adalah singkatan dari *kernel density estimation*. *Kdeplot* merupakan *density plot* (histogram) yang berdasarkan *probability density function* (Cohen dan Cohen 2006).

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{(-x^2/2)} \quad (29)$$

$$f(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) \quad (30)$$

$$h \approx 1.06 \hat{\sigma} n^{(-\frac{1}{5})} \quad (31)$$

$K(x)$  = Fungsi kernel (*probability density function*)

$f(x)$  = Nilai Kde terhadap titik x

$n$  = Banyak Data

$h$  = Bandwidth

Persamaan (29) merupakan persamaan dari fungsi *Gaussian Distribution* (fungsi normal). Persamaan (30) merupakan nilai KDE. Untuk Bandwidth yang digunakan adalah Persamaan (31) yakni Scott’s rule (Scott 1992). Pada penelitian

ini, parameter yang diubah adalah hanyalah Parameter *kde* bernilai True untuk menampilkan grafik histogram yang disertai dengan grafik *kdeplot*.

Analisis eksplorasi data ini menjadi penting untuk memastikan persiapan data yang optimal sebelum melanjutkan ke tahap analisis atau pemodelan data berikutnya. Pra-proses yang tepat akan memastikan kualitas data, mengurangi *noise* atau *outlier*, dan memastikan hasil analisis yang diperoleh valid.

### 3.2.4 Pengolahan data

Dalam analisis data ini, dilakukan pra-proses menggunakan beberapa metode, termasuk imputasi (mengisi data yang hilang) data dengan metode *drift* menggunakan Polynomial Trend Forecaster dari pustaka *sktime*. Metode *drift* ini memungkinkan kita untuk mengisi nilai yang hilang dalam dataset berdasarkan *trend* polinomial data sebelum dan sesudah nilai yang hilang tersebut. Selain itu, dilakukan pembagian data menggunakan metode *cross-validation k-fold* sebanyak 3 kali, yang membagi dataset menjadi tiga subset yang berbeda untuk melatih dan menguji model secara berulang. Pendekatan ini membantu menghindari *overfitting* dan memberikan gambaran yang lebih objektif tentang kinerja model pada dataset yang berbeda. Hasil dari pra-proses data dan *cross-validation* ini akan memberikan landasan yang solid untuk analisis lebih lanjut atau pemodelan data yang dapat dipercaya dan relevan.

### 3.2.5 Melatih data dengan 3 model

Dalam pelatihan data ini, tiga model berbeda digunakan untuk peramalan data dengan tingkat akurasi yang tinggi. Model pertama adalah Facebook Prophet, yang memiliki kemampuan untuk mengatasi *trend non-linear*, efek musiman, dan perubahan struktural dalam data (Taylor dan Letham 2017). Model Facebook Prophet digunakan dari *library pycaret* dengan tambahan pra-proses. Nilai *default* parameter dari model Facebook Prophet berdasarkan *library pycaret* adalah sebagai berikut :

Tabel 2 Parameter *default* Facebook Prophet

Parameter	Nilai default
<i>changepoint_prior_scale</i>	0,05
<i>holidays_prior_scale</i>	10
<i>seasonality_prior_scale</i>	10

Pada Tabel 2, nilai ini diperoleh dari modul FBProphet yang menjadi dependensi dari *library pycaret*. Model kedua adalah XGBoost, sebuah model berbasis *gradient boosting tree* yang sangat kuat dalam menghasilkan peramalan yang akurat (Chen dan Guestrin 2016). Model XGboost digunakan dari *library pycaret* dengan tambahan pra-proses dan juga dengan dekomposisi *deseasonalisasi*



dan *detrending*. Nilai *default* parameter dari model XGBoost berdasarkan *library pycaret*.

Tabel 3 Parameter *Default* XGBoost

Parameter	Nilai default
<i>learning_rate</i>	0,3
<i>max_depth</i>	6
<i>max_leaves</i>	0
<i>n_estimators</i>	100
<i>Sp</i>	1
<i>deseasonal_model</i>	additive
<i>degree</i>	1
<i>window_length</i>	10

Pada Tabel 3, nilai ini diperoleh dari modul XGBoost yang menjadi dependensi dari *library pycaret*. Model ketiga yang digunakan adalah STL, yang menggabungkan metode *smoothing* dengan pendekatan LOESS untuk mengidentifikasi komponen musiman, trend, dan residu dalam data (Cleveland *et al* 1990).

Tabel 4 Parameter *Default* STL

Parameter	Nilai default
<i>seasonal</i>	7
<i>seasonal_deg</i>	1
<i>seasonal_jump</i>	1
<i>sp</i>	2
<i>trend</i>	None
<i>trend_deg</i>	1
<i>trend_jump</i>	1

Pada Tabel 4, nilai *default* ini diperoleh dari modul sktime yang menjadi dependensi dari *library pycaret*. Dengan mengadopsi ketiga model ini, diharapkan dapat memberikan peramalan yang lebih handal dan akurat, serta meningkatkan kualitas analisis dan pengambilan keputusan dalam konteks peramalan data.

Pada library *pycaret* digunakan beberapa sintaks yang meliputi seluruh proses pembuatan model *machine learning*. Langkah-langkah pembuatan model machine learning dengan menggunakan library *pycaret*:

1. *Setup*
2. *Create Model*
3. *Tune Model*
4. *Finalize Model*
5. *Predict Model*

*Setup* pada library *pycaret* memiliki tujuan untuk membuat *pipeline* yang digunakan untuk pra-proses data. Pada modul analisis *time series* bahwa pra-proses yang dilakukan meliputi menentukan fitur, imputasi data, transformasi data dan *cross validation* data. *Setup* memiliki beberapa parameter yakni sebagai berikut:

1. *data*
2. *data\_func*
3. *target*
4. *index*
5. *ignore\_features*
6. *numeric\_imputation\_target*
7. *numeric\_imputation\_exogenous*
8. *transform\_target*
9. *transform\_exogenous*
10. *scale\_target*
11. *scale\_exogenous*
12. *fe\_target\_rr*
13. *fe\_exogenous*
14. *fold\_strategy*
15. *fold*
16. *fh*
17. *hyperparameter\_split*
18. *seasonal\_period*
19. *ignore\_seasonality\_test*
20. *sp\_detection*
21. *max\_sp\_to\_consider*
22. *remove\_harmonics*
23. *harmonic\_order\_method*
24. *num\_sps\_to\_use*
25. *seasonality\_type*
26. *point\_alpha*
27. *coverage*
28. *enforce\_exogenous*
29. *n\_jobs*
30. *use\_gpu*

- 31. *custom\_pipeline*
- 32. *html*
- 33. *session\_id*
- 34. *system\_log*
- 35. *log\_experiment*
- 36. *experiment\_name*
- 37. *experiment\_custom\_tags*
- 38. *log\_plots*
- 39. *log\_profile*
- 40. *log\_data*
- 41. *engine*
- 42. *verbose*
- 43. *profile*
- 44. *profile\_kwargs*
- 45. *fig\_kwargs*

Parameter berikut terdapat pada fungsi *Setup*. Parameter yang wajib pada fungsi setup adalah Parameter *data*, Sedangkan parameter yang lain merupakan parameter yang opsional. Berikut ini merupakan keterangan parameter. Parameter *data* digunakan untuk input data. Data yang digunakan berupa *dataframe* yang memiliki indeks *datetime*. Parameter *data\_func* digunakan untuk membuat fungsi pengolahan dari Parameter *data*. Nilai *default* dari Parameter *data\_func* adalah *None*. Parameter *target* digunakan untuk menentukan kolom target dari Parameter *data*. Nilai *default* dari Parameter *target* adalah *None*. Parameter *index* digunakan untuk acuan *datetime* pada Parameter *data*. Nilai *default* dari Parameter *index* adalah *None* (Ali 2023).

Terdapat parameter yang digunakan dalam pemilihan fitur seperti Parameter *ignore\_features* dan imputasi data yaitu Parameter *numeric\_imputation\_target*. Parameter *ignore\_features* digunakan untuk mengabaikan fitur tertentu pada Parameter *data*. Nilai *default* dari Parameter *ignore\_features* adalah *None*. Parameter *numeric\_imputation\_target* digunakan untuk menentukan metode imputasi target. Nilai *default* dari Parameter *numeric\_imputation\_target* adalah *None*. Parameter *numeric\_imputation\_exogenous* digunakan untuk menentukan metode imputasi fitur. Nilai *default* dari Parameter *numeric\_imputation\_exogenous* adalah *None* (Ali 2023).

Lalu parameter yang mengatur transformasi data seperti Parameter *transform\_target* dan normalisasi data seperti *scale\_target*. Parameter *transform\_target* digunakan untuk menentukan metode transformasi target. Nilai *default* dari Parameter *transform\_target* adalah *None*. Parameter *transform\_exogenous* digunakan untuk menentukan metode untuk transformasi fitur. Nilai *default* dari Parameter *transform\_exogenous* adalah *None*. Parameter *scale\_target* digunakan untuk menentukan metode untuk *scaling* target. Nilai *default* dari Parameter *scale\_target* adalah *None*. Parameter *scale\_exogenous*

digunakan untuk menentukan metode untuk *scaling* fitur. Nilai *default* dari Parameter *scale\_exogenous* adalah *None* (Ali 2023).

Beberapa pengguna juga melakukan *feature engineering* sehingga pembuat *library pycaret* membuat Parameter *fe\_target\_rr* dan Parameter *fe\_exogenous*. Parameter *fe\_target\_rr* digunakan untuk mengekstrak target menjadi beberapa fitur. Nilai *default* dari Parameter *fe\_target\_rr* adalah *None*. Parameter *fe\_exogenous* digunakan untuk mengekstrak fitur. Nilai *default* dari Parameter *fe\_exogenous* adalah *None* (Ali 2023).

*Cross validation* diperlukan dalam proses pembuatan *machine learning* untuk meningkatkan akurasi model. Oleh karena itu dibutuhkan parameter untuk mengatur *cross validation*. Parameter *fold\_strategy* digunakan untuk menentukan metode *fold\_strategy* pada Parameter *data*. Nilai *default* dari Parameter *fold\_strategy* adalah 'expanding'. Parameter *fold* digunakan untuk menentukan banyaknya *fold*. Nilai *default* dari Parameter *fold* adalah 3. Parameter *fh* digunakan untuk menentukan banyaknya *datetime* yang akan diramal. Nilai *default* dari Parameter *fh* adalah 1. Parameter *hyperparameter\_split* yang digunakan untuk menentukan pembagian hiperparameter pada data train atau seluruh data. Nilai *default* dari Parameter *hyperparameter\_split* adalah 'all' (Ali 2023).

Data *time series* tidak terlepas dari komponen *seasonal*. Oleh karena itu, modul *analysis time series* menyiapkan parameter yang berkaitan dengan *seasonal*. Parameter *seasonal\_period* digunakan untuk menentukan *periode seasonal*. Nilai *default* dari Parameter *seasonal\_period* adalah *None*. Parameter *ignore\_seasonality\_test* digunakan untuk menolak atau menyetujui *seasonality*. Nilai *default* dari Parameter *ignore\_seasonality\_test* adalah *False*. Parameter *sp\_detection* digunakan untuk mendeteksi *seasonal*. Nilai *default* dari Parameter *sp\_detection* adalah 'auto'. Parameter *max\_sp\_to\_consider* digunakan untuk menentukan maksimum periode yang dideteksi. Nilai *default* dari Parameter *max\_sp\_to\_consider* adalah 60 (Ali 2023).

Periode-periode yang memiliki kelipatan dari periode lainnya disebut dengan *harmonic*. *Harmonic* ini perlu dihilangkan agar tidak terjadi *overfitting*. Parameter *remove\_harmonics* digunakan untuk menghapus atau tidaknya harmonik. Nilai *default* dari Parameter *remove\_harmonics* adalah *False*. Parameter *harmonic\_order\_method* digunakan untuk menentukan metode penghapusan harmonik. Nilai *default* dari Parameter *harmonic\_order\_method* adalah 'harmonic\_max' (Ali 2023).

Parameter *num\_sps\_to\_use* digunakan untuk menentukan banyaknya *periode seasonal* yang diterapkan pada *pipeline*. Nilai *default* dari Parameter *num\_sps\_to\_use* adalah 1. Parameter *seasonality\_type* digunakan untuk menentukan tipe *seasonality*. Nilai *default* dari Parameter *seasonality\_type* adalah 'mul'. Parameter *point\_alpha* digunakan untuk menentukan kuartil yang diterapkan pada ramalan dengan interval tertentu. Nilai *default* dari Parameter *point\_alpha*

adalah None. Parameter *coverage* digunakan untuk menentukan ramalan interval. Nilai *default* dari Parameter *coverage* adalah None (Ali 2023).

Parameter *enforce\_exogenous* digunakan untuk menentukan penggunaan fitur eksternal. Nilai *default* dari Parameter *enforce\_exogenous* adalah True. Parameter *n\_jobs* digunakan untuk menentukan banyaknya processor yang digunakan. Nilai *default* dari Parameter *n\_jobs* adalah -1. Parameter *use\_gpu* digunakan untuk menentukan apakah gpu digunakan atau tidak. Nilai *default* dari Parameter *use\_gpu* adalah False. Parameter *custom\_pipeline* digunakan untuk membuat *pipeline* sesuai dengan pengguna. Nilai *default* dari Parameter *custom\_pipeline* adalah None. Parameter *html* digunakan untuk menampilkan html. Nilai *default* dari Parameter *html* adalah True. Parameter *session\_id* digunakan untuk mengatur pengacakan. Nilai *default* dari Parameter *session\_id* adalah None. Parameter *system\_log* digunakan untuk menyimpan file *log*. Nilai *default* dari Parameter *system\_log* adalah True. Parameter *log\_experiment* digunakan untuk menaruh parameter dan metrik pada model ke MLflow server. Nilai *default* dari Parameter *log\_experiment* adalah False. Parameter *experiment\_name* digunakan untuk memberi nama pada file log di dalam server MLflow. Nilai *default* dari Parameter *experiment\_name* adalah None. Parameter *log\_plots* digunakan untuk menentukan apakah bisa diplot pada server MLflow. Nilai *default* dari Parameter *log\_plots* adalah False. Parameter *log\_profile* digunakan untuk menampilkan profil file log pada server MLflow. Nilai *default* dari Parameter *log\_profile* adalah False. Parameter *engine* digunakan untuk menentukan *engine backend* pada suatu model. Nilai *default* dari Parameter *engine* adalah None. Parameter *verbose* digunakan untuk menampilkan informasi. Nilai *default* dari Parameter *verbose* adalah True. Parameter *profile* digunakan untuk menampilkan interaktif analisis data eksplorasi. Nilai *default* dari Parameter *profile* adalah False. Parameter *profile\_kwargs* digunakan untuk menambahkan argumen pada profile. Nilai *default* dari Parameter *profile\_kwargs* adalah {}. Parameter *fig\_kwargs* digunakan untuk mengatur beberapa plot yang dikeluarkan pada *library pycaret*. Nilai *default* dari Parameter *fig\_kwargs* adalah {} (Ali 2023).

Create model pada *library pycaret* digunakan untuk membuat model *machine learning*. Pada modul *time series*, model yang digunakan berasal dari *library sktime*. Parameter yang digunakan pada model *sktime* sama dengan parameter turunan seperti model Facebook Prophet dari *library prophet*. Fungsi create model pada *library pycaret* memiliki parameter sebagai berikut:

1. *estimator*
2. *fold*
3. *round*
4. *cross\_validation*
5. *fit\_kwargs*
6. *engine*
7. *verbose*

Parameter *estimator* digunakan untuk menentukan model yang digunakan. Jika nilai estimator ‘stlf’ maka model yang dibuat adalah model STL. Jika nilai estimator ‘prophet’ maka model yang dibuat adalah model Facebook Prophet. Jika nilai estimator *xgboost\_cds\_dt* maka model yang dibuat adalah model XGBoost yang telah dimodifikasi. Parameter *fold* digunakan untuk menentukan banyaknya *fold*. Nilai *default* dari Parameter *fold* adalah None. Parameter *round* digunakan untuk menentukan banyak desimal pada hasil metrik. Nilai *default* dari Parameter *round* adalah 4. Parameter *cross\_validation* digunakan untuk menentukan apakah menerapkan *cross validation* atau tidak. Nilai *default* dari Parameter *cross\_validation* adalah True. Parameter *fit\_kwargs* digunakan untuk membuat argumen untuk metode fit pada *library sktime*. Nilai *default* dari Parameter *fit\_kwargs* adalah {}. Parameter *engine* digunakan untuk menentukan *engine backend* pada model tertentu. Nilai *default* dari Parameter *engine* adalah None. Parameter *verbose* digunakan untuk menampilkan hasil model. Nilai *default* dari Parameter *verbose* adalah True (Ali 2023).

Tune model memiliki tujuan untuk tuning hyperparameter pada model tersebut. Fungsi tune model juga bertujuan untuk meningkatkan akurasi pada model. Berikut ini adalah daftar parameter pada fungsi tune model:

1. *estimator*
2. *fold*
3. *round*
4. *n\_iter*
5. *custom\_grid*
6. *optimize*
7. *custom\_scorer*
8. *search\_algorithm*
9. *choose\_better*
10. *fit\_kwargs*
11. *return\_tuner*
12. *verbose*
13. *tuner\_verbose*

Parameter *estimator* digunakan untuk model yang sudah dilatih. Parameter *fold* digunakan untuk menentukan banyaknya *fold*. Nilai *default* dari Parameter *fold* adalah None. Parameter *round* digunakan untuk menentukan banyak desimal pada hasil metrik. Nilai *default* dari Parameter *round* adalah 4. Parameter *n\_iter* untuk menentukan iterasi pada *tuning hyperparameter*. Nilai *default* dari Parameter *n\_iter* adalah 10. Parameter *custom\_grid* digunakan untuk mendefinisikan rentang parameter yang digunakan untuk *tune model*. Nilai *default* dari Parameter *custom\_grid* adalah None. Parameter *optimize* digunakan untuk metrik evaluasi dari *tune model*. Nilai *default* dari Parameter *optimize* adalah ‘MASE’. Parameter *custom\_scorer* digunakan untuk menambahkan metrik terbaru. Nilai *default* dari Parameter *custom\_scorer* adalah None. Parameter *search\_algorithm* digunakan

untuk menentukan metode *tuning hyperparameter*, apakah menggunakan random atau grid search. Nilai *default* dari Parameter *fold* adalah 'random'. Parameter *choose\_better* digunakan untuk menentukan model yang dipilih adalah terbaik.. Nilai *default* dari Parameter *choose\_better* adalah True. Parameter *fit\_kwargs* digunakan untuk menentukan argumen fit. Nilai *default* dari Parameter *fit\_kwargs* adalah {}. Parameter *return\_tuner* digunakan untuk mengembalikan tuner object. Nilai *default* dari Parameter *return\_tuner* adalah False. Parameter *verbose* digunakan untuk menunjukkan metrik dapat ditampilkan atau tidak.. Nilai *default* dari Parameter *verbose* adalah True. Parameter *tuner\_verbose* digunakan untuk menunjukkan keterangan tuner dapat ditampilkan atau tidak.. Nilai *default* dari Parameter *tuner\_verbose* adalah True (Ali 2023).

Fungsi *finalize\_model* merupakan fungsi akhir dalam pembuatan model *machine learning* karena pada penelitian ini tidak mendeploy model. *Finalize* model bertujuan melatih ulang model pada seluruh data termasuk data uji. Fungsi ini memiliki parameter seperti *estimator* dan *fit\_kwargs*. Parameter *estimator* digunakan untuk model yang dipilih sertadilatih ulang. Parameter *fit\_kwargs* digunakan untuk menambahkan argumen pada metode fit. Nilai *default* dari Parameter *fit\_kwargs* adalah {} (Ali 2023).

Fungsi *predict\_model* menghasilkan *dataframe* dengan kolom *target* yang berindeks *datetime*. Indeks *datetime* dimulai dari indeks *datetime* terakhir pada data *input*, dan berukuran sesuai dengan Parameter *fh* yang ditentukan pada fungsi *setup*. Fungsi *predict\_model* membutuhkan dua parameter, yaitu Parameter *estimator* dan Parameter *X*. Parameter *estimator* merupakan Parameter yang mengatur model yang telah difinalisasi dengan fungsi *finalize\_model*, sedangkan Parameter *X* adalah parameter yang mengatur fitur data yang digunakan untuk meramalkan kolom *target*. Nilai default dari Parameter *X* adalah None.

Parameter yang digunakan untuk fungsi *setup* pada penelitian ini yakni Parameter *target* bernilai 'harga\_emas', Parameter *numeric\_imputation\_target* bernilai 'linear', Parameter *max\_sp\_to\_consider* bernilai None, Parameter *num\_sps\_to\_use* bernilai -1, Parameter *session\_id* bernilai 42, Parameter *fold* bernilai 3 dan Parameter *fh* bernilai 730 (Ali 2023).

### 3.2.6 Tuning parameter

Proses tuning parameter sangat penting untuk meningkatkan performa dan akurasi ketiga model peramalan ini. Pada model Facebook Prophet, parameter *changepoint\_prior\_scale*, *holidays\_prior\_scale*, dan *seasonality\_prior\_scale* perlu disesuaikan. Sedangkan pada model XGBoost, tuning parameter melibatkan *learning\_rate*, *depth*, dan *n\_estimators*. Untuk model STL, parameter-parameter terkait *smoothing* dan tingkat keakuratan dalam pendekatan LOESS harus diatur. *library pycaret* dapat mencari kombinasi parameter terbaik untuk 3 model (Ali, 2020).

Tabel 5 *Tuning* parameter Facebook Prophet

Parameter	Facebook Multivariate	Prophet Facebook Univariate
<i>changepoint_prior_scale</i>	0,040820721203399565	0,016439324494196616
<i>holidays_prior_scale</i>	0,21751953118777642	0,01095960453692584
<i>seasonality_prior_scale</i>	0,014936568554617629	7,886714129990491

Pada Tabel 5 menunjukkan bahwa ada perubahan parameter pada *Multivariate*. Pada nilai parameter *univariate* dan *multivariate* bernilai koma tidak beraturan karena dalam tuning parameter menggunakan metode *random search*. Algoritma *random search* merupakan algoritma sederhana untuk mencari parameter secara acak (Fordana dan Rochmawati 2022).

Tabel 6 *Tuning* parameter STL

Parameter	STL Univariate
<i>seasonal</i>	7
<i>seasonal_deg</i>	1
<i>seasonal_jump</i>	1
<i>sp</i>	8
<i>trend</i>	None
<i>trend_deg</i>	1
<i>trend_jump</i>	1

Tabel 6 menunjukkan bahwa parameter memiliki peran penting terhadap model STL. Parameter *seasonal* merupakan parameter untuk *smoothing seasonal* dan harus bernilai ganjil. Parameter *seasonal\_deg* merupakan derajat untuk *seasonal LOESS*. Parameter *seasonal\_jump* merupakan parameter yang menentukan langkah interpolasi secara linear pada *seasonal*. Parameter *trend* merupakan parameter untuk *smoothing trend* dan harus bernilai ganjil. Parameter *trend\_deg* merupakan derajat untuk *trend*. Parameter *trend\_jump* merupakan parameter yang menentukan langkah interpolasi secara linear pada *trend*. Parameter *sp* merupakan periode pada data tersebut. Maka, pada model STL hanya bisa menggunakan *univariate*.



Tabel 7 *Tuning* parameter XGBoost

Parameter	XGBoost Multivariate	XGBoost Univariate
<i>learning_rate</i>	0,3	0,3
<i>max_depth</i>	6	6
<i>max_leaves</i>	0	0
<i>n_estimators</i>	100	100

Pada Tabel 7 menunjukkan bahwa ada beberapa parameter yang perlu diketahui. Pertama, *learning\_rate* adalah ukuran langkah yang digunakan dalam setiap pembaruan untuk mencegah *overfitting*. Parameter *max\_depth* menentukan kedalaman maksimum dari pohon yang dibangun, dengan nilai yang lebih tinggi akan membuat model lebih kompleks dan berpotensi *overfitting*. Selanjutnya, parameter *max\_leaves* mengontrol jumlah maksimum simpul yang ditambahkan, tetapi tidak digunakan dalam metode pohon. Terakhir, *n\_estimators* merupakan jumlah pohon *gradient boost* yang akan dibangun.

### 3.2.7 Evaluasi Model

Untuk membandingkan model forecasting, terdapat beberapa metode evaluasi statistik yang dapat digunakan seperti *Symmetric Mean Absolute Percentage Error* (SMAPE) dan *Mean Absolute Scaled Error* (MASE). SMAPE dipilih karena tahan terhadap *outlier* dan memiliki sifat *scale-independence* yaitu, pengukuran tidak berpengaruh dengan skala data (Chen *et al.* 2017).

### 3.2.8 Peramalan Data

Peramalan data dilakukan untuk meramalkan data hingga 2 tahun ke depan. Hasil peramalan tersebut akan disimpan dalam bentuk file CSV. Proses peramalan menggunakan ketiga model yakni, model Facebook Prophet, XGBoost, dan STL.

## 3.3 Deskripsi Data

Dataset yang dikonstruksi pada penelitian ini meliputi lima data utama yaitu data harga beli emas, data kurs jual dollar, data kurs beli dollar, data persentase inflasi, dan data suku bunga BI. Data kurs beli dan kurs jual dollar yang digunakan dalam penelitian ini berasal dari situs resmi informasi kurs Bank Indonesia (Bank Indonesia 2023a).

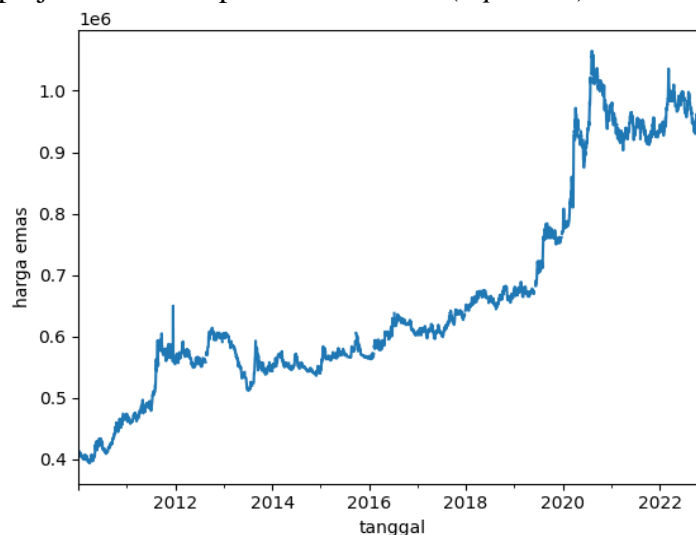
Data persentase inflasi yang digunakan dalam penelitian ini berasal dari situs resmi informasi data inflasi Bank Indonesia (Bank Indonesia 2023b). Data suku bunga BI yang digunakan dalam penelitian ini berasal dari situs informasi Suku bunga Bank Indonesia (Bank Indonesia 2023c). Data harga emas yang digunakan dalam penelitian ini diperoleh dari situs Logam Mulia PT.Antam

(PT.Antam 2022). Pada penelitian ini, data yang digunakan adalah data tanggal 4 Januari 2010 hingga 30 April 2023. Harga emas dihitung berdasarkan harga beli emas dari mata uang rupiah per 1 gram emas.

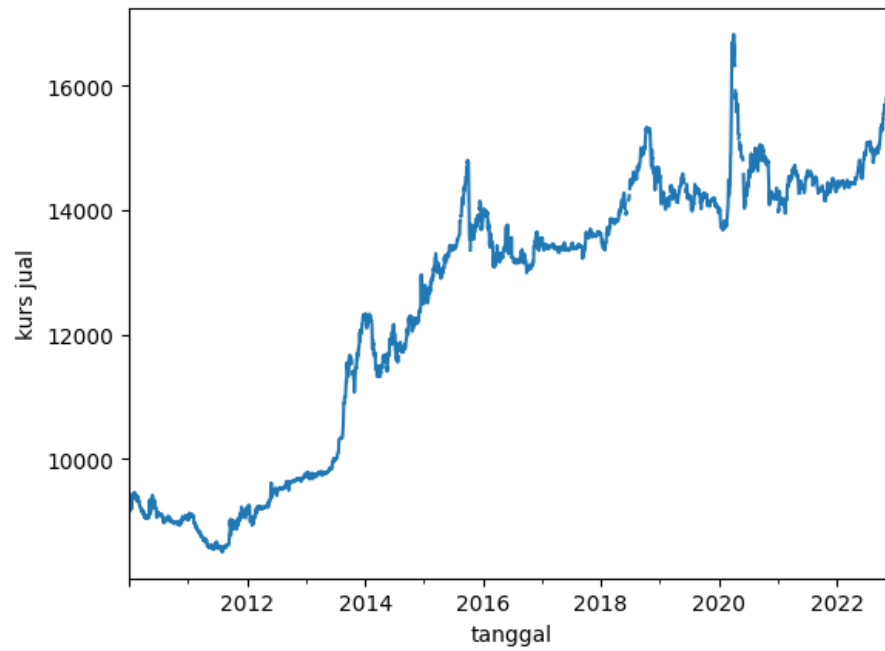
Tabel 8 Master data

Tanggal	Harga emas (Rp)	Kurs jual dollar (Rp)	Kurs beli dollar (Rp)	Data (%)	inflasi	Suku bunga BI (%)
04-01-2010	408.000	9.377	9.283	3,72		6,5
05-01-2010	410.000	9.355	9.261	3,72		6,5
06-01-2010	410.000	9355	9.261	3,72		6,5
07-01-2010	412.000	9274	9182	3,72		6,5
08-01-2010	410.000	9286	9194	3,72		6,5

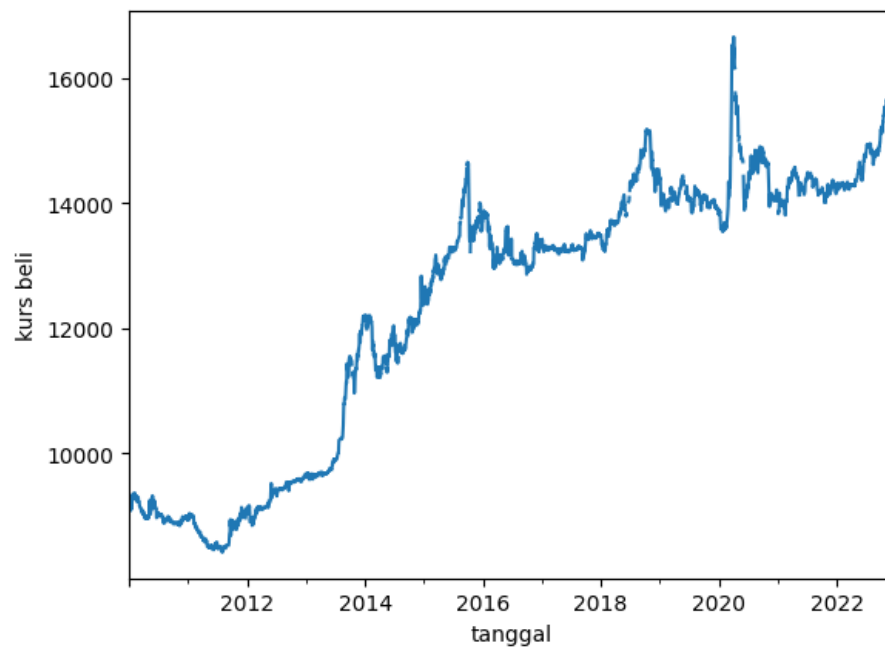
Tabel 8 merupakan tabel master yang dapat menunjukkan bahwa data yang diperoleh terdiri atas enam *field* utama. Secara total, *record* data yang diperoleh sebanyak 3412 baris. Enam *field* data meliputi *Tanggal*, *Harga Emas*, *Kurs Jual Dollar*, *Kurs Beli Dollar*, *Data Inflasi*, dan *Suku Bunga BI*. *Field* Harga Emas menyatakan nilai harga beli emas ANTAM per 1 gram. *Field* kurs jual dollar menyatakan kurs yang ditawarkan oleh bank untuk menjual mata uang dollar (Yusnaini 2023). *Field* kurs beli dollar menyatakan kurs dimana bank bersedia membeli mata uang (Yusnaini 2023). *Field* data inflasi menyatakan persentase kenaikan harga barang dan jasa secara umum dalam jangka waktu tertentu di Indonesia. *Field* suku bunga BI menyatakan suku bunga acuan BI untuk memberikan pinjaman terhadap bank komersial (*repo rate*).



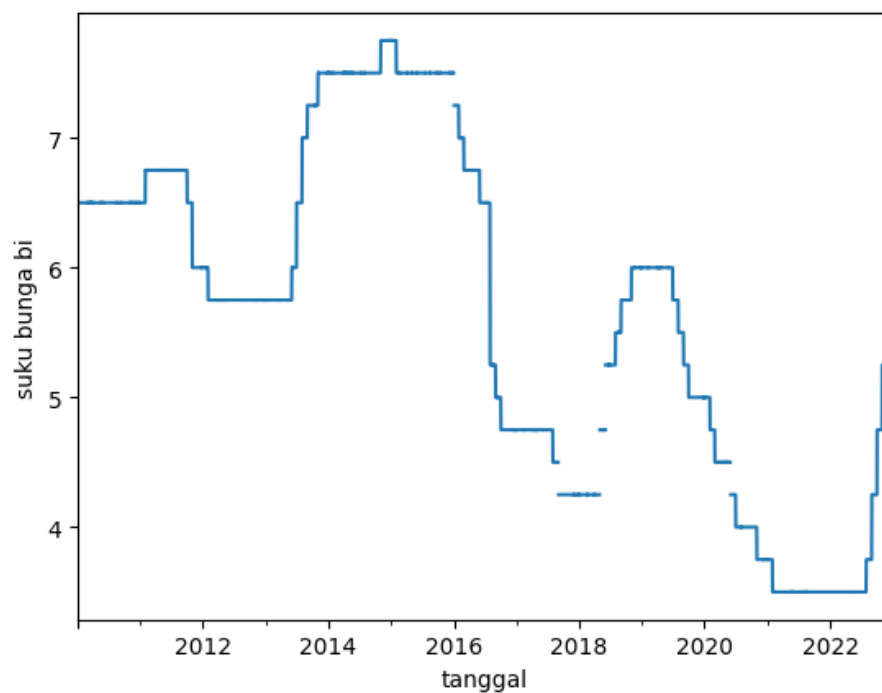
Gambar 3 Grafik harga emas berdasarkan waktu



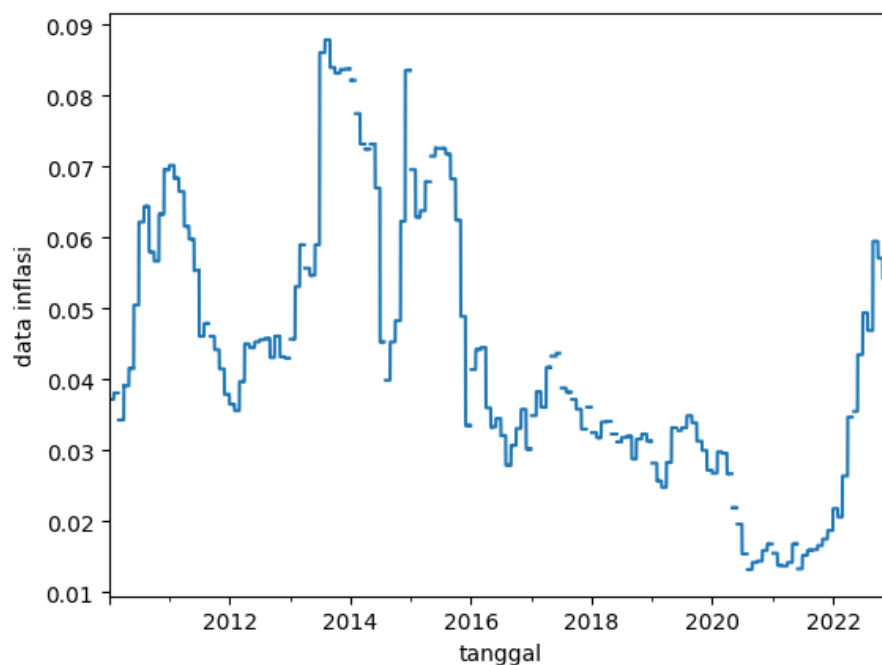
Gambar 4 Grafik kurs jual berdasarkan tanggal



Gambar 5 Grafik kurs beli berdasarkan waktu



Gambar 6 Grafik suku bunga BI berdasarkan waktu



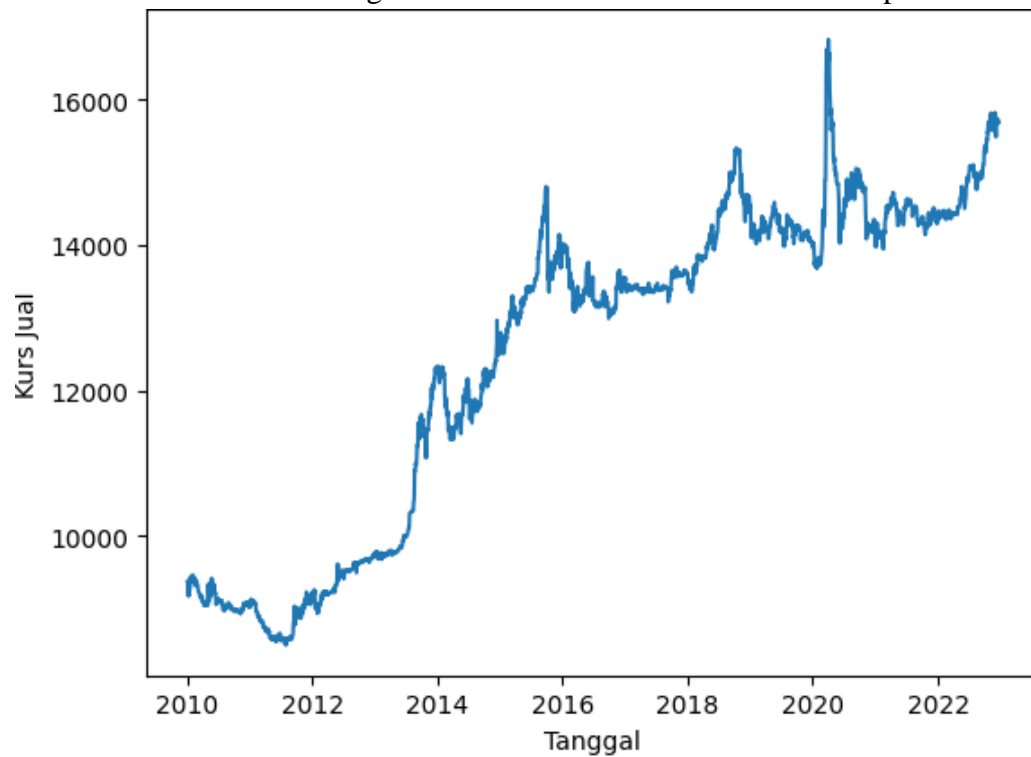
Gambar 7 Grafik inflasi di Indonesia berdasarkan waktu

Pada Gambar 3, terlihat peningkatan yang signifikan dalam harga emas dari tahun 2019 hingga 2020. Grafik ini menunjukkan kenaikan yang stabil. Gambar 4 dan Gambar 5 menampilkan grafik yang serupa untuk kurs jual dan kurs beli. Titik puncak untuk keduanya terjadi pada tahun 2020. Pada Gambar 6, grafik suku bunga terlihat tidak lengkap. Nilai tertinggi tercatat dari tahun 2014 hingga 2016. Pada

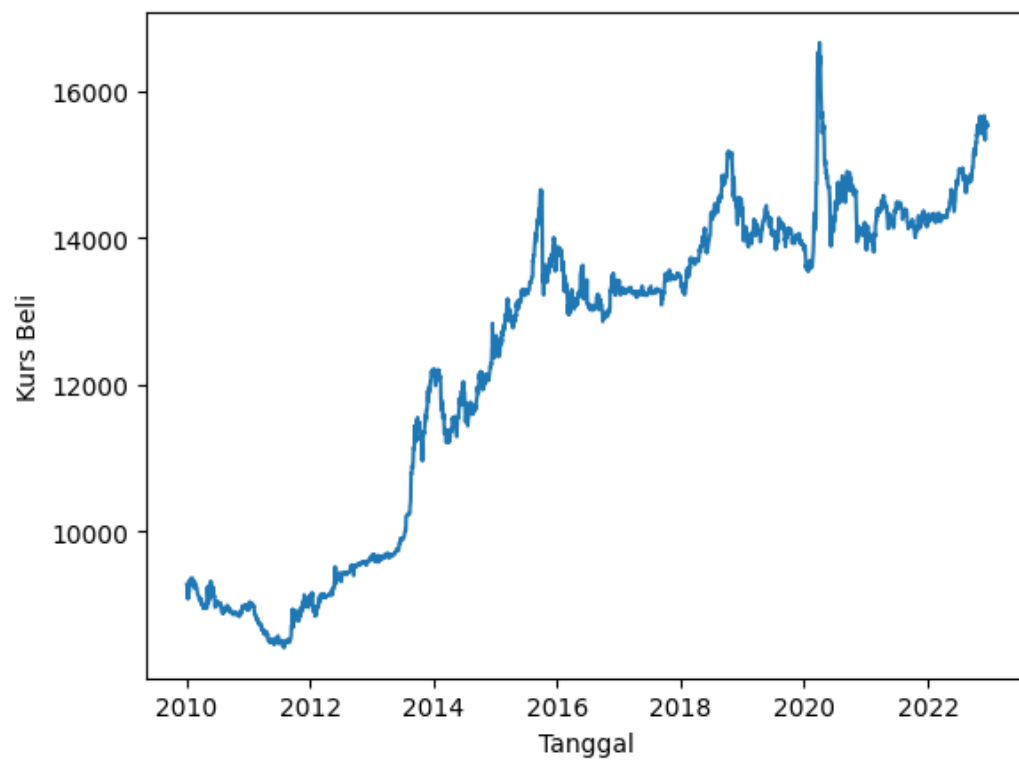
Gambar 7, inflasi mengalami peningkatan pada tahun 2013-2014 dan kemudian mengalami penurunan pada tahun 2020-2021.



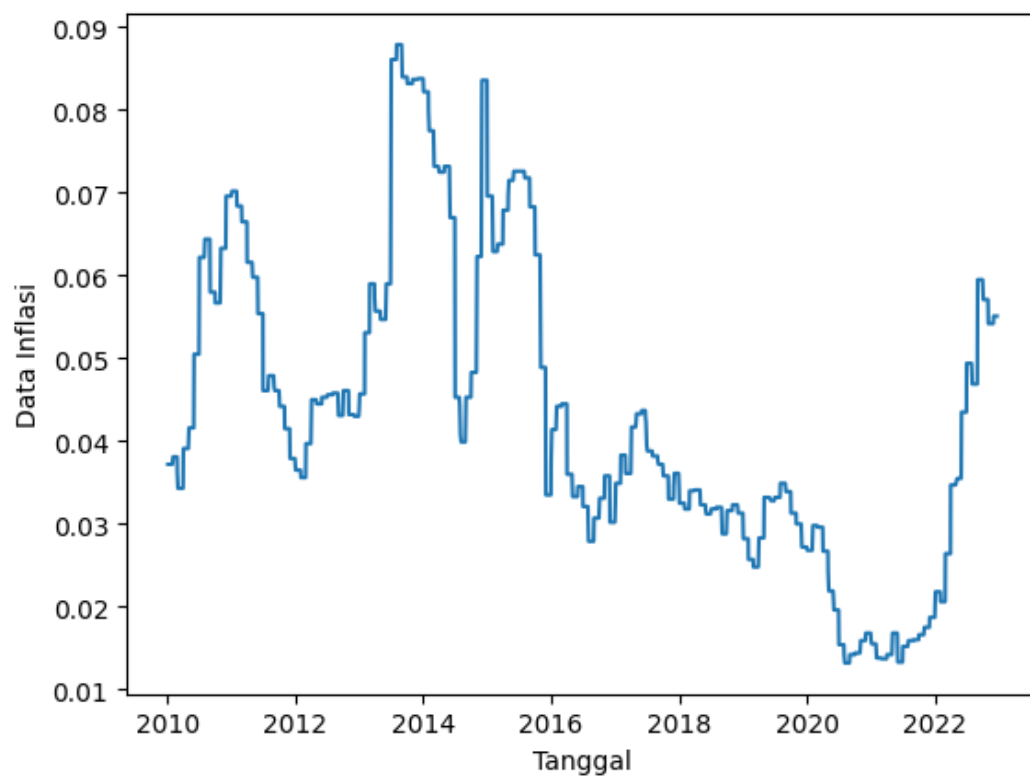
Gambar 8 Grafik harga emas berdasarkan waktu setelah diimputasi



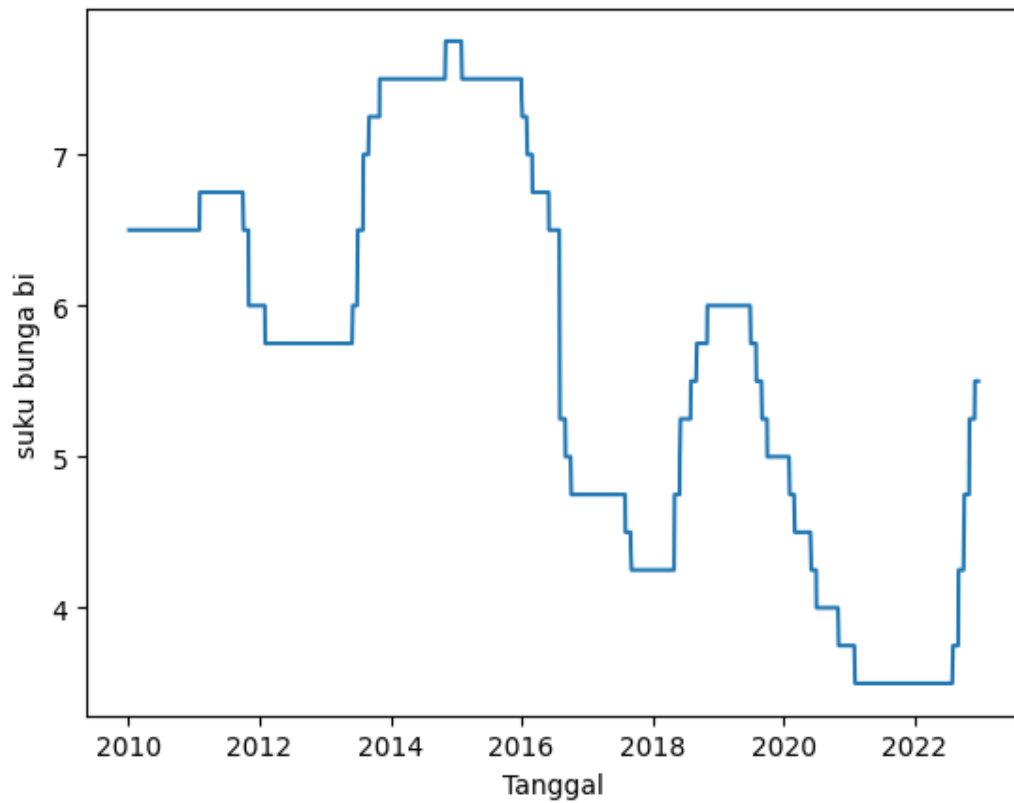
Gambar 9 Grafik kurs jual berdasarkan waktu setelah diimputasi



Gambar 10 Grafik kurs beli berdasarkan waktu setelah diimputasi

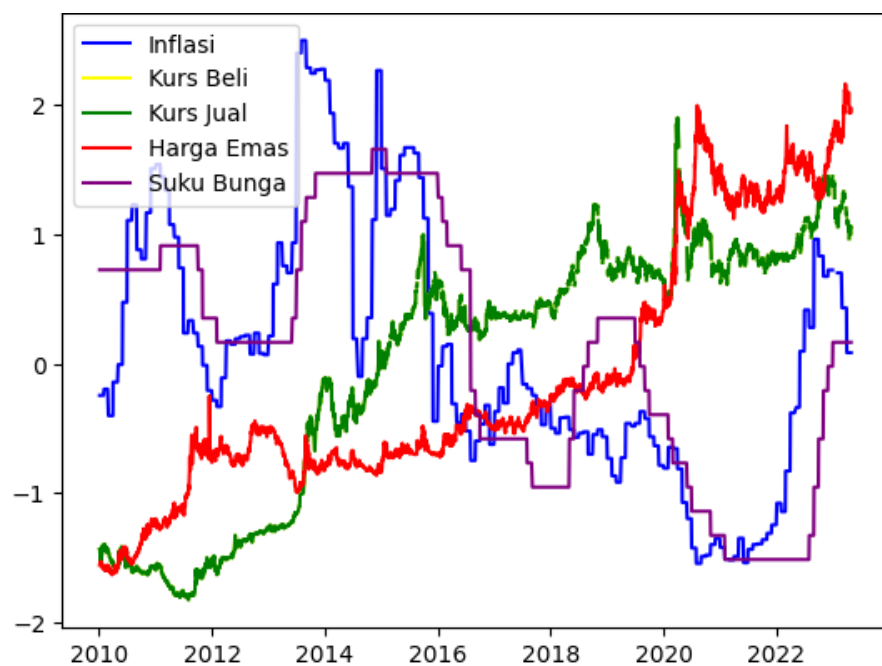


Gambar 11 Grafik data inflasi berdasarkan waktu setelah diimputasi



Gambar 12 Grafik suku bunga BI berdasarkan waktu setelah diimputasi

Pada Gambar 8 hingga 12, data yang hilang telah diimputasi dengan pendekatan interpolasi linear. Interpolasi dilakukan dengan tanggal sebagai variabel independen.

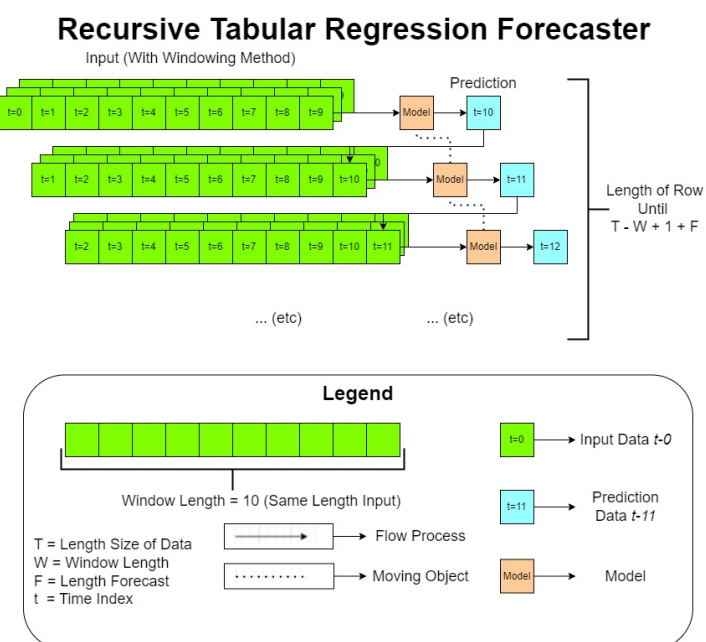


Gambar 13 Grafik normalisasi master data

Gambar 13 menunjukkan bahwa grafik kurs beli dan jual dolar terhadap rupiah memiliki tren yang cenderung berdekatan, meskipun terdapat selisih kecil di antara keduanya. Sementara itu, grafik inflasi dan suku bunga juga menunjukkan adanya hubungan yang berdekatan, meskipun inflasi cenderung lebih fluktuatif. Pada tahun 2020, harga emas dan kurs jual memiliki puncak yang berdekatan.

### 3.4 Skenario Eksperimen

Skenario eksperimen dilakukan dengan menggunakan tiga model *machine learning*, yakni XGBoost, STL, dan Facebook Prophet. Khususnya untuk XGBoost dan Facebook Prophet, akan diterapkan analisis *univariate* (ramalan dengan sebuah fitur) dan analisis *multivariate* (ramalan dengan lebih dari satu fitur, pada penelitian ini digunakan lima fitur). Adapun untuk model STL hanya digunakan analisis *univariate*. *Input* data untuk analisis *univariate* terdiri dari data harga emas masa lalu. *Input* data untuk analisis *multivariate* terdiri dari data harga emas, persentase inflasi, harga kurs jual, harga beli dollar, dan suku bunga BI. Data yang digunakan pada analisis *multivariate* berupa data *multivariate time series* bukan data *panel*. Data *multivariate time series* merupakan data *time series* yang memiliki beberapa variabel pada suatu objek tertentu diamati seiring waktu. Sementara data *panel* merupakan data yang mengukur variabel yang sama dari waktu ke waktu, tetapi dari beberapa unit yang berbeda. Unit-unit ini dapat berupa individu, perusahaan, atau bahkan negara (Löning *et al.* 2019). *Output* dari seluruh model merupakan hasil ramalan harga emas 730 hari ke depan.

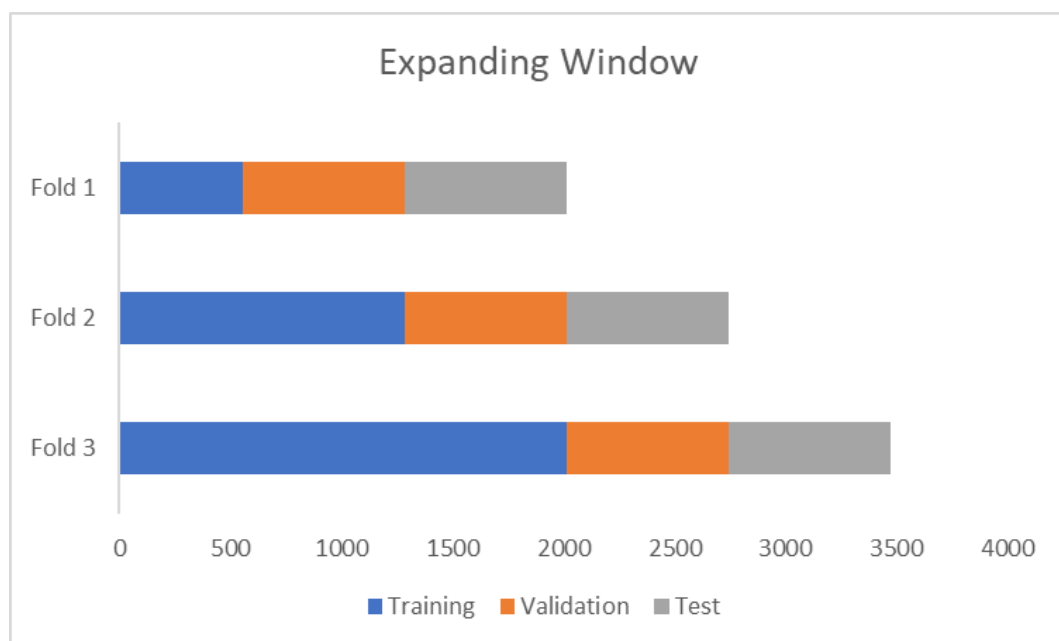


Gambar 14 Ilustrasi Metode *Recursive Tabular Regression Forecaster* pada *library sktime* (Löning *et al.* 2019)



Gambar 14 menunjukkan ilustrasi metode *Recursive Tabular Regression Forecaster* yang menggunakan *input* data dari 10 hari terakhir untuk memprediksi 1 hari ke depan. *Input* data dapat berupa satu fitur atau lebih. Fitur tersebut dibagi menjadi bagian-bagian kecil dengan metode *windowing*. *Windowing* adalah teknik membagi data deret waktu menjadi lebih kecil, baik tumpang tindih maupun tidak tumpang tindih. Model yang digunakan adalah model regresi.

Dalam penelitian ini, model XGBoost menggunakan metode *Recursive Tabular Regression Forecaster* dengan panjang *window* 10 hari ke belakang. Sementara, model Facebook Prophet dan model STL langsung terhubung pada library Facebook Prophet dan *statsmodels* tanpa menggunakan metode *windowing* seperti pada metode *Recursive Tabular Regression Forecaster*.



Gambar 15 *Expanding Window* dari Penelitian Kingphai dan Moshfeghi (2023)

Pada Skenario eksperimen, penelitian ini melakukan validasi data. Validasi data dilakukan dengan metode *cross-validation* khusus data deret waktu dengan menggunakan *expanding window* dengan *fold* sebanyak 3 kali untuk menghindari *overfitting* dan *underfitting*. Penelitian Kingphai dan Moshfeghi (2023) menemukan bahwa model yang menggunakan *cross validation* dengan menggunakan metode *expanding window* lebih baik dibandingkan metode lainnya. Pada Penelitian de Barbosa Ferreira (2021), *expanding window* merupakan teknik resampling dimana penggunaan ukuran data latih yang semakin bertambah pada setiap *fold* berikutnya, sementara ukuran data uji tetap konstan namun bergeser dengan setiap *fold* berikutnya. Ilustrasi *expanding window* ada di Gambar 15.

Penelitian Dhake *et al.* (2023) mendapatkan akurasi lebih tinggi dengan *Tuning hyperparameter*. *Tuning hyperparameter* dilakukan pada ketiga model

dengan menggunakan metode *random search*. Keseluruhan skenario eksperimen diberikan pada Tabel 9.

Tabel 9 Skenario Eksperimen

Model	Data	Fold			
		Nama Fold	Training	Validation	Test
Facebook Prophet	Univariate	Fold 1	555	730	730
		Fold 2	1285	730	730
		Fold 3	2015	730	730
	Multivariate	Fold 1	555	730	730
		Fold 2	1285	730	730
		Fold 3	2015	730	730
STL	Univariate	Fold 1	555	730	730
		Fold 2	1285	730	730
		Fold 3	2015	730	730
XGBoost	Univariate	Fold 1	555	730	730
		Fold 2	1285	730	730
		Fold 3	2015	730	730
	Multivariate	Fold 1	555	730	730
		Fold 2	1285	730	730
		Fold 3	2015	730	730

Tabel 9 menunjukkan bahwa keseluruhan skenario eksperimen ada 15. Pada data *training* dan *validation* bersifat berubah-ubah berdasarkan metode *expanding window*, tetapi pada data *test* tidak ada yang berubah.

### 3.5 Lingkungan Penelitian

Spesifikasi perangkat lunak, perangkat keras dan lingkungan buatan yang akan digunakan untuk penelitian ini adalah sebagai berikut:

1. Perangkat Keras:
  - Memory memiliki RAM 8 GB
  - Processor memiliki tipe AMD Ryzen 5 3500U dengan Radeon Vega Mobile Gfx (8 CPU) ~ 2.1 GHz
  - GPU menggunakan tipe AMD Radeon(TM) Vega 8 Graphics
2. Perangkat Lunak
  - Sistem Operasi menggunakan Windows 11

- Bahasa Pemrograman menggunakan Python
3. Lingkungan Buatan
- Kaggle Environment (server online)
    - 12 jam untuk waktu eksekusi pada prosesor CPU/GPU dan 9 jam untuk waktu eksekusi pada prosesor TPU
    - 20 GB penyimpanan disk
    - Spesifikasi CPU
      - 4 Cores CPU
      - RAM 30 GB
    - P100 GPU Specifications
      - 1 Nvidia Tesla P100 GPU
      - 2 CPU cores
      - 13 Gigabytes of RAM
    - T4 x2 GPU Specifications
      - 2 Nvidia Tesla T4 GPUs
      - 2 CPU cores
      - 13 Gigabytes of RAM
    - TPU Specifications
      - 4 CPU cores
      - 16 Gigabytes of RAM
    - TPU IVM Specifications
      - 96 CPU cores
      - 330 Gigabytes of RAM
    - Penginstalan *default library* ada di <https://github.com/Kaggle/docker-python>

## IV HASIL DAN PEMBAHASAN

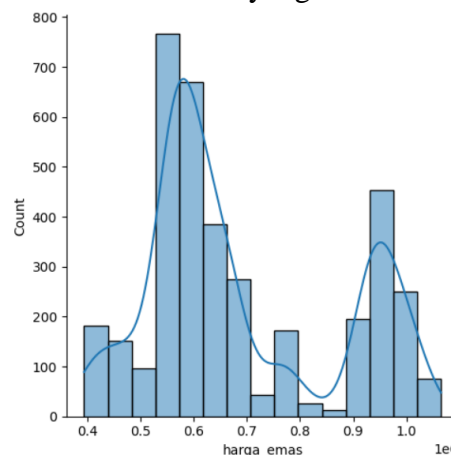
### 4.1 Analisis Statistik Data

Analisis statistik adalah proses pemrosesan data yang melibatkan penyaringan, pembersihan, transformasi, pemrosesan, dan pemodelan data. Tujuannya adalah untuk menemukan informasi yang berguna yang dapat digunakan sebagai dasar untuk kesimpulan atau keputusan (Sahabuddin *et al.* 2021).

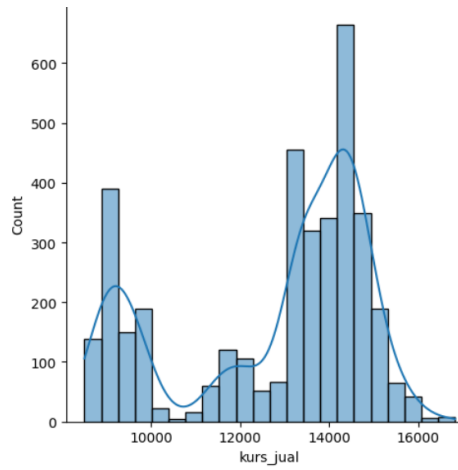
Tabel 10 Statistik Master Data

	Harga Emas (Rp)	Kurs Dollar (Rp)	Jual (Rp)	Kurs Dollar (Rp)	Beli (Rp)	Data (%)	Inflasi	Suku (%)	Bunga	BI
<b>count</b>	3.475	3.475	3.475	3.475	3.475	3.475		3.475		
<b>mean</b>	674.049,7	1.257,75	12450.58	12450.58	4,321	5,652				
<b>std</b>	179.028,8	2.229,803	2207.624	2207.624	1,854	1,317				
<b>min</b>	394.000	8.502	8418	8418	1,32	3,5				
<b>25%</b>	560.000	9.789,5	9691.5	9691.5	3,16	4,5				
<b>50%</b>	607.000	13.405	13271	13271	3,97	5,75				
<b>75%</b>	777.000	14.330,8	14188.35	14188.35	5,54	6,75				
<b>max</b>	1.096.000	16.824,71	16657.3	16657.3	8,79	7,75				

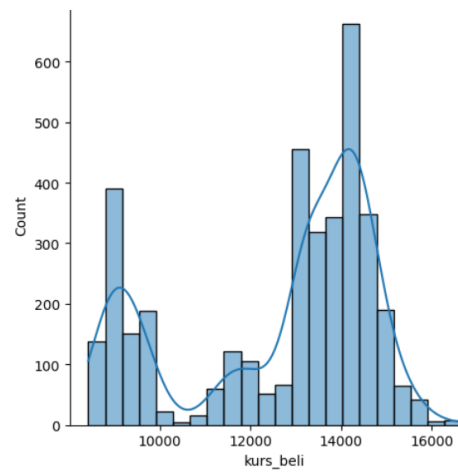
Tabel 10 merupakan contoh tabel data yang berisi nilai rata-rata dan median untuk harga emas, kurs jual, kurs beli, data inflasi, dan suku bunga. Hasil analisis menunjukkan bahwa nilai rata-rata dan median pada fitur tersebut cenderung mendekati nilai mean yang berarti bahwa distribusinya cenderung mendekati distribusi normal. Pendekatan distribusi normal pada data penting dalam analisis statistik dan model *machine learning* karena memungkinkan penggunaan teknik statistik lanjutan dengan asumsi distribusi yang terdefinisi.



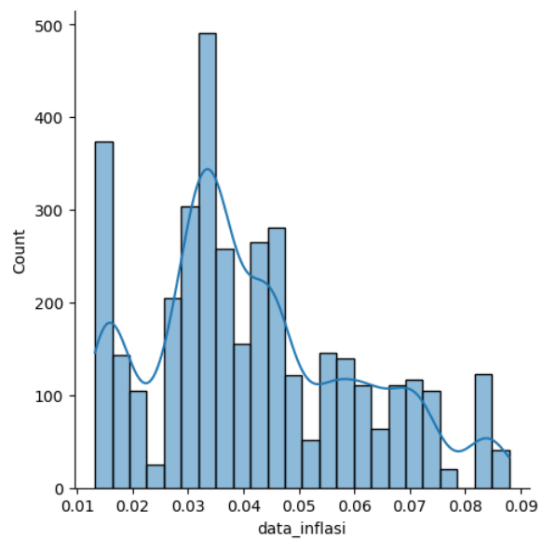
Gambar 16 Persebaran data harga emas



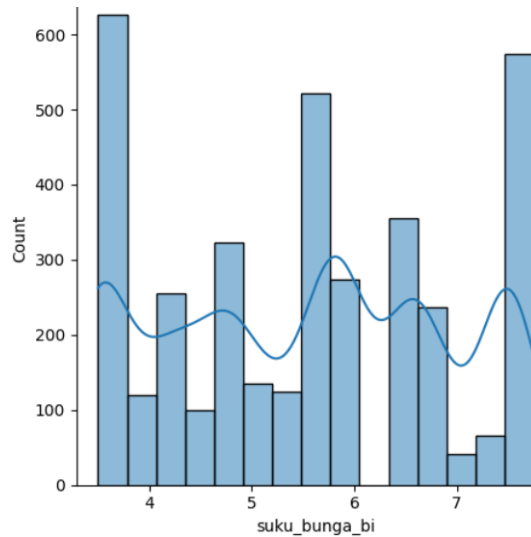
Gambar 17 Persebaran kurs jual dollar



Gambar 18 Persebaran kurs beli dollar

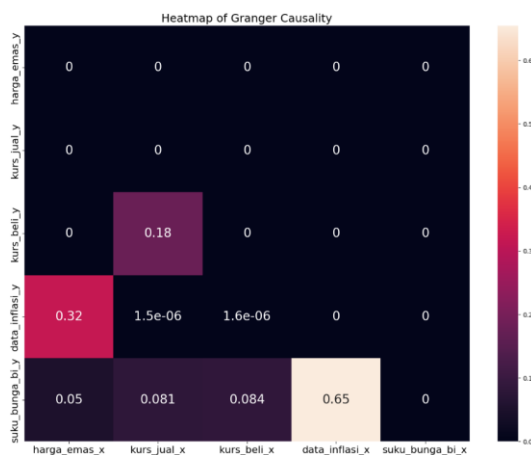


Gambar 19 Persebaran data inflasi



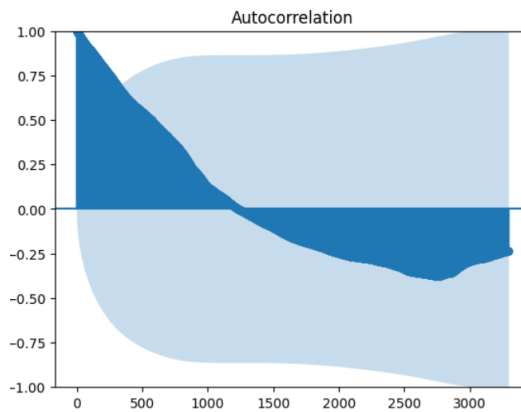
Gambar 20 Persebaran data suku bunga BI

Pada Gambar 16 - 18 terdapat pola distribusi bimodal dalam grafik harga emas, kurs jual, dan kurs beli yang ditandai dengan dua puncak data. Kondisi ini berpengaruh dalam analisis statistik untuk penentuan model yang sesuai dengan asumsi. Pada Gambar 16, *binwidth* pada histogram tersebut sebesar Rp 43.875,00. Berarti modus dari harga emas yakni antara Rp 525.625,00 - Rp 569.500,00. Pada Gambar 17, *binwidth* pada histogram tersebut sebesar Rp 346,78. Berarti modus dari harga kurs jual dollar yakni antara Rp 14.050,47 - Rp 14.397,25. Pada Gambar 18, *binwidth* pada histogram tersebut sebesar Rp 343.30. Berarti modus dari harga kurs beli dollar yakni antara Rp 13.910,86 - Rp 14.254,17. Pada Gambar 19, *binwidth* pada histogram tersebut sebesar 0,3%. Berarti modus dari persentase inflasi yakni antara 3,19% - 3,50%. Pada Gambar 20, *binwidth* pada histogram tersebut sebesar 0,28%. Berarti modus dari persentase suku bunga yakni antara 3,5% - 3,78%.

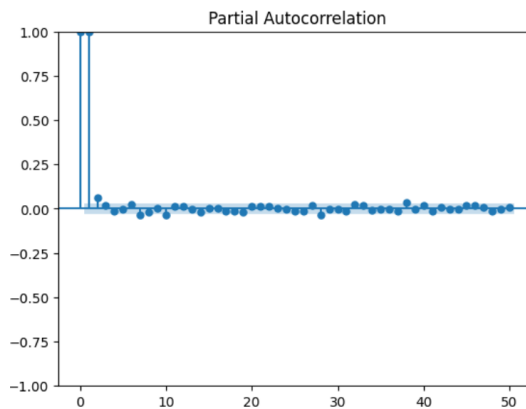


Gambar 21 Heatmap *p-value* Granger Causality

Pada Gambar 21 terlihat bahwa sumbu-x sebagai variabel independensi dan sumbu-y menjadi variabel dependensi. Nilai ini didasarkan pada hasil p-value dari uji *Granger causality* pada variabel x dan variabel y. Pada baris *harga\_emas\_y* yang menunjukkan nilai sebesar 0. Dengan mengacu pada hipotesis alternatif, yaitu bahwa nilai p-value kurang dari 0,05 berarti menunjukkan kurs dollar, inflasi, dan suku bunga secara signifikan mempengaruhi perubahan harga emas. Hal ini memiliki implikasi penting dalam analisis ekonomi karena memahami hubungan kausal antara fitur-fitur tersebut dapat membantu dalam pengambilan keputusan investasi dan kebijakan ekonomi yang lebih tepat.



Gambar 22 Grafik autocorrelation



Gambar 23 Grafik partial autocorrelation

Pada Gambar 22 dan 23 menampilkan plot *autocorrelation* dan *partial autocorrelation*. Grafik ini memungkinkan harga emas tidak stasioner dan kemungkinan terdapat pola musiman. Harga emas tidak stasioner sebab plot *autocorrelation* tidak menurun secara cepat mendekati nol (Tinungki 2019). Pola musiman bisa saja menyebabkan harga emas tidak stasioner.

#### 4.2 Perbandingan Model dan Hasil Peramalan

Tabel 11 dan Tabel 12 menampilkan tabel kinerja model Facebook Prophet dengan metrik dan jenis data yang berbeda. Fold yang digunakan keduanya sama. Berdasarkan tabel tersebut, nilai *error* pada Tabel 11 lebih kecil dibanding Tabel 12. Hal ini bisa disimpulkan bahwa model Facebook Prophet dengan data *univariate* lebih baik dibandingkan model Facebook Prophet dengan data *multivariate*.

Tabel 11 Kinerja Model Facebook Prophet pada Data *univariate* dengan Metode *cross validation (expanding window)*

Fold	MASE	SMAPE
Fold 0	2,6642	0,0353
Fold 1	5,6967	0,0675
Fold 2	9,7766	0,0855

Tabel 12 Kinerja Model Facebook Prophet pada Data *multivariate* dengan Metode *cross validation (expanding window)*

Fold	MASE	SMAPE
Fold 0	8,7202	0,1092
Fold 1	13,0384	0,1647
Fold 2	14,5302	0,1330

Tabel 13 dan Tabel 14 menampilkan tabel kinerja model XGBoost dengan metrik dan jenis data yang berbeda. Fold yang digunakan keduanya sama. Berdasarkan tabel tersebut, nilai *error* pada sebagian besar Tabel 10 lebih kecil dibanding Tabel 11. Hasil ini menunjukkan bahwa pada kasus ini model XGBoost untuk data *univariate* lebih baik dibandingkan dengan model XGBoost untuk data *multivariate*.

Tabel 13 Kinerja Model XGBoost pada Data *univariate* dengan Metode *cross validation (expanding window)*

Fold	MASE	SMAPE
Fold 0	18,5216	0,2133
Fold 1	1,7189	0,0197
Fold 2	10,3050	0,0907



Tabel 14 Kinerja Model XGBoost pada Data *multivariate* dengan Metode *cross validation (expanding window)*

<b>Fold</b>	<b>MASE</b>	<b>SMAPE</b>
Fold 0	20,8881	0,2377
Fold 1	6,4214	0,0714
Fold 2	8,1354	0,0697

Tabel 15 menunjukkan kinerja model STL dengan beragam metrik. Pada *fold 1*, nilai *error* lebih rendah dibanding *fold* yang lain. Pada hal ini, *fold 1* memiliki pola yang cocok dengan model STL dibandingkan *fold* lainnya. *Fold 2* memiliki ketidakaturan lebih tinggi dibandingkan dengan yang lain.

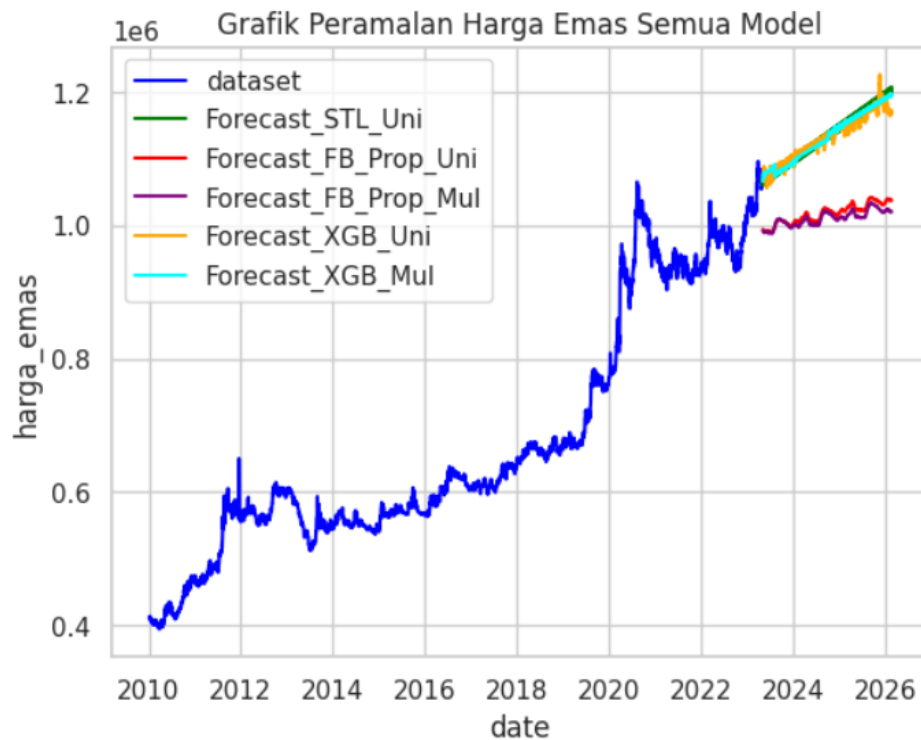
Tabel 15 Kinerja Model STL pada Data *univariate* dengan Metode *cross validation (expanding window)*

<b>Fold</b>	<b>MASE</b>	<b>SMAPE</b>
Fold 0	14,8528	0,1762
Fold 1	2,3609	0,0276
Fold 2	7,5521	0,0642

Tabel 16 menunjukkan hasil evaluasi seluruh model dengan data *univariate* dan *multivariate*. Hasil perhitungan MASE dan SMAPE pada Tabel 16 merupakan nilai rata-rata dari fold 0 sampai fold 2 pada setiap model. Model yang terbaik adalah model Facebook Prophet dengan MASE sebesar 6,0458 dan SMAPE sebesar 0,0628. Berdasarkan hasil ini, model Facebook Prophet memiliki nilai akurasi yang lebih tinggi dalam memprediksi nilai harga emas dibandingkan dengan model yang lain.

Tabel 16 Kinerja seluruh model

<b>Model</b>	<b>MASE</b>	<b>SMAPE</b>	<b>Jenis Data</b>
Facebook Prophet	6,0458	0,0628	<i>Univariate</i>
XGBoost	10,1818	0,1079	<i>Univariate</i>
STL	8,2552	0,0893	<i>Univariate</i>
Facebook Prophet	12,0963	0,1356	<i>Multivariate</i>
XGBoost	11,8150	0,1263	<i>Multivariate</i>



Gambar 24 Grafik Peramalan Harga Emas Semua Model

Pada Gambar 24 terlihat grafik hasil peramalan harga emas dari semua model. Dataset yang digunakan untuk melatih model ini mencakup periode tahun 2010 hingga 2023. Dataset ini digunakan untuk meramalkan harga emas selama dua tahun berikutnya, yaitu tahun 2024 hingga 2026. Grafik hasil peramalan dari model XGBoost *univariate*, model XGBoost *multivariate*, dan model STL *univariate* menunjukkan tingkat kesamaan yang sangat tinggi. Sedangkan peramalan dari model Facebook Prophet *univariate* memiliki hasil yang cukup serupa dengan model Facebook Prophet *multivariate*. Pada Gambar 23 terlihat adanya dua kluster pada grafik hasil peramalan tersebut.

Tabel 17 Hasil Peramalan Seluruh Model

Tanggal	Harga emas Facebook Prophet <i>Univariate</i>	Harga emas Facebook Prophet <i>Multivariate</i>	Harga emas XGBoost <i>Univariate</i>	Harga emas XGBoost <i>Multivariate</i>	Harga emas STL <i>Univariate</i>
01-05-2023	993.819	992.117	1.066.227	1.065.853	1.066.059
02-05-2023	992.870	991.196	1.073.365	1.075.033	1.070.750
03-05-2023	992.441	990.771	1.079.915	1.077.176	1.059.778
04-05-2023	992.238	990.611	1.082.113	1.076.967	1.066.035
05-05-2023	992.045	990.477	1.085.596	1.074.917	1.061.929

Pada Tabel 17 ditunjukkan bahwa harga emas memiliki nilai yang beragam pada seluruh model. Seperti yang terlihat pada Tabel 17 bahwa harga emas Facebook Prophet *univariate* dan *multivariate* memiliki nilai yang serupa, yakni sekitar Rp 990.000,00. Pada harga emas XGBoost *univariate*, XGBoost *multivariate*, dan STL *univariate* memiliki nilai yang serupa, yakni sekitar Rp 1.070.000,00.

## V KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Penelitian ini berhasil membangun tiga model *time series*, yakni Facebook Prophet, XGBoost, dan STL dengan membandingkan ketiga model berdasarkan nilai metrik SMAPE dan MASE untuk data *multivariate* dan *univariate*. Pada data *univariate*, model yang digunakan adalah Facebook Prophet, XGBoost, dan STL. Akan tetapi, untuk data *multivariate* model yang digunakan adalah XGBoost dan Facebook Prophet.

Pada data *univariate*, model Facebook Prophet menunjukkan kinerja yang paling baik dengan metrik MASE sebesar 6,0458 dan metrik SMAPE sebesar 6,28%. Dari metrik SMAPE dapat disimpulkan bahwa model Facebook Prophet dengan data *univariate* memiliki akurasi yang sangat tinggi (*highly accurate forecasting*).

Pada data *multivariate*, model XGBoost menunjukkan kinerja yang paling baik dengan metrik MASE sebesar 11,815 dan metrik SMAPE sebesar 12,63%. Dari metrik SMAPE dapat disimpulkan bahwa model XGBoost dengan data *multivariate* memiliki akurasi yang tinggi (*good forecasting*).

Jadi, model yang lebih baik digunakan berdasarkan penelitian ini adalah model yang telah dilatih dengan data *univariate*. Dalam penelitian ini diharapkan bahwa penggunaan model Facebook Prophet dengan data *univariate* dan XGBoost dengan data *multivariate* dapat memberikan kontribusi yang signifikan dalam pengambilan keputusan investasi yang lebih akurat dan efektif pada pasar emas.

### 5.2 Saran

Terdapat beberapa saran untuk penelitian ke depan agar dapat meningkatkan ramalan harga emas di masa depan melalui beberapa cara. Pertama, perlu dipertimbangkan faktor-faktor eksternal lain yang kemungkinan dapat mempengaruhi pergerakan harga emas agar ramalan menjadi lebih akurat. Selain itu, untuk mengatasi masalah imputasi nilai yang hilang dapat dicoba metode seperti Random Forest atau MICE (*Multivariate Imputation By Chained Equations*). Analisis ekonometrika dan analisis spektral juga dapat dipertimbangkan untuk meningkatkan akurasi. Dengan menggunakan cara tersebut, diharapkan hasil ramalan harga emas akan menjadi lebih handal dan relevan yang dapat mendukung pengambilan keputusan investasi yang lebih baik di masa depan.

## DAFTAR PUSTAKA

- Ali, M. (2023, April). PyCaret: An open source, low-code machine learning library in Python (PyCaret version 1.0.0). <https://www.pycaret.org>
- Alameer, Z., Abd Elaziz, M., Ewees, A. A., Ye, H., dan Jianhua, Z. (2019). Forecasting gold price fluctuations using improved multilayer perceptron neural network and whale optimization algorithm. *Resources Policy*, 61, 250-260.
- Anandasayanan, S., Thevananth, J., dan Mathuranthy, A. (2019). The Relationship Between Inflation and Gold Price: Evidence From Sri Lanka. *International Journal of Accounting and Financial Reporting*, 9(4), 322. <https://doi.org/10.5296/ijafr.v9i4.15750>
- Hyndman, R. J., dan Athanasopoulos, G. (2018). *Forecasting: Principles and practice* (2nd ed.). OTexts.
- Armstrong, J. S. (1985). *Long-range forecasting: From crystal ball to computer* (2nd ed.). Wiley.
- Ben Jabeur, S., Mefteh-Wali, S., dan Viviani, J.-L. (2022). Forecasting gold price with the XGBoost algorithm and SHAP interaction values. *Journal of Financial Forecasting*, 10(2), 25-40. <https://doi.org/10.1000/jff.2022.10.2.25>.
- Chai, J., Zhao, C., Hu, Y., dan Zhang, Z. G. (2021). Structural analysis and forecast of gold price returns. In *Journal of Management Science and Engineering* (Vol. 6, Issue 2, pp. 135–145). Elsevier BV. <https://doi.org/10.1016/j.jmse.2021.02.011>
- Chen, T., dan Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). New York, NY, USA: ACM. <https://doi.org/10.1145/2939672.2939785>
- Cleveland, R. B., Cleveland, W. S., dan Terpenning, I. (1990). STL: A Seasonal-Trend Decomposition Procedure Based on Loess. *Journal of Official Statistics*, 6(1), 3. <https://www.proquest.com/scholarly-journals/stl-seasonal-trend-decomposition-procedure-based/docview/1266805989/se-2>
- Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368), 829-836. <https://doi.org/10.1080/01621459.1979.10481038>
- Farhat, S., dan Ghalayini, L. (2020). *Modeling and Forecasting Gold Prices*. Research Square Platform LLC.
- Kishann, H., dan RamaParvathy, L. (2022). A Novel Approach for Correlation Analysis on FBProphet to Forecast Market Gold Rates with Linear Regression. In *Advances in Parallel Computing Algorithms, Tools and Paradigms*. IOS Press. <https://doi.org/10.3233/apc220037>
- Korstanje, J. (2021). *The Prophet Model*. In: *Advanced Forecasting with Python*. Apress, Berkeley, CA.

- Logam Mulia. (2022). Harga Emas Hari Ini. Diakses pada 2 Juli 2023 dari <https://www.logammulia.com/id/harga-emas-hari-ini>
- Makala, D., dan Li, Z. (2021). Prediction of gold price with ARIMA and SVM. In *Journal of Physics: Conference Series* (Vol. 1767, Issue 1, p. 012022). IOP Publishing. <https://doi.org/10.1088/1742-6596/1767/1/012022>
- Makridakis, S., Spiliotis, E., dan Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLOS ONE*, 13(3), e0194889. <https://doi.org/10.1371/journal.pone.0194889>
- Mohtasham Khani, M., Vahidnia, S., dan Abbasi, A. (2022). A deep learning-based method for forecasting gold price with respect to pandemics. *Journal of Financial Forecasting*, 10(2), 25-40.
- Risse, M. (2019). Combining wavelet decomposition with machine learning to forecast gold returns. In *International Journal of Forecasting* (Vol. 35, Issue 2, pp. 601–615). Elsevier BV. <https://doi.org/10.1016/j.ijforecast.2018.11.008>
- Sahabuddin, R., Idrus, M. I., dan Karim, A. (2021). *Pengantar Statistika* (1st ed.). Liyan Pustaka Ide.
- Shi, Y., Li, J., dan Li, Z. (2018). Gradient Boosting With Piece-Wise Linear Regression Trees (Version 3). arXiv. <https://doi.org/10.48550/ARXIV.1802.05640>
- Taylor, S. J., dan Letham, B. (2018). Forecasting at scale. *The American Statistician*, 72(1), 37-45. <https://doi.org/10.1080/00031305.2017.1380080>
- Tinungki, G. M. (2019). The analysis of partial autocorrelation function in predicting maximum wind speed. *IOP Conference Series: Earth and Environmental Science*, 235(1), 012097.
- Cleveland, W. S., dan Devlin, S. J. (1988). Locally weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83(403), 596–610. <https://doi.org/10.1080/01621459.1988.10478639>
- Ouyang, Z., Ravier, P., dan Jabloun, M. (2021). STL Decomposition of Time Series Can Benefit Forecasting Done by Statistical Methods but Not by Machine Learning Ones. *Engineering Proceedings*, 5(1), 42. <https://doi.org/10.3390/engproc2021005042>
- Das, S., Nayak, J., Kamesh Rao, B., Vakula, K., dan Ranjan Routray, A. (2021). Gold price forecasting using machine learning techniques: Review of a decade. *Computational intelligence in pattern recognition* (pp. 679–695). Springer Singapore. [https://doi.org/10.1007/978-981-16-2543-5\\_58](https://doi.org/10.1007/978-981-16-2543-5_58)
- Syrris, I., dan Shenai, V. (2021). Forecasting Gold Prices with ARIMA and GARCH Models. *Journal of Quantitative Finance and Economics*, 3(1), 49-78.
- [WGC] World Gold Council. (2023). Gold spot price and cost of gold. Gold spot prices. <https://www.gold.org/goldhub/data/gold-prices>

- World Bank. (2013). Gold futures end of day settlement price. IndexMundi. <https://www.indexmundi.com/commodities/?commodity=gold>
- Federal Reserve Bank of St. Louis. (1991). *Welcome to Fred, your trusted source for economic data since 1991*. FRED. <https://fred.stlouisfed.org/>
- Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M. Z., Barrow, D. K., Ben Taieb, S., Bergmeir, C., Bessa, R. J., Bijak, J., Boylan, J. E., Browell, J., Carnevale, C., Castle, J. L., Cirillo, P., Clements, M. P., Cordeiro, C., Oliveira, F. L. C., De Baets, S., Dokumentov, A., Ellison, J., Fiszeder, P., Franses, P. H., Frazier, D. T., Gilliland, M., Gönül, M. S., Goodwin, P., Grossi, L., Grushka-Cockayne, Y., Guidolin, M., Guidolin, M., Gunter, U., Guo, X., Guseo, R., Harvey, N., Hendry, D. F., Hollyman, R., Januschowski, T., Jeon, J., Jose, V. R. R., Kang, Y., Koehler, A. B., Kolassa, S., Kourentzes, N., Leva, S., Li, F., Litsiou, K., Makridakis, S., Martin, G. M., Martinez, A. B., Meeran, S., Modis, T., Nikolopoulos, K., Önköl Dilek, Paccagnini A, Panagiotelis A, Panapakidis I, Pavía J.M, Pedio M, Pedregal D.J, Pinson P, Ramos P, Rapach D.E, Reade J.J, Rostami-Tabar B, Rubaszek M, Sermpinis G, Shang H.L, Spiliotis E, Syntetos A.A, Talagala P.D, Talagala T.S, Tashman L, Thomakos D, Thorarindottir T, Todini E, Trapero Arenas J.R, Wang X, Winkler R.L, Yusupova A, Ziel F., (2022). Forecasting: theory and practice. *International Journal of Forecasting* 38(3), 705-871. <https://doi.org/10.1016/j.ijforecast.2021.11.001>
- Hyndman, R. J., dan Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4), 679–688. <https://doi.org/10.1016/j.ijforecast.2006.03.001>
- Nawaz, S. A., Li, J., Bhatti, U. A., Bazai, S. U., Zafar, A., Bhatti, M. A., Mehmood, A., Ain, Q. ul, danamp; Shoukat, M. U. (2021). A hybrid approach to forecast the COVID-19 epidemic trend. *PLOS ONE*. <https://doi.org/10.1371/journal.pone.0256971>
- Yusnaini, Y. (2023). The effect of inflation rate and foreign exchange on company value at Bank Mandiri Tbk. *RIGGS: Journal of Artificial Intelligence and Digital Business*, 1(2), 45–50. <https://doi.org/10.31004/riggs.v1i2.58>
- Kingphai, K., Moshfeghi, Y. (2023). On Time Series Cross-Validation for Deep Learning Classification Model of Mental Workload Levels Based on EEG Signals. In: Nicosia, G., et al. *Machine Learning, Optimization, and Data Science. LOD 2022. Lecture Notes in Computer Science*, vol 13811. Springer, Cham. [https://doi.org/10.1007/978-3-031-25891-6\\_30](https://doi.org/10.1007/978-3-031-25891-6_30)
- Dhake, H., Kashyap, Y., dan Kosmopoulos, P. (2023). Algorithms for hyperparameter tuning of LSTMs for time series forecasting. *Remote Sensing*, 15(8), 2076. <https://doi.org/10.3390/rs15082076>
- Chen, C., Twycross, J., dan Garibaldi, J. M. (2017). A new accuracy measure based on bounded relative error for time series forecasting. In Z.-K. Gao (Ed.),

- PLOS ONE (Vol. 12, Issue 3, p. e0174202). Public Library of Science (PLOS). <https://doi.org/10.1371/journal.pone.0174202>
- Shojaie, A., dan Fox, E. B. (2022). Granger causality: A review and recent advances. *Annual Review of Statistics and Its Application*, 9(1), 289-319. <https://doi.org/10.1146/annurev-statistics-040120-010930>
- Löning, M., Bagnall, A., Ganesh, S., Kazakov, V., Lines, J., & Király, F. J. (2019). sktime: A Unified Interface for Machine Learning with Time Series (Version 1). arXiv. <https://doi.org/10.48550/ARXIV.1909.07872>
- Cowles, M., dan Davis, C. (1982). On the origins of the .05 level of statistical significance. *American Psychologist*, 37(5), 553–558. <https://doi.org/10.1037/0003-066X.37.5.553>
- Scott, D.W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons.
- Cohen, D. J., & Cohen, J. (2006). The Sectioned Density Plot. *The American Statistician*, 60(2), 167–174. <http://www.jstor.org/stable/27643757>
- Aydin, D. (2008). A Comparison of the Sum of Squares in Linear and Partial Linear Regression Models. *International Journal of Mathematical and Computational Sciences*, 2(6), 342-348
- Bank Indonesia. (2023a). Kurs Transaksi BI. <https://www.bi.go.id/id/statistik/informasi-kurs/transaksi-bi/default.aspx>
- Bank Indonesia. (2023b). Data Inflasi. <https://www.bi.go.id/id/statistik/indikator/data-inflasi.aspx>
- Bank Indonesia. (2023c). BI 7-day (Reverse) Repo Rate. <https://www.bi.go.id/id/statistik/indikator/data-inflasi.aspx>
- de Barbosa Ferreira, N. M. (2021). Time series vs regression, which is the best approach?.
- Fordana, M. D., & Rochmawati, N. (2022). Optimisasi Hyperparameter CNN Menggunakan random search Untuk Deteksi COVID-19 Dari Citra X-ray dada. *Journal of Informatics and Computer Science (JINACS)*, 4(01), 10-18. <https://doi.org/10.26740/jinacs.v4n01.p10-18>
- Aleemi, A. R., Tariq, M., & Ahmed, S. (2016). THE ROLE OF GOLD PRICES, EXCHANGE RATE AND INTEREST RATE ON THE CONTINUITY OF INFLATION IN PAKISTAN. Institute of Business Management, Karachi, Pakistan: *Journal Management System*. <https://doi.org/10.22555/PBR.V18I1.660>
- He, D. (2023). Analysis of applications for Taylor series expansion: Evidence from machine learning, mathematics and engineering. *Proceedings of the 2023 International Conference on Mathematical Physics and Computational Simulation*, 10(1), 216-223. <https://doi.org/10.54254/2753-8818/10/20230346>



## **Riwayat Hidup**

Penulis dilahirkan di Depok pada tanggal 6 April 2001 sebagai anak kedua dari Bapak Cecep Gunawan dan Ibu Minah Mintarsih. Selama masa sekolah menengah di SMAN 2 Depok dari tahun 2016 hingga 2019, penulis menikmati perjalanan belajar dan proses tumbuh menjadi pribadi yang lebih matang. Tetapi yang paling membuat penulis bersemangat adalah ketika penulis mulai mengenal dunia teknologi, khususnya dalam bidang machine learning, data science, dan artificial intelligence.

Setelah meraih kelulusan, penulis dengan penuh antusiasme melanjutkan pendidikan ke tingkat perguruan tinggi. Tahun 2019 menjadi momen berarti dalam hidup penulis karena berhasil diterima di salah satu universitas impian, yaitu Institut Pertanian Bogor (IPB). Memilih Fakultas Matematika dan Ilmu Pengetahuan Alam adalah keputusan yang tepat karena penulis memiliki minat yang besar dalam dunia ilmu pengetahuan, terutama yang berkaitan dengan machine learning, data science, dan artificial intelligence.

Selama menempuh studi di kampus, penulis tidak hanya fokus pada pembelajaran, tetapi juga aktif dalam berbagai kegiatan di organisasi IEEE IPB. Pengalaman berharga ini membantu penulis dalam mengembangkan keterampilan sosial dan kepercayaan diri, yang menjadi pondasi penting dalam karir masa depan sebagai seorang profesional di bidang teknologi yang penulis geluti.

Salah satu pencapaian paling mengesankan bagi penulis adalah menjadi finalis dalam lomba gemastik data mining. Rasanya luar biasa melihat usaha keras yang telah dilakukan berbuah manis, dan itu semakin memotivasi penulis untuk terus berusaha dan mengembangkan kemampuan analitis yang dimiliki pada bidang *machine learning*, *data science*, dan *artificial intelligence*.

Seluruh perjalanan hidup, baik di sekolah maupun di perguruan tinggi, telah membentuk penulis menjadi individu yang bersemangat dan siap menghadapi tantangan baru dalam dunia teknologi. Dengan passion yang membara di bidang machine learning, data science, dan artificial intelligence, penulis berkomitmen untuk memberikan kontribusi positif dan inovatif dalam dunia akademis dan masyarakat, serta terus mengeksplorasi potensi diri untuk meraih masa depan yang lebih cerah dalam industri yang sangat penulis cintai.