

Perbandingan Algoritma Temporal Convolutional Neural (TCN) dan Long Short-term Memory (LSTM) untuk Memprediksi Harga Saham Menggunakan Time Series Data

Ryan Syahriel Maulana Putra¹, Aisyah Larasati^{*2}, Titalia Trias Salsabila³, Andiny Trie Oktavia⁴

^{1,2,3,4}Program Studi Teknik Industri, Universitas Negeri Malang

e-mail: ¹ryan.syahriel.2005166@students.um.ac.id, ^{*2}aisyah.larasati.ft@um.ac.id,

³titalia.trias.2305166@students.um.ac.id, ⁴andinytrieo.ft@um.ac.id

Abstrak

Penelitian ini bertujuan untuk memberikan wawasan tambahan bagi investor yang berinvestasi di pasar modal dengan membandingkan Algoritma TCN dan LSTM dalam memprediksi data deret waktu. Data yang digunakan mencakup periode 1 Januari 2023 hingga 31 Desember 2023 dan diambil dari Yahoo Finance, dengan variabel-variabel seperti harga pembukaan, penutupan, dan volume. Proses penelitian melibatkan pembersihan data, pembagian data latih dan uji, serta pemodelan dengan pencarian parameter optimal menggunakan Hyperband. Hasil menunjukkan bahwa TCN lebih efisien dengan RSME sebesar 167.06 dan MAPE 2,58%, sementara LSTM memperoleh RMSE 467.52 dan MAPE 7,05% dengan waktu pelatihan TCN yang lebih singkat (40.8 detik) dibandingkan LSTM (252.5 detik).

Kata kunci—Saham, Prediksi, Timeseries Data, Algoritma TCN, Algoritma LSTM

1. PENDAHULUAN

Saham merupakan tanda penyertaan modal dalam suatu badan usaha pada perusahaan atau perseroan terbatas mengimplikasikan kepemilikan saham di perusahaan tersebut [1]. Penyertaan modal tersebut memiliki potensi untuk diperdagangkan di pasar saham seiring dengan perubahan harga sahamnya, yang dapat menghasilkan keuntungan dari pertumbuhan modal (*capital gain*). Selain itu, pendapatan juga dapat diperoleh melalui pembagian laba yang dihasilkan oleh perusahaan, yang sering disebut sebagai *dividen* [2]. Dalam penelitiannya, [2] juga menegaskan bahwa investor dapat berinvestasi menggunakan strategi jangka panjang maupun jangka pendek. Data statistik yang dipublikasikan oleh PT Kustodian Sentral Efek Indonesia (KSEI) pada tahun 2020 hingga bulan September 2023 menunjukkan pertumbuhan yang signifikan dalam jumlah investor di pasar modal. Pada tahun 2020 KSEI mencatat sebanyak 3.880.753 investor yang menanam modal pada pasar modal. Pada akhir tahun 2021 KSEI mencatat pertumbuhan investor menjadi 7.389.337. Kemudian, pada tahun 2022 jumlah investor meningkat menjadi 10.311.152. Hingga pada bulan September 2023, KSEI mencatat jumlah investor meningkat menjadi 11.729.977 [3].

Merujuk pada data KSEI yang mengindikasikan banyaknya investor pemula, penelitian ini bertujuan untuk memberikan instrumen tambahan bagi investor yang ingin berinvestasi di pasar saham. Instrumen alternatif ini memanfaatkan *machine learning* untuk mengolah data *timeseries*, menemukan pola yang tersembunyi, dan membantu membuat prediksi. Kemajuan teknologi dan ketersediaan informasi yang melimpah telah membuka jalan bagi pengembangan metode analisis yang lebih canggih dan efisien. Salah satu pendekatan yang populer adalah memanfaatkan algoritma dan teknik komputer [4], [5]. *Timeseries data* atau data historis

merupakan sekumpulan data tentang peristiwa atau keadaan masa lalu yang berkaitan dengan subjek tertentu [6], [7]. Dari berbagai algoritma yang dapat digunakan dalam memproses data *timeseries*, Algoritma *Temporal Convolution Neural* (TCN) dan *Long Short-Term Memory* (LSTM) menjadi algoritma yang banyak digunakan dalam memproses data *timeseries*.

TCN dan LSTM memiliki performansi yang baik dalam proses memprediksi data *timeseries* [8]. Oleh karena itu, pada penelitian ini algoritma tersebut akan dilakukan perbandingan dengan menggunakan data saham. Data saham yang digunakan adalah data PT Astra International Tbk. Proses perbandingan akan mencoba membandingkan parameter paling optimal pada masing-masing algoritma menggunakan matriks evaluasi berupa RMSE, dan MAPE.

2. METODE PENELITIAN

Tahapan dari penelitian ini dimulai dari pendahuluan, *scrapping data*, *preprocessing data*, membuat model pada masing-masing algoritma, pencarian parameter paling optimal, proses prediksi menggunakan parameter optimal, pembahasan, dan penutup. Tahap pendahuluan dimulai dengan mempelajari data dan mempelajari masing-masing algoritma. Algoritma TCN merupakan pengembangan dari Algoritma *Convolutional Neural Network* (CNN) [9]. Algoritma TCN mengintegrasikan modul *fully convolutional network*, *causal convolution*, *dilational convolutional*, dan *residual block* [10]. Dari penelitian yang dilakukan oleh [10], TCN mampu memproses *timeseries* data dengan efektif dan reliabel.

One-dimentional convolution network merupakan suatu proses konvolusi yang biasa digunakan untuk data yang memiliki satu dimensi seperti *sequential data* [11]. Pada dasarnya *convolution network* memanfaatkan beberapa data untuk menghasilkan *dot product* [11].

Dilational convolution digunakan untuk menambah “*the field of perception*” sehingga model akan mempertimbangkan lebih banyak informasi dalam data [12]. Persamaan *dilation convolution* dapat dilihat pada persamaan (1).

$$F(s) = (x * df)(s) = \sum^i = 0k - 1f(i) \cdot xs - d \cdot i \quad (1)$$

Di mana:

d = dilationn factor,

k = kernel size atau filter size, dan

s-d·i = perhitungan ke masa lalu

Hal tersebut kemudian diintegrasikan dengan *residual block*. Digunakan untuk mengurangi kompleksitas suatu model yang akan mengakibatkan model akan kehilangan banyak informasi berharga [10]. Dengan penambahan fungsi tersebut TCN memiliki tingkat efisiensi yang tinggi [10]. Dalam kesimpulan yang disebutkan oleh [10] mengatakan bahwa, seluruh model yang telah terintegrasi disebut TCN.

Algoritma LSTM merupakan penyempurnaan dari Algoritma RNN (*Recurrent Neural Network*) [13]. Pada dasarnya Algoritma LSTM memiliki dua komponen utama yaitu *internal state* dan *gates* [14]. LSTM memiliki tiga *gates* yaitu *input gates*, *output gates*, dan *forget gates*. Sedangkan, fungsi *state* memiliki fungsi seperti konveyor berjalan [14].

Persamaan dari model LSTM dapat dilihat pada persamaan (2).

$$\Gamma_{\{(i, f, o)\}} = \sigma(W_{\{(i, f, o)\}}[x^{\wedge\{< t >\}}; h^{\wedge\{< t - 1 >\}}] + b_{\{(i, f, o)\}} \quad (2)$$

Di mana:

W = Bobot pada input gate

b = Bias pada input gate

h<t-1> = Hidden state pada time step sebelumnya

Γ = Bentuk dari gate

i, f, o = Input, forget, output

Dari persamaan tersebut, jika hasil dari Γ merupakan 0 atau mendekati 0 maka value akan dihilangkan, namun jika 1 atau mendekati 1 maka *value* akan dipertahankan. *Timestep* tersebut akan bertambah seiring dengan panjang data.

Setelah mempelajari masing-masing algoritma bekerja, dilanjutkan dengan tahap *scrapping data*, pada tahapan ini akan mengambil data sekunder dari situs yahoofinance.com dengan rentang 1 Januari 2003 hingga 31 Desember 2023. Pada data tersebut terdapat 7 variabel antara lain, *Date* (Tanggal), *Open* (Harga pembukaan), *High* (Harga tertinggi), *Low* (Harga terendah), *Close* (Harga penutupan), *Adj. Close* (Harga penutupan yang telah disesuaikan), dan *Volume* (Volume penjualan). Hasil scrapping data mendapatkan data sebanyak 5222 data. Data yang didapatkan dapat dilihat pada Tabel 1.

Tabel 1. Hasil *Scrapping Data*

Date	Open	High	Low	Close	Adj Close	Volume
2003-01-01	243	243	243	243	106	0
2003-01-02	243	243	226	226	98	85700560
2003-01-03	231	247	231	243	106	111560830
2003-01-06	243	247	235	241	105	124992384
2003-01-07	241	245	231	231	101	123129047
...	...	...	...	...	...	...
2003-01-21	5600	5600	5475	5550	5550	114749900
2003-01-22	5550	5600	5500	5550	5550	38055800
2003-01-27	5550	5575	5500	5550	5550	34332400
2003-01-28	5575	5600	5550	5600	5600	31643200

Langkah selanjutnya adalah tahapan preprocessing data, dengan melakukan pemilihan variabel “*Close*”, dilanjutkan dengan *exploratory data analysis* untuk mengetahui karakteristik data.

Dilanjutkan dengan proses *scaling data* yaitu menjadikan nilai suatu data menjadi rentang 0-1 bertujuan untuk normalisasi data. Dilanjutkan dengan membagi data menjadi *training data* dan *testing data*. Tahapan selanjutnya adalah membangun model pada masing-masing algoritma, dimulai dengan penentuan rentang parameter pada masing-masing model. Penentuan parameter disetiap model dapat dilihat pada Tabel 2 dan Tabel 3.

Tabel 2. Parameter Algoritma TCN

Parameter	Nilai	Deskripsi
Dense_units	64-128	Mengontrol jumlah unit dalam lapisan dense (<i>dense layer</i>) yang digunakan dalam model TCN.
Dropout_rate	0.1-0.5	Digunakan untuk mencegah overfitting dengan cara menghapus secara acak sebagian unit selama pelatihan.
Activation	ReLU	Fungsi aktivasi digunakan untuk membantu jaringan mempelajari pola yang lebih kompleks dan tidak linear.
Epoch	100	Jumlah iterasi pelatihan yang dilakukan pada data latih.

Tabel 3. Parameter Algoritma LSTM

Parameter	Nilai	Deskripsi
Dense_units	64-128	Mengontrol jumlah unit dalam lapisan dense (<i>dense layer</i>) yang digunakan dalam model TCN.
Dropouts_rate	0.1-0.5	Digunakan untuk mencegah overfitting dengan cara menghapus secara acak sebagian unit selama pelatihan.

Parameter	Nilai	Deskripsi
Activation	ReLU	Fungsi aktivasi digunakan untuk membantu jaringan mempelajari pola yang lebih kompleks dan tidak linear.
Epoch	100	Jumlah iterasi pelatihan yang dilakukan pada data latih.

Setelah parameter selesai ditentukan, dilanjutkan dengan pencarian parameter paling optimal menggunakan *Hyperband*. *Hyperband* merupakan metode pencarian parameter yang nantinya akan mencari konfigurasi *hyperparameter* dengan meminimalisir pencarian pada konfigurasi yang tidak menghasilkan nilai yang baik [15]. *Hyperband* memiliki keunggulan dibanding dengan pencarian parameter lain yaitu waktu pencarian yang relatif singkat karena memiliki pengaturan *earlystop* ketika konfigurasi tidak menunjukkan hasil yang baik [16]. Oleh karena itu, *Hyperband* digunakan untuk mencari konfigurasi parameter terbaik dengan pengaturan seperti Tabel 4.

Tabel 4. *Setting Hyperband*

Setting	Nilai	Deskripsi
Objective	Val_loss	Menentukan metrik yang akan dioptimalkan dalam pencarian.
Max_epochs	100	Menentukan jumlah maksimal epoch.
Factor	3	Menentukan faktor untuk kemudian menentukan parameter terbaik.

Tahapan selanjutnya adalah proses prediksi menggunakan model yang telah ditentukan oleh *Hyperband*. Pembahasan hasil yang diperoleh model akan dilakukan dengan melihat hasil dari matriks evaluasi yang digunakan. Matriks evaluasi yang digunakan adalah RMSE dan MAPE untuk melihat proses pembelajaran model. RMSE dan MAPE juga digunakan untuk melihat persentase hasil prediksi yang didapatkan selama 30 hari kedepan.

3. HASIL DAN PEMBAHASAN

Hasil scrapping data sebelumnya menghasilkan 5222 data. Setelah dilakukan EDA, variable "*Close*" dipilih untuk digunakan dalam pembagian data menjadi *training data* (80%) dan *testing data* (20%), menghasilkan 4178 data untuk *training data*. Data pelatihan akan dinormalisasi menggunakan pustaka "MinMaxScaler". Selanjutnya, data akan dibagi menjadi "x_train" dan "y_train", dengan "x_train" direshape menjadi bentuk 3D, sesuai kebutuhan algoritma TCN dan LSTM.

3.1 Algoritma TCN

Selanjutnya, akan dilakukan pemodelan algoritma TCN dan LSTM. Pemodelan TCN dimulai dengan pembentukan variabel "build_tcn_model" untuk mempermudah pemanggilan model. Variabel ini berisi parameter yang digunakan pada algoritma TCN, yang dapat dilihat pada Tabel 5.

Tabel 5. *Setting Parameter TCN*

Parameter	Konfigurasi
Conv1D	Filters: [64, 96, 128], Kernel Size: [3, 5, 7], Activation: ReLU, Regularizer: L1=0.01, L2=0.01, Input Shape: (x_train.shape[1], x_train.shape[2])
MaxPooling1D	Pool Size: 2
Dense	Units: [32, 64, 96], Activation: ReLU, Regularizer: L1=0.01, L2=0.01
Dropout	Rate: [0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5]
Dense (Output)	Units: [32, 96, 128]

Parameter	Konfigurasi
Optimizer	Choices: ['adam']
Loss	Mean Squared Error
Metrics	MSE, RMSE

Setelah parameter disesuaikan, langkah selanjutnya adalah menjalankan fungsi *Hyperband* untuk mencari parameter optimal. Mengacu pada penelitian [17], variabel "*Loss*" dipilih untuk mengukur seberapa baik model dapat menggeneralisasi data yang belum pernah dilihat. Model akan dipilih berdasarkan nilai kerugian terendah. Tabel 6 menampilkan 5 model paling optimal hasil pencarian *Hyperband*.

Tabel 6. Hasil Pencarian Parameter

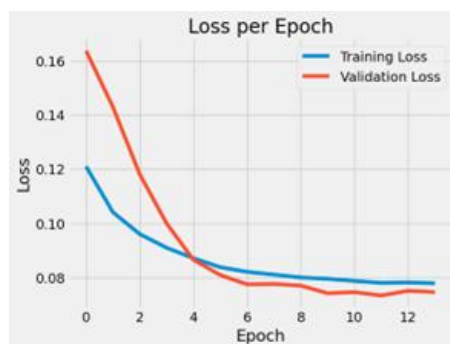
	Cony_filter	Kernel_size	Dense_unit	Dropout_rate	Optimizer	Score
best_parameter_1	64	5	32	0.1	adam	0.2160
best_parameter_2	64	3	32	0.15	adam	0.2415
best_parameter_3	64	7	32	0.25	adam	0.2695
best_parameter_4	64	7	32	0.25	adam	0.3008
best_parameter_5	64	7	32	0.15	adam	0.3124

Hyperband menemukan model paling optimal pada TCN dengan conv_filter sebesar 64, kernel_size sebesar 5, dengan dense_unit sebesar 32, dropout menggunakan 0.1, menggunakan optimizer adam. Mendapatkan score error sebesar 0.2160. Model ini akan digunakan dalam proses pelatihan data, dengan pengaturan seperti yang tertera pada Tabel 7.

Tabel 7. Pengaturan Proses Training Data

Setting	Value	Deskripsi
Epoch	100	Model akan mempelajari sebuah dataset sebanyak 100 kali
Batch Size	64	Dataset akan dibagi menjadi 64 sehingga akan mempercepat model mempelajari sebuah pola data
Callbacks	Earlystopping	Hal tersebut dilakukan untuk mencegah adanya overfitting. Dengan Pengaturan tingkat patience sebanyak 5 dan memantau metrik loss

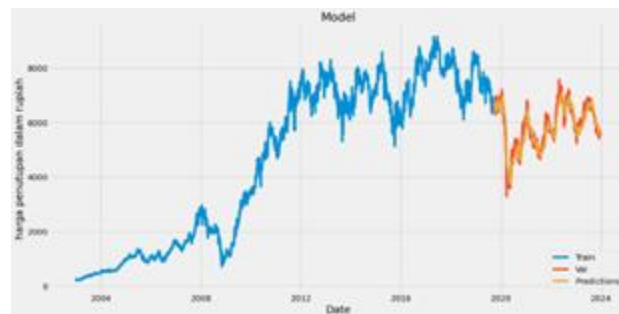
Pengaturan pada Tabel 7 mengacu pada penelitian [18], yang menyimpulkan epoch optimal adalah 100 dengan batch size 64. Penelitian lain [19] menyarankan penggunaan early stopping untuk mencegah overfitting, dengan pengaturan patience 5 kali dan memantau metrik loss. Proses pelatihan model berhenti pada epoch ke-13 karena early stopping. Grafik hasil pelatihan ditampilkan pada Gambar 1.



Gambar 1. Hasil Training Model

Pada Gambar 1, terdapat dua garis: merah untuk validation loss dan biru untuk training loss. Grafik validation loss menunjukkan penurunan signifikan dari epoch 1 hingga 5, kemudian menurun perlahan hingga epoch 13. Training loss juga menurun signifikan dari epoch 1 hingga 4, lalu menurun perlahan hingga epoch 13. Proses pelatihan berhenti pada epoch 13, menandakan bahwa callbacks tidak lagi mendeteksi penurunan pada metrik loss selama 5 epoch berturut-turut.

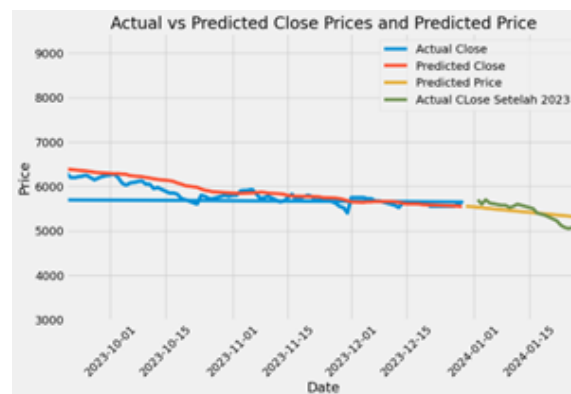
Langkah selanjutnya adalah mengimplementasikan model untuk prediksi. Pertama, test data diubah menjadi 3D, kemudian dijalankan menggunakan pengaturan pada Tabel 7. Grafik hasil pengujian ditampilkan pada Gambar 2.



Gambar 2. Hasil Prediksi Algoritma TCN

Grafik menunjukkan bahwa prediksi TCN sangat mendekati data asli, yang mengindikasikan prediksi berjalan baik. Evaluasi dilakukan dengan perhitungan RMSE dan MAPE, di mana RMSE mencapai 167,06, yang dianggap cukup baik. Mengacu pada penelitian [20], RMSE dipengaruhi oleh jumlah epoch, dengan nilai RMSE sebesar 139 pada penelitian tersebut. MAPE yang dihasilkan oleh TCN adalah 2,58%.

Langkah terakhir adalah melakukan prediksi 30 hari ke depan menggunakan fungsi loop pada Python, yang menghasilkan grafik seperti Gambar 3.



Gambar 3. Harga Aktual Dibandingkan Dengan Harga Prediksi

Pada Gambar 3, terdapat empat garis: biru untuk harga aktual hingga akhir 2023, merah untuk harga prediksi TCN hingga akhir 2023, hijau untuk harga aktual 30 hari ke depan, dan kuning untuk harga prediksi TCN selama 30 hari ke depan. Grafik menunjukkan bahwa TCN berhasil memprediksi harga dengan hasil yang cukup mendekati harga aktual. MAPE yang diperoleh sebesar 3,82%. Mengacu pada penelitian [21], jika MAPE di bawah 10%, prediksi dapat dianggap sangat baik.

3.2 Algoritma LSTM

Pada proses model LSTM memiliki langkah-langkah yang sama seperti model TCN. Pada penelitian ini, parameter LSTM diatur seperti Tabel 8.

Tabel 8. Setting Parameter LSTM

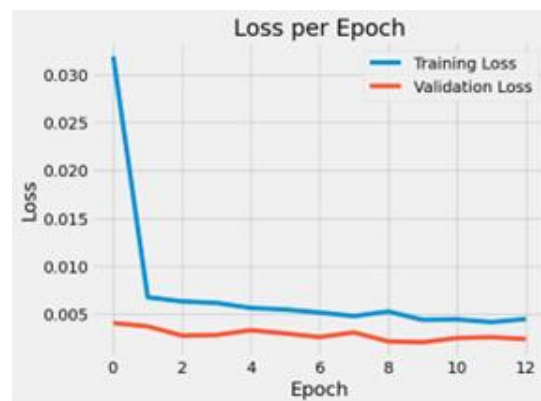
Layer	Parameters
LSTM	Units: [64, 96, 128], Regularizer: L1=0.01, L2=0.01, Input Shape: (x_train.shape[1], x_train.shape[2])
Dropout	Rate: [0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5]
Dense	Units: [32, 64, 96], Activation: tanh, Regularizer: L1=0.01, L2=0.01
Dense (Output)	Units: [64, 96, 128]
Optimizer	Choices: ['adam']
Loss	Mean Squared Error
Metrics	MSE, RMSE

Langkah selanjutnya adalah pencarian parameter optimal. Pengaturan Hyperband dapat dilihat pada Tabel 5, sehingga mendapatkan parameter optimal seperti Tabel 9.

Tabel 9. Hasil Parameter LSTM

	Best parameter 1	Best parameter 2	Best parameter 3	Best parameter 4	Best parameter 5
LSTM Layer 1	64	64	64	64	64
Drop out_1	0.2	0.3	0.3	0.3	0.5
LSTM Layer 2	96	64	64	96	64
Drop out_2	0.2	0.2	0.2	0.2	0.1
LSTM Layer 3	32	64	96	128	96
Drop out_3	0.2	0.3	0.3	0.3	0.15
Dense	96	96	96	96	96
Optimizer	Adam	Adam	Adam	Adam	Adam
Score	0.1128	0.1231	0.1233	0.1285	0.1335

Setelah parameter optimal telah ditentukan, langkah selanjutnya adalah proses training model sama seperti model TCN. Sehingga, mendapatkan grafik seperti Gambar 4.



Gambar 4. Hasil Training Model LSTM

Pada Gambar 4, terlihat dua garis: merah untuk validation loss dan biru untuk training loss. Grafik menunjukkan bahwa training loss menurun signifikan dari epoch 1 hingga 2, kemudian menurun halus hingga epoch 12. Sementara itu, validation loss mengalami penurunan yang kurang signifikan hingga epoch 12. Hal ini mengindikasikan bahwa model cukup efektif tanpa overfitting.

3.3 Analisis Model TCN

Model TCN pada penelitian ini memiliki performa yang baik pada proses training model. Hal tersebut dibuktikan dengan hasil RMSE yang relatif rendah, yaitu RMSE sebesar 167,06 dan MAPE sebesar 2,58%. Terdapat beberapa analisis yang akan dilakukan, antara lain:

1. Pada pengaturan parameter, jumlah filter merupakan faktor penting yang mempengaruhi keakuratan model [22]. Dalam penelitian ini, digunakan 64 filter, sementara penelitian [22] menunjukkan filter optimal sebanyak 200, dan 400 filter menghasilkan hasil yang kurang optimal. Hal ini mengindikasikan bahwa jumlah filter tidak harus besar, tetapi harus disesuaikan agar optimal, yang berdampak pada keakuratan model.
2. Pengaturan parameter lain yang perlu diperhatikan adalah jumlah kernel. Penelitian [23] menunjukkan bahwa kernel optimal berjumlah 3, dibandingkan dengan kernel lebih dari 3. Namun, pada penelitian ini, kernel optimal adalah 5. Jumlah kernel mempengaruhi detail data yang diproses, sehingga model ini memerlukan jumlah kernel yang cukup besar untuk mempelajari data time series.
3. Pengaturan epoch, dropout, regularizer, dense unit, dan optimizer mempengaruhi akurasi model. Penelitian [24] mencapai akurasi 80,22% dengan epoch 60, aktivasi ReLU, dropout 0,5, regularizer pada setiap layer, dense unit 64, dan optimizer Adam. Penelitian tersebut menunjukkan bahwa meskipun percobaan dilakukan 30 kali, besarnya epoch tidak menjamin akurasi lebih tinggi. Akurasi terbaik diperoleh dari kombinasi parameter yang optimal.
4. Proses pencarian parameter mempengaruhi kemampuan model menemukan parameter optimal. Metode Hyperband dirancang untuk mencari konfigurasi terbaik dengan efisien, hanya memproses parameter dengan "prospek" terbaik. Ini berbeda dengan Bayesian Optimization, yang juga memproses kandidat terbaik tetapi mencoba berulang kali [15]. Penelitian [15] menyimpulkan bahwa pada dataset berjumlah 117, Hyperband mengungguli RandomSearch dalam error test meskipun kinerjanya lebih buruk dalam validation error. Metode Bayesian mengungguli Hyperband dan RandomSearch dalam error test, namun menunjukkan tanda-tanda overfitting. Pada dataset berjumlah 21, Hyperband memberikan hasil lebih baik dibandingkan model lainnya. Meskipun pencarian parameter menggunakan Hyperband tidak seoptimal gridsearch, yang mencoba seluruh kombinasi parameter, Hyperband unggul dalam hal waktu komputasi [15].
5. Performa model TCN menunjukkan bahwa TCN dapat memprediksi dengan sangat baik, terbukti dengan MAPE sebesar 3,677%

3.4 Analisis Model LSTM

Model LSTM pada penelitian ini memiliki performa yang cukup baik pada proses training model. Hal tersebut dibuktikan dengan hasil RMSE sebesar 467,52 dan MAPE sebesar 7,05%. Terdapat beberapa analisis yang akan dilakukan, antara lain:

1. Pada pengaturan parameter, salah satu hal yang perlu diperhatikan adalah jumlah units. Penelitian [13] dan [20] menggunakan 50 units pada setiap lapisannya. Sementara itu, penelitian [25] menemukan bahwa jumlah units optimal adalah 64. Pada penelitian ini, jumlah units bervariasi di setiap lapisan: lapisan pertama 64 units, lapisan kedua 96 units, lapisan ketiga 32 units, dan 1 unit dense sebesar 96. Pengaturan ini tentu mempengaruhi hasil prediksi yang dilakukan oleh model.

2. Penggunaan dropout juga perlu diperhatikan. Penelitian [13] dan [25] menggunakan dropout sebesar 0,2, yang juga merupakan nilai optimal yang ditemukan oleh Hyperband. Dropout digunakan untuk mencegah model bergantung pada unit tertentu, sehingga menghindari overfitting.
3. Proses pencarian parameter juga mempengaruhi hasil akhir sebuah model dan perlu dipertimbangkan dengan baik.
4. Performa model LSTM menunjukkan bahwa LSTM dapat memprediksi dengan sangat baik, terbukti dengan MAPE sebesar 2,84%.

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, berikut merupakan kesimpulan dari hasil yang didapatkan:

1. *Algoritma TCN:*

Parameter optimal pada algoritma TCN untuk memprediksi harga saham ditemukan dengan conv_filter 64, kernel_size 5, dense_unit 32, dropout 0,1, dan optimizer Adam, menghasilkan skor 0,2160. TCN unggul dalam hal efisiensi waktu dan performa prediksi. Waktu proses pelatihan TCN jauh lebih cepat dibandingkan LSTM, yaitu 40,8 detik berbanding 252,5 detik. Meskipun MAPE pada proses pelatihan sama (3,82%), TCN memberikan prediksi yang lebih akurat dengan MAPE 2,58%. Selain itu, nilai RMSE pada TCN untuk prediksi lebih rendah, yaitu 167,06, yang menunjukkan kesalahan prediksi yang lebih kecil dibandingkan LSTM.

2. *Algoritma LSTM*

Parameter optimal untuk LSTM menggunakan empat lapisan: lapisan pertama dengan 64 unit dan dropout 0,2, lapisan kedua dengan 96 unit dan dropout 0,2, serta lapisan terakhir dengan 32 unit dan dropout 0,2. Output lapisan terakhir diikuti oleh Dense unit sebesar 96. Model LSTM menggunakan aktivasi tanh dan optimizer Adam. LSTM memerlukan waktu pelatihan lebih lama dibandingkan TCN, yaitu 252,5 detik. Meskipun MAPE pada pelatihan sama dengan TCN (3,82%), performa prediksi LSTM menurun signifikan dengan MAPE 7,05%, yang menunjukkan prediksi yang kurang akurat. Nilai RMSE LSTM pada prediksi juga lebih tinggi, yaitu 467,52, menandakan tingkat kesalahan yang lebih besar dibandingkan TCN.

3. *Kinerja Algoritma TCN dan LSTM*

Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa algoritma Temporal Convolutional Network (TCN) lebih unggul dibandingkan Long Short-Term Memory (LSTM) dalam memprediksi harga saham PT. Astra International, baik dari segi efisiensi waktu maupun akurasi prediksi. TCN lebih cepat dalam proses pelatihan dan menghasilkan prediksi yang lebih akurat, dengan MAPE dan RMSE yang lebih rendah dibandingkan LSTM. Meskipun LSTM memiliki MAPE yang sama pada tahap pelatihan, performanya menurun signifikan pada proses prediksi. Oleh karena itu, TCN lebih efektif untuk aplikasi prediksi harga saham dalam penelitian ini.

UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih kepada Universitas Negeri Malang yang telah memberikan dukungan pada pelaksanaan penelitian ini melalui pendanaan penelitian internal UM dengan nomer kontrak Nomor: 24.2.527/UN32.14.1/LT/2025.

DAFTAR PUSTAKA

- [1] BEI, "Saham," 2023. [Online]. Available: <https://www.idx.co.id/id/produk/saham>. [Accessed: Sep. 11, 2023].
- [2] Juniarto, "Analisis teknikal terhadap pergerakan harga saham subsektor perbankan di LQ45 (dengan pendekatan candlestick, rasio Fibonacci, dan moving average convergence divergence (MACD))," 2022. DOI: 10.29313/v7i1.27026.
- [3] KSEI, "Kustodian Efek Sentral Indonesia," 2023. [Online]. Available: https://www.ksei.co.id/archive_download/holding_composition. [Accessed: Oct. 1, 2023].
- [4] D. Ismiyana and M. Y. Putra, "Komparasi algoritma dalam memprediksi perubahan harga saham GOTO menggunakan Rapidminer," *Jurnal Komputer Indonesia*, vol. 11, no. 1, 2023. DOI: 10.31294/jki.v11i1.16153.
- [5] A. Larasati, A. M. Hajji, Y. W. Chen, E. Anandya, A. T. Wahyu, and V. E. B. Darmawan, "Building regression model to estimate NOX emission pollutant of backhoe equipment," in *AIP Conf. Proc.*, vol. 2227, no. 1, 2020.
- [6] K. Hanna and I. Wigmore, "Historical Data," 2022. [Online]. Available: <https://www.techtarget.com/whatis/definition/historical-data>. [Accessed: Oct. 15, 2023].
- [7] A. Larasati, Y. W. Chen, A. M. Hajji, A. F. P. Mahardika, and V. E. B. Darmawan, "Determining predictor variables of HC, CO, and CO₂ emissions using decision tree models," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 669, no. 1, p. 012032, 2019.
- [8] Z. He, X. Wang, and C. Li, "A time series intrusion detection method based on SSAE, TCN and Bi-LSTM," *Computers, Materials and Continua*, vol. 78, no. 1, pp. 845–871, 2024.
- [9] Y. Chen, Y. Kang, Y. Chen, and Z. Wang, "Probabilistic forecasting with temporal convolutional neural network," *Neurocomputing*, vol. 399, pp. 491–501, 2020.
- [10] J. Yao, Z. Cai, Z. Qian, and B. Yang, "A novel approach based on TCN-LSTM network for predicting waterlogging depth with waterlogging monitoring station," *PLOS ONE*, vol. 18, no. 10, 2023. DOI: <https://doi.org/10.1371/journal.pone.0286821>.
- [11] F. Lässig, "Temporal Convolutional Networks and Forecasting," 2020. [Online]. Available: <https://medium.com/unit8-machine-learning-publication/temporal-convolutional-networks-and-forecasting-5ce1b6e97ce4>. [Accessed: Nov. 20, 2023]
- [12] R. Wan, S. Mei, J. Wang, M. Liu, and F. Yang, "Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting," *Electronics*, vol. 8, no. 8, p. 876, 2019. DOI: <https://doi.org/10.3390/electronics8080876>.
- [13] A. Satyo, J. K. Noer Ali, and K. Bekasi, "Prediksi data time series saham Bank BRI dengan mesin belajar LSTM (Long Short-Term Memory)," *J. Inf. and Inf. Security (JIFORTY)*, vol. 1, no. 1, pp. 1–8, 2020. DOI: 10.31599/jiforty.v1i1.133.
- [14] S. Ashraf, "Introduction to Sequence Learning Models: RNN, LSTM, GRU," 2021. [Online]. Available: <https://doi.org/10.13140/RG.2.2.36370.99522>.

- [15] L. Li, K. Jamieson, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *J. Mach. Learn. Res.*, vol. 18, 2018. [Online]. Available: <http://jmlr.org/papers/v18/16-558.html>.
- [16] I. Mahesastra, I. Darmawan, and I. Kadyanan, "Klasifikasi gambar dalam dan luar ruangan dengan metode CNN dan Hyperband," vol. 1, no. 3, 2023.
- [17] J. P. Mohanty and S. Dash, "Forecasting energy consumption using hybrid CNN and LSTM auto-encoder network with Hyperband optimization," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 10, pp. 1041–1055, 2022. DOI: <https://doi.org/10.22214/ijraset.2022.47136>.
- [18] A. Wulandari, "Analisis kinerja algoritma CNN dan LSTM untuk memprediksi tinggi muka air di DKI Jakarta, Indonesia," 2022. [Online]. Available: <http://digilib.mercubuana.ac.id/>. [Accessed: Feb. 3, 2023].
- [19] O. Ulfah, H. Ricky, A. Alfian, and W. Galih, "Klasifikasi penyakit padi berdasarkan citra daun menggunakan model terlatih ResNet101," *J. RESTI*, vol. 5, no. 6, pp. 1216–1222, 2021. DOI: <https://doi.org/10.29207/resti.v5i6.3607>.
- [20] R. Julian and M. R. Pribadi, "Peramalan harga saham pertambangan pada Bursa Efek Indonesia (BEI) menggunakan Long Short-Term Memory (LSTM)," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 8, no. 3, 2021. [Online]. Available: <http://jurnal.mdp.ac.id>.
- [21] I. Nabillah and I. Ranggadara, "Mean absolute percentage error untuk evaluasi hasil prediksi komoditas laut," *JOINS*, vol. 5, no. 2, pp. 250–255, 2020. DOI: <https://doi.org/10.33633/joins.v5i2.3900>
- [22] I. G. L. Yudha, N. A. Sanjaya ER, A. A. I. N. E. Karyawati, I. B. G. Dwidasmara, I. G. N. A. C. Putra, and I. B. M. Mahendra, "Article classification using convolutional neural network (CNN) and chi-square feature selection," *JELIKU*, vol. 11, no. 3, p. 529, 2022. DOI: <https://doi.org/10.24843/JLK.2023.v11.i03.p08>.
- [23] L. Listyalina, "Penentuan kombinasi kernel terbaik menggunakan median filter," *Teknoin*, vol. 22, 2016.
- [24] Y. Hidayat, E. Erwin, and D. Handayani, "Penerapan 1D-CNN untuk analisis sentimen ulasan produk kosmetik berdasar Female Daily Review," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 8, no. 3, pp. 153–163, 2023. DOI: [10.25077/TEKNOSI.v8i3.2022.153-163](https://doi.org/10.25077/TEKNOSI.v8i3.2022.153-163).
- [25] Y. Findawati, U. Indahyanti, Y. Rahmawati, dan R. Puspitasari, "Sentiment Analysis of Potential Presidential Candidates 2024: A Twitter-Based Study," *Academia Open*, vol. 8, no. 1, 2023. DOI: [10.21070/acopen.8.2023.7138](https://doi.org/10.21070/acopen.8.2023.7138).