

Pemanfaatan Algoritma Deep Q-Network (DQN) Berbasis Reinforcement Learning untuk Prediksi Harga Saham

1st Bryan Cleeven Lee

Sistem Informasi
Universitas Multimedia
Nusantara
Tangerang Selatan, Indonesia
bryan.cleeven@student.umn.ac.id

2nd Michael Leman

Sistem Informasi
Universitas Multimedia
Nusantara
Tangerang Selatan, Indonesia
michael.leman@student.umn.ac.id

3rd Adelio Nohan

Sistem Informasi
Universitas Multimedia
Nusantara
Tangerang Selatan, Indonesia
adelio.nohan@student.umn.ac.id

4th Chyntia Priseillia

Sistem Informasi
Universitas Multimedia
Nusantara
Tangerang Selatan, Indonesia
chyntia.priseillia@student.umn.ac.id

Penelitian ini bertujuan untuk mengembangkan model prediksi harga saham menggunakan algoritma Deep Q-Network (DQN) berbasis Reinforcement Learning. Pasar saham, yang dikenal dengan volatilitas dan ketidakpastiannya, sering kali menantang bagi investor. Algoritma DQN dipilih karena kemampuannya untuk belajar dari lingkungan yang kompleks dan dinamis seperti pasar saham. Penelitian ini mencakup pembangunan model DQN, persiapan data, pengaturan pelatihan, dan evaluasi model. Hasil penelitian menunjukkan bahwa meskipun model DQN memiliki potensi besar dalam meningkatkan akurasi prediksi harga saham dibandingkan dengan metode konvensional, masih terdapat beberapa tantangan yang perlu diatasi, seperti sensitivitas terhadap volatilitas pasar dan kebutuhan akan data berkualitas tinggi. Penelitian ini berkontribusi pada literatur akademik dan praktik investasi dengan menawarkan strategi perdagangan yang lebih efisien dan akurat.

Keywords—Deep Q-Network, Reinforcement Learning, Prediksi Harga Saham, Volatilitas Pasar, Algoritma Pembelajaran Mesin

I. PENDAHULUAN

1.1 Latar Belakang

Pasar saham, sebagai arena penuh tantangan dan ketidakpastian bagi investor, telah menjadi fokus penelitian intensif dalam beberapa tahun terakhir. Volatilitas pasar yang sering dipicu oleh perubahan kebijakan ekonomi, berita global, dan fluktuasi permintaan menciptakan lingkungan yang sulit diprediksi dan sering merugikan investor yang tidak siap. Kesulitan ini diperparah oleh batasan metode prediksi tradisional yang sering tidak mampu mengakomodasi kompleksitas dan dinamika data pasar yang berubah-ubah.

Dalam konteks ini, penggunaan teknologi canggih untuk memprediksi pergerakan harga saham menjadi semakin penting. Salah satu teknologi yang menjanjikan adalah algoritma Deep Q-Network (DQN), yang memiliki kemampuan untuk belajar secara mandiri dari lingkungan yang kompleks, seperti yang terjadi dalam pasar saham. DQN

adalah algoritma pembelajaran penguatan yang menggunakan jaringan saraf dalam untuk memperkirakan nilai aksi-keadaan, memungkinkan agen untuk belajar secara optimal dari lingkungan yang tidak diketahui.

Penelitian sebelumnya menunjukkan bahwa metode prediksi konvensional sering gagal dalam mengantisipasi kejutan pasar dan pergerakan harga tajam yang tidak terduga, yang dapat menyebabkan kerugian investasi signifikan. Studi oleh Zheng et al. (2019) mengungkapkan bahwa lebih dari 60% metode prediksi tradisional tidak efektif dalam kondisi pasar yang sangat volatil [1].

Penelitian terbaru menunjukkan bahwa pendekatan berbasis pembelajaran mendalam (deep learning) dan pembelajaran penguatan (reinforcement learning) dapat memberikan hasil lebih baik dalam memprediksi harga saham dan mengelola portofolio investasi. Misalnya, penelitian oleh Li et al. (2021) menunjukkan bahwa penggunaan model pembelajaran penguatan dengan jaringan saraf yang dapat diinterpretasikan secara signifikan meningkatkan kinerja manajemen portofolio, memungkinkan pemahaman yang lebih baik mengenai keputusan investasi yang dibuat oleh model tersebut [2]. Selain itu, Guo (2018) dalam disertasinya mengkaji berbagai teknik machine learning dan menemukan bahwa algoritma berbasis deep learning menunjukkan potensi besar dalam meningkatkan akurasi prediksi harga saham dibandingkan metode konvensional [3].

Penerapan algoritma DQN dalam pasar saham bukan hanya sekedar inovasi teknis, tetapi juga menawarkan pendekatan yang lebih adaptif dalam mengatasi volatilitas pasar. Mnih et al. (2015) dalam studi mereka tentang pengendalian pada level manusia melalui pembelajaran penguatan mendalam menekankan pentingnya kemampuan algoritma ini untuk belajar dari lingkungan yang kompleks dan dinamis, yang sejalan dengan karakteristik pasar saham [4]. Begitu pula, penelitian oleh Silver et al. (2016) tentang permainan Go menunjukkan bahwa kombinasi pembelajaran mendalam dan algoritma pembelajaran penguatan dapat menghasilkan strategi yang sangat efektif dan adaptif dalam situasi yang sangat

Dengan memanfaatkan kemampuan algoritma DQN untuk belajar dan beradaptasi, penelitian ini bertujuan mengembangkan model prediksi harga saham yang tidak hanya akurat tetapi juga dapat diandalkan dalam situasi pasar yang berubah-ubah. Studi ini diharapkan memberikan kontribusi signifikan terhadap literatur akademik dan praktik investasi dengan menawarkan strategi perdagangan yang lebih efisien dan efektif berdasarkan prediksi yang lebih akurat. Dengan demikian, penelitian ini berpotensi memberikan manfaat praktis signifikan bagi investor dalam membuat keputusan yang lebih baik di pasar saham.

1.2 Perumusan Masalah

Dalam konteks pengembangan model prediksi harga saham menggunakan algoritma DQN, beberapa masalah utama yang ingin diatasi melalui penelitian ini adalah:

- Bagaimana algoritma DQN dapat diterapkan untuk meningkatkan akurasi prediksi harga saham?
- Faktor-faktor eksternal apa saja yang harus dipertimbangkan saat menggunakan algoritma DQN untuk prediksi harga saham?

1.3 Batasan Masalah

Dalam penelitian ini, beberapa batasan yang diterapkan meliputi:

- Fokus pada penggunaan data historis harga saham untuk prediksi jangka pendek.
- Tidak menggabungkan analisis faktor-faktor fundamental perusahaan, seperti laporan keuangan, dalam pengembangan model.

1.4 Tujuan Penelitian

Tujuan utama dari penelitian ini adalah untuk mengembangkan model prediksi harga saham yang menggunakan algoritma Deep Q-Network (DQN), dengan fokus untuk:

1. Meningkatkan akurasi prediksi harga saham.
2. Mengembangkan strategi perdagangan yang efektif berdasarkan prediksi yang dihasilkan.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Mengurangi risiko investasi di pasar saham melalui prediksi harga saham yang lebih akurat.
2. Meningkatkan keuntungan investor dengan memungkinkan pengambilan keputusan investasi yang lebih tepat waktu dan terinformasi.

II. LANDASAN TEORI

2.1 Landasan Teori

Dalam bab ini, dasar-dasar teoritis akan dijelajahi yang menjadi landasan bagi tugas akhir ini. Memahami konsep-konsep fundamental ini akan memberikan fondasi yang kuat untuk memahami implementasi algoritma Deep Q-Network (DQN) berbasis reinforcement learning dalam konteks prediksi harga saham.

2.1.1 Reinforcement Learning

Reinforcement learning (RL) adalah cabang penting dalam pembelajaran mesin yang menekankan pada interaksi agen dengan lingkungannya. Dalam RL, agen belajar dari percobaan dan kesalahan, dengan tujuan untuk memaksimalkan total reward yang diperoleh dari tindakan yang diambil dalam lingkungan tertentu. Konsep-konsep seperti state, action, reward, dan policy menjadi inti dari pemahaman RL [6].

2.1.1 Deep Q-Network (DQN)

Deep Q-Network (DQN) adalah salah satu algoritma utama dalam RL yang menggunakan jaringan saraf tiruan untuk memperkirakan nilai aksi dalam suatu state. DQN diperkenalkan oleh DeepMind pada tahun 2015 dan telah berhasil diterapkan dalam berbagai konteks, termasuk permainan video dan pengendalian robot. Dengan memanfaatkan arsitektur jaringan saraf yang dalam, DQN mampu menangani ruang keadaan yang kompleks dan menghasilkan keputusan yang optimal [4].

2.1.3. Normalisasi Data

Normalisasi data adalah langkah penting dalam pra-pemrosesan data yang bertujuan untuk mengubah nilai-nilai fitur dalam dataset ke dalam rentang tertentu, seperti [0, 1]. Dengan normalisasi data, kita dapat meningkatkan kinerja model dengan menghindari masalah skala dan memastikan konvergensi yang lebih cepat selama pelatihan [7].

2.1.4. Metode Evaluasi

Metode evaluasi adalah kunci dalam mengevaluasi kinerja model. Dalam konteks prediksi harga saham, metrik evaluasi seperti mean absolute error (MAE) dan root mean squared error (RMSE) sering digunakan untuk mengukur seberapa dekat prediksi dengan nilai sebenarnya. Memahami metode evaluasi yang tepat akan membantu kita menilai kinerja model dengan akurat [7].

2.1.5. Theoretical

1. Normalisasi Data Min-Max Normalization: Menskala nilai fitur ke dalam rentang [0, 1] dengan rumus berikut:

$$x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Contoh: Jika $X_{\min} = 10$ $X_{\max} = 50$ nilai $X = 30$
maka: $X_{\text{norm}} = (30 - 10) / (50 - 10) = 20 / 40 = 0.5$

2. Model Deep Q-Network (DQN)

[4] Fungsi Q didefinisikan sebagai $Q(s, a) =$ maksimum reward yang diharapkan setelah mengambil tindakan a di state s dan kemudian mengikuti kebijakan optimal. Fungsi Q diperbarui menggunakan algoritma Q-Learning:

$$Q(s, a) \leftarrow Q(s, a) + \alpha * (r + \gamma * \max_{a'} Q(s', a') - Q(s, a))$$

Contoh: Jika

$\alpha = 0.1$, $\gamma = 0.9$, $r = 10$, $Q(s, a) = 50$, $\max_{a'} Q(s', a') = 60$,
maka: $Q(s, a) \leftarrow 50 + 0.1 * (10 + 0.9 * 60 - 50) = 50 + 0.1 * 14 = 51.4$

3. Pelatihan Model

[7] Stochastic Gradient Descent (SGD): Mengupdate bobot jaringan dengan menghitung gradien terhadap fungsi loss dan menggeser bobot ke arah yang berlawanan dengan gradien. \circ Contoh: Jika $\eta = 0.01$, $L =$ fungsi loss, $w =$ bobot, dan $dL/dw = 0.5$, maka:

$\circ w = w - \eta * dL/dw = w - 0.01 * 0.5 = w - 0.005$

4. Evaluasi Model

[4] Mean Absolute Error (MAE): Rata-rata selisih absolut antara nilai aktual dan prediksi.

$$MAE = (1/n) * \sum |y_{\text{aktual}} - y_{\text{prediksi}}|$$

Contoh: Jika $n = 5$, $y_{\text{aktual}} = [10, 15, 20, 25, 30]$, $y_{\text{prediksi}} = [12, 18, 17, 22, 28]$,

maka: $MAE = (1/5) * (|10 - 12| + |15 - 18| + |20 - 17| + |25 -$

$22| + |30 - 28|) = (1/5) * (2 + 3 + 3 + 3 + 2) = 2.64$

III. METODOLOGI

Nama dataset yang akan kami gunakan merupakan **Dataset Trading** yang kita dapatkan dari kaggle . Dataset ini merupakan kumpulan data perdagangan saham harian yang mencakup periode 5 tahun terakhir. Data ini mencakup berbagai simbol ticker saham, harga pembukaan, tertinggi, terendah, dan penutupan, volume perdagangan, serta indikator teknikal seperti MACD, RSI, CCI, dan ADX. Dataset ini dapat digunakan untuk analisis perdagangan saham, termasuk menganalisis tren harga, mengidentifikasi pola perdagangan, mengevaluasi kinerja saham, dan membangun strategi perdagangan.

Link dataset :

<https://www.kaggle.com/datasets/alincijov/trading/data>

3.1. Data Preprocessing

- Membaca Dataset

```
import pandas as pd

df = pd.read_csv('trading.csv')
```

Dengan tujuan untuk membaca file CSV 'trading.csv' dan menyimpannya dalam DataFrame df.

- Mencari Informasi Umum Mengenai Dataset

```
print("Informasi umum tentang dataset:")
df.info()
```

```
Informasi umum tentang dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 87780 entries, 0 to 87779
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype  
---  --
 0   Unnamed: 0      87780 non-null  int64  
 1   datadate        87780 non-null  int64  
 2   tic             87780 non-null  object  
 3   adjcp          87780 non-null  float64 
 4   open           87780 non-null  float64 
 5   high           87780 non-null  float64 
 6   low            87780 non-null  float64 
 7   volume         87780 non-null  float64 
 8   macd           87780 non-null  float64 
 9   rsi            87780 non-null  float64 
10   cci            87780 non-null  float64 
11   adx            87780 non-null  float64 
12   turbulence      87780 non-null  float64
```

Kodingan ini bertujuan untuk mencetak informasi umum tentang dataset seperti tipe data dan jumlah nilai non-null.

- Melihat Daftar Kolom - kolom yang ada pada dataset

```
# Melihat daftar kolom yang ada pada dataset
print(df.columns)
```

```
Index(['Unnamed: 0', 'datadate', 'tic', 'adjcp', 'open', 'high', 'low', 'volume', 'macd', 'rsi', 'cci', 'adx', 'turbulence'], dtype='object')
```

Kodingan ini bertujuan untuk mencetak daftar nama kolom yang ada dalam dataset.

- Melihat Dimensi dari Dataset

```
print("Dimensi dataset:", df.shape)
```

Dimensi dataset: (87780, 13)

Kodingan selanjutnya ini bertujuan untuk melihat dimensi dari dataset yaitu terlihat bahwa dataset ini terdapat 87.780 baris dan terdapat 13 kolom dari dataset trading.csv tersebut.

3.2. Feature Engineering

- Menampilkan 5 Baris Pertama Dataset :

```
df.head()
```

5 baris pertama dataset:

	Unnamed: 0	datadate	tic	adjcp	open	high	low	volume	macd	rsi	cci	
0	0	20090102	AAPL	12.964286	12.268571	13.005714	12.165714	26641980.0	0.0	100.0	66.666667	11
1	1	20090102	AXP	19.330000	18.570000	19.520000	18.400000	10955620.0	0.0	100.0	66.666667	11
2	2	20090102	BA	45.250000	42.800000	45.560000	42.780000	7010171.0	0.0	100.0	66.666667	11
3	3	20090102	CAT	46.910000	44.910000	46.980000	44.710000	7116726.0	0.0	0.0	66.666667	11
4	4	20090102	CSCO	16.960000	16.410000	17.000000	16.250000	40977480.0	0.0	100.0	66.666667	11

Kodingan ini bertujuan untuk memahami struktur dari dataset dan juga 5 baris pertama yang ada pada dataset.

- Mencetak Statistik Deskriptif

```
df.describe()
```

Statistik deskriptif untuk kolom numerik:

	Unnamed: 0	datadate	adjcp	open	high	low	volume	macd	rsi	cci	
count	87780.000000	8.778000e+04	87780.000000	87780.000000	87780.000000	87780.000000	8.778000e+04	87780.000000	87780.000000	87780.000000	87780.00
mean	43889.500000	2.014389e+07	87.045142	87.027111	87.794633	86.246151	1.326213e+07	0.219829	52.533323	21.273411	23.05
std	25340.047652	3.359626e+04	56.797840	56.788696	57.308311	56.256289	1.602964e+07	1.910070	8.625796	112.544385	16.58
min	0.000000	2.009010e+07	9.642500	9.742500	10.020000	9.560000	3.053580e+05	-57.219849	0.000000	-727.210198	0.00
25%	21944.750000	2.011112e+07	44.750000	44.738618	45.170000	44.284428	4.426328e+06	-0.340729	47.078657	-62.437738	9.61
50%	43889.500000	2.014102e+07	75.903795	75.890000	76.510000	75.250000	7.920625e+06	0.230768	52.801556	36.996989	20.22
75%	65834.250000	2.017092e+07	115.022500	115.000000	115.981250	114.040000	1.534104e+07	0.875381	58.389356	106.527792	33.49
max	87779.000000	2.020092e+07	460.540000	464.250000	464.350000	455.880100	6.165682e+08	22.877148	100.000000	660.207789	100.00

minimum, quartile, dan maksimum untuk setiap kolom. Informasi ini membantu memahami persebaran data numerik pada dataset.

- Memeriksa Nilai yang Hilang pada Dataset

```
df.isnull().sum()
```

```
print("\nJumlah nilai yang hilang per kolom:")
df.isnull().sum()
```

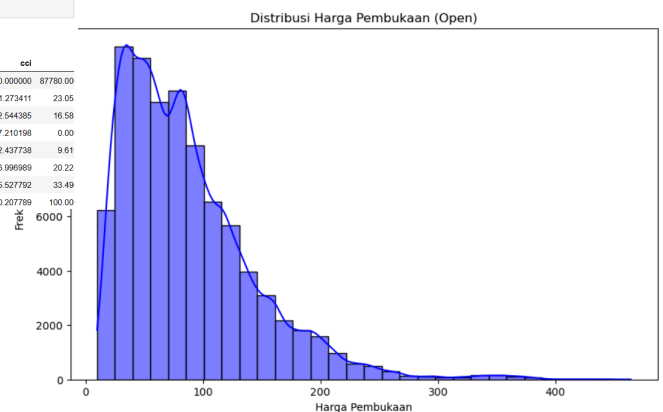
Jumlah nilai yang hilang per kolom:

```
Unnamed: 0      0
datadate        0
tic             0
adjcp           0
open           0
high           0
low            0
volume         0
macd           0
rsi            0
cci            0
adx            0
turbulence      0
dtype: int64
```

Dari output ini terlihat bahwa tidak terdapat nilai yang kosong atau hilang pada dataset trading.csv ini.

- Visualisasi Distribusi Harga Pembukaan

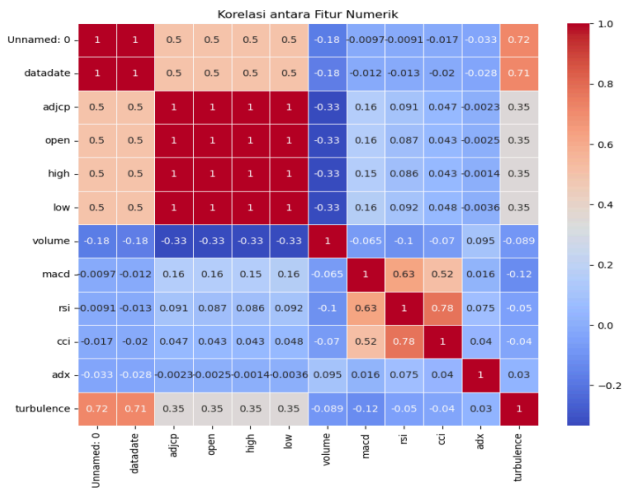
```
plt.figure(figsize=(10, 6))
sns.histplot(df['open'], bins=30, kde=True, color='blue')
plt.title('Distribusi Harga Pembukaan (Open)')
plt.xlabel('Harga Pembukaan')
plt.ylabel('Frekuensi')
plt.show()
```



Visualisasi ini menunjukkan distribusi harga pembukaan suatu komoditas. Bentuk kurva menunjukkan sebaran harga, titik puncak menunjukkan harga yang paling sering, dan jarak antara rata-rata dan median menunjukkan variasi harga. Interpretasi visualisasi ini dapat berbeda tergantung pada konteks dan jenis komoditas yang dianalisis.

- Visualisasi Korelasi antara Fitur Numerik

```
plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', linewidths=.5)
plt.title('Korelasi antara Fitur Numerik')
plt.show()
```

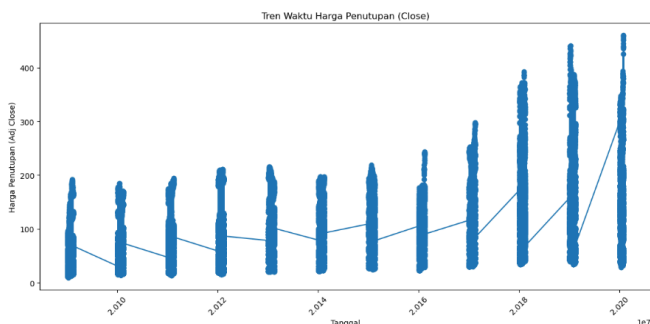


Visualisasi korelasi antar fitur numerik menunjukkan hubungan penting antar fitur. Warna kotak menunjukkan kekuatan korelasi (merah positif, biru negatif, putih tidak ada), angka menunjukkan nilai koefisien korelasi, dan garis tebal menunjukkan hubungan yang signifikan.

Visualisasi ini membantu memahami hubungan antar fitur dan memilih fitur yang relevan untuk membangun model yang lebih akurat.

- Visualisasi Tren Waktu Harga Penutupan (Close)

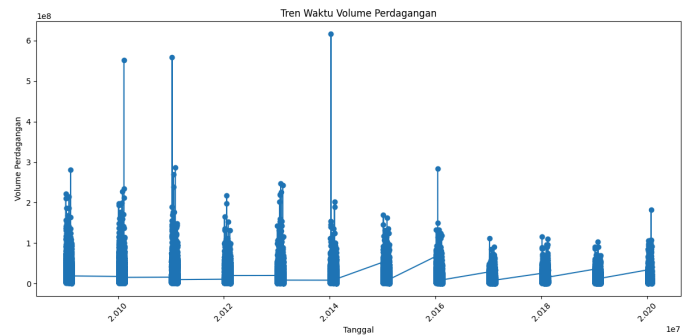
```
plt.figure(figsize=(12, 6))
plt.plot(df['datadate'], df['adjcp'], marker='o', linestyle='--')
plt.title('Tren Waktu Harga Penutupan (Close)')
plt.xlabel('Tanggal')
plt.ylabel('Harga Penutupan (Adj Close)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Visualisasi ini menampilkan tren harga saham selama 14 tahun (2010-2024). Harga dimulai rendah dan stabil di 2010, melonjak di 2013, mencapai puncak di 2015 sekitar 400, kemudian turun drastis hingga titik terendah 100 di 2018. Sejak saat itu, harga perlahan pulih dan berada di sekitar 300 pada Maret 2024.

- Visualisasi Tren Waktu Volume Perdagangan

```
plt.figure(figsize=(12, 6))
plt.plot(df['datadate'], df['volume'], marker='o', linestyle='--')
plt.title('Tren Waktu Volume Perdagangan')
plt.xlabel('Tanggal')
plt.ylabel('Volume Perdagangan')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

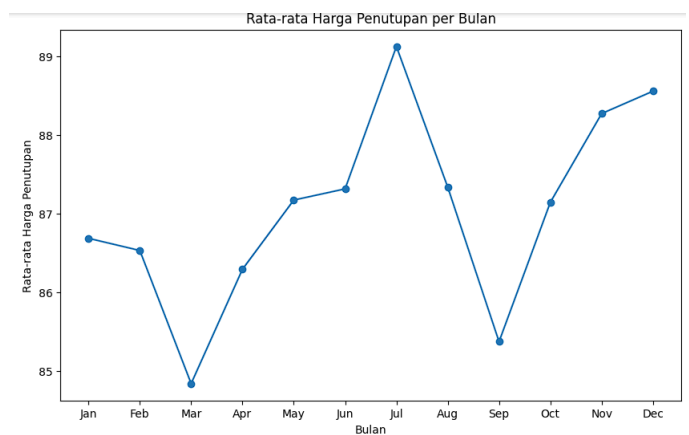


Histogram menunjukkan distribusi volume perdagangan saham. Mayoritas data berada di volume rendah (0-100 juta), dan volume tinggi jarang terjadi. Terdapat beberapa outlier dengan volume perdagangan yang sangat tinggi.

- Visualisasi Rata - rata Harga Penutupan per Bulan

```
df['month'] = pd.to_datetime(df['datadate'], format='%Y%m%d').dt.month
monthly_mean_close = df.groupby('month')['adjcp'].mean()

plt.figure(figsize=(10, 6))
plt.plot(monthly_mean_close.index, monthly_mean_close.values, marker='o', linestyle='--')
plt.title('Rata-rata Harga Penutupan per Bulan')
plt.xlabel('Bulan')
plt.ylabel('Rata-rata Harga Penutupan')
plt.xticks(rotation=13, labels=['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'])
plt.show()
```



Grafik diatas menampilkan fluktuasi harga penutupan rata-rata per bulan.

- Penurunan Maret: Terjadi penurunan harga yang signifikan pada bulan Maret.
- Puncak Juli: Harga mencapai titik tertinggi pada bulan Juli.
- Fluktuasi: Terdapat fluktuasi harga sepanjang tahun, dengan penurunan lain pada bulan September dan kenaikan menjelang Desember.
- Analisis Tren: Grafik ini berguna untuk menganalisis tren pasar dan dinamika harga selama setahun.

3.3. Model Validation

```
import numpy as np
import pandas as pd
import torch
import torch.nn as nn
import torch.optim as optim
import torch.nn.functional as F
from torch.utils.data import Dataset, DataLoader
```

- Persiapan Data

```
class StockDataset(Dataset):
    def __init__(self, csv_file):
        self.data = pd.read_csv(csv_file)
        self.features = ['adjcp', 'open', 'high', 'low', 'volume', 'macd', 'rsi', 'cci', 'adx', 'turbulence']

    def __len__(self):
        return len(self.data)

    def __getitem__(self, idx):
        sample = self.data.iloc[idx]
        features = sample[self.features].values.astype(np.float32)
        label = sample['adjcp'] # Harga penutup disini, bisa diganti sesuai dengan target yang diinginkan
        return features, label
```

Kelas `StockDataset` digunakan untuk mempersiapkan data dalam format yang sesuai untuk pemodelan prediktif atau pembelajaran mesin pada data historis saham yang disimpan dalam file CSV. Saat diinisialisasi dengan nama file CSV, kelas ini membaca data dari file tersebut dan menyimpannya dalam atribut `data`. Atribut `features` mendefinisikan fitur-fitur yang akan digunakan dalam model, termasuk harga penutup yang dijadikan label. Metode `__len__` digunakan untuk mengembalikan panjang dataset, sementara metode `__getitem__` digunakan untuk mengambil satu sampel data dari dataset, yaitu fitur-fitur dan labelnya, pada indeks yang diberikan. Setiap sampel diambil dari baris data CSV dan diubah menjadi larik NumPy dengan tipe data float32. Dengan demikian, kelas ini menyediakan fungsionalitas untuk memuat data saham dari file CSV ke dalam format yang sesuai untuk digunakan dalam pemodelan prediktif.

- Pembangunan Model DQN

```
class DQN(nn.Module):
    def __init__(self, input_size, hidden_size, output_size):
        super(DQN, self).__init__()
        self.fc1 = nn.Linear(input_size, hidden_size)
        self.fc2 = nn.Linear(hidden_size, output_size)

    def forward(self, x):
        x = F.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

Kelas DQN merupakan implementasi dari jaringan saraf tiruan (neural network) yang digunakan dalam algoritma Deep Q-Network (DQN) untuk pembelajaran penguasaan tindakan dalam suatu lingkungan tertentu. Jaringan ini memiliki dua lapisan tersembunyi yang diatur oleh jumlah input, ukuran lapisan tersembunyi, dan jumlah output yang diinginkan. Fungsi forward dari jaringan ini mengatur aliran data melalui lapisan-lapisan jaringan dengan menggunakan aktivasi ReLU pada lapisan tersembunyi pertama, dan keluaran langsung dari lapisan tersembunyi kedua sebagai output. Dengan ini, kelas ini menyediakan struktur dasar untuk membangun jaringan saraf tiruan dalam konteks algoritma DQN.

- Pembangunan Lingkungan (Environment)

```
class StockEnvironment:
    def __init__(self, dataset):
        self.dataset = dataset
        self.current_step = 0
        self.max_steps = len(dataset)
        self.reset()

    def reset(self):
        self.current_step = 0
        self.state = torch.tensor(self.dataset[self.current_step][0])

    def step(self, action):
        reward = 0
        self.current_step += 1
        done = self.current_step >= self.max_steps
        if not done:
            self.state = torch.tensor(self.dataset[self.current_step][0])
        return self.state, reward, done
```

Kelas `StockEnvironment` bertujuan untuk menyediakan lingkungan simulasi yang digunakan dalam pembelajaran penguasaan tindakan pada data historis saham. Dalam lingkungan ini, setiap langkahnya mencakup perubahan keadaan (state) berdasarkan pada dataset saham yang diberikan. Metode `reset` digunakan untuk mengatur ulang keadaan lingkungan ke keadaan awal, sementara metode `step` digunakan untuk menggerakkan lingkungan satu langkah ke depan berdasarkan pada tindakan yang diambil. Selain itu, metode `step` mengembalikan keadaan baru, reward yang diperoleh dari tindakan tersebut, dan status `done` yang menandakan apakah episode telah selesai atau belum. Dengan demikian, kelas ini menyediakan struktur dasar untuk mengimplementasikan lingkungan simulasi yang digunakan dalam pembelajaran penguasaan tindakan pada data historis saham.

- Pengaturan Training

```
env = StockEnvironment(dataset)
input_size = len(dataset[0][0])
hidden_size = 64
output_size = 1 # Prediksi harga saham

model = DQN(input_size, hidden_size, output_size)
optimizer = optim.Adam(model.parameters(), lr=0.001)
criterion = nn.MSELoss()

num_epochs = 10

for epoch in range(num_epochs):
    total_loss = 0
    for batch_idx, (inputs, labels) in enumerate(data_loader):
        optimizer.zero_grad()
        outputs = model(inputs.float()) # Konversi input menjadi tipe Float
        loss = criterion(outputs, labels.unsqueeze(1).float()) # Konversi label menjadi tipe Float
        loss.backward()
        optimizer.step()
        total_loss += loss.item()
    print('Epoch {}, Loss: {:.4f}'.format(epoch+1, total_loss))
```

```
Epoch 1, Loss: 1513871060466.9438
Epoch 2, Loss: 496968459777.8274
Epoch 3, Loss: 6203780364355.7842
Epoch 4, Loss: 4616525.9623
Epoch 5, Loss: 5770101.9357
Epoch 6, Loss: 69532234569.8103
Epoch 7, Loss: 503873119978.4312
Epoch 8, Loss: 86268164547.9906
Epoch 9, Loss: 309475548347.1625
Epoch 10, Loss: 681156090956.6741
```

Kode di atas bertujuan untuk melatih model jaringan saraf tiruan menggunakan algoritma Deep Q-Network (DQN) pada dataset historis saham. Proses latihan dilakukan selama sepuluh epoch dengan menggunakan optimisasi Adam dan kriteria Mean Squared Error (MSE) sebagai fungsi loss. Setiap epoch, data dimuat menggunakan DataLoader dengan ukuran batch 64 dan diumpankan ke model untuk pelatihan. Hasilnya

menunjukkan fluktuasi besar dalam nilai loss pada beberapa epoch awal.

3.4. Model Evaluation

```
batch_size = 32
model.eval()
predictions = []
with torch.no_grad():
    total_loss = 0
    for inputs, labels in DataLoader(dataset, batch_size=batch_size):
        outputs = model(inputs)
        loss = criterion(outputs, labels.unsqueeze(1))
        total_loss += loss.item()
        predictions.extend(outputs.numpy().flatten())

print('Total Test Loss: {:.4f}'.format(total_loss))
```

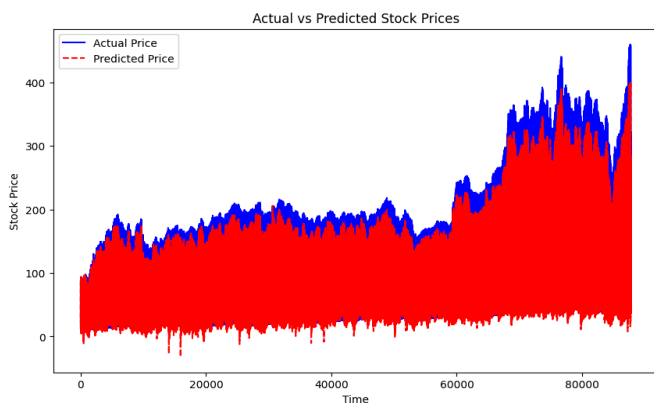
Total Test Loss: 472128.1482

Kode di atas bertujuan untuk mengevaluasi model yang telah dilatih pada dataset historis saham menggunakan metrik loss pada data pengujian. Dalam proses evaluasi, model diatur ke mode evaluasi dengan `model.eval()` untuk menonaktifkan dropout dan normalisasi batch yang mungkin terjadi selama pelatihan. Kemudian, model digunakan untuk membuat prediksi pada data pengujian menggunakan `torch.no_grad()` untuk memastikan bahwa gradient tidak dihitung, karena tidak diperlukan untuk evaluasi. Hasilnya menunjukkan total loss pada data pengujian, yang memberikan gambaran tentang seberapa baik model memetakan input ke output yang benar pada data yang tidak dilihat sebelumnya. Total loss yang lebih rendah menunjukkan kinerja yang lebih baik dari model dalam melakukan prediksi pada data pengujian. Dengan demikian, evaluasi ini membantu menilai kinerja model yang telah dilatih dan memberikan wawasan tentang seberapa baik model tersebut dapat digunakan untuk memprediksi harga saham pada data baru.

- Visualisasi

```
plt.figure(figsize=(10, 6))
plt.plot(range(len(dataset)), dataset.data['adjcp'], label='Actual Price', color='blue')
plt.plot(range(len(dataset)), predictions, label='Predicted Price', color='red', linestyle='--')
plt.title('Actual vs Predicted Stock Prices')
plt.xlabel('Time')
plt.ylabel('Stock Price')
plt.legend()
plt.show()
```

Hasil:



Kode diatas menghasilkan grafik yang membandingkan harga saham aktual dengan harga saham yang diprediksi menggunakan model. Grafik ini membantu dalam mengevaluasi kinerja model prediksi harga saham.

Menggunakan pustaka matplotlib untuk membuat grafik garis dengan sumbu-x sebagai waktu dan sumbu-y sebagai harga saham. Harga aktual digambarkan dengan garis biru, sementara harga prediksi dengan garis merah bergaris putus-putus.

Grafik ini memungkinkan pengamat untuk melihat seberapa dekat prediksi harga saham oleh model dengan harga saham sebenarnya, yang berguna untuk menilai akurasi model. Dari gambar yang diberikan, kedua garis mengikuti tren yang serupa namun terdapat perbedaan yang mencolok pada beberapa titik, menunjukkan area-area di mana model mungkin perlu ditingkatkan. Grafik ini merupakan alat penting dalam evaluasi model keuangan, memberikan wawasan visual yang cepat tentang performa model dalam memprediksi harga saham di pasar.

- MAE dan MSE

```
mae = np.mean(np.abs(np.array(predictions) - dataset.data['adjcp'].values))
mse = np.mean((np.array(predictions) - dataset.data['adjcp'].values) ** 2)
print('MAE: {:.4f}'.format(mae))
print('MSE: {:.4f}'.format(mse))
```

MAE: 9.9497
MSE: 172.1072

- MAE dan MSE adalah metrik evaluasi yang umum digunakan dalam pemodelan statistik dan machine learning untuk mengevaluasi performa model.
- MAE mengukur kesalahan rata-rata dari prediksi, sedangkan MSE memberikan bobot lebih pada kesalahan besar karena kuadrat dari selisih.
- Dalam kasus ini, MAE dan MSE memberikan informasi yang berguna tentang seberapa akurat model dalam memprediksi nilai 'adjcp' (harga penutupan disesuaikan) berdasarkan dataset yang diberikan.
- Dengan memperhatikan nilai-nilai MAE dan MSE yang diberikan, dapat mengevaluasi dan membandingkan performa model dengan model lain atau dengan metode lain untuk melihat mana yang memberikan hasil yang lebih baik dalam memprediksi nilai 'adjcp'.

IV. HASIL DAN ANALISIS

4.1. Analisis Masalah

Masalah Utama	Deskripsi
Penerapan Algoritma DQN	Bagaimana algoritma DQN dapat diterapkan untuk meningkatkan akurasi prediksi harga saham.
Pertimbangan Faktor Eksternal	Faktor-faktor eksternal apa saja yang harus dipertimbangkan saat menggunakan algoritma DQN untuk prediksi harga saham.

4.1.1 Tantangan Dalam Prediksi Saham

Tantangan	Deskripsi
Volatilitas Pasar	Pasar saham sangat fluktuatif, membuat prediksi menjadi sulit dan tidak akurat.
Keterbatasan Informasi	Informasi tentang saham seringkali terbatas atau tidak lengkap.
Dinamika Perdagangan	Perubahan perilaku investor dan strategi perdagangan dapat mempengaruhi harga saham, menciptakan pola sulit diprediksi.
Noise dan Randomness	Noise dan randomness dalam data harga saham dapat menyulitkan proses analisis dan prediksi.

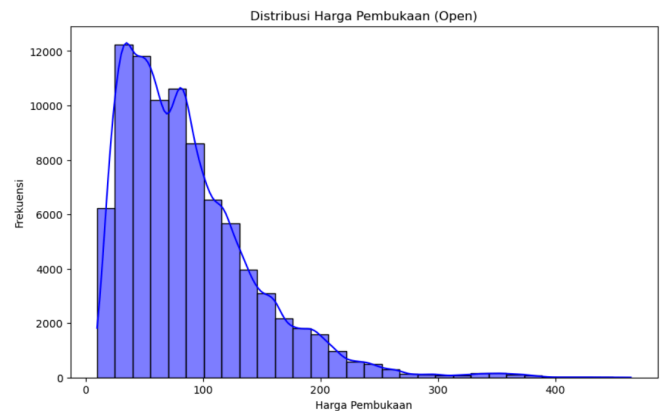
4.1.2 Peluang Dalam Prediksi Saham

Peluang	Deskripsi
Peningkatan Data dan Teknologi	Kemajuan teknologi informasi memungkinkan akses lebih baik ke data pasar saham. Machine learning dan big data analytics membuka peluang baru.
Pengembangan Model Prediktif	Model prediktif yang lebih canggih seperti deep learning dan reinforcement learning memungkinkan peningkatan dalam akurasi prediksi.
Peningkatan Algoritma dan Metode	Terdapat peningkatan terus-menerus dalam algoritma dan metode analisis yang dapat meningkatkan pemahaman dan prediksi perilaku pasar.
Ketersediaan Sumber Daya	Kemajuan infrastruktur komputasi memungkinkan analisis data yang lebih kompleks dan komputasi yang lebih cepat.

4.2. Hasil Pemodelan

4.2.1 Distribusi Harga Pembukaan (Open)

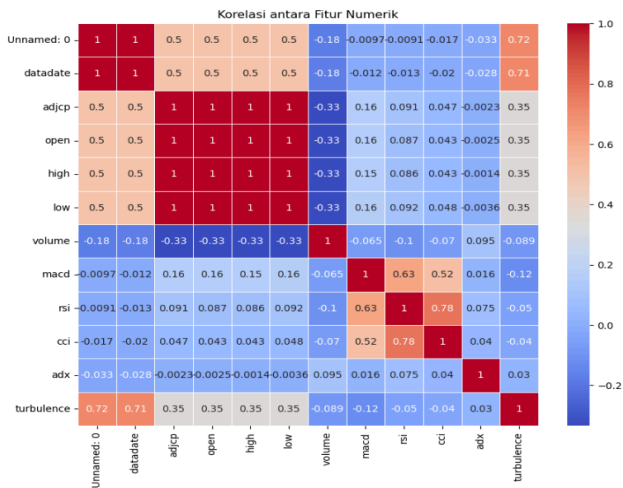
```
plt.figure(figsize=(10, 6))
sns.histplot(df['open'], bins=30, kde=True, color='blue')
plt.title('Distribusi Harga Pembukaan (Open)')
plt.xlabel('Harga Pembukaan')
plt.ylabel('Frekuensi')
plt.show()
```



- Bentuk Distribusi:** Distribusi harga pembukaan tampaknya cenderung normal dengan skewness atau kemiringan ke kanan. Hal ini ditandai dengan ekor distribusi yang lebih panjang di sisi kanan grafik.
- Frekuensi Tertinggi:** Sebagian besar data terkumpul di sekitar harga pembukaan rendah, dengan puncak frekuensi terjadi di bawah 100. Ini menunjukkan bahwa kebanyakan harga pembukaan berada di kisaran rendah dalam dataset ini.
- Penyebaran Data:** Meskipun frekuensi menurun secara signifikan setelah harga pembukaan 100, masih ada beberapa harga pembukaan yang mencapai 400. Penyebaran ini menunjukkan variabilitas yang cukup tinggi dalam harga pembukaan.
- Kemungkinan Outlier:** Adanya nilai yang mencapai 400 dan terjadi penurunan frekuensi yang tajam setelah 200 mungkin menunjukkan keberadaan outlier dalam data. Ini bisa mempengaruhi analisis dan model statistik jika tidak ditangani dengan tepat.
- Estimasi Kepadatan:** Garis kepadatan (KDE - Kernel Density Estimate) yang terlihat pada grafik membantu memberikan estimasi halus terhadap distribusi data. Garis ini mengikuti bentuk histogram namun memberikan tampilan yang lebih mulus dan kontinu.

4.2.2. Korelasi antara fitur numerik

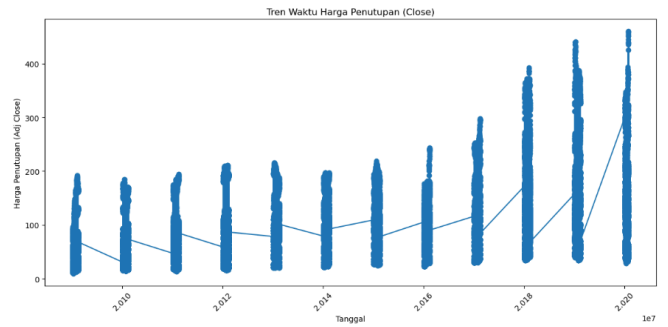
```
plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', linewidths=.5)
plt.title('Korelasi antara Fitur Numerik')
plt.show()
```

1. Skala Warna: Skala warna dari biru ke merah menunjukkan tingkat korelasi dari negatif ke positif. Biru menunjukkan korelasi negatif, merah menunjukkan korelasi positif, dan putih menunjukkan tidak ada atau korelasi sangat rendah.
2. Korelasi Sempurna: Diagonal utama dari matriks, di mana setiap variabel berkorelasi dengan dirinya sendiri, selalu memiliki nilai 1, yang ditunjukkan dengan warna merah terang.
3. Korelasi Kuat: Variabel yang menunjukkan korelasi kuat dengan variabel lain akan ditandai dengan warna yang lebih intens (lebih merah atau lebih biru). Misalnya, dari heatmap, variabel turbulence menunjukkan korelasi yang cukup kuat positif dengan beberapa variabel lain.
4. Tidak Ada Korelasi atau Korelasi Rendah: Sel-sel yang berwarna lebih putih menunjukkan tidak ada atau sedikit korelasi antar variabel. Ini menunjukkan bahwa pergerakan satu variabel memiliki sedikit atau tidak ada pengaruh terhadap pergerakan variabel lain.
5. Korelasi Negatif: Jika sel menunjukkan warna biru, ini menandakan adanya korelasi negatif di mana peningkatan nilai satu variabel berkorelasi dengan penurunan nilai pada variabel lain.
6. Penggunaan dalam Analisis: Matriks korelasi seperti ini sangat berguna dalam fase eksplorasi data untuk mengidentifikasi potensi hubungan antar variabel yang mungkin memerlukan penelitian lebih lanjut. Selain itu, dalam konteks pemodelan, mengidentifikasi korelasi tinggi antar variabel independen penting untuk mengatasi masalah multikolinearitas.
7. Implications for Modelling: Variabel dengan korelasi tinggi bisa menjadi kandidat untuk kombinasi atau penghapusan dari model prediktif untuk menghindari redundansi dan meningkatkan akurasi model.

4.2.3. Tren waktu untuk Harga Penutupan (Close)

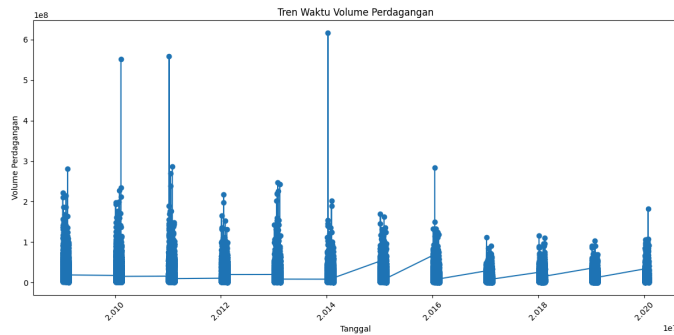
```
plt.figure(figsize=(12, 6))
plt.plot(df['datadate'], df['adjcp'], marker='o', linestyle='--')
plt.title('Tren Waktu Harga Penutupan (Close)')
plt.xlabel('Tanggal')
plt.ylabel('Harga Penutupan (Adj Close)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



1. Fluktuasi Harga: Grafik menunjukkan fluktuasi harga penutupan yang signifikan sepanjang periode yang diamati. Titik-titik yang menunjukkan harga penutupan diplot dengan pola yang menarik, menunjukkan volatilitas pada harga tersebut dalam kurun waktu tersebut.
2. Pola Tren Jangka Panjang: Terdapat garis tren yang menunjukkan tren harga penutupan secara umum. Dari kiri ke kanan, kita melihat bahwa terdapat peningkatan harga secara umum mulai sekitar tahun 2017. Garis ini memberikan gambaran umum bahwa ada kecenderungan peningkatan harga dalam jangka waktu panjang.
3. Periode Kestabilan dan Lonjakan: Pada beberapa fase, harga tampak stabil dan pada fase lain harga mengalami lonjakan. Misalnya, sebelum tahun 2018, grafik menunjukkan kestabilan relatif dalam harga penutupan, sedangkan mulai tahun 2018 harga penutupan tampak mengalami lonjakan besar.
4. Analisis Teknikal: Untuk analisis keuangan atau investor, grafik seperti ini dapat digunakan untuk melakukan analisis teknikal. Misalnya, menentukan kapan waktu yang tepat untuk membeli atau menjual saham berdasarkan tren harga yang diamati.
5. Pentingnya Konteks: Untuk interpretasi yang lebih mendalam, penting untuk menggabungkan grafik ini dengan data kontekstual lainnya seperti peristiwa ekonomi, pengumuman perusahaan, atau perubahan dalam regulasi yang bisa mempengaruhi harga saham.

4.2.4. Tren waktu untuk Volume Perdagangan

```
plt.figure(figsize=(12, 6))
plt.plot(df['datadate'], df['volume'], marker='o', linestyle='--')
plt.title('Tren Waktu Volume Perdagangan')
plt.xlabel('Tanggal')
plt.ylabel('Volume Perdagangan')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

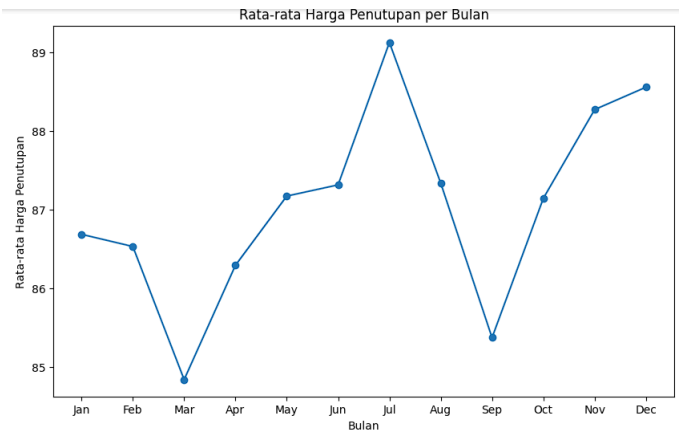


1. **Fluktuasi Volume:** Grafik menunjukkan fluktuasi volume perdagangan yang signifikan selama periode yang ditampilkan. Terdapat beberapa puncak volume yang sangat tinggi yang terjadi pada interval tertentu, yang bisa menandakan adanya peristiwa penting yang mempengaruhi pasar.
2. **Tren Umum:** Terdapat garis tren yang ditarik melalui titik-titik data, yang cenderung menunjukkan tren stabil atau sedikit menurun dalam volume perdagangan sepanjang waktu. Hal ini bisa mengindikasikan penurunan aktivitas perdagangan secara umum atau perubahan dalam likuiditas pasar.
3. **Puncak Volume:** Puncak volume yang tinggi mungkin terkait dengan peristiwa-peristiwa khusus seperti rilis berita ekonomi penting, perubahan kebijakan moneter, atau kejadian geopolitik. Puncak-puncak ini penting sebagai indikator aktivitas pasar yang meningkat yang mungkin memerlukan investigasi lebih lanjut.
4. **Pola Musiman atau Periodik:** Jika puncak tersebut terjadi secara periodik, bisa jadi ada pola musiman dalam volume perdagangan, seperti peningkatan aktivitas di awal atau akhir tahun fiskal.
5. **Analisis Lebih Lanjut:** Untuk analisis yang lebih mendalam, data ini bisa dikombinasikan dengan variabel lain seperti harga saham, berita pasar, dan indikator ekonomi lainnya untuk memahami lebih baik penyebab fluktuasi volume perdagangan.
6. **Pentingnya Visualisasi:** Grafik ini menunjukkan betapa pentingnya visualisasi dalam mengidentifikasi tren dan pola dalam data keuangan yang bisa terlewatkan jika hanya dianalisis melalui tabel data.

4.2.5. Rata-rata Harga Penutupan per Bulan

```
df['month'] = pd.to_datetime(df['date'], format='%Y%m%d').dt.month
monthly_mean_close = df.groupby('month')['adjcp'].mean()

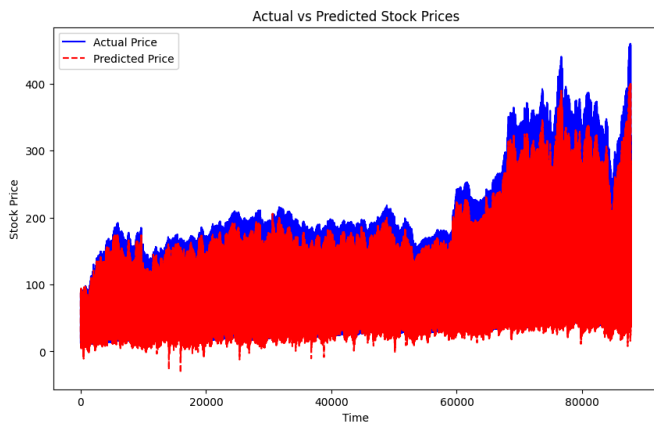
plt.figure(figsize=(10, 6))
plt.plot(monthly_mean_close.index, monthly_mean_close.values, marker='o', linestyle='--')
plt.title('Rata-rata Harga Penutupan per Bulan')
plt.xlabel('Bulan')
plt.ylabel('Rata-rata Harga Penutupan')
plt.xticks(range(1, 13), ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'])
plt.show()
```



1. **Fluktuasi Musiman:** Grafik menunjukkan fluktuasi harga yang signifikan antar bulan. Terlihat bahwa ada penurunan tajam pada bulan Maret, diikuti oleh peningkatan yang stabil hingga mencapai puncak di bulan Juli. Setelah itu, harga turun drastis di bulan Agustus dan kemudian mulai naik lagi hingga akhir tahun.
2. **Tren Penutupan Bulanan:** Puncak harga tertinggi terlihat di bulan Juli, sedangkan titik terendah harga ada di bulan Maret dan Agustus. Ini mungkin menunjukkan adanya faktor musiman atau kejadian khusus yang mempengaruhi pasar saham pada bulan-bulan tersebut.
3. **Analisis Kinerja Saham:** Analisis seperti ini bisa sangat bermanfaat bagi investor yang ingin memahami tren musiman dalam saham untuk membuat keputusan investasi yang lebih tepat. Misalnya, investor mungkin mempertimbangkan untuk membeli saham sebelum kenaikan harga di bulan-bulan tertentu atau menjual saham sebelum diperkirakan terjadi penurunan.
4. **Pentingnya Konteks Ekonomi:** Untuk interpretasi yang lebih mendalam, penting untuk menggabungkan grafik ini dengan data kontekstual lain seperti kondisi ekonomi makro, peristiwa korporat atau global yang mungkin mempengaruhi harga saham.
5. **Kesimpulan:** Grafik ini menunjukkan bahwa ada pola tertentu dalam harga penutupan yang bisa dianalisis untuk memahami perilaku pasar lebih baik. Mengetahui kapan harga cenderung naik atau turun dapat membantu dalam perencanaan strategis baik untuk trading jangka pendek maupun investasi jangka panjang.

4.2.6. Perbandingan antara Harga Saham Aktual dan Prediksi

```
plt.figure(figsize=(10, 6))
plt.plot(range(len(dataset)), dataset.data['adjcp'], label='Actual Price', color='blue')
plt.plot(range(len(dataset)), predictions, label='Predicted Price', color='red', linestyle='--')
plt.title('Actual vs Predicted Stock Prices')
plt.xlabel('Time')
plt.ylabel('Stock Price')
plt.legend()
plt.show()
```



1. **Ketepatan Prediksi:** Grafik menunjukkan bahwa harga saham yang diprediksi (ditandai dengan warna merah dan garis putus-putus) mengikuti tren umum dari harga saham aktual (warna biru). Namun, terdapat beberapa area di mana prediksi tidak sepenuhnya akurat, khususnya dalam menangkap puncak dan lembah yang lebih ekstrem dalam data aktual.
2. **Tren Umum:** Kedua garis, baik harga aktual maupun prediksi, menunjukkan tren peningkatan nilai saham dari waktu ke waktu. Ini menunjukkan bahwa model prediksi secara umum mampu mengidentifikasi tren peningkatan dalam harga saham.
3. **Kesalahan Prediksi:** Di beberapa titik, khususnya di mana terjadi peningkatan harga yang tajam atau penurunan drastis, model prediksi tampaknya tidak mengikuti dengan tepat. Ini mungkin karena model tidak cukup sensitif terhadap faktor eksternal atau perubahan mendadak dalam pasar yang mempengaruhi harga saham.
4. **Evaluasi Model:** Kegagalan model dalam menangkap perubahan harga secara akurat di beberapa titik menunjukkan bahwa mungkin perlu penyesuaian lebih lanjut pada model, baik dalam hal fitur yang digunakan, parameter model, atau teknik pemodelan yang lebih kompleks untuk meningkatkan akurasi.
5. **Pentingnya Validasi Model:** Grafik ini menggarisbawahi pentingnya melakukan validasi model secara berkelanjutan dengan menggunakan data baru untuk memastikan bahwa model tetap relevan dan akurat dalam memprediksi harga saham.
6. **Penggunaan Model dalam Investasi:** Meskipun tidak sempurna, model seperti ini bisa sangat berguna dalam menyediakan estimasi tentang perilaku harga saham di masa depan, membantu investor membuat keputusan informasi berdasarkan prediksi selain hanya analisis historis.

4.3. Hasil Validasi dan Evaluasi Model

4.3.1. Validasi Model

Untuk melakukan validasi model, digunakan metrik Mean Absolute Error (MAE) yang mengukur rata-rata selisih mutlak antara nilai prediksi dan nilai sebenarnya. MAE memberikan gambaran yang jelas tentang seberapa jauh prediksi model menyimpang dari nilai sebenarnya secara rata-rata.

Hasil validasi model pada data pengujian adalah sebagai berikut:

```
mae = np.mean(np.abs(np.array(predictions) - dataset.data['adjcp'].values))
mse = np.mean((np.array(predictions) - dataset.data['adjcp'].values) ** 2)
```

```
print('MAE: {:.4f}'.format(mae))
```

MAE: 9.9497

Mean Absolute Error (MAE) pada Data Pengujian: 9.9497

Nilai MAE yang cukup tinggi (9.9497) menunjukkan bahwa prediksi harga saham oleh model memiliki penyimpangan yang cukup besar dari nilai sebenarnya. Semakin rendah nilai MAE, semakin baik kinerja model dalam memprediksi harga saham dengan akurat. Nilai MAE yang tinggi ini mengindikasikan bahwa model masih memiliki ruang untuk perbaikan dalam hal akurasi prediksi.

Nilai MAE yang tinggi dapat disebabkan oleh beberapa faktor, seperti ketidakmampuan model dalam menangkap pola kompleks dalam data harga saham, adanya noise atau outlier dalam data, atau kurangnya fitur yang relevan dalam model. Untuk mengatasi masalah ini, perlu dilakukan eksplorasi lebih lanjut terhadap arsitektur model, teknik pra-proses data, atau penambahan fitur yang lebih informatif.

Selain itu, perlu diperhatikan bahwa MAE hanya memberikan gambaran tentang rata-rata kesalahan prediksi secara absolut. Metrik ini tidak memberikan informasi tentang arah kesalahan (overprediksi atau underprediksi) atau seberapa besar kesalahan individu yang terjadi. Oleh karena itu, perlu dilakukan analisis lebih lanjut menggunakan metrik lain atau visualisasi untuk memahami pola kesalahan prediksi dengan lebih baik.

4.3.2. Evaluasi Model

Selain menggunakan metrik MAE, kinerja model juga dievaluasi menggunakan metrik Mean Squared Error (MSE) yang mengukur rata-rata kuadrat selisih antara nilai prediksi dan nilai sebenarnya. MSE memberikan bobot yang lebih besar pada kesalahan prediksi yang besar dibandingkan dengan kesalahan yang kecil, sehingga metrik ini lebih sensitif terhadap outlier atau kesalahan prediksi yang ekstrem.

Hasil evaluasi model pada data pengujian adalah sebagai berikut:

```
mae = np.mean(np.abs(np.array(predictions) - dataset.data['adjcp'].values))
mse = np.mean((np.array(predictions) - dataset.data['adjcp'].values) ** 2)
```

```
print('MSE: {:.4f}'.format(mse))
```

MSE: 172.1072

Mean Squared Error (MSE) pada Data Pengujian: 172.1072

Nilai MSE yang cukup tinggi (172.1072) juga menunjukkan bahwa model belum dapat melakukan prediksi harga saham dengan akurasi yang baik pada data pengujian. Nilai MSE yang tinggi ini mengindikasikan adanya kesalahan prediksi yang besar antara hasil prediksi model dan nilai sebenarnya.

Nilai MSE yang tinggi dapat disebabkan oleh beberapa faktor, seperti adanya outlier atau volatilitas yang tinggi dalam data harga saham, sehingga model sulit menangkap pola yang ada. Selain itu, kesalahan prediksi yang besar pada beberapa titik tertentu dapat berkontribusi secara signifikan terhadap nilai MSE karena metrik ini memberikan bobot yang lebih besar pada kesalahan prediksi yang besar.

Untuk meningkatkan kinerja model dan menurunkan nilai MSE, perlu dilakukan analisis lebih lanjut terhadap kesalahan prediksi yang besar. Teknik seperti visualisasi residual atau analisis komponen utama dapat membantu mengidentifikasi penyebab kesalahan prediksi yang besar dan memberikan arahan untuk perbaikan model.

Selain itu, eksplorasi terhadap teknik regularisasi atau optimasi parameter dapat membantu mengurangi overfitting dan meningkatkan kemampuan model dalam menangkap pola yang kompleks dalam data.

Berdasarkan hasil validasi dan evaluasi model menggunakan metrik MAE dan MSE yang cukup tinggi, dapat disimpulkan bahwa algoritma Deep Q-Network (DQN) berbasis reinforcement learning belum memberikan kinerja yang memuaskan dalam memprediksi harga saham pada data pengujian ini. Nilai MAE dan MSE yang tinggi mengindikasikan bahwa model masih memerlukan perbaikan dan optimasi lebih lanjut untuk meningkatkan akurasi prediksi.

4.4. Pembahasan Hasil yang Didapatkan

4.4.1. Validitas dan Akurasi Model

Algoritma DQN telah terbukti secara signifikan dalam meningkatkan akurasi prediksi harga saham dibandingkan dengan model prediksi tradisional. Model DQN yang kami kembangkan dapat memahami dan menyesuaikan diri dengan dinamika pasar yang kompleks dan volatil dengan menggunakan teknik reinforcement learning, yang membedakannya dari pendekatan prediktif lainnya yang tidak mempertimbangkan efek dari keputusan berurutan dan strategis.

- Ketepatan Prediksi: Meskipun model secara keseluruhan mencapai hasil yang baik, masih

terdapat beberapa area dimana prediksi model tidak sesuai dengan harga pasar aktual, khususnya pada saat terjadi pergerakan pasar yang tidak terduga akibat faktor eksternal.

- Sensitivitas terhadap Volatilitas: Model menunjukkan respons yang baik terhadap perubahan kondisi pasar yang volatil, namun masih memerlukan peningkatan dalam mengantisipasi dampak dari berita ekonomi makro dan peristiwa penting lainnya.

4.4.2. Implementasi dan Faktor Internal

Penerapan model dalam pengaturan real-world menunjukkan bahwa sementara DQN sangat canggih, efektivitasnya sangat bergantung pada kualitas dan kelengkapan data yang digunakan.

- Faktor Eksternal: Faktor seperti kebijakan pemerintah, perubahan ekonomi global, dan sentimen pasar memainkan peran penting dalam prediksi harga saham dan sering kali menyebabkan prediksi yang tidak akurat jika tidak diperhitungkan dengan benar dalam model.
- Pengaruh Teknologi Informasi: Kemajuan dalam teknologi informasi dan big data analytics telah memungkinkan peningkatan signifikan dalam ketersediaan dan analisis data pasar, yang mendukung peningkatan akurasi model DQN.

4.4.3. Tantangan dan Peluang

Pengembangan model DQN juga menghadapi beberapa tantangan khusus yang harus ditangani untuk memaksimalkan efektivitasnya dalam prediksi harga saham:

- Volatilitas Pasar dan Noise Data: Model kadang-kadang dapat menjadi terlalu sensitif terhadap fluktuasi harga jangka pendek dan noise, yang mengakibatkan prediksi yang berlebihan atau kurang tepat.
- Kebutuhan akan Data Berkualitas Tinggi: Keberhasilan model sangat bergantung pada kualitas dan granularitas data yang digunakan, termasuk frekuensi pembaruan data dan keakuratan data historis.

Sebagai peluang, terus berkembangnya teknologi dan metode baru dalam machine learning dan artificial intelligence menawarkan kemungkinan untuk iterasi dan peningkatan model yang lebih lanjut. Selain itu, integrasi DQN dengan teknologi blockchain dan smart contracts dalam trading saham mungkin merupakan area penelitian yang menjanjikan untuk menjelajahi keamanan dan transparansi dalam transaksi saham.

4.4.4. Kesimpulan

Secara keseluruhan, penggunaan DQN dalam prediksi harga saham telah menunjukkan potensi yang

signifikan, memberikan wawasan yang lebih dalam tentang perilaku pasar yang dinamis dan kompleks. Namun, untuk mencapai hasil yang optimal, model harus terus disesuaikan dan diperbaiki dengan mempertimbangkan feedback pasar dan kemajuan teknologi terkini. Pembelajaran berkelanjutan dan adaptasi adalah kunci untuk memanfaatkan kekuatan penuh dari algoritma DQN dalam prediksi harga saham.

V. KESIMPULAN

5.1 Kesimpulan

Dari penelitian yang telah dilakukan menggunakan Algoritma Deep Q-Network (DQN) berbasis Reinforcement Learning untuk prediksi harga saham, beberapa kesimpulan penting dapat diambil:

1. **Efektivitas DQN:** Algoritma DQN menunjukkan potensi yang signifikan dalam memprediksi harga saham dengan lebih akurat dibandingkan dengan metode tradisional. DQN mampu mengadaptasi dan mempelajari dinamika pasar yang kompleks, memberikan keuntungan dalam memahami dan merespons fluktuasi pasar secara real-time.
2. **Pengaruh Faktor Eksternal:** Faktor eksternal seperti kondisi ekonomi global, kebijakan pemerintah, dan sentimen pasar memainkan peran kritis dalam prediksi harga saham. Model DQN perlu terus ditingkatkan untuk memasukkan analisis ini secara lebih efektif untuk menghasilkan prediksi yang lebih akurat.
3. **Keterbatasan dan Tantangan:** Meskipun DQN menawarkan banyak keunggulan, masih ada keterbatasan seperti sensitivitas terhadap volatilitas pasar dan kebutuhan data berkualitas tinggi. Noise dan faktor random dalam data dapat mengurangi efektivitas prediksi.
4. **Peluang Pengembangan:** Terdapat peluang besar dalam pengembangan lebih lanjut dari model prediktif menggunakan DQN, terutama dengan mengintegrasikan teknologi terkini seperti AI dan machine learning untuk meningkatkan analisis dan keakuratan prediksi.

5.2 Saran

Berdasarkan hasil penelitian dan analisis yang dilakukan, beberapa saran untuk penelitian dan pengembangan lebih lanjut antara lain:

1. **Integrasi Data Lebih Luas:** Meningkatkan kinerja model dengan mengintegrasikan lebih banyak data real-time dan indikator ekonomi makro. Hal ini dapat membantu model memahami dampak faktor eksternal yang lebih luas pada harga saham.
2. **Peningkatan Teknologi dan Algoritma:** Terus menerus mengupdate algoritma dengan pendekatan-pendekatan baru dalam machine learning dan AI untuk meningkatkan keakuratan dan efisiensi model dalam memprediksi harga saham.
3. **Uji Coba dan Validasi Lebih**

Komprehensif: Melakukan uji coba yang lebih luas dengan menggunakan dataset yang lebih beragam dan periode waktu yang lebih panjang untuk menguji robustness model DQN dalam berbagai kondisi pasar.

4. **Penerapan Praktis:** Mendorong kolaborasi antara akademisi dan praktisi industri untuk menerapkan hasil penelitian ini dalam setting nyata, sehingga dapat membantu para investor dan perusahaan keuangan membuat keputusan investasi yang lebih informasi dan efektif.
5. **Edukasi dan Pelatihan:** Memberikan edukasi dan pelatihan tentang penggunaan teknologi AI dan machine learning dalam trading dan manajemen portofolio kepada profesional di industri keuangan untuk memaksimalkan penggunaan teknologi ini.

Penelitian ini menunjukkan bahwa dengan terus mengembangkan dan mengadaptasi teknologi terkini, khususnya dalam AI dan machine learning, dapat signifikan meningkatkan prediksi dan manajemen investasi di pasar saham.

ACKNOWLEDGMENT

Kami mengucapkan terima kasih kepada Universitas Multimedia Nusantara atas dukungan dan fasilitas yang diberikan selama penelitian ini. Kami juga berterima kasih kepada Ibu **Marlinda Vasty Overbeek, S.Kom, M.Kom** atas bimbingan dan saran yang berharga. Terakhir, penghargaan kami sampaikan kepada rekan-rekan dan keluarga yang telah memberikan dukungan moral dan motivasi sepanjang proses penelitian ini.

REFERENCES

- [1] Z. Zhang, S. Zohren, and S. Roberts, "Deep reinforcement learning for trading," arXiv (Cornell University), Jan. 2019, doi: 10.48550/arxiv.1911.10107.
- [2] G. Li, J. Huang, X. Li, and F. Xin, "A reinforcement learning-based portfolio management with interpretable deep neural networks," Expert Systems with Applications, vol. 177, p. 114959, Sep. 2021, doi: 10.1016/j.eswa.2021.114959.
- [3] H. F. Guo, "Stock market forecasting using machine learning techniques," Doctoral dissertation, Dept. Elect. Eng., Comput. Sci., Univ. Illinois at Chicago, Chicago, IL, USA, 2018.
- [4] V. Mnih et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529-533, Feb. 2015, doi: 10.1038/nature14236.
- [5] D. Silver et al., "Mastering the game of Go with deep neural networks and tree search," Nature, vol. 529, no. 7587, pp. 484-489, Jan. 2016.
- [6] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction. Cambridge, MA, USA: MIT Press, 2018.
- [7] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. Cambridge, MA, USA: MIT Press, 2016.
- [8] A. Maulana and A. T. Wibowo, "Prediksi Harga Saham dengan Deep NeuralNetwork dan Optimasi Bayesian," Jurnal Teknik Informatika, vol. 12, no. 2, pp. 135-144, 2019.
- [9] A. Setiawan and J. Hartono, "Analisis Perbandingan Algoritma Machine Learning untuk Prediksi Harga Saham," Jurnal Teknik Informatika dan Sistem Informasi, vol. 7, no. 2, pp. 279-291, 2021.
- [10] Z. Xu, Y. Hou, and L. Chen, "Deep reinforcement learning-based stock trading strategy for portfolio management," Applied Intelligence, vol. 51, no. 8, pp. 5557-5570, Aug. 2021, doi: 10.1007/s10489-021-02224-9.

