Seminar Nasional Sains Data 2025 (SENADA 2025)
UPN "Veteran" Jawa Timur
E-ISSN 2808-5841
P-ISSN 2808-7283

Analisis Feasibilitas Model CNN untuk Prediksi Aksi pada Pasar Saham di Indonesia

Aryadharma Wibowo[1], Rudy Kusdiantara[2], Nuning Nuraini[3]
[1,2,3]*Program Studi Sarjana Matematika, Institut Teknologi Bandung*
[1]aryabelajarterus@gmail.com
[2]rudy_kusdiantara@itb.ac.id
[3]nunnura@itb.ac.id
Corresponding author email: aryabelajarterus@gmail.com

Abstract: *Many researchers have attempted to predict future prices using regression. However, this regression-based approach often yields delayed price predictions relative to the actual price, making the results unreliable. Instead of viewing this as a regression task, we propose a classification approach to predict the best action: sell, buy, or hold. Inspired by previous research, this article aims to build a CNN model that can generalize phenomena, avoiding both overfitting and underfitting. We also evaluate the model's feasibility in real-world scenarios. This study utilized stock prices from BBCA, BBRI, BMRI, and BBNI. Results indicate that the CNN model successfully learned, rather than merely memorizing patterns, as evidenced by decreasing loss and increasing accuracy on the training data. Furthermore, the CNN model demonstrated a limited ability to generalize, proven by decreasing validation loss and a Macro Average F1-Score consistently higher than that of a naive model. However, the model's accuracy remains unstable and Sharpe Ratio values stay below 0.1 across all trading simulations. These findings show that building an artificial intelligence model to generate daily profits consistently is highly challenging. This study hopes to raise public awareness that claims of guaranteed profit from AI-based systems or robot trading should be viewed critically to avoid falling into unreliable investment schemes.*
Keywords: *Convolutional Neural Network (CNN), F1-Score, Financial Simulation, Stock Price Prediction, Classification of a Trading Action*
Abstrak: Banyak peneliti berlomba-lomba melakukan regresi harga saham di masa depan. Namun, pendekatan regresi seperti ini sering kali hanya menghasilkan prediksi yang tertunda dibanding harga aslinya, sehingga kurang dapat diandalkan. Sebagai alternatif, penelitian ini memandangnya sebagai masalah klasifikasi untuk memprediksi aksi terbaik antara jual, beli, atau tahan. Dengan data saham BBCA, BBRI, BMRI, dan BBNI, dibangunlah model Convolutional Neural Network (CNN) yang dirancang untuk mampu melakukan generalisasi dan menghindari overfitting. Hasil menunjukkan bahwa CNN berhasil belajar dari data latih, dibuktikan dengan penurunan kerugian dan peningkatan akurasi. Model juga mampu melakukan generalisasi sampai titik tertentu, ditunjukkan oleh nilai Macro Average F1-Score pada data uji yang konsisten lebih tinggi dibanding baseline prediktor kelas mayoritas. Namun, akurasi model pada data validasi tetap fluktuatif, dan nilai Sharpe Ratio pada simulasi jual-beli seluruh saham berada di bawah 0.1. Hal ini menunjukkan bahwa meskipun model dapat menghasilkan keuntungan secara terbatas, tingkat risikonya terlalu tinggi. Penelitian ini memberikan wawasan bahwa membangun sistem otomatis seperti robot trading yang menjanjikan keuntungan harian secara konsisten adalah tugas yang kompleks dan tidak selalu layak diterapkan dalam dunia nyata.
Kata kunci: *Convolutional Neural Network* (CNN), *Macro Average F1-Score*, Simulasi Finansial, Prediksi pasar saham, Klasifikasi Tindakan *Trading*

I. PENDAHULUAN

Kecerdasan buatan dalam satu dekade ini telah berkembang secara pesat dan terbukti berjalan dengan baik dalam berbagai macam aplikasi. Misalnya pada pengenalan citra wajah atau pada asisten virtual berbasis suara. Namun, ada satu bidang yang cukup menantang untuk diteliti, yakni: penerapannya dalam dunia finansial khususnya dunia pasar keuangan. Banyak peneliti yang mencoba untuk melakukan prediksi harga menggunakan algoritma pembelajaran mendalam. Parameter keberhasilan model untuk masalah ini biasanya adalah *mean squared error, mean absolute error* dan sejenisnya yang dihitung dengan membandingkan seberapa berbeda harga yang sebenarnya dan harga yang diprediksi. Sayangnya, cara seperti ini sangat riskan untuk *overfitting* meskipun *plot* kurva kerugian dan akurasi terhadap *epoch* menyatakan tidak. Bagaimana jika hasil prediksi harga yang ada

itu hanyalah merupakan lag hari ke-i dari harga sebenarnya? Tentu model seperti ini menjadi tidak dapat diandalkan karena dengan "prediksi yang telat" seperti ini, kita bisa saja membeli instrumen investasi tersebut disaat instrumen sudah terindikasi *overbuy* atau menjual innstrumen tersebut disaat instrumen sudah terindikasi *oversell*.

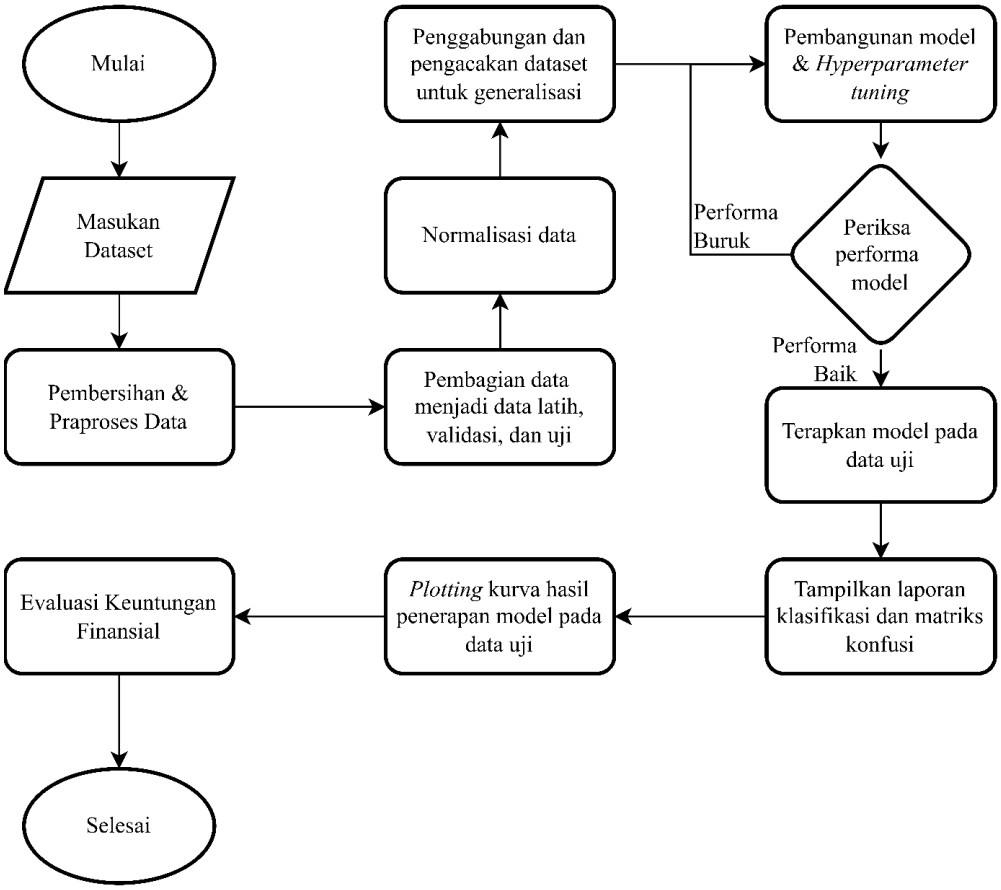
Oleh karena-nya muncul sudut pandang baru dari berbagai peneliti mengenai masalah ini. Alih-alih memandang masalah ini sebagai masalah regresi, mengapa tidak memandang masalah ini sebagai masalah klasifikasi. Idenya adalah berdasarkan data harga historis, akan diprediksi posisi terbaik yang dapat kita lakukan pada kemudian hari. Langkah ini bisa berupa beli, jual, dan tahan posisi terhadap instrumen investasi tersebut.

Sezer, O.B. *et al.* dalam artikelnya yang berjudul *Algorithmic Financial Trading with Deep Convolutional Neural Network: Time Series to Image Conversion Approach* [1] mengusulkan sudut pandang baru untuk menyelesaikan masalah klasifikasi ini. Dalam artikelnya, Sezer menggunakan analisis teknikal untuk membangun matriks dua dimensi, dimana matriks ini akan menjadi masukan untuk model jaringan konvolusinya. Fischer, C. *et al.* dalam artikelnya yang berjudul *Deep Learning with long-term short memory network for financial market predictions* [2] mencoba untuk melakukan generalisasi terhadap kurang lebih 500 saham dari rentang tahun 1995-2015 dengan cara mengacak data latih yang dikumpulkan dari berbagai macam saham untuk tujuan generalisasi yang lebih baik.

Terinspirasi dari artikel ini, penulis mencoba untuk mengeksplorasi terkait masalah ini. Penulis akan membahas hal-hal yang cukup penting dalam pelatihan algoritma pembelajaran mendalam yang belum dibahas oleh Sezer, O.B. *et al.,* seperti plot kurva kerugian dan akurasi terhadap *epoch*, apakah model dapat melakukan generalisasi serta penulis akan menguji apakah model ini sebenarnya realistis untuk diterapkan dalam dunia nyata khususnya di Indonesia melalui simulasi jual-beli. Dalam eksplorasi kali ini, penulis akan menggunakan data harga saham dari empat perusahaan perbankan yang memiliki *market cap* terbesar dan likuiditas terbaik di Indonesia, yakni: BBCA (Bank BCA), BBRI (Bank BRI), BMRI (Bank Mandiri), dan BBNI (Bank BNI).

Lebih jauh lagi, penting untuk memahami bahwa pembuatan model CNN untuk melakukan tugas klasifikasi seperti ini sebenarnya sama saja dengan mencoba untuk membangun *robot trading*, yakni suatu sistem kecerdasan buatan yang dapat mengetahui kapan waktu yang terbaik untuk melakukan jual, beli, ataupun tahan pada suatu instrumen investasi tertentu. Hasil dari penelitian ini akan memiliki implikasi besar terhadap pengambilan keputusan investasi oleh masyarakat berhubung dalam kurung 5 tahun terakhir, banyak masyarakat yang tergiur dan terjebak dalam skema investasi bodong. Oleh karena itu, penelitian ini tidak hanya mengevaluasi performa model dari sisi metrik teknis, tetapi juga meninjau kelayakan model untuk diterapkan dalam praktik nyata, guna mencegah ekspektasi berlebihan terhadap kemampuan kecerdasan buatan dalam menghasilkan keuntungan instan.

II. METODE PENELITIAN



Gambar 1. Diagram alir metode penelitian

Gambar di atas menunjukkan proses eksplorasi yang dilakukan dalam peneliti. Adapun penjelasan dari masing masing proses adalah sebagai berikut:

2.1. Data

Data yang digunakan adalah data historis harga saham perusahaan perbankan di Indonesia yang terdiri dari: BBCA (Bank BCA), BBRI (Bank BRI), BMRI (Bank Mandiri), BBNI (Bank BNI). Alasan penulis memilih saham dari perusahaan ini adalah karena saham perusahaan ini memiliki *market cap* yang besar dan likuiditas yang baik untuk pasar saham Indonesia. Penulis mengambil data ini dari *yahoo finance* dengan fitur-fitur sebagai berikut:

- Data BBCA, BBRI, BMRI dan BBNI berturut-turut diambil sejak tanggal 8 Juni 2004, 10 November 2003, 14 Juli 2003, 23 Desember 2002 sampai dengan 20 Mei 2025.
- Fitur-fitur pada data yang terambil diantaranya adalah: *Date, Close, High, Low, Open,* dan *Volume*.

2.2. Pembersihan, praproses, pembagian, normalisasi, dan pengacakan data

Keempat data saham ini yang berasal dari *yahoo finance* ini sebenarnya umumnya sudah bersih. Namun, penulis tetap melakukan pembersihan data pada tanggal-tanggal dimana pasar saham Indonesia tutup. Setelah diteliti terbukti bahwa pada saat dimana pasar saham tutup, *yahoo finance* tetap mencatat harga dari saham tersebut, hanya saja entri pada kolom *Volume*-nya tercatat nol saja. Baris baris seperti

ini yang akan dibuang karena untuk keperluan perhitungan indikator analisis teknisnya pada tahapan praproses.

Selanjutnya, tahapan praproses data dilakukan. Penulis menggunakan pustaka technical-analysis, pustaka ta dan pustaka pandas_ta untuk membantu proses perhitungan indikator analisis teknikal. Adapun indikator analisis teknikal yang digunakan diberikan oleh tabel berikut:

Tabel 1. Indikator Analisis Teknikal yang digunakan

15 Indikator Analisis Teknikal yang digunakan	
Nama Indikator Analisis Teknis	*Variabel yang digunakan di python*
Relative Strength Index	*rsi*
Williams %R	*willr*
Weighted Moving Average	*wma*
Exponential Moving Average	*ema*
Simple Moving Average	*sma*
Hull Moving Average	*hma*
Triple Exponential Moving Average	*tema*
Commodity Channel Index	*cci*
Change Momentum Oscillator Index	*cmo*
Moving Average Convergence and Divergence	*macd*
Percentage Price Oscillator	*ppo*
Rate of Change	*roc*
Chaikin Money Flow Indicator	*cmf*
Directional Movement Indicator	*adx*
Parabolic SAR	*psar*

Rentang hari yang digunakan dalam perhitungan indikator analisis teknikal ini adalah pada rentang 6-20 hari. Sehingga tiap tiap indikator memiliki 15 variasi. Sehingga totalnya sejumlah $15 \times 15 = 225$ fitur. Selanjutnya, karena fokus artikel ini bukan mengupas tentang indikator analisis teknikal, maka hanya akan diberikan satu contoh perhitungan dari indikator analisis teknikal ini. Penulis akan memilih indikator analisis teknikal berupa *Simple Moving Average*. Ilustrasinya adalah sebgai berikut:

Misal data historis dari harga *close* sejumlah T hari yang terdefinisi pada suatu deret $\{P_t\}\ \forall t \in [1, T]$. Maka sma dengan periode n pada waktu t didefinisikan sebagai:

$$sma_n(t) = \frac{1}{n}\sum_{k=0}^{n-1} P_{t-k}$$

Pada eksplorasi kali ini, nilai t akan akan selalu statis dan sama dengan t. Sedangkan n akan divariasikan dengan rentang nilai 6-20. Setelah kita menghitung indikator analisis teknis ini, pada data latih, kita dapatkan rentang hasil sebagai berikut:

Pada akhirnya kita akan dapatkan nilai $sma_6(6), sma_7(7), \ldots, sma_{20}(20)$ dimana untuk simplifikasi, nilai-nilai ini akan langsung saja direpresentasikan dengan variabel $sma_6, sma_7, \ldots, sma_{20}$. Fitur-fitur inilah yang nanti akan kita bentuk menjadi matriks dua dimensi. Ilustrasinya adalah sebagai berikut:

$$\begin{bmatrix} rsi_6 & rsi_7 & \cdots & \cdots & rsi_{20} \\ willr_6 & willr_7 & \cdots & \cdots & willr_{20} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ sma_6 & sma_7 & \cdots & \ddots & sma_{20} \\ psar_6 & psar_7 & \cdots & \cdots & psar_{20} \end{bmatrix}_{15\times15}$$

Setelah didapat matriks seperti di atas, kita akan lakukan normalisasi pada tiap tiap fitur. Penulis menggunakan normalisasi skala atau *min-max*. Ilustrasi dari normalisasi fitur ini dapat dilihat pada ilustrasi berikut:

$$\begin{bmatrix} 0-100 & 0-100 & \cdots & \cdots & 13.55-88.88 \\ 0-1 & 0-1 & \cdots & \cdots & 0-1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 115.15-4779.49 & 115.34-4775.71 & \cdots & \ddots & 116.96-4731.93 \\ 113.29-4942.08 & 113.29-4942.08 & \cdots & \cdots & 113.29-4942.08 \end{bmatrix}_{15\times15}$$

$$\downarrow$$

$$\begin{bmatrix} 0-1 & 0-1 & \cdots & \cdots & 0-1 \\ 0-1 & 0-1 & \cdots & \cdots & 0-1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0-1 & 0-1 & \cdots & \ddots & 0-1 \\ 0-1 & 0-1 & \cdots & \cdots & 0-1 \end{bmatrix}_{15\times15}$$

Hal yang patut diperhatikan adalah dalam langlah normalisasi ini kita hanya melakukan normalisasi dengan rujukan nilai minimum dan maksimum pada data latih untuk menghindari kebocoran informasi yang dapat membuat model pada data lati seolah-olah menyontek ke data validasi ataupun uji. Matriks dua dimensi seperti di atas yang nanti akan menjadi masukan bagi arsitektur CNN kita.

Selanjutnya, karena tidak ada label aksi pada data mentah, terlebih dahulu akan dilakukan proses pelabelan untuk mengkategorikan suatu hari tertentu masuk ke dalam kategori jual, beli, atau tahan. Proses pelabelan akan mengikuti algoritma berikut:

Algoritma 1. Pelabelan Harga

Prosedur pelabelan_harga(df, kolom_harga = "close", ukuran_jendela)
masukan dari pengguna : df, kolom_harga, ukuran_jendela
  jika ukuran_jendela adalah bilangan genap:
    tampilkan error: "ukuran jendela harus bilangan ganjil"
price_series ← ambil deret harga dari kolom_harga dalam df
buat array labels dengan panjang sama seperti price_series, inisialisasi nilai awan dengan NaN
untuk counter dari ukuran_jendela hingga panjang price_series-1 lakukan:
  awal_jendela ← counter – ukuran_jendela
  akhir_jendela ← awal_jendela + ukuran_jendela
  window ← potongan harga dari indeks awal_jendela hingga
      akhir_jendela
  tengah ← floor(ukuran_jendela dibagi 2) + 1
  indeks_min ← indeks harga terkecil dalam window
  indeks_maks ← indeks harga terbesar dalam window
  jika indeks_maks = tengah:

```
14:             labels[counter]← 0 //label: jual
15:         jika indeks_maks != tengah dan indeks_min = tengah:
16:             labels[counter]← 1 // label: beli
17:         jika lainnya:
18:             labels[counter]← 2 // label: tahan
```

Pada eksperimen kali ini, penulis menggunakan ukuran jendela sebesar 29 hari. Setelah dilakukan pemberian label, berarti sejauh ini kita punyai 225 kolom tambahan dari perhitungan indikator analisis teknikal dan label sebagai terget yang akan kita prediksi. Tahap selanjutnya adalah membuang baris yang kolom-kolomnya mengandung entri kosong dengan metode *dropna()*.

Pada akhirnya, kita punyai *dataframe* bersih yang mengandung 225 fitur yang berada di dalam matriks 2 dimensi berukuran 15×15 dan 1 target aksi. Selanjutnya, akan dilakukan tahapan yang umum dilakukan dalam praktik persiapan pelatihan pembelajaran mendalam, yakni: memisahkan data menjadi data latih, validasi, dan uji. Setelah dilakukan pembagian, data latih dan validasi dari keempat saham ini akan diacak sedemikian rupa sehingga kita memiliki suatu dataset latihan gabungan yang siap untuk dilatih. Ada satu hal yang perlu diperhatikan dalam melakukan pengacakan dataset gabungan ini, yakni: inisialisasi *seed* agar reprodusibilitas dapat terjaga. Selanjutnya, perhatikan bahwa kita tidak mengacak data uji dikarenakan urutan data uji mennjadi penting karena data uji akan digunakan untuk pengujian performa model CNN hasil pelatihan.

Sebelum mengaplikasikan algoritma di atas pada data yang dimiliki, penulis membagi data menjadi 3 bagian, yakni: sebanyak 70% menjadi data latih, 15% menjadi data validasi, dan menjadi data 15% uji. Perlu diketahui bahwa terjadi ketidak seimbangan data dari hasil pengaplikasian algoritma pelabelan ini. Adapun hasil dari ketidakseimbangan ini data dapat dilihat sebagai berikut:

Tabel 2. Hasil pengaplikasian algoritma pelabelan

Dataset	Label	Data Latih	Data Validasi	Data Uji
Saham BBCA	Jual	84	18	19
	Beli	76	17	18
	Tahan	3339	714	714
Sahan BBRI	Jual	86	18	16
	Beli	75	15	15
	Tahan	3432	737	740
Saham BMRI	Jual	82	20	17
	Beli	71	17	14
	Tahan	3498	745	753
Saham BBNI	Jual	77	22	17
	Beli	76	17	17
	Tahan	3408	724	730

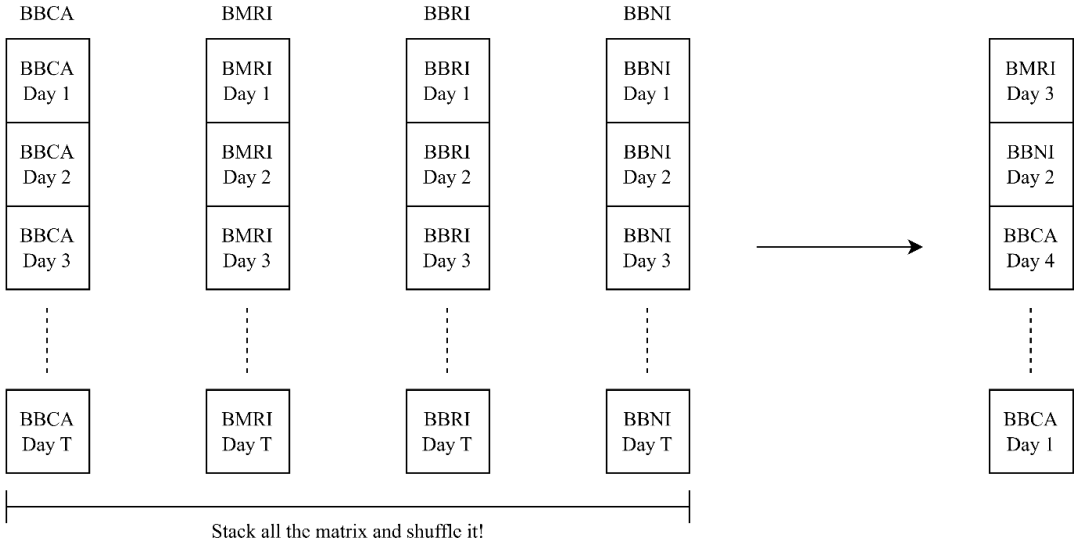
Karena terjadi ketidakseimbangan pada data, maka akan dilakukan penanganan lebih lanjut secara khusus. Penanganan ketidakseimbangan pada data ini hanya dilakukan pada data latih untuk menghindari *leakage of info* atau menghindari model menyontek pada saat validasi dan proses uji berlangsung. Misalkan n adalah total sampel, n_c, jumlah sampel untuk kelas c, C adalah banyaknya variasi dari c, dan y_i adalah label ke-i, maka bobot untuk sampel ke-i diberikan oleh:

$$w_i = \frac{n}{C \cdot n_{y_i}} = \frac{1}{f_{y_i}} \cdot \frac{1}{C}$$

Dimana:

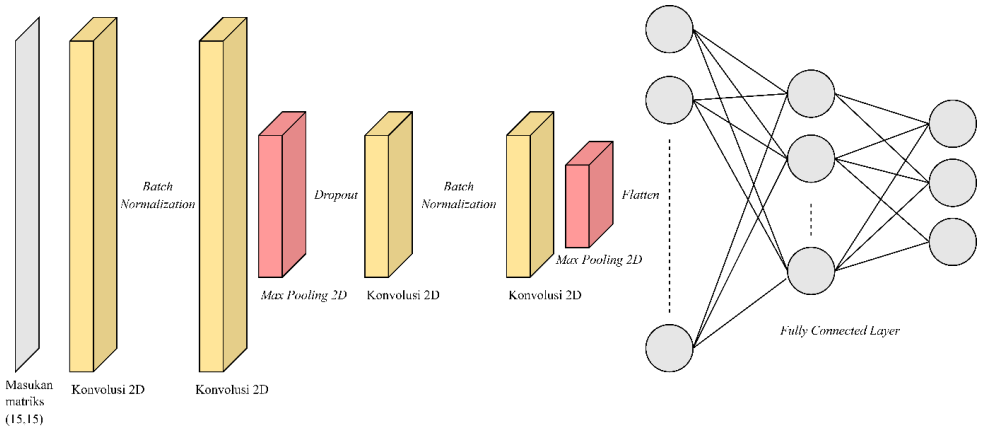
- $f_{y_i} = \frac{n_{y_i}}{n}$ adalah proporsi kelas dari y_i
- Semakin jarang suatu kelas, f_{y_i} akan semakin kecil, artinya w_i akan semakin besar, demikian sebaliknya. Ini berarti makin jarang suatu kelas, makin besar pembobotannya.

Setelah kita lakukan langkah-langkah yang telah dijelaskan, data sebenarnya sudah siap untuk dimasukan ke model untuk dilatih. Namun masih ada satu langkah lagi yang tidak boleh terlewat. Langkah itu adalah penggabungan dan pengacakan pada data. Pendekatan ini harus dilakukan karena penulis ingin membuat "satu model untuk semua" dengan harapan generalisasi yang baik seperti hasil yang didapat oleh Fischer, C. *et al*[2]. Ilustrasi dari pendekatan ini adalah sebagai berikut:



Bayangkan setiap kotak di atas berisi masukan dan luaran yang bersesuaian beserta bobotnya. Agar kode reprodusibel, kita akan melakukan inisialisasi *seed* sebelum dilakukan penggabungan dan pengacakan. Hasil dari penggabungan pengacakan inilah yang nanti akan menjadi masukan bagi model. Harapan dari pendekatan ini adalah agar model dapat generalisasi yang baik untuk semua kasus yang sebenarnya mirip-mirip dalam hal ini melakukan tugas klasifikasi JUAL, BELI atau TAHAN pada instrumen investasi perbankan.

2.3. Pelatihan model arsitektur CNN



Gambar 2. Visualisasi Arsitektur Model CNN yang digunakan

Gambar di atas merupakan visualisasi arsitektur model CNN yang digunakan dalam eksperimen kali ini. Arsitektur di atas memiliki total parameter sebanyak 135,587 Parameter dimana 135.489 parameter diantaranya merupakan parameter yang dapat dilatih sedangkan sisanya sebanyak 128 parameter merupakan parameter yang tidak dapat dilatih. Penulis mendapatkan arsitektur seperti ini dengan cara melakukan *hyperparameter-tuning* secara manual, dimana secara garis besar, penulis mencoba untuk melihat plot kurva kerugian dan akurasi terhadap *epoch.* Jika terindikasi overfit, maka kompleksitas model akan dikurangi sehingga parameternya juga berkurang. Sebaliknya jika terindikasi *underfit,* maka kompleksitas akan ditambah sehingga parameternya bertambah. Selanjutnya, akan disajikan lebih detil terkait ukuran luaran dan jumlah parameternya pada tabel berikut beserta apa yang sebenarnya terjadi pada tiap tiap lapisannya.

Tabel 3. Ukuran Luaran dan Jumlah Parameter pada tiap Lapisan Arsitektur CNN

Lapisan ke-i	Nama lapisan (tipe lapisan)	Ukuran luaran	Jumlah parameter
Lapisan ke-0	input_layer ($InputLayer$)	(None, 15, 15, 1)	0
Lapisan ke-1	block1_conv1 ($Conv2d$)	(None, 15, 15, 32)	320
Lapisan ke-2	block1_bn1 ($BatchNormalization$)	(None, 15, 15, 32)	128
Lapisan ke-3	block1_conv2 ($Conv2D$)	(None, 15, 15, 32)	9,248
Lapisan ke-4	block1_pool ($MaxPooling2D$)	(None, 7, 7, 32)	0
Lapisan ke-5	block1_dropout ($dropout$)	(None, 7, 7, 32)	0
Lapisan ke-6	block2_conv1 ($Conv2D$)	(None, 7, 7, 32)	9,248
Lapisan ke-7	block2_bn1 ($BatchNormalization$)	(None, 7, 7, 32)	128
Lapisan ke-8	block2_conv2 ($Conv2D$)	(None, 7, 7, 32)	9,248
Lapisan ke-9	block2_pool ($Maxpooling2D$)	(None, 3, 3, 32)	0
Lapisan ke-10	block2_dropout ($Dropout$)	(None, 3, 3, 32)	0
Lapisan ke-11	flatten ($Flatten$)	(None, 288)	0
Lapisan ke-12	dense_relu_1 ($Dense$)	(None, 256)	73,984
Lapisan ke-13	dense_dropout_1 ($Dropout$)	(None, 256)	0
Lapisan ke-14	dense_relu_2 ($Dense$)	(None, 128)	32,896
Lapisan ke-15	dense_dropout_2 ($Dense$)	(None, 128)	0
Lapisan ke-16	output ($Dense$)	(None, 3)	387

Mula-mula inisialisasikan masukan pada lapisan ke-0. Perhatikan bahwa pada keterangan di Tabel 3, keterangan None pada entri kolom ukuran luaran dari lapisan ke-0 menyatakan jumlah pelatihan pada tiap *batch* dalam suatu *epoch*. Perhitungan paralel seperti ini menjadikan proses pelatihan lebih cepat, namun menjadi kurang deterministik. Oleh karena itu, pada awal dari kode, ditambahkan inisialisasi *seed* agar reprodusibilitas dapat dihasilkan.

Selain itu, karena pada eksperimen kali ini penulis menggunakan *gpu* pada *google colab* untuk menjalankan pelatihan, maka untuk memastikan hasil pelatihan yang *reproducible*, diperlukan konfigurasi sintaks kode tambahan yang mengatur determinisme perhitungan pada pustaka *tensorflow* dan operasi acak *python* . Kode berikut dapat ditambahkan sebelum memanggil model atau jika ingin lebih aman, jalankan di awal kode sebelum mengimpor kumpulan pustaka.

Kode 1. Inisialisasi Deterministik Untuk Memastikan Reprodusibilitas

```
import os
os.environ['TF_DETERMINISTIC_OPS'] = '1'
os.environ['TF_CUDNN_DETERMINISTIC'] = '1'
os.environ['PYTHONHASHSEED'] = str(42)
```

Arsitektur di atas adalah arsitektur dengan hasil terbaik yang didapat setelah penulis melakukan *hyperparameter-tuning* secara manual. Parameter-parameter yang penulis coba ganti-ganti adalah sebagai berikut:

- Jumlah *layer* atau lapisan. Jika hasil *overfit* parah artinya model terlalu kompleks sehingga penulis akan mencoba untuk mengurangkan jumlah lapisan yang digunakan. Demikian berlaku sebaliknya.
- Penggunaan *layer* atau lapisan *dropout* dan regularisasi. Jika hasil tidak *overfit* ataupun *underfit* tetapi masih kurang stabil, penulis akan mencoba untuk menambahkan lapisan *dropout* dan regularisasi setelah lapisan-lapisan tertentu.
- *learning rate* dicoba pada ruang $[10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}]$.
- Pemberhentian awal atau *early stop* juga digunakan dengan batas kesabaran atau *patience* 10 *epoch* dengan total maksimal *epoch* diatur sebesar 100. Indikator pemberhentian awal adalah nilai kerugian dari validasi, ini berarti jika nilai kerugian dari validasi tidak kunjung turun selama 10 *epoch* berturut-turut, maka proses pelatihan akan dipaksa berhenti dan hasil parameter terbaik sebelum *epoch* terakhir akan disimpan.

2.4. Proses pelatihan

Fungsi kerugian yang dipakai dalam proses pelatihan adalah fungsi *categorical cross-entropy*. Secara matematis, misal θ menyatakan kumpulan parameter yang dapat dilatih dari arsitektur, maka fungsi kerugian didefiniskan sebagai:

$$\mathcal{L}(\theta) = -\frac{\sum_{n=1}^{N} \lambda_i \times \sum_{i=1}^{C} \log\left(\hat{y}_{n,i}\right)}{\sum_{i=1}^{N} w_i}$$

dimana:

- λ_i : bobot tiap sampel ke-i
- N : jumlah sampel tiap *batch*
- C : jumlah kelas
- $y_{n,i}$: label pada data asli dalam bentuk *one-hot encoded*
- $\hat{y}_{n,i}$: hasil prediksi untuk sampel ke-n dan kelas ke-i

Selain itu, proses pelatihan juga memanfaatkan algoritma dari fungsi optimisasi untuk mempercepat konvergensi-nya. Dalam proses pelatihan ini, fungsi optimisasi yang dipakai adalah fungsi adam. Algoritma dari fungsi adam diberikan pada ilustrasi berikut:

Algoritma 2. Algoritma Optimisasi Adam

1: $\boldsymbol{g}_t \leftarrow \nabla_{\theta_{t-1}} f(\theta_{t-1})$
2: $\boldsymbol{m}_t \leftarrow \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\boldsymbol{g}_t$

3: $\widehat{\boldsymbol{m}}_t \leftarrow \dfrac{\boldsymbol{m}_t}{1-\beta_1^t}$

4: $\boldsymbol{v}_t \leftarrow \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2)\boldsymbol{g}_t^2$

5: $\widehat{\boldsymbol{v}}_t \leftarrow \dfrac{v_t}{1-\beta_2^t}$

6: $\theta_{t+1} = \theta_t - \eta \dfrac{\widehat{\boldsymbol{m}}_t}{\sqrt{\widehat{\boldsymbol{v}}_t} + \epsilon}$

dimana:

- η : nilai *learning rate*
- ϵ : konstanta cukup kecil untuk menghindari pembagian oleh nol
- $\boldsymbol{m}_t$: estimasi bias dari gradien *moving average*
- β_1 : parameter momentum (secara *default* bernilai 0.9)
- $\boldsymbol{v}_t$: *exponential moving average* dari kuadrat gradien
- β_2 : konstanta peluruhan (*decay*) dari $\boldsymbol{v}_t$ (secara *default* bernilai 0.999)

2.5. *Evaluasi finansial*

Dalam menguji apakah label hasil prediksi model menguntungkan atau tidak, kita akan melakukan evaluasi finansial terhadap kumpulan label yang berisikan langkah-langkah jual (0), beli (1), dan tahan (2). Dalam melakukan evaluasi finansial ini dengan melakukan simulasi jual-beli. Namun, sebelum dilakukan simulasi, akan dibuat beberapa aturan jual-beli terlebih dahulu, diantaranya:

- Label beli
  Jika prediksi menginstruksikan beli dan sebelumnya belum pernah membeli, maka semua uang tunai digunakan untuk membeli saham saat itu juga.
- Label jual
  Jika prediksi menginstruksikan jual dan sebelumnya sudah pernah membeli, maka akan dilakukan aksi jual semua saham yang dimiliki saat itu juga
- Label tahan
  Tidak melakukan aksi apapun.
- Label sama muncul berturut-turut
  Hanya label pertama yang dianggap. Misalnya prediksi label sebagai berikut: beli, tahan, beli. Maka aksi beli yang terakhir tidak akan dilakukan, jika prediksi label sebagai berikut: beli, jual, jual, tahan. Maka aksi jual yang terakhir tidak akan dilakukan karena pada penjualan pertama sudah harus habis terjual.

Setelah aturan jual-beli ditetapkan, simulasi dilakukan untuk keseluruhan hari yang ada pada data uji. Pada saat melakukan simulasi tersebut, akan dihitung beberapa indikator evaluasi finansial sebagai berikut:

- *Annualized return* (AR)

$$\text{AR} = \left(\frac{\text{jumlah uang di akhir simulasi}}{\text{jumlah uang di awal simulasi}}\right)^{\frac{1}{\text{total tahun simulasi}}} - 1$$

- *Average trades per year* (AnT)

$$\text{AnT} = \frac{\text{jumlah aksi jual beli}}{\text{total tahun simulasi}}$$

- *Success Rate* (PoS)

$$\text{PoS} = \frac{\text{jumlah transaksi yang untung}}{\text{jumlah keseluruhan transaksi}} \times 100$$

- *Average Profit per Trade* (ApT)

$$\text{ApT} = \frac{\text{total persentase keuntungan}}{\text{jumlah transaksi}} \times 100$$

- *Idle Rate* (IdleR)

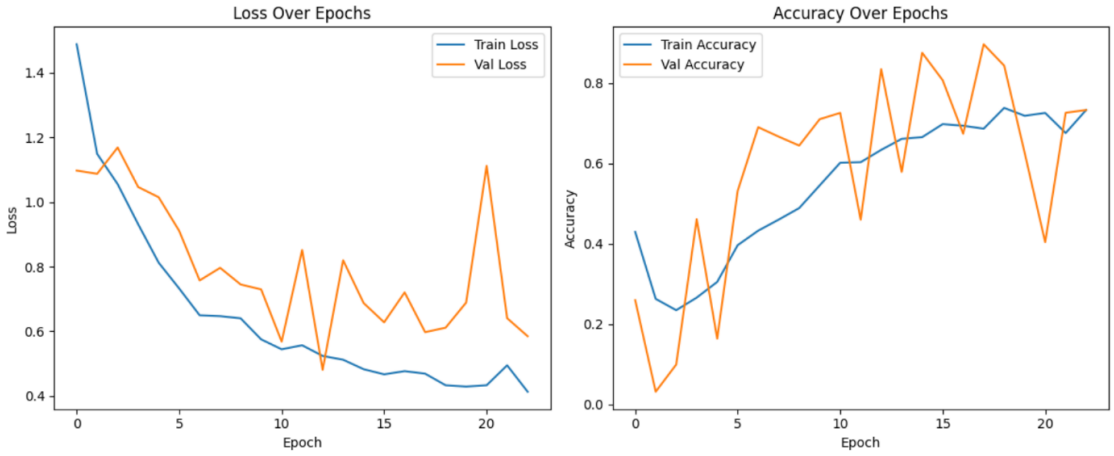
$$\text{IdleR} = \frac{\text{jumlah hari tidak memiliki saham}}{\text{total hari}} \times 100$$

- *Daily Sharpe Ratio* (sharpe_ratio)

$$\text{sharpe}_{\text{ratio}} = \frac{(\text{rata} - \text{rata keuntungan harian} - risk\ free\ rate\ tahunan)}{\sigma_p\ (\text{volatilitas keuntungan portofolio})}$$

III. Hasil dan Pembahasan

3.1. Hasil pelatihan data latih dan validasi



Gambar 3. Plot Kurva Kerugian dan Akurasi terhadap *epochs*

Gambar kurva di atas adalah hasil pelatihan model CNN terhadap dataset gabungan Saham BBCA, BBRI, BMRI, dan BBNI. Model yang digunakan untuk keempat data historis saham tersebut adalah sama karena data latih dari keempat saham ini digabung dan diacak seperti yang sudah dijelaskan pada bagian sebelumnya.

Jika melihat gambar kurva di atas, didapat bahwa plot kerugian dari data latih monoton turun, ini menunjukkan model berhasil belajar dari data pelatihan. Tidak hanya itu, plot akurasi dari data latih juga monoton naik. Dari hasil kurva kerugian dan akurasi terhadap *epoch* pada data latih, dapat disimpulkan bahwa model memiliki kapasitas yang cukup untuk mempelajari pola-pola yang ada dalam data pelatihan.

Sedangkan untuk plot kerugian data validasi, walaupun nilai kerugian memiliki tren turun, penurunan nilai kerugian-nya sangat tidak stabil dimulai saat *epoch* ke-11. Meskipun demikian, penurunan nilai kerugian di bawah 10 *epoch* sangatlah stabil. Hal ini menandakan model masih berhasil melakukan generalisasi pada 10 *epoch* pertama. Hal ini menunjukkan model masih sulit untuk melakukan generalisasi. Meskipun model berhasil melakukan generalisasi, akurasi model pada data validasi sangat tidak stabil. Hal ini mengindikasikan bahwa performa akurasi untuk data yang belum pernah dilihat model masih belum stabil.

3.2. Hasil pengujian model pada data uji

Setelah model berhasil dilatih, model akan diuji pada 4 data uji dari Saham BBCA, BBRI, BMRI, dan BBNI. Berikut adalah hasil konfusi matriks dari keempat data uji:

Tabel 4. Konfusi Matriks dari Data Uji Saham BBCA

		Label yang diprediksi		
		Jual	Beli	Tahan
Label yang benar	Jual	8	0	11
	Beli	0	9	9
	Tahan	19	27	668

Tabel 5. Laporan Klasifikasi dari Data Uji Saham BBCA

	Precision	*Recall*	*F1-Score*	*Support*
Label Jual	0.30	0.42	0.35	19
Label Beli	0.25	0.50	0.33	18
Label Tahan	0.97	0.94	0.95	714
Accuracy			0.91	751
Macro Average	0.51	0.62	0.54	751
Weighted Average	0.94	0.91	0.92	751

Tabel 6. Konfusi Matriks dari Data Uji Saham BBRI

		Label yang diprediksi		
		Jual	Beli	Tahan
Label yang benar	Jual	5	0	11
	Beli	0	7	8
	Tahan	22	20	698

Tabel 7. Laporan Klasifikasi dari Data Uji Saham BBRI

	Precision	*Recall*	*F1-Score*	*Support*
Label Jual	0.19	0.31	0.23	16
Label Beli	0.26	0.47	0.33	15
Label Tahan	0.97	0.94	0.96	740
Accuracy			0.92	771
Macro Average	0.47	0.57	0.51	771
Weighted Average	0.94	0.92	0.93	771

Tabel 8. Konfusi Matriks dari Data Uji Saham BMRI

		Label yang diprediksi		
		Jual	Beli	Tahan
Label yang benar	Jual	7	0	10
	Beli	0	5	9
	Tahan	27	26	700

Tabel 9. Laporan Klasifikasi dari Data Uji Saham BMRI

	Precision	*Recall*	*F1-Score*	*Support*
Label Jual	0.21	0.41	0.27	17
Label Beli	0.16	0.36	0.22	14
Label Tahan	0.97	0.93	0.95	753
Accuracy			0.91	784
Macro Average	0.45	0.57	0.48	784
Weighted Average	0.94	0.91	0.92	784

Tabel 10. Laporan Klasifikasi dari Data Uji Saham BBNI

		Label yang diprediksi		
		Jual	Beli	Tahan
Label yang benar	Jual	10	0	7
	Beli	0	15	2
	Tahan	50	70	610

Tabel 11. Laporan Klasifikasi dari Data Uji Saham BBNI

	Precision	*Recall*	*F1-Score*	*Support*
Label Jual	0.17	0.59	0.26	17
Label Beli	0.18	0.88	0.29	17
Label Tahan	0.99	0.84	0.90	730
Accuracy			0.83	764
Macro Average	0.44	0.47	0.49	764
Weighted Average	0.95	0.83	0.88	784

Tabel-tabel di atas menunjukkan hasil dari model terlatih pada data uji. Sebenarnya pada sintaks kode dan pelatihan sudah diaplikasikan bobot untuk target, namun ternyata hasil prediksi masih menghasilkan ketidakseimbangan jumlah kelas. Oleh karena itu, untuk menyimpulkan baik atau tidaknya model ini, kita tidak dapat menggunakan nilai dari akurasi karena bias terhadap kelas mayoritas. Sehingga akan digunakan nilai dari *Macro Average F1-Score.* Nilai ini akan kita bandingkan terhadap *macro average F1-Score* dari model naif, yakni model yang selalu memprediksi kelas mayoritas. Telebih dahulu kita hitung *Macro Average F1-Score* dari model naif untuk masing masing saham lalu bandingkan dengan *Macro Average F1-Score* dari model yang sudah dilatih. Perbandingannya sebagai berikut:

- Interpretasi Metrik Saham BBCA
  Diketahui $N_{jual} = 19, N_{beli} = 18,$ dan $N_{tahan} = 714$ sehingga $N_{total} = 751$. Sedangkan untuk *F1-Score* dari kelas tahan diberikan oleh:
  $$2 \times \frac{\text{jumlah tahan sebenarnya}}{\text{jumlah tahan sebenarnya} + \text{jumlah total saham}} = 2 \times \frac{714}{714 + 751} \approx 0.9747$$
  Karena tidak pernah memprediksi jual dan beli, akibatnya, *F1-Score* dari kelas jual dan beli bernilai nol sehingga *Macro Average F1-Score* dari model naif diberikan oleh:
  $$\frac{0.9747}{3} \approx 0.3249$$

Karena *Macro Average F1-Score* model terlatih adalah sebesar 0.54, dimana lebih besar daripada 0.3249, dapat disimpulkan model bekerja lebih baik dari sekadar menebak kelas mayoritas pada data uji saham BBCA.

- Interpretasi Metrik Saham BBRI

  Diketahui $N_{jual} = 16, N_{beli} = 15, \text{dan } N_{tahan} = 740$ sehingga $N_{total} = 771$. Sedangkan untuk *F1-Score* dari kelas tahan diberikan oleh:

$$2 \times \frac{\text{jumlah tahan sebenarnya}}{\text{jumlah tahan sebenarnya} + \text{jumlah total saham}} = 2 \times \frac{740}{740 + 771} \approx 0.9795$$

  Karena tidak pernah memprediksi jual dan beli, akibatnya, *F1-Score* dari kelas jual dan beli bernilai nol sehingga *Macro Average F1-Score* model naif diberikan oleh:

$$\frac{0.9795}{3} \approx 0.3265$$

  Karena *Macro Average F1-Score* model terlatih adalah sebesar 0.51, dimana lebih besar daripada 0.3265, dapat disimpulkan model bekerja lebih baik dari sekadar menebak kelas mayoritas pada data uji saham BBRI.

- Interpretasi Metrik Saham BMRI

  Diketahui $N_{jual} = 17, N_{beli} = 14, \text{dan } N_{tahan} = 753$ sehingga $N_{total} = 784$. Sedangkan untuk *F1-Score* dari kelas tahan diberikan oleh:

$$2 \times \frac{\text{jumlah tahan sebenarnya}}{\text{jumlah tahan sebenarnya} + \text{jumlah total saham}} = 2 \times \frac{753}{753 + 784} \approx 0.9798$$

  Karena tidak pernah memprediksi jual dan beli, akibatnya, *F1-Score* dari kelas jual dan beli bernilai nol sehingga *Macro Average F1-Score* model naif diberikan oleh:

$$\frac{0.9747}{3} \approx 0.3266$$

  Karena *Macro Average F1-Score* model terlatih adalah sebesar 0.48, dimana lebih besar daripada 0.3266, dapat disimpulkan model bekerja lebih baik dari sekadar menebak kelas mayoritas pada data uji saham BMRI.

- Interpretasi Metrik Saham BBNI

  Diketahui $N_{jual} = 17, N_{beli} = 17, \text{dan } N_{tahan} = 730$ sehingga $N_{total} = 764$. Sedangkan untuk *F1-Score* dari kelas tahan diberikan oleh:

$$2 \times \frac{\text{jumlah tahan sebenarnya}}{\text{jumlah tahan sebenarnya} + \text{jumlah total saham}} = 2 \times \frac{730}{730 + 764} \approx 0.9772$$

  Karena tidak pernah memprediksi jual dan beli, akibatnya, *F1-Score* dari kelas jual dan beli bernilai nol sehingga *Macro Average F1-Score* model naitf diberikan oleh:

$$\frac{0.9772}{3} \approx 0.3257$$

  Karena *Macro Average F1-Score* model terlatih adalah sebesar 0.49, dimana lebih besar daripada 0.3257, dapat disimpulkan model bekerja lebih baik dari sekadar menebak kelas mayoritas pada data uji saham BBNI.

3.3. Evaluasi Finansial

Evaluasi finansial akan dilakukan dengan cara melakukan simulasi berdasarkan aturan jual beli yang telah dijelaskan sebelumnya dan menghitung beberapa metrik finansial saat simulasi sedang berlangsung. Pada evaluasi finansial pada data uji, digunakan asumsi sebagai berikut: uang awal simulasi adalah Rp10,000,000.00., admin jual sebesar 0.25%, admin beli sebesar 0.15% dan *Risk Free*

Rate (RFR) yang biasa ditetapkan oleh Bank Indonesia adalah sebesar 5%. Simulasi dilakukan sebanyak 751, 771, 784 dan 764 hari berturut-turut untuk saham BBCA, BBRI, BMRI dan BBNI. Setelah dilakukan simulasi, didapat hasil sebagai berikut:

Tabel 12. Evaluasi Finansial Pada Simulasi Data Uji Saham BBCA, BBRI, BMRI, Dan BBNI

Metode Evaluasi	BBCA	BBRI	BMRI	BBNI
Annualized Return (%)	10.97	10.86	15.24	12.90
Average Trades per Year	3.49	3.40	7.65	2.26
Average Profit per Trade (%)	3.54	3.87	7.65	2.26
Final Money	Rp13,887,935.84	Rp13,959,024.27	Rp15,944,494.86	Rp14,744,774.17
Idle Rate (%)	49.4	42.41	63.9	44.5
Max Loss per Trade (%)	-2.71	-6.59	-2.54	-12.71
Max Profit per Trade (%)	13.44	25.02	25.6	15.1
Sharpe Ratio (Daily)	0.031	0.0246	0.04271	0.031
Success Rate (%)	81.82	54.55	85.71	62.5

Dari tabel di atas, kita dapat menyimpulkan bahwa dengan menggunakan aturan jual beli dan hasil prediksi label dari model yang telah dilatih dapat menghasilkan keuntungan terlihat dari evaluasi *Annualized Return* yang konsisten menghasilkan keuntungan di atas 10% dan *Success Rate,* persentase kesuksesan simulasi semua saham yang berada di atas 50%, ini berarti mengungguli model naif. Meskipun demikian, model ini tetap mengandung risiko, terlihat dari jarak antara *Max loss per Trade* dan *Max Profit per Trade* yang cukup terpaut jauh dan *Sharpe Ratio* yang konsisten dibawah 0.1 mengindikasikan risiko yang jauh lebih tinggi daripada potensi keuntungan yang didapat.

Cara jual-beli seperti ini juga tidak cocok bagi *trader* harian karena *Average Trades per Year* bernilai kecil yang berarti tiap tahun sebenarnya simulasi ini tidak sering melakukan transaksi. *Average Profit per Trade* juga tidak terlalu besar, menjadikan ini sebagai kekurangan karena dengan jumlah jual-beli per tahunnya yang sedikit, untung yang didapat juga tidak signifikan.

IV. KESIMPULAN

Berdasarkan hasil dan pembahasan, didapat bahwa model CNN berhasil untuk belajar pada data latih dibuktikan dengan kerugian yang terus turun dan akurasi yang terus naik seiring dengan bertambahnya *epoch*. Selain itu, model CNN juga terindikasi dapat melakukan generalisasi pada data validasi yang dibuktikan dengan kerugian yang terus turun sebelum akhirnya mulai mengalami *overfit* pada *epoch* ke-11 yang dibuktikan dengan kerugian yang tidak stabil (walaupun jika dirata-rata masih, tren kerugian masih dapat dikatakan turun). Meskipun model sempat melakukan generalisasi sampai *epoch* ke-10, akurasi pada data validasi terindikasi tidak stabil dari awal hingga akhir *epoch*. Hal ini mengindikasikan bahwa model mengalami kesulitan dalam melakukan prediksi pada data yang belum ditemuinya.

Selanjutnya, ketika model yang telah dilatih diuji pada data uji dari keempat saham, didapat bahwa

keseluruhan *Macro Average F1-Score* dari model terlatih yang diuji pada data uji, selalu di atas *Macro Average F1-Score* dari model naif prediktor kelas mayoritas. Hal ini memperkuat kesimpulan sementara yang didapat dari plot kurva kerugian terhadap *epoch* pada data latih dan validasi, yakni: model berhasil belajar sesuatu dari data latih, bukan sekadar menghafal pola data latih serta dapat melakukan generalisasi pada data validasi.

Pengujian feasibilitas model pada kasus nyata juga dilakukan dengan melakukan simulasi aksi berdasarkan prediksi yang dihasilkan model. Secara keseluruhan, dengan pendekatan dan cara jual-beli yang tepat, model yang telah dilatih ini sebenarnya bisa saja menghasilkan keuntungan, terbukti dari nilai *Annualized Return* yang didapat dari data uji keempat saham ini yang semuanya di atas 10%. Namun, risiko untuk mendapatkan keuntungan yang hanya berkisar diantara 10-15% sangatlah besar terbukti dari keseluruhan nilai *Sharpe Ratio* yang di bawah 0.1. Hal ini memperkuat kesimpulan yang didapat dari plot kurva akurasi terhadap *epoch* pada data validasi, yakni: model kesulitan memprediksi data yang belum pernah dilihat sebelumnya yang dibuktikan dengan akurasinya yang tidak stabil meskipun tetap dalam tren naik.

Lebih lanjut, hasil dari eksperimen ini menunjukkan bahwa untuk membangun model kecerdasan buatan yang dapat secara konsisten menghasilkan keuntungan setiap hari itu adalah tugas yang sangat sulit. Meskipun model CNN yang dilatih bisa melakukan generalisasi sampai tahap tertentu dan menghasilkan keuntungan, sayangnya hasil ini tidak feasibel untuk diterapkan dalam simulasi nyata *trading* karena risikonya tidak sebanding dengan keuntungannya dibuktikan dengan *Sharpe Ratio* yang bernilai kurang dari 0.1. *Sharpe Ratio* yang bernilai 0.1 lebih jauh juga memberikan arti bahwa dengan usaha membuat model kecerdasan buatan yang sedemikian penuh tantangan dan kompleks, hasil yang didapat tidak bisa sepadan atau menyamai risiko yang terkandung dan keuntungan yang didapat dari melakukan investasi pada instrumen investasi seperti obligasi negara, *sukuk*, deposito atau semacamnya yang memberikan keuntungan sebesar 5% (asumsi yang digunakan oleh penulis).

Hasil penelitian ini diharapkan dapat memberikan pemahaman yang lebih luas kepada masyarakat dan pelaku pasar bahwa membangun model kecerdasan buatan untuk otomatisasi perdagangan, seperti robot trading yang diklaim mampu memberikan keuntungan harian secara konsisten, merupakan tantangan yang sangat kompleks dan tidak mudah diwujudkan. Oleh karena itu, dengan adanya bukti empiris yang disajikan dalam penelitian ini, diharapkan masyarakat dapat bersikap lebih kritis dan bijak dalam menilai berbagai skema investasi yang menawarkan keuntungan instan dengan mengatasnamakan kecerdasan buatan, robot trading, atau teknologi serupa, agar terhindar dari investasi yang tidak kredibel.

V. RENCANA PENELITIAN SELANJUTNYA

Rencana selanjutnya dari penelitian kami adalah untuk melakukan penelitian terhadap aturan pelabelan sedemikian sehingga hasil yang didapatkan dari algoritma pelabelan tidak menghasilkan hasil *imbalance.* Selain itu, kami juga akan melakukan *hyperparameter tuning* lanjutan dengan harapan hasil pada evaluasi finansial dari simulasi jual-beli yang dilakukan dapat menaikkan nilai *sharpe ratio*. Dengan begitu diharapkan bisa didapat model yang lebih dapat diandalkan untuk membantu seseorang dalam melakukan *trading* sehingga dapat menghasilkan keuntungan yang lebih konsisten dengan risiko yang seminim mungkin.

REFERENSI

1. O. B. Sezer dan A. M. Ozbayoglu, "Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach," *Applied Soft Computing*, vol. 70, hlm. 525-538, 2018.
2. T. Fischer dan C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, hlm. 654-669, 2018.
3. O. B. Sezer, A. M. Ozbayoglu, dan E. Dogdu, "An Artificial Neural Network-based Stock Trading System Using Technical Analysis and Big Data Framework," dalam *Proceedings of the 2017 ACM Southeast Conference*, Kennesaw, GA, USA, 2017, hlm. 223–226.
4. J. F. Chen, W. L. Chen, C. P. Huang, S.-H. Huang, dan A. P. Chen, "Financial time-series data analysis using deep convolutional neural networks," dalam *Proceedings - 2016 7th International Conference on Cloud Computing and Big Data, CCBD 2016*, 2017, hlm. 87–92.
5. K. Johnson, "Pandas TA - A technical analysis library in Python 3." [Online]. Tersedia: https://github.com/twopirllc/pandas-ta [Diakses: 20 Mei 2025].
6. R. Kusdiantara dan M. Islahuddin, "Lecture notes on CNNs for image classification."
7. T. McGuire, "Python library for technical analysis." [Online]. Tersedia: https://github.com/trevormcguire/technical-analysis [Diakses: 20 Mei 2025].
8. M. S. Z. Rizvi, "CNN image classification: Image classification using CNN," Analytics Vidhya, 2020. [Online]. Tersedia: https://www.analyticsvidhya.com/blog/2020/02/learn-image-classification-cnn-convolutional-neural-networks-3-datasets/ [Diakses: 20 Mei 2025].
9. Ding, X., Zhang, Y., Liu, T., & Duan, J. (2015). Deep Learning for Event-Driven Stock Prediction. *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, 2327–2333.
10. Yoshihara, A., Fujikawa, K., Seki, K., & Uehara, K. (2014). Predicting Stock Market Trends by Recurrent Deep Neural Networks. *Proceedings of the 2014 International Conference on Neural Information Processing (ICONIP 2014)*, 759–769.
11. X. Huang, J. Song, dan Z. Qin, "Hybrid CNN–BiGRU–AM model for stock forecasting based on risk-adjusted metrics," *Expert Systems with Applications*, vol. 235, 2025. [Online]. Tersedia: https://doi.org/10.3390/electronics14071275
12. MD S. M. Bhuiyan, M. A. Rafi, G. N. Rodrigues, M. N. H. Mir, A. Ishraq, M. F. Mridha, dan J. Shin, "Deep learning for algorithmic trading: A systematic review of predictive models and optimization strategies," Array, vol. 26, 2025. [Online]. Tersedia: https://doi.org/10.1016/j.array.2025.100390