ISSN: xxxx-xxxx
Prosiding SAINTEK: Sains dan Teknologi Vol.1 No.1 Tahun 2022
Call for papers dan Seminar Nasional Sains dan Teknologi Ke-1 2022
Fakultas Teknik, Universitas Pelita Bangsa, Juli 2022

Analisis dan perbandingan algoritma data maining dalam Prediksi Harga Saham PT Telekomunikasi Indonesia Tbk

Analysis and comparison of data mining algorithms in PT Telekomunikasi Indonesia Tbk Share Price Predictions

Taufik Hidayat
Institusi Teknik Informatika, Fakultas Teknik, Universitas Pelita Bangsa
taufik.hidayat100700@gmail.com

Abstract

This research aims to analyze and compare the achievements of several data mining algorithms in predicting the share value of PT Telekomunikasi Indonesia Tbk. This research examines the effectiveness of data mining algorithms such as Linear Regression, Support Vector Machines (SVM), Decision Trees, and Random Forests in the context of stock price prediction. In this research, data preprocessing methods are used to prepare the dataset. The test results show that Linear Regression has an accuracy rate of 98.16%, with RMSE and MAE values close to zero. SVM achieved an accuracy of 80.70%, while Decision Tree and Random Forest had an accuracy of 87.67% and 93.84%, respectively. Even if the results are positive, keep in mind that unexpected events in the stock market can still result in potential errors in predictions.

Keywords: *Linear Regression Algorithm, SVM, Decision Tree, Random Forest, Stock Price Prediction, Historical Data, TLKM*

Abstrak

Penelitian ini dimaksudkan untuk menganalisis serta membandingkan prestasi beberapa algoritma data mining dalam meramalkan nilai saham PT Telekomunikasi Indonesia Tbk. Riset ini mengevaluasi efektivitas algoritma data mining seperti Regresi Linear, Mesin Vector Pendukung (SVM), Pohon Keputusan, dan Hutan Acak dalam konteks prediksi harga saham. Pada penelitian ini, metode preprocessing data digunakan untuk menyiapkan dataset. Hasil pengujian menunjukkan bahwa Regresi Linear memiliki tingkat akurasi sebesar 98.16%, dengan nilai RMSE dan MAE mendekati nol. SVM mencapai akurasi sebesar 80.70%, sementara Pohon Keputusan dan Hutan Acak masing-masing memiliki akurasi sebesar 87.67% dan 93.84%. Meskipun hasilnya positif, perlu diingat bahwa fluktuasi tak terduga di pasar saham masih dapat mengakibatkan potensi kesalahan dalam prediksi.

Kata kunci: Algoritma Linear Regression, SVM, Decisio Tree, Random Forest, Prediksi Harga Saham, Data Historis, TLKM

Pendahuluan

Dalam zaman sekarang, investasi di pasar modal di setiap negara menjadi aset yang memiliki signifikansi tinggi bagi semua perusahaan global. Hal ini disebabkan oleh kemampuan investor dari berbagai penjuru dunia untuk memberikan pengaruh, baik secara langsung maupun tidak langsung, terhadap kondisi ekonomi di negara tempat mereka melakukan investasi. Saham, yang merupakan instrumen keuangan yang dikeluarkan oleh perusahaan dalam bentuk Perseroan Terbatas (PT) atau yang biasa disebut emitmen,

menjadi salah satu bentuk investasi yang sangat penting. Saham mencerminkan kepemilikan sebagian dari perusahaan oleh pemegang saham tersebut[1].

Dalam upaya melaksanakan pembangunan ekonomi nasional suatu negara, sumber pembiayaan menjadi hal yang krusial, baik dari pemerintah maupun masyarakat. Pasar modal muncul sebagai salah satu opsi pendanaan yang menjadi alternatif bagi kedua sektor tersebut. Pemerintah yang memerlukan dana dapat menerbitkan obligasi atau surat utang, kemudian menjualnya ke masyarakat melalui pasar modal. Hal serupa juga berlaku untuk sektor swasta, khususnya perusahaan, yang dapat menerbitkan efek, baik dalam bentuk saham maupun obligasi, dan menjualnya kepada masyarakat melalui pasar modal [2]

Bagi para investor yang ingin melakukan transaksi saham di pasar modal, sangat penting untuk melakukan penelitian dan teliti dalam pengambilan keputusan, baik itu untuk pembelian atau penjualan saham, serta dalam melindungi investasinya (Nurlina, 2017). Data keuangan PT. Telekomunikasi dapat diakses melalui www.idx.com dan situs http://www.indotelko.com, yang diukur oleh Indonesia Stock Exchange (IDX). Harga saham dari tahun 2017 hingga tahun 2021 mengalami fluktuasi setiap bulannya. Meskipun terdapat periode stabil, namun investor perlu memahami faktor-faktor yang dapat mempengaruhi pergerakan nilai saham sebelum membuat keputusan untuk membeli atau menjual (Agustina & Sumartio, 2014). Sayangnya, sebagian investor baru terkadang terlalu tergesa-gesa dalam mengambil keputusan jual-beli saham tanpa mempertimbangkan faktor ekonomi yang relevan. Sebagai solusi, perlu dilakukan upaya edukasi kepada masyarakat agar calon investor tidak hanya mengikuti perkembangan zaman tetapi juga memahami dengan baik kondisi perusahaan sebelum mengambil keputusan investasi[3].

Penelitian lain yang dilaksanakan oleh L. Septiningrum, H. Yasin, S. Sugito (2015) memiliki tujuan untuk melakukan prediksi terhadap harga saham gabungan dengan menggunakan metode Support Vector Regression (SVR) melalui algoritma Grid Search. Hasil dari penelitian ini mengindikasikan bahwa SVR yang menggunakan fungsi kernel linier mampu memberikan tingkat akurasi yang sangat baik dalam melakukan prediksi terhadap Indeks Harga Saham Gabungan (JCI). Hasil tersebut mencakup R2 sebesar 98,4% dan MAPE sebesar 0,873% pada data latih, sedangkan pada data uji, diperoleh R2 sebesar 90,9% dengan MAPE sebesar 0,613%[4].

Dalam penelitian yang dilakukan oleh Maulana & Kumalasari pada tahun 2019, dilakukan analisis perbandingan kinerja model algoritma data mining untuk memprediksi harga saham GGRM. Model yang dibandingkan melibatkan Linear Regression, Neural Network, SVM, Gaussian Process, dan Polynomial Regression dengan menggunakan lima atribut, yaitu tanggal (date), harga pembukaan (open), harga tertinggi (high), harga terendah (low), dan harga penutupan (close).Hasil penelitian menyimpulkan bahwa model algoritma Neural Network memberikan kinerja terbaik dalam memprediksi harga saham GGRM. Dengan akurasi dan nilai Root Mean Squared Error (RMSE) sebesar 612.474 +/- 89.402 (mikro: 618.916 +/- 0.000), Neural Network menunjukkan performa yang lebih baik dibandingkan dengan model algoritma lainnya. Temuan ini memiliki implikasi positif dalam membantu prediksi harga saham GGRM di pasar modal[5].

Random Forest merupakan salah satu teknik dalam machine learning yang dianggap efektif dan telah diterapkan oleh para peneliti dalam beberapa penelitian. Penelitian awal ini dilakukan oleh Phase Tejas dan Patil Suhas pada tahun 2020, yang fokus pada permasalahan yang dihadapi oleh beberapa program sosial dalam pendistribusian dana bantuan kepada penerima yang tepat. Penelitian tersebut melibatkan 9,557 data dengan 143 fitur.Hasil dari penelitian ini menunjukkan akurasi sebesar 89.97%. Temuan ini mengindikasikan bahwa Random Forest mampu menyelesaikan permasalahan tersebut secara efektif, memberikan kontribusi positif dalam mengoptimalkan distribusi dana bantuan kepada pihak yang membutuhkan[6].

Sachin Kamley, Shailesh Jaloree, dan R. S. Thakur menyusun ulasan penelitian terkait kinerja dalam melakukan prediksi pasar saham menggunakan metode machine learning. Keunggulan dari Decision Tree disebabkan oleh kemudahan dan kapasitasnya dalam mengidentifikasi contoh data yang memiliki nilai signifikan, baik besar maupun kecil, serta melakukan prediksi terhadap nilai-nilainya. Meskipun memiliki

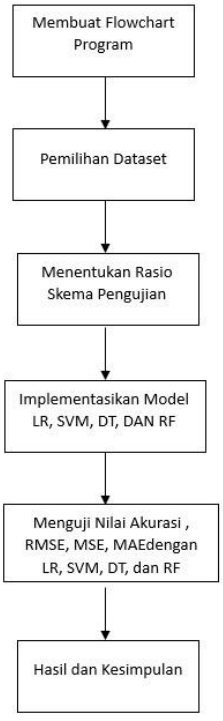
kekurangan berupa akurasi prediksi yang relatif rendah, hasil penelitian menunjukkan bahwa teknik Decision Tree mampu melakukan prediksi dengan tingkat akurasi yang lebih tinggi dibandingkan dengan model lain. Hal ini disebabkan oleh kemampuannya dalam melakukan prediksi jangka pendek dan signifikansi variabel yang digunakan dalam proses prediksi[7].

Pasar saham dikenal karena fluktuasinya yang tinggi dan dipengaruhi oleh berbagai faktor eksternal yang kompleks. Oleh karena itu, penting untuk melakukan analisis yang cermat menggunakan algoritma data mining guna meningkatkan pemahaman dan meminimalkan ketidakpastian terkait pergerakan harga saham.

Mengidentifikasi dan menganalisis faktor-faktor yang berpengaruh terhadap pergerakan harga saham PT Telekomunikasi Indonesia Tbk. Tujuan ini bertujuan untuk mendapatkan wawasan yang lebih baik mengenai variabilitas pasar saham dan potensi faktor-faktor yang dapat mempengaruhi perubahan harga saham.Melakukan evaluasi kinerja beberapa algoritma data mining yang berbeda dalam konteks prediksi harga saham. Hal ini bertujuan untuk menentukan algoritma yang paling efektif dan akurat dalam meramalkan pergerakan harga saham PT Telekomunikasi Indonesia Tbk.Membandingkan hasil prediksi yang diperoleh dari berbagai algoritma data mining, seperti Linear Regression, Decision Trees, Random Forest, SVM, atau algoritma lainnya. Tujuan ini memberikan pemahaman yang lebih baik mengenai kelebihan dan kekurangan masing-masing algoritma dalam konteks aplikasi prediksi harga saham.

METODOLOGI

Proses ini mengggunakan metode penelitian yang dijelaskan dalam Gambar 1. Pembuatan flowchart program bertujuan untuk memahami alur kerja program secara visual. Langkah berikutnya melibatkan pemilihan dataset sebagai input yang akan diprediksi oleh model regresi LR, SVM, DT, dan RF. Setelah itu, perancangan rasio skema pengujian ditentukan untuk proses pelatihan dan pengujian model. Implementasi model LR, SVM, DT, dan RF pada dataset merupakan langkah selanjutnya.Selanjutnya, tahap uji kesalahan dilakukan untuk mengevaluasi hasil implementasi metode LR, SVM, DT, dan RF. Pengujian niali akurasi dengan metode LR, SVM, DT, dan RF dari hasil implementasi tersebut digunakan untuk menilai tingkat akurasi prediksi. Tahap terakhir mencakup penyusunan hasil dan pembuatan kesimpulan berdasarkan temuan dari penelitian yang dilaksanakan.



Gambar 1. Metode Penelitian

Data set

Penulis menggunakan dataset harga saham dari PT. Telekomunikasi Indonesia Tbk dengan kode saham TLKM dari tanggal 1 Januari 2019 hingga tanggal 28 Desember 2023 sebanyak 1232 data, yang penulis peroleh dari https://finance.yahoo.com/quote/TLKM.JK/history?p=TLKM.JK. Yahoo Finance merupakan salah satu situs yang menyajikan data harga historis saham dalam kurun waktu yang cukup panjang. Berdasarkan dataset harga saham dari PT. Telekomunikasi Indonesia Tbk terdapat 3 atribut yang terdiri dari open, high, low sedangkan sebagai labelnya adalah close. Semua atribut tersebut selain label merupakan hal- hal yang mempengaruhi penutupan harga saham atau close.

Selanjutnya, data yang diperoleh akan disesuaikan menggunakan teknik preprocessing data. Sebelumnya, dilakukan analisis terhadap dataset yang menunjukkan adanya dua jenis tipe data, yaitu float dan integer. Atribut Volume, yang merupakan jumlah saham yang terjual, merupakan tipe data integer. Oleh karena itu, pada dataset yang digunakan, tidak perlu dilakukan proses konversi tipe data. Selain itu, dilakukan pembersihan data untuk beberapa nilai atribut yang tidak lengkap, dengan tujuan mengurangi potensi kesalahan dalam proses klasifikasi data. Jika terdapat nilai nol pada atribut Volume, data atau baris tersebut tidak dihapus, mengingat bahwa nilai nol menunjukkan tidak adanya penjualan saham pada hari tersebut.

Persiapan Data

Pada fase ini, setelah data dianggap telah memadai setelah proses preprocessing, langkah selanjutnya adalah menentukan data training dan testing. Penentuan ini menggunakan teknik Split Validation, di mana dataset dibagi menjadi dua bagian, dengan 20% data untuk pengujian dan 80% untuk pelatihan. Hasil dari pembagian data tersebut akan diuji menggunakan metode yang telah ditetapkan, yaitu LINEAR REGRESSION, SVM, DECISION TREES, dan RANDOM FORESTS. dan hasilnya akan dibandingkan dengan setiap metode untuk menilai sejauh mana metode yang memiliki akurasi yang baik dalam melakukan prediksi harga saham.

Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) merupakan hasil perhitungan akar kuadrat dari rata-rata kuadrat selisih antara data aktual dan data prediksi, yang kemudian dibagi dengan jumlah data. Rumus RMSE dapat dinyatakan dalam persamaan, di mana "Aktual" merujuk kepada data sebenarnya, "prediksi" adalah nilai prediksi dari variabel Aktual[8] , dan "n" menunjukkan jumlah observasi:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

Gambar 2. RMSE

Mean Squared Error (MSE)

Mean Squared Error (MSE) adalah nilai rata-rata dari kesalahan kuadrat antara nilai data aktual dan hasil prediksi. Mean Square Error didasarkan pada perbedaan nilai antara data aktual dan data prediksi, di mana nilai-nilai tersebut dikuadratkan, dijumlahkan secara keseluruhan, dan kemudian dibagi dengan jumlah data yang tersedia. Ilustrasi perhitungan MSE dapat ditemukan dalam gambar di bawah ini, dengan "Aktual" yang

merujuk pada data sebenarnya, "prediksi" adalah nilai prediksi dari variabel Aktual [8], dan "n" menunjukkan jumlah observasi:

$$\text{MSE} = \underbrace{\frac{1}{n}\sum_{i=1}^{n}}_{\text{Mean}}\underbrace{(Y_i - \hat{Y}_i)^2}_{\text{Error}\quad\text{Squared}}$$

Gambar 3. MSE

Dimana :

- N adalah jumlah sampel atau observasi
- Y i adalah nilai sebenarnya dari sampel i
- Y i adalah nilai prediksi dari sampel i

Mean Absolute Error (MAE)

Pengukuran kinerja model menggunakan Mean Absolute Error (MAE) bertujuan untuk mengevaluasi sejauh mana perbedaan absolut antara nilai prediksi dan nilai aktual. MAE dihitung dengan menambahkan selisih absolut antara setiap nilai prediksi dan nilai sebenarnya, kemudian diambil rata-rata dari nilai-nilai tersebut. Hasil ini mencerminkan tingkat kesalahan dari model asosiasi. Semakin mendekati nilai nol, menandakan bahwa model tersebut semakin baik dalam memprediksi[8] :

$$MAE = \frac{1}{N}\sum_{i=1}^{i=N}|y\_true_i - y\_pred_i|$$

$$MAE = \frac{1}{N}\sum_{i=1}^{i=N}|y_i - \widehat{y_i}|$$

$$\Longrightarrow MAE = \frac{1}{3}[|y_1 - \widehat{y_1}| + |y_2 - \widehat{y_2}| + |y_3 - \widehat{y_3}|]$$

Gambar 4. MAE

Algoritma prediksi

Dalam rangka penelitian ini, peneliti memanfaatkan beberapa algoritma yang akan diadu, termasuk

Linear Regression (LR)

Algoritma statistik yang dikenal sebagai regresi linear digunakan untuk menentukan pengaruh dari satu atau beberapa variabel pada suatu variabel tunggal. Variabel yang nilainya mengalami perubahan disebut sebagai variabel, dan variabel-variabel yang memiliki dampak dianggap sebagai variabel independen atau variabel bebas[9].

Support Vector Machine (SVM)

Support Vector Machine (SVM) merupakan suatu teknik klasifikasi yang terdiri dari dua kategori, yaitu Support Vector Machine classification dan Support Vector Machine Regression. SVM diperkenalkan pertama kali oleh Vapnik pada tahun 1992 sebagai konsep unggulan dalam bidang pengenalan pola. Algoritma ini mampu memilih model secara otomatis dan tidak mengalami masalah overfitting. Sebuah penelitian lain yang dilakukan oleh Kyoung-jae juga menunjukkan bahwa metode SVM sangat efektif untuk prediksi karena mampu mengurangi kesalahan klasifikasi dan penyimpangan data pada data pelatihan.

SVM beroperasi berdasarkan prinsip dasar dari pemisahan linier untuk kasus klasifikasi. Meskipun demikian, SVM telah mengalami pengembangan agar dapat menangani permasalahan non-linier dengan memperkenalkan konsep kernel dalam ruang kerja yang memiliki dimensi tinggi. Dalam ruang berdimensi tinggi ini, SVM bertujuan untuk menemukan hyperplane yang dapat memaksimalkan jarak antara kelas data[10].

Decision Trees (DT)

Pohon adalah struktur data yang terdiri dari node dan edge. Terdapat tiga jenis node dalam pohon, yaitu simpul akar (root/node), simpul percabangan/internal (branch/simpul internal), dan simpul daun (leaf/node). Pohon keputusan merupakan metode simplifikasi untuk klasifikasi dengan jumlah kelas yang terbatas. Pada pohon keputusan, simpul akar dan simpul internal diberi label sebagai atribut, edge diberi label sebagai nilai atribut yang mungkin, dan simpul daun diberi label sebagai kelas yang berbeda. Pohon keputusan merupakan teknik pembelajaran mesin yang menggunakan hierarki aturan klasifikasi dengan mempartisi dataset pelatihan secara rekursif. Pohon keputusan memiliki struktur diagram alur berbentuk pohon, dimana setiap simpul internal mewakili pengujian suatu atribut, cabangnya menunjukkan hasil tes, dan simpul daun mencerminkan distribusi kelas[11].

Random Forests (RF)

Random Forest adalah sebuah ensemble dari decision trees yang dibangun dengan menggunakan sampel yang dipilih secara acak, namun tetap mengikuti aturan pembagian simpul yang berbeda. Model ini bekerja dengan menggunakan subset dari fitur pada setiap pohon, kemudian mencoba menemukan ambang batas terbaik untuk memisahkan data. Akibatnya, model ini menghasilkan banyak pohon yang dilatih secara lemah, dan setiap pohon memberikan prediksi yang berbeda. Interpretasi hasil dapat dilakukan dengan dua cara, yang paling umum adalah berdasarkan mayoritas suara, di mana kelas yang paling banyak dianggap sebagai prediksi akhir. Namun, dalam implementasi scikit-learn, algoritma ini menghasilkan prediksi berdasarkan rata-rata dari hasil pohon-pohon tersebut, sehingga menghasilkan prediksi yang sangat akurat. Meskipun secara teoritis berbeda, rata-rata probabilitas dari Random Forest yang terlatih tidak akan sangat berbeda dari sebagian besar prediksi. Oleh karena itu, kedua metode ini sering kali menghasilkan hasil yang sebanding. Pada model Random Forest di scikit-learn, terdapat beberapa parameter yang dapat diatur, seperti jumlah pohon yang ingin dibangun oleh model (n_estimators)[12].

Model Evaluasi

Pengujian dilakukan untuk memverifikasi bahwa model mampu memberikan prediksi yang akurat dan dapat diandalkan dalam menghadapi data uji. Dalam rangka pengujian ini, beberapa metrik evaluasi seperti Mean Squared Error (MSE), Root Mean Squared Error (RMSE), dan Mean Absolute Error (MAE) digunakan untuk mengukur sejauh mana model mampu memprediksi data dengan keakuratan. Langkah-langkahnya mencakup :

1. mengimpor mean_squared_error dan mean_absolute_error dari perpustakaan scikit-learn,
2. membentuk model evaluasi untuk MSE, RMSE, dan MAE
3. menampilkan hasilnya

HASIL DAN PEMBAHASAN

Hasil pengujian Model Algoritma Linear Regression

```
!pip install pyspark
```

```
Collecting pyspark
  Downloading pyspark-3.5.0.tar.gz (316.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 316.9/316.9 MB 3.5 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.10/dist-packages (from pyspark) (0.10.9.7)
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.5.0-py2.py3-none-any.whl size=317425345 sha256=4972c27d3a3af2912bc1eaa3a96355597d935cf7ce83020f26be30c
  Stored in directory: /root/.cache/pip/wheels/41/4e/10/c2cf2467f71c678cfc8a6b9ac9241e5e44a01940da8fbb17fc
Successfully built pyspark
Installing collected packages: pyspark
Successfully installed pyspark-3.5.0
```

```
!pip install findspark
```

```
Collecting findspark
  Downloading findspark-2.0.1-py2.py3-none-any.whl (4.4 kB)
Installing collected packages: findspark
Successfully installed findspark-2.0.1
```

Prediksi saham algoritma Linear Reggresion

```
[ ] from pyspark.sql import SparkSession
    from pyspark.ml.feature import VectorAssembler
    from pyspark.ml.regression import LinearRegression
    from pyspark.ml import Pipeline
    from pyspark.sql.types import StructType, StructField, DateType, FloatType, IntegerType
    from pyspark.sql.functions import col
```

```
# Inisialisasi sesi Spark
spark = SparkSession.builder.appName("SahamPrediction").getOrCreate()
```

```
[ ] # Membaca data dari file CSV atau sumber data lainnya
    df = spark.read.csv("TLKM.JK.csv", header=True, inferSchema=True)
```

```
# Skema DataFrame
schema = StructType([
    StructField("Date", DateType(), True),
    StructField("Open", FloatType(), True),
    StructField("High", FloatType(), True),
    StructField("Low", FloatType(), True),
    StructField("Close", FloatType(), True),
    StructField("Adj_Close", FloatType(), True),
    StructField("Volume", IntegerType(), True),
])
```

```
[ ] # Menampilkan skema data
    df.printSchema()
```

```
root
 |-- Date: date (nullable = true)
 |-- Open: string (nullable = true)
 |-- High: string (nullable = true)
 |-- Low: string (nullable = true)
 |-- Close: string (nullable = true)
 |-- Adj Close: string (nullable = true)
 |-- Volume: string (nullable = true)
```

```
# Skema DataFrame
schema = StructType([
    StructField("Date", DateType(), True),
    StructField("Open", FloatType(), True),
    StructField("High", FloatType(), True),
    StructField("Low", FloatType(), True),
    StructField("Close", FloatType(), True),
    StructField("Adj_Close", FloatType(), True),
    StructField("Volume", IntegerType(), True),
])
```

```
# Menampilkan skema data
df.printSchema()
```

```
root
 |-- Date: date (nullable = true)
 |-- Open: string (nullable = true)
 |-- High: string (nullable = true)
 |-- Low: string (nullable = true)
 |-- Close: string (nullable = true)
 |-- Adj Close: string (nullable = true)
 |-- Volume: string (nullable = true)
```

```
# Menampilkan DataFrame setelah transformasi
data.show()
```

```
+----------+------+------+------+------+---------+---------+--------------------+
|      Date|  Open|  High|   Low| Close|Adj Close|   Volume|            features|
+----------+------+------+------+------+---------+---------+--------------------+
|2019-01-01|3750.0|3750.0|3750.0|3750.0|3102.0913|        0|[3750.0,3750.0,37...|
|2019-01-02|3750.0|3760.0|3700.0|3730.0|3085.5466| 31355300|[3750.0,3760.0,37...|
|2019-01-03|3710.0|3770.0|3690.0|3740.0| 3093.819| 83842400|[3710.0,3770.0,36...|
|2019-01-04|3690.0|3740.0|3690.0|3710.0|3069.0022| 73936900|[3690.0,3740.0,36...|
|2019-01-07|3760.0|3790.0|3750.0|3770.0|3118.6357| 83678100|[3760.0,3790.0,37...|
|2019-01-08|3770.0|3800.0|3750.0|3800.0|3143.4524| 67963700|[3770.0,3800.0,37...|
|2019-01-09|3820.0|3830.0|3730.0|3730.0|3085.5466| 98529400|[3820.0,3830.0,37...|
|2019-01-10|3760.0|3800.0|3740.0|3800.0|3143.4524|126396700|[3760.0,3800.0,37...|
|2019-01-11|3820.0|3860.0|3800.0|3860.0| 3193.086|116753700|[3820.0,3860.0,38...|
|2019-01-14|3810.0|3850.0|3790.0|3850.0|3184.8137| 68487000|[3810.0,3850.0,37...|
|2019-01-15|3860.0|3930.0|3850.0|3930.0|3250.9915| 83464900|[3860.0,3930.0,38...|
|2019-01-16|3940.0|4000.0|3920.0|3990.0| 3300.625|139056700|[3940.0,4000.0,39...|
|2019-01-17|3970.0|3990.0|3960.0|3990.0| 3300.625| 78766100|[3970.0,3990.0,39...|
|2019-01-18|4010.0|4020.0|3970.0|4020.0| 3325.442| 76350400|[4010.0,4020.0,39...|
|2019-01-21|4040.0|4050.0|4020.0|4030.0|3333.7139| 83317600|[4040.0,4050.0,40...|
|2019-01-22|4000.0|4010.0|3930.0|4000.0|3308.8972| 95420500|[4000.0,4010.0,39...|
|2019-01-23|3950.0|3980.0|3920.0|3920.0|3242.7195| 72518500|[3950.0,3980.0,39...|
|2019-01-24|3900.0|3910.0|3830.0|3860.0| 3193.086|118836500|[3900.0,3910.0,38...|
|2019-01-25|3880.0|3930.0|3840.0|3880.0|3209.6304| 75844000|[3880.0,3930.0,38...|
|2019-01-28|3880.0|3890.0|3730.0|3780.0| 3126.908|168983100|[3880.0,3890.0,37...|
```

```
# Membuat kolom label
data = data.withColumnRenamed("Adj Close", "label")
```

```
# Membagi data menjadi set pelatihan dan pengujian
train_data, test_data = data.randomSplit([0.8, 0.2], seed=123)
```

```
# Membuat model Linear Regression
lr = LinearRegression(featuresCol="features", labelCol="label", maxIter=10, regParam=0.1)
```

```
# Membuat pipeline untuk melatih model
pipeline = Pipeline(stages=[lr])
```

```
# Melatih model
model = pipeline.fit(train_data)
```

```
# Menguji model
predictions = model.transform(test_data)
```

```
[ ] # Menampilkan hasil prediksi
    predictions.select("Date", "features", "label", "prediction").show()
```

```
+----------+--------------------+---------+------------------+
|      Date|            features|    label|        prediction|
+----------+--------------------+---------+------------------+
|2019-01-03|[3710.0,3770.0,36...| 3093.819|  3094.14571803847|
|2019-01-09|[3820.0,3830.0,37...|3085.5466|3085.8717856414246|
|2019-01-17|[3970.0,3990.0,39...| 3300.625|3301.0243826017804|
|2019-01-18|[4010.0,4020.0,39...| 3325.442| 3325.831143491819|
|2019-01-25|[3880.0,3930.0,38...|3209.6304|3209.9784347320574|
|2019-02-01|[3910.0,3930.0,38...| 3201.358| 3201.696877340695|
|2019-02-11|[3870.0,3940.0,38...|3250.9915|3251.3552185405824|
|2019-02-12|[3900.0,3910.0,38...| 3159.997|3160.3289508290054|
|2019-02-15|[3770.0,3800.0,37...|3135.1802|3135.5100867789733|
|2019-02-18|[3830.0,3920.0,38...| 3226.175|3226.5271035598516|
|2019-02-19|[3930.0,3940.0,39...|3242.7195|3243.1023335536593|
|2019-02-22|[3870.0,3870.0,38...|3176.5415| 3176.893999597038|
|2019-02-26|[3920.0,3930.0,38...|3250.9915|3251.3524943444545|
|2019-03-07|[3820.0,3820.0,38...| 3159.997| 3160.351350072239|
|2019-03-19|[3800.0,3820.0,37...| 3126.908|3127.2422910676214|
|2019-03-27|[3840.0,3840.0,38...| 3159.997|3160.3477905031114|
|2019-04-18|[3920.0,3950.0,38...| 3193.086|3193.4383406972993|
|2019-04-23|[3780.0,3870.0,37...|3184.8137| 3185.122673895841|
|2019-04-30|[3890.0,3890.0,37...|3135.1802|3135.5006999970515|
```

```
[ ] # Menguji model
    predictions = model.transform(test_data)
```

```
    # Menampilkan hasil prediksi
    predictions.select("Date", "features", "label", "prediction").show()
```

```
+----------+--------------------+---------+------------------+
|      Date|            features|    label|        prediction|
+----------+--------------------+---------+------------------+
|2019-01-03|[3710.0,3770.0,36...| 3093.819|  3094.14571803847|
|2019-01-09|[3820.0,3830.0,37...|3085.5466|3085.8717856414246|
|2019-01-17|[3970.0,3990.0,39...| 3300.625|3301.0243826017804|
|2019-01-18|[4010.0,4020.0,39...| 3325.442| 3325.831143491819|
|2019-01-25|[3880.0,3930.0,38...|3209.6304|3209.9784347320574|
|2019-02-01|[3910.0,3930.0,38...| 3201.358| 3201.696877340695|
|2019-02-11|[3870.0,3940.0,38...|3250.9915|3251.3552185405824|
|2019-02-12|[3900.0,3910.0,38...| 3159.997|3160.3289508290054|
|2019-02-15|[3770.0,3800.0,37...|3135.1802|3135.5100867789733|
|2019-02-18|[3830.0,3920.0,38...| 3226.175|3226.5271035598516|
|2019-02-19|[3930.0,3940.0,39...|3242.7195|3243.1023335536593|
|2019-02-22|[3870.0,3870.0,38...|3176.5415| 3176.893999597038|
|2019-02-26|[3920.0,3930.0,38...|3250.9915|3251.3524943444545|
|2019-03-07|[3820.0,3820.0,38...| 3159.997| 3160.351350072239|
|2019-03-19|[3800.0,3820.0,37...| 3126.908|3127.2422910676214|
```

```
# Evaluasi model
evaluator = RegressionEvaluator(labelCol="label",
predictionCol="prediction")
accuracy = evaluator.evaluate(predictions)
print("Model Accuracy: {:.2%}".format(accuracy))
Model Accuracy: 98.16%
```

Hasil Pengujian model Algoritma Support Vector Mechine

```
from pyspark.sql import SparkSession
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.classification import LinearSVC
from pyspark.ml import Pipeline
from pyspark.ml.evaluation import BinaryClassificationEvaluator
from pyspark.sql.types import DoubleType
from pyspark.sql.functions import lag, when
from pyspark.sql.window import Window
from pyspark.sql.functions import when
```

```
# Inisialisasi Spark session
spark = SparkSession.builder.appName("StockPrediction").getOrCreate()
```

```
# Mengganti "your_dataset.csv" dengan nama dataset yang sesuai
data = spark.read.csv("TLKM.JK.csv", header=True, inferSchema=True)
```

```
# Menampilkan skema dataset
data.printSchema()
```

```
# Menampilkan skema dataset
data.printSchema()
```

```
root
 |-- Date: date (nullable = true)
 |-- Open: string (nullable = true)
 |-- High: string (nullable = true)
 |-- Low: string (nullable = true)
 |-- Close: string (nullable = true)
 |-- Adj Close: string (nullable = true)
 |-- Volume: string (nullable = true)
```

```
# Mengonversi tipe data kolom-kolom yang dibutuhkan
data = data.withColumn("Open", data["Open"].cast(DoubleType()))
data = data.withColumn("High", data["High"].cast(DoubleType()))
data = data.withColumn("Low", data["Low"].cast(DoubleType()))
data = data.withColumn("Close", data["Close"].cast(DoubleType()))
data = data.withColumn("Volume", data["Volume"].cast(DoubleType()))
```

```
# Mengganti nilai null dengan nilai default
data = data.fillna(0)
```

```
# Menyiapkan kolom fitur
feature_cols = ["Open", "High", "Low", "Close", "Volume"]
vector_assembler = VectorAssembler(inputCols=feature_cols, outputCol="features")
data = vector_assembler.transform(data)
```

```
# Menambahkan kolom label
data = data.withColumn("label", when(data["Close"] > lag("Close").over(Window.orderBy("Date")), 1).otherwise(0))
```

```
# Memilih kolom label dan fitur
data = data.select("features", "label")
```

```
# Membagi dataset menjadi data latih dan data uji
train_data, test_data = data.randomSplit([0.8, 0.2], seed=42)
```

```
# Membuat model SVM
svm = LinearSVC(maxIter=10, regParam=0.1)
pipeline = Pipeline(stages=[svm])
```

```
[ ] # Melatih model
    model = pipeline.fit(train_data)

[ ] # Membuat prediksi pada data uji
    predictions = model.transform(test_data)

[ ] # Mengukur akurasi
    evaluator = BinaryClassificationEvaluator()
    accuracy = evaluator.evaluate(predictions)
    print(f"Accuracy: {accuracy}")

    Accuracy: 0.8070480549199085

    # Mengukur akurasi
    evaluator = BinaryClassificationEvaluator()
    accuracy = evaluator.evaluate(predictions)
    print(f"Accuracy: {accuracy}")

    Accuracy: 0.8070480549199085
```

```
# Evaluasi model
evaluator =
BinaryClassificationEvaluator(rawPredictionCol="rawPrediction")
accuracy = evaluator.evaluate(predictions)
print("Model Accuracy: {:.2%}".format(accuracy))
```

```
Model Accuracy: 80.70%
```

Hasil Pengujian Model Algoritma Decision Tree dan Random Forests

```
[ ] from pyspark.sql import SparkSession
    from pyspark.ml.feature import VectorAssembler
    from pyspark.ml.classification import DecisionTreeClassifier, RandomForestClassifier
    from pyspark.ml import Pipeline
    from pyspark.ml.evaluation import MulticlassClassificationEvaluator

    # Inisialisasi sesi Spark
    spark = SparkSession.builder.appName("StockPrediction").getOrCreate()

    + Code  + Text

[ ] # Membaca data dari CSV atau sumber lainnya
    data = spark.read.csv("TLKM.JK.csv", header=True, inferSchema=True)

[ ] # Memilih fitur yang akan digunakan untuk prediksi
    feature_cols = ["Open", "High", "Low", "Close", "Volume"]
    vector_assembler = VectorAssembler(inputCols=feature_cols, outputCol="features")
    data = vector_assembler.transform(data)

[ ] # Membuat kolom label biner
    threshold_value = data.agg({"Adj Close": "avg"}).collect()[0][0]
    data = data.withColumn("label", (data["Adj Close"] > threshold_value).cast("double"))

[ ] # Membagi data menjadi set pelatihan dan pengujian
    train_data, test_data = data.randomSplit([0.8, 0.2], seed=123)

    # Membuat model Decision Tree
    dt = DecisionTreeClassifier(featuresCol="features", labelCol="label", maxDepth=5)
    dt_pipeline = Pipeline(stages=[dt])
    dt_model = dt_pipeline.fit(train_data)

    + Code  + Text

[ ] # Membuat model Random Forest
    rf = RandomForestClassifier(featuresCol="features", labelCol="label", numTrees=10)
    rf_pipeline = Pipeline(stages=[rf])
    rf_model = rf_pipeline.fit(train_data)

[ ] # Menguji model Decision Tree
    dt_predictions = dt_model.transform(test_data)
```

```
[ ] # Menguji model Random Forest
    rf_predictions = rf_model.transform(test_data)

[ ] # Evaluasi model Decision Tree
    dt_evaluator = MulticlassClassificationEvaluator(labelCol="label", predictionCol="prediction", metricName="accuracy")
    dt_accuracy = dt_evaluator.evaluate(dt_predictions)
    print("Decision Tree Model Accuracy: {:.2%}".format(dt_accuracy))

    Decision Tree Model Accuracy: 87.67%

[ ] # Evaluasi model Random Forest
    rf_evaluator = MulticlassClassificationEvaluator(labelCol="label", predictionCol="prediction", metricName="accuracy")
    rf_accuracy = rf_evaluator.evaluate(rf_predictions)
    print("Random Forest Model Accuracy: {:.2%}".format(rf_accuracy))

    Random Forest Model Accuracy: 93.84%
```

```
# Evaluasi model Decision Tree
dt_evaluator = MulticlassClassificationEvaluator(labelCol="label",
predictionCol="prediction", metricName="accuracy")
dt_accuracy = dt_evaluator.evaluate(dt_predictions)
print("Decision Tree Model Accuracy: {:.2%}".format(dt_accuracy))
Decision Tree Model Accuracy: 87.67%
```

```
# Evaluasi model Random Forest
rf_evaluator = MulticlassClassificationEvaluator(labelCol="label",
predictionCol="prediction", metricName="accuracy")
rf_accuracy = rf_evaluator.evaluate(rf_predictions)
print("Random Forest Model Accuracy: {:.2%}".format(rf_accuracy))
Random Forest Model Accuracy: 93.84%
```

Hasil Evaluasi Model

Pengujian ini dilaksanakan dengan tujuan memverifikasi kemampuan model dalam memberikan prediksi yang tepat dan dapat dipercaya ketika dihadapkan pada data uji. Dalam rangkaian pengujian ini menggunakan beberapa metrik evaluasi seperti MeanSqauredError(MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) dan Accuracy yang digunakan untuk mengukur sejauh mana model mampu memprediksi data secara akurat.

Tabel 1. Hasil Evaluasi Model

ALGORITMA	ACCURACY	Nilai RMSE	Nilai MAE	Nilai MSE
LR	LinearRegression Model Accuracy: 98.16%	Root Mean Squared Error (LinearRegression): 7.924761663479602e-12	mae Absolute Error (LinearRegression): 6.497411102456338e-12	mse Squared Error (LinearRegression): 6.280184742295598e-23
SVM	Support Vector Machine Model	Root Mean Squared Error	MAE Absolute Error(SVM):	MSE Absolute Error (SVM):

	Model Accuracy: 80.70%	(SVM): 0.655572483329 5524	0.429775280 89887634	0.4297752808 9887634
DT	Decision Tree Model Accuracy: 87.67%	Root Mean Squared Error (Decision Tree): 30.00487755293 6833	mae Absolute Error (Decision Tree): 19.42891319 847204	mse Squared Error (Decision Tree): 768.35954698 2856
RF	Random Forest Model Accuracy: 93.84%	Root Mean Squared Error (Random Forest): 27.71929917914 333	mae Absolute Error (Random Forest): 19.42891319 847204	mse Squared Error (Random Forest): 768.3595469 82856

Kesimpulan

1. penelitain ini bertujuan untuk pengenalan dan penelitian terhadap faktor-faktor yang memengaruhi fluktuasi harga saham PT Telekomunikasi Indonesia Tbk, penilaian efektivitas beberapa algoritma data mining dalam meramalkan perubahan harga saham, perbandingan hasil prediksi yang dihasilkan oleh berbagai algoritma, termasuk Linear Regression, Decision Trees, Random Forests, dan SVM.
2. Penelitian ini bertujuan untuk menganalisis dan membandingkan kinerja beberapa algoritma data mining dalam memprediksi harga saham PT Telekomunikasi Indonesia Tbk. Beberapa algoritma yang digunakan meliputi Linear Regression, Support Vector Machine (SVM), Decision Trees, dan Random Forests
3. Hasil Evaluasi Model Linear Regression: Akurasi 98.16%, RMSE dan MAE mendekati nol.
   SVM: Akurasi 80.70%, RMSE dan MAE dengan nilai tertentu.
   Decision Trees: Akurasi 87.67%, RMSE dan MAE dengan nilai tertentu.
   Random Forests: Akurasi 93.84%, RMSE dan MAE dengan nilai tertentu. Meskipun demikian, tetap ada potensi kesalahan prediksi yang berasal dari fluktuasi pasar saham yang bersifat tidak terduga. Untuk mengatasi hal ini, diperlukan analisis fundamental yang menyeluruh dan pemahaman mendalam terhadap faktor-faktor eksternal yang dapat memengaruhi pasar, seperti kebijakan pemerintah, kondisi ekonomi global, dan peristiwa geopolitik.

Saran

1. Pemanfaatan Data dengan Cakupan yang Lebih Luas: Penelitian ini dapat diperluas dengan mempertimbangkan penggunaan dataset yang lebih luas, mencakup berbagai industri atau sektor lainnya. Langkah ini dapat memberikan pemahaman yang lebih komprehensif tentang generalisasi model dalam berbagai kondisi pasar.
2. Analisis Mendalam terhadap Faktor-Faktor Eksternal: Sebagai tahap berikutnya, penelitian dapat mendalami analisis terhadap faktor-faktor eksternal yang dijelaskan dalam pendahuluan, seperti kebijakan pemerintah, kondisi ekonomi global, dan peristiwa geopolitik. Penelitian ini dapat memberikan konteks yang lebih kaya mengenai dampak variabel-variabel tersebut terhadap pergerakan harga saham.

3. Pertimbangan Penggunaan Model Ensemble: Mengingat kinerja yang positif dari Random Forest, penelitian dapat mempertimbangkan penggunaan model ensemble yang menggabungkan berbagai algoritma untuk meningkatkan performa prediksi. Ini dapat mencakup kombinasi SVM, Decision Trees, dan model lainnya.

DAFRAT PUSTAKA

[1] Reza Maulana and Devy Kumalasari, "Analisis Dan Perbandingan Algoritma Data Mining Dalam Prediksi Harga Saham Ggrm," *J. Inform. Kaputama*, vol. 3, no. 1, pp. 22–28, 2019, [Online]. Available: https://finance.yahoo.com/quote/GGRM.J.

[2] P. C. Puspa and M. A. Ghoni, "Peranan Pasar Modal Dalam Perekonomian Negara," *Hum. FALAH J. Ekon. dan Bisnis Islam*, vol. 5, no. 2, p. 52, 2013.

[3] T. Sulastri and D. Suselo, "Pengaruh Inflasi, Suku Bunga Dan Nilai Tukar Terhadap Harga Saham PT. Telekomunikasi Indonesia Tbk.," *JPEKA J. Pendidik. Ekon. Manaj. dan Keuang.*, vol. 6, no. 1, pp. 29–40, 2022, doi: 10.26740/jpeka.v6n1.p29-40.

[4] W. R. U. Fadilah, D. Agfiannisa, and Y. Azhar, "Analisis Prediksi Harga Saham PT. Telekomunikasi Indonesia Menggunakan Metode Support Vector Machine," *Fountain Informatics J.*, vol. 5, no. 2, p. 45, 2020, doi: 10.21111/fij.v5i2.4449.

[5] E. Fitri and D. Riana, "Analisa Perbandingan Model Prediction Dalam Prediksi Harga Saham Menggunakan Metode Linear Regression, Random Forest Regression Dan Multilayer Perceptron," *METHOMIKA J. Manaj. Inform. dan Komputerisasi Akunt.*, vol. 6, no. 1, pp. 69–78, 2022, doi: 10.46880/jmika.vol6no1.pp69-78.

[6] M. E. Bastian, B. Rahayudi, and D. E. Ratnawati, "Prediksi Trend Harga Saham Jangka Pendek berdasarkan Fitur Technical Analysis dengan menggunakan Algoritma Random Forest," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 10, pp. 4536–4542, 2021, [Online]. Available: http://j-ptiik.ub.ac.id.

[7] A. Thabibi and R. Supriyanto, "Perbandingan Model Multiple Linear Regression Dan Decision Tree Regression (Studi Kasus: Prediksi Harga Saham Telkom, Indosat, Dan Xl)," *J. Ilm. Teknol. dan Rekayasa*, vol. 28, no. 1, pp. 78–92, 2023, doi: 10.35760/tr.2023.v28i1.6081.

[8] M. I. Baihaqi, A. Syaripudin, and F. A. Nugroho, "Implementation Of The Random Forest Algorithm In Stock Price Predictions Based On Historical Data Implementasi Algoritma Random Forest Pada Prediksi Harga Saham Berdasarkan Data Historis," vol. 1, pp. 42–51, 2023.

[9] L. Alpianto, A. Hermawan, and Junaedi, "Moving Average untuk Prediksi Harga Saham dengan Linear Regression," *J. Buana Inform.*, vol. 14, no. 02, pp. 117–126, 2023, doi: 10.24002/jbi.v14i02.7446.

[10] P. A. Octaviani, Yuciana Wilandari, and D. Ispriyanti, "Penerapan Metode Klasifikasi Support Vector Machine (SVM) pada Data Akreditasi Sekolah Dasar (SD) di Kabupaten

Magelang," *J. Gaussian*, vol. 3, no. 8, pp. 811–820, 2014, [Online]. Available: http://download.portalgaruda.org/article.php?article=286497&val=4706&title=PENERAPAN METODE KLASIFIKASI SUPPORT VECTOR MACHINE (SVM) PADA DATA AKREDITASI SEKOLAH DASAR (SD) DI KABUPATEN MAGELANG.

[11] N. Nadiah, S. Soim, and S. Sholihin, "Implementation of Decision Tree Algorithm Machine Learning in Detecting Covid-19 Virus Patients Using Public Datasets," *Indones. J. Artif. Intell. Data Min.*, vol. 5, no. 1, p. 37, 2022, doi: 10.24014/ijaidm.v5i1.17054.

[12] D. P. Sinambela, H. Naparin, M. Zulfadhilah, and N. Hidayah, "Implementasi Algoritma Decision Tree dan Random Forest dalam Prediksi Perdarahan Pascasalin," *J. Inf. dan Teknol.*, vol. 5, no. 3, pp. 58–64, 2023, doi: 10.60083/jidt.v5i3.393.