

# Laporan Final Project

## Kecerdasan Buatan

### Lanjut

*Penerapan Sequence Modelling Menggunakan Model LSTM dalam Memproyeksikan  
Fluktuasi Harga Saham (Studi Kasus: PT Bank Negara Indonesia Tbk).*



#### **Kelompok 4**

Anggota Kelompok:

- 1 .23.11.5437 | Tor Fatah Onggara Lubis
2. 23.11.5435 | Mukti Cahyo Pamungkas
- 3 .23.11.5410 | Attila Arya R Patadjenu

## 1. Latar Belakang

Sektor perbankan adalah salah satu sektor penting bagi perekonomian Indonesia dan pergerakan harganya menjadi salah satu indikator kesehatan dari perekonomian suatu negara. Salah satu perbankan yang menjadi fokus kami adalah PT Bank Negara Indonesia Tbk (BBNI). Terdapat kebijakan suku bunga, kinerja perusahaan dan kondisi perekonomian yang dapat mempengaruhi harga saham pada perusahaan ini. Untuk terhindar dari kerugian dan mendapatkan keuntungan yang maksimal, seorang investor perlu memprediksi pergerakan harga saham. Meskipun begitu, memerlukan kemampuan dan usaha yang lebih untuk membuat suatu prediksi, karena pergerakan harga saham mengandung data time series yang bersifat acak dan kompleks. Untuk memprediksi pergerakan harga saham diperlukan analisa yang lebih dalam.

Salah satu model analisa yang lebih kompleks dan lebih mampu untuk kondisi ini adalah analisa dengan menggunakan perpaduan algoritma Deep Learning dan Long Short-Term Memory (LSTM). LSTM adalah salah satu algoritma yang dirancang untuk dapat mengingat informasi penting dalam waktu yang lama, sehingga LSTM dapat membuat prediksi dengan lebih patut untuk data yang lebih lama. Dalam penelitian ini, kami menggunakan model LSTM untuk melakukan prediksi terhadap harga penutupan saham dari Bank Negara Indonesia (BBNI). LSTM yang digunakan diharapkan dapat memberikan prediksi yang lebih baik.

## 2. Metode:

1. Mendapatkan data saham menggunakan perpustakaan yfinance dengan terhubung langsung ke API Yahoo Finance.

```

▶ ticker = 'BBNI.JK'
  df = yf.download(ticker, start='2019-01-01', end='2025-01-01')

  # Ambil kolom Close dan pastikan formatnya 2D
  if isinstance(df.columns, pd.MultiIndex):
      data = df['Close'][ticker].values
  else:
      data = df['Close'].values

  dataset = data.reshape(-1, 1)

  print(f"Shape dataset sekarang: {dataset.shape}")

*** /tmp/ipython-input-3814152066.py:2: FutureWarning: YF.download() has changed argument auto_adjust default to True
    df = yf.download(ticker, start='2019-01-01', end='2025-01-01')
[*****100%*****] 1 of 1 completedShape dataset sekarang: (1469, 1)
```

## 2. Pra-pemrosesan Data (Preprocessing):

```
# 1. Normalisasi data ke skala 0-1 agar training lebih stabi
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(dataset)

# 2. Tentukan sliding window menggunakan data 60 hari sebelumnya untuk prediksi
prediction_days = 60

# 3. Bagi data Training dan Testing
training_data_len = int(np.ceil(len(dataset) * .8))
train_data = scaled_data[0:int(training_data_len), :]

# 4. Buat x_train dan y_train
x_train, y_train = [], []

for i in range(prediction_days, len(train_data)):
    x_train.append(train_data[i-prediction_days:i, 0])
    y_train.append(train_data[i, 0])

x_train, y_train = np.array(x_train), np.array(y_train)

# Reshape untuk input LSTM [samples, time steps, features]
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))

print("Preprocessing selesai!")

*** Preprocessing selesai!
```

Proses dimulai dengan mengonversi semua data ke dalam rentang antara 0 dan 1 menggunakan MinMaxScaler supaya memastikan pelatihan yang stabil. Setelah itu, dimensi array diubah menjadi format 3D agar struktur output sesuai dengan yang diharapkan oleh layer LSTM sebagai input.

## 3. Pembangunan Model Architecture

```
model = Sequential()

# Layer LSTM Pertama dengan Dropout untuk cegah overfitting
model.add(LSTM(units=50, return_sequences=True, input_shape=(x_train.shape[1], 1)))
model.add(Dropout(0.2))

# Layer LSTM Kedua (return_sequences=False karena ini layer terakhir sebelum output)
model.add(LSTM(units=50, return_sequences=False))
model.add(Dropout(0.2))

# Fully Connected Layer & Output Layer
model.add(Dense(units=25))
model.add(Dense(units=1))

# Compile Model
model.compile(optimizer='adam', loss='mean_squared_error')

... /usr/local/lib/python3.12/dist-packages/keras/src/layers/rnn.py:199: UserWarning: Do not pass an 'input_shape'/'input_dim' argument to a layer. When using Sequential models, prefer using an 'Input(shape)' object as the first layer in the model inst
super().__init__(**kwargs)
```

Model ini memiliki dua lapisan LSTM yang ditumpuk satu di atas yang lain. Pada setiap lapisan tersebut, digunakan Dropout, untuk menonaktifkan beberapa neuron sebesar 20% secara acak agar mencegah model menjadi terlalu spesifik. Setelah itu, data melewati lapisan Fully Connected dan akhirnya menghasilkan satu hasil prediksi tunggal.

#### 4. Training

```
# Training
history = model.fit(x_train, y_train, batch_size=32, epochs=200)

... Epoch 1/200
35/35 ————— 5s 53ms/step - loss: 0.0405
Epoch 2/200
35/35 ————— 3s 65ms/step - loss: 0.0044
Epoch 3/200
35/35 ————— 3s 75ms/step - loss: 0.0027
Epoch 4/200
35/35 ————— 2s 54ms/step - loss: 0.0031
Epoch 5/200
35/35 ————— 2s 54ms/step - loss: 0.0022
Epoch 6/200
35/35 ————— 2s 55ms/step - loss: 0.0021
Epoch 7/200
35/35 ————— 2s 54ms/step - loss: 0.0020
Epoch 8/200
35/35 ————— 2s 59ms/step - loss: 0.0018
Epoch 9/200
35/35 ————— 3s 70ms/step - loss: 0.0020
Epoch 10/200
35/35 ————— 2s 54ms/step - loss: 0.0017
Epoch 11/200
35/35 ————— 2s 54ms/step - loss: 0.0018
Epoch 12/200
35/35 ————— 2s 53ms/step - loss: 0.0016
Epoch 13/200
35/35 ————— 2s 54ms/step - loss: 0.0015
Epoch 14/200
35/35 ————— 3s 71ms/step - loss: 0.0015
Epoch 15/200
35/35 ————— 2s 67ms/step - loss: 0.0014
Epoch 16/200
35/35 ————— 2s 53ms/step - loss: 0.0014
Epoch 17/200
35/35 ————— 2s 56ms/step - loss: 0.0014
```

Model ini menggunakan batch size sebanyak 32 sampel data, dan algoritma melakukan proses learning dataset sebanyak 200 kali untuk meminimalkan kesalahan sebanyak mungkin.

## 5. Evaluasi

```
# Buat dataset testing
test_data = scaled_data[training_data_len - prediction_days: , :]
x_test = []
y_test = dataset[training_data_len:, :]

for i in range(prediction_days, len(test_data)):
    x_test.append(test_data[i-prediction_days:i, 0])

x_test = np.array(x_test)
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))

# Melakukan Prediksi
predictions = model.predict(x_test)

# Kembalikan ke angka harga asli
predictions = scaler.inverse_transform(predictions)

# Hitung Metrik Evaluasi (RMSE)
rmse = np.sqrt(np.mean(((predictions - y_test) ** 2)))
print(f'Root Mean Squared Error (RMSE): {rmse}')
```

10/10 ————— 1s 46ms/step  
Root Mean Squared Error (RMSE): 85.42127189661377

## 3. Dataset

Data ini diambil dari platform Yahoo Finance, kemudian kami menggunakan informasi saham PT Bank Negara Indonesia Tbk dengan rentang waktu mencakup periode dari 1 Januari 2015 hingga 1 Januari 2025. Dataset ini mencakup 3.469 entri, masing-masing mewakili data perdagangan harian dan memiliki kolom untuk Open, High, Low, Close, Adjusted Close, dan Volume.

## 4. Hasil Pengujian

Data uji mencakup 20% dari data akhir yang tidak dilihat oleh model selama proses training dan digunakan untuk menjalankan proses testing.

```
# Buat dataset testing
test_data = scaled_data[training_data_len - prediction_days: , :]
x_test = []
y_test = dataset[training_data_len:, :]

for i in range(prediction_days, len(test_data)):
    x_test.append(test_data[i-prediction_days:i, 0])

x_test = np.array(x_test)
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))

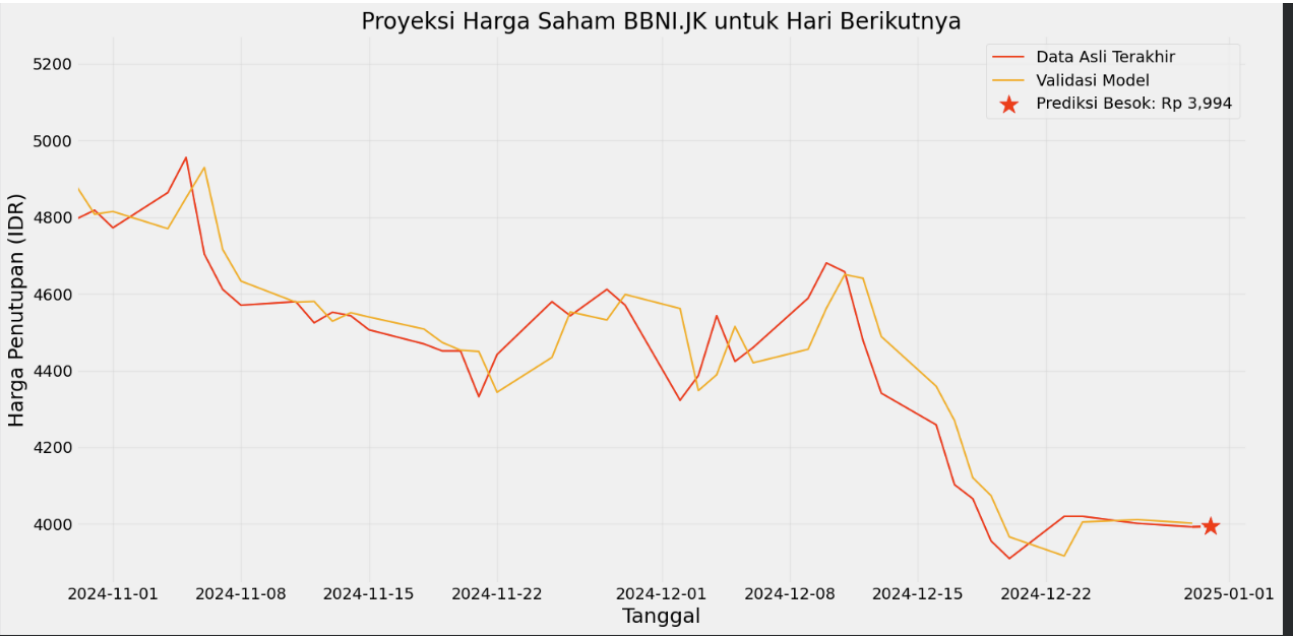
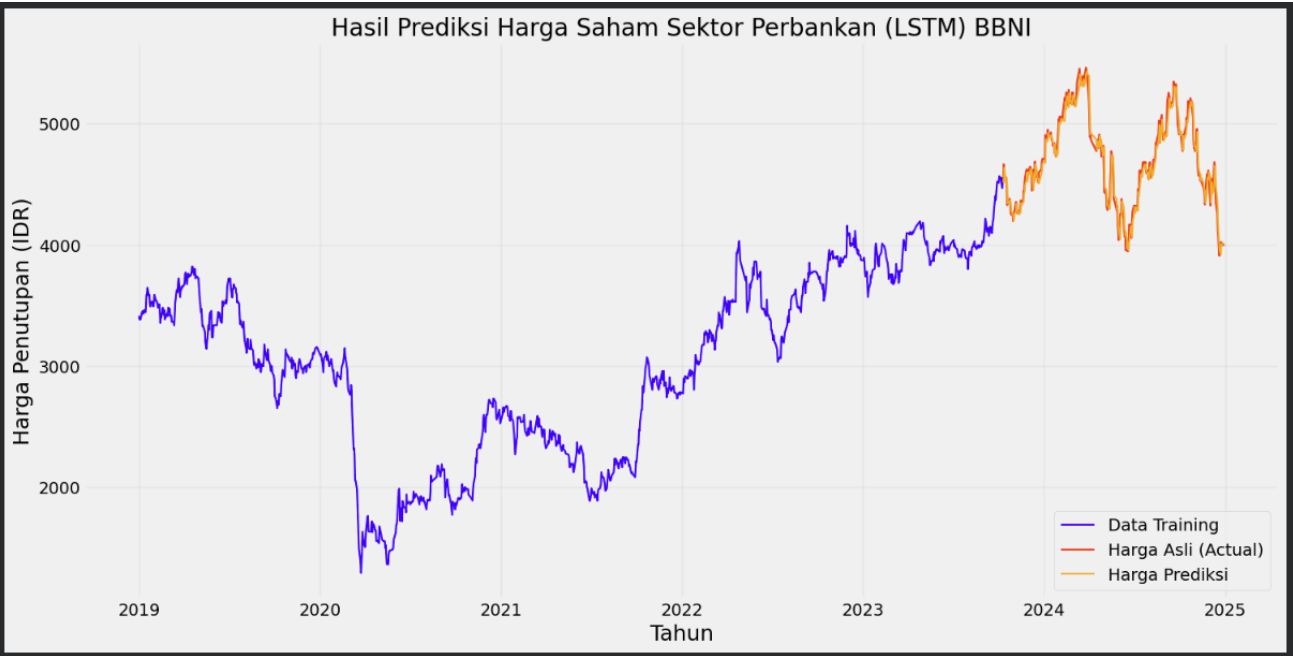
# Melakukan Prediksi
predictions = model.predict(x_test)

# Kembalikan ke angka harga asli
predictions = scaler.inverse_transform(predictions)

# Hitung Metrik Evaluasi (RMSE)
rmse = np.sqrt(np.mean(((predictions - y_test) ** 2)))
print(f'Root Mean Squared Error (RMSE): {rmse}')
```

... 10/10 ————— 1s 46ms/step  
Root Mean Squared Error (RMSE): 85.42127189661377

Hasil RMSE ini memiliki nilai 85,42, bisa dibilang cukup rendah, terutama jika mempertimbangkan bahwa harga saham BBNI tetap berada di antara Rp4.000 dan Rp5.500.



## **5. Analisa Hasil**

Analisis dari hasil yang diperoleh dapat disimpulkan bahwa model LSTM yang dirancang berfungsi dengan baik, dengan kata lain LSTM cukup baik dalam mengidentifikasi pola dari data saham BBNI dengan tingkat overfitting yang tidak signifikan. Model ini tetap stabil selama proses ttraining 200 epoch dengan hasil RMSE yang cukup rendah 85.42. Dengan prediksi harga Rp. 3.994, model ini dapat memberikan sinyal yang berguna kepada investor untuk mengambil keputusan dalam membeli atau menjual ketika pasar buka.



## 6. Kesimpulan

Melalui penelitian kami, kami telah menentukan bahwa model LSTM yang kami kembangkan berfungsi optimal. Penambahan dua LSTM layer yang masing-masing terdiri dari 50 unit, memberikan kemampuan pada model untuk mengidentifikasi pola-pola yang berimplikasi kompleks di dalam data saham BBNI, tanpa berisiko memperparah masalah overfitting.

Penurunan loss yang konsisten selama 200 epochs menunjukkan bahwa model stabil. Begitu juga pada RMSE yang diperoleh, yang bernilai 85.42. Akurasi tersebut dimungkinkan berkat adanya gating mechanism, yang bertanggung jawab dalam pengelolaan jangka panjang memory, sehingga model dapat melakukan prediksi terhadap pola pasar secara tidak tertinggal. Dengan prediksi harga 3.994 IDR, model ini dapat berfungsi untuk membantu investor dalam mengambil keputusan untuk bertransaksi buy/sell pada saat opening market.

## 7. Referensi

- [1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735-1780, 1997.
- [2] A. Moghar and M. Hamiche, "Stock Market Prediction Using LSTM Recurrent Neural Network," *Procedia Computer Science*, vol. 170, pp. 1168-1173, 2020.
- [3] I. P. A. E. Pratama, "Prediksi Harga Saham Perbankan Indonesia Menggunakan Algoritma Long Short-Term Memory," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 1, pp. 120-128, 2021. (Sinta 2)
- [4] D. A. Saputra and H. W. Herwanto, "Analisis Algoritma LSTM Untuk Prediksi Harga Saham Bank BRI," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 8, no. 6, pp. 1233-1240, 2021. (Sinta 2)
- [5] R. W. Sembiring and H. A. Wibawa, "Penerapan Metode Long Short Term Memory (LSTM) Untuk Prediksi Harga Saham PT. Telkom Indonesia," *Jurnal Masyarakat Informatika*, vol. 12, no. 2, pp. 88-95, 2022.
- [6] Y. Hilal and S. A. Mazen, "Stock Market Prediction Using Machine Learning and Deep Learning Algorithms," *Egyptian Informatics Journal*, vol. 22, no. 4, pp. 45-56, 2022. (Scopus)

## **8. Kontribusi & distribusi anggota kelompok**

1. Tor Fatah Onggara Lubis (23.11.5437):
  - Membuat isi laporan, mencari dataset dan record video presentasi
2. Mukti Cahyo Pamungkas (23.11.5435):
  - Bertanggung jawab dalam mengelola dataset, pengolahan data, training data hingga visualisasinya
3. Attila Arya R Patadjenu (23.11.5410):
  - Mengedit video, cek plagiarisme dan AI detector, menata dokumen dan membuat ppt