

# **PENGUNAAN CONVOLUTIONAL NEURAL NETWORK UNTUK MEMBERIKAN REKOMENDASI JUAL BELI SAHAM**

## **MAKALAH**

**disusun untuk memenuhi tugas mata kuliah MA4072**

**Pembelajaran Mendalam pada semester II tahun ajaran 2021/2022**

oleh

Yayan Febriansyah 10118017

Arya Pratama Putra 10118104



**PROGRAM STUDI MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
INSTITUT TEKNOLOGI BANDUNG  
2022**

## ABSTRAK

Saham adalah surat berharga yang dimiliki seseorang sebagai bukti kepemilikan perusahaan. Dalam melakukan analisis saham, terdapat dua pendekatan yaitu analisis fundamental dan analisis teknikal. Analisis teknikal berfokus pada pergerakan harga saham dari waktu ke waktu. Untuk melakukan analisis teknikal biasanya menggunakan alat bantu yang disebut dengan indikator saham. Indikator ini berguna untuk memberikan tanda bahwa saham akan bergerak naik atau turun. Ada banyak indikator saham, dan biasanya digunakan beberapa indikator saham. Seiring berkembangnya *Deep Learning*, analisis teknikal saham dapat dilakukan dengan berbagai macam model *Deep Learning*. Pada penelitian ini dibuat model untuk melakukan prediksi pengambilan keputusan dari pergerakan harga saham menggunakan *Convolutional Neural Network*. Input yang digunakan adalah matriks dari 15 indikator saham dan 15 rentang hari yang berbeda sehingga dihasilkan gambar berukuran  $15 \times 15$  piksel. Berdasarkan penelitian ini, kita dapat menggunakan 15 indikator teknikal untuk dapat mengkonstruksi suatu gambar berukuran  $15 \times 15$  pixel sedemikian sehingga gambar tersebut dapat digunakan untuk memberikan prediksi pergerakan harga saham dan rekomendasi keputusan jual beli saham dengan model *Convolutional Neural Network*. Berdasarkan hasil yang diperoleh, arsitektur yang penulis gunakan dapat memberikan hasil yang mirip dengan tidak terpengaruh oleh perubahan hyperparameter maupun input data. Namun, arsitektur yang digunakan masih mengalami overfitting sehingga performanya saat diuji tidak sebaik performa pada saat traning. Selain itu, arsitektur ini hanya mampu memberikan hasil yang bagus untuk rekomendasi 'Buy' saja, sedangkan untuk rekomendasi 'Hold' dan 'Sell' memberikan hasil yang buruk. Untuk rekomendasi 'Buy' model mampu memberikan nilai precision, recall, dan F1 score lebih dari 80%. Sedangkan untuk rekomendasi 'Hold' dan 'Sell', model hanya mampu memberikan nilai precision, recall, dan F1 score kurang dari 30%. Jadi, jika model memberikan rekomendasi 'Buy', kita dapat mengikuti rekomendasi model. Sedangkan untuk rekomendasi 'Hold' dan 'Sell' sebaiknya tidak diikuti.

Kata kunci: *Deep learning, Convolutional Neural Network, Algorithmic trading*

## DAFTAR ISI

ABSTRAK.....	ii
DAFTAR ISI.....	iii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	1
1.3 Tujuan Penelitian.....	2
1.4 Manfaat Penelitian.....	2
BAB II METODOLOGI.....	3
2.1 Pengambilan Data .....	3
2.2 Indikator Teknikal .....	3
2.3 Convolutional Neural Network .....	7
2.4 Arsitektur Model CNN Yang Digunakan.....	18
BAB III HASIL PENELITIAN .....	20
3.1 Hasil Transformasi Data.....	20
3.2 Evaluasi Pengambilan Keputusan Jual Beli Saham .....	20
BAB IV DISKUSI .....	27
4.1 Pola Yang Mirip Terlepas dari Input Data dan Hyperparameter .....	27
4.2 Terjadinya Overfitting.....	27
4.3 Hasil Rekomendasi yang Hanya Baik Untuk Rekomendasi ‘Buy’ Saja.....	27
BAB V PENUTUP .....	28
5.1 Kesimpulan.....	28
5.2 Saran.....	28
REFERENSI .....	29
KONTRIBUSI .....	31

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Prediksi harga saham menggunakan model kecerdasan komputasional telah menjadi bagian dari sistem perdagangan saham selama beberapa dekade terakhir. Pada saat yang sama, banyak instrumen keuangan seperti ETF, *option*, dan *leverage system* (seperti forex) telah diperkenalkan untuk investor dan *trader* individu. Akibatnya, sistem perdagangan berdasarkan model pengambilan keputusan cerdas yang otonom semakin mendapatkan perhatian di berbagai pasar keuangan yang berbeda secara global.

Dalam beberapa tahun terakhir, model prediksi/klasifikasi berbasis *deep learning* mulai banyak muncul sebagai pencapaian kinerja terbaik di berbagai aplikasi, mengungguli metode kecerdasan komputasi klasik seperti *Support Vector Machine*. Namun, pemrosesan gambar dan masalah berbasis *vision* mendominasi jenis penerapan yang model *deep learning* ini mengungguli teknik lainnya.

Dalam literatur, metode pembelajaran mendalam sudah mulai muncul pada studi keuangan. Ada beberapa implementasi teknik *deep learning* seperti *Recurrent Neural Network* (RNN), *Convolutional Neural Network* (CNN), dan *Long Short Term Memory* (LSTM). Secara khusus, penerapan jaringan saraf dalam pada model *forecasting* keuangan sangat terbatas.

CNN sejauh ini merupakan model *deep learning* yang paling umum diadaptasi. Sementara itu, sebagian besar implementasi CNN dalam literatur dipilih untuk mengatasi tantangan *computer vision* dan analisis citra. Dengan keberhasilan implementasi model CNN, tingkat kesalahan model terus menurun selama bertahun-tahun. Meskipun menjadi salah satu model yang diusulkan awal, AlexNet mencapai tingkat keberhasilan 50-55%. Baru-baru ini, versi berbeda dari algoritma Inception (v3, v4) dan ResNet (v50, v101, v152) mencapai tingkat keberhasilan sekitar 75-80%. Saat ini, hampir semua peneliti visi komputer, dengan satu atau lain cara, mengimplementasikan CNN dalam masalah klasifikasi gambar.

### **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini adalah sebagai berikut.

1. Bagaimana penggunaan Convolutional Neural Network untuk membuat pengambilan keputusan jual beli saham?
2. Bagaimana pengaruh perubahan input data terhadap hasil evaluasi model Convolutional Neural Network?
3. Bagaimana pengaruh perubahan hyperparameter terhadap hasil evaluasi model Convolutional Neural Network?

### **1.3 Tujuan Penelitian**

Tujuan yang hendak dicapai melalui penelitian ini adalah sebagai berikut.

1. Membuat prediksi pengambilan keputusan jual beli saham menggunakan Convolutional Neural Network
2. Melihat pengaruh perubahan input data terhadap hasil evaluasi model Convolutional Neural Network
3. Melihat pengaruh perubahan hyperparameter terhadap hasil evaluasi model Convolutional Neural Network

### **1.4 Manfaat Penelitian**

Manfaat dari adanya penelitian ini adalah dapat digunakannya hasil penelitian sebagai salah satu metode untuk melakukan analisis teknikal saham dan membantu dalam membuat keputusan jual beli saham.

## BAB II

### METODOLOGI

#### 2.1 Pengambilan Data

Data yang digunakan untuk penelitian ini adalah data historis 5 saham dari bursa saham Amerika Serikat yang terekam pada *Yahoo Finance*. *Ticker* dan nama perusahaan ditunjukkan pada Tabel 3.1. Pengambilan data dilakukan menggunakan API (*Application Programming Interface*) dari *Yahoo Finance* pada situsnya. Data yang diperoleh berbentuk tabular, seperti contoh pada Gambar 3.1.

Ticker	Nama Perusahaan
IBM	International Business Machines Corporation
AAPL	Apple Inc.
MSFT	Microsoft Corporation
JPM	JPMorgan Chase & Co.
KO	The Coca-Cola Company
HMC	Honda Motor Co., Ltd.

	Date	Open	High	Low	Close	Adj Close	Volume
0	1962-01-02	7.374124	7.374124	7.291268	7.291268	1.673325	407940
1	1962-01-03	7.291268	7.355003	7.291268	7.355003	1.687950	305955
2	1962-01-04	7.355003	7.355003	7.278521	7.281708	1.671130	274575
3	1962-01-05	7.272148	7.272148	7.125558	7.138305	1.638220	384405
4	1962-01-08	7.131931	7.131931	6.947100	7.004461	1.607503	572685
...	...	...	...	...	...	...	...
15191	2022-05-06	135.470001	137.990005	135.470001	137.669998	136.020004	7302400
15192	2022-05-09	134.410004	136.350006	133.320007	134.440002	134.440002	7647900
15193	2022-05-10	135.000000	135.410004	128.429993	129.130005	129.130005	8642400
15194	2022-05-11	129.860001	132.960007	129.860001	130.750000	130.750000	5301100
15195	2022-05-12	131.270004	133.619995	130.410004	132.899994	132.899994	5382000

*Gambar 1. Contoh data saham IBM*

#### 2.2 Indikator Teknikal

Indikator merupakan suatu penghitungan harga saham menggunakan garis atau data grafik dan memberikan suatu tanda tertentu pada saham, apakah harga saham akan

bergerak naik atau turun. Ada banyak indikator saham, namun dalam penelitian ini hanya akan digunakan 15 indikator saham, yaitu:

- Simple Moving Average (SMA)

Simple Moving Average (SMA) menunjukkan rata-rata pergerakan harga untuk periode tertentu. Dalam interpretasi yang paling banyak diterima, perpotongan nilai SMA dengan nilai interval yang berbeda digunakan untuk menentukan arah tren. Akibatnya, beberapa SMA dapat digabungkan untuk digunakan bersama-sama, atau satu nilai SMA tunggal dapat digunakan bersama dengan saham yang mendasarinya, yaitu jika harga saham lebih tinggi dari SMA (misalnya 50 hari), diasumsikan bahwa saham sedang uptrend, menandakan harga saham akan terus naik (buy trigger), sedangkan jika harga saham lebih rendah dari SMA, diasumsikan saham sedang downtrend, menandakan harga saham akan turun (sell trigger). Perhitungan SMA dapat dituliskan dalam persamaan berikut.

$$SMA(M, n) = \sum_{k=a+1}^{a+n} \frac{M(k)}{n}$$

- Exponential Moving Average (EMA)

Exponential Moving Average (EMA) adalah jenis indikator rata-rata pergerakan yang menunjukkan rata-rata pergerakan harga, lebih menekankan pada hari-hari terakhir. Data terbaru memiliki bobot lebih saat menghitung rata-rata bergerak. Pentingnya data terbaru meningkat secara eksponensial dalam perhitungan EMA. Perhitungan EMA dapat dituliskan dalam persamaan berikut.

$$(M(t) - EMA(M, t - 1, \tau)), \frac{2}{\tau + 1} + EMA(M, t - 1, \tau)$$

- Triple Exponential Moving Average (TEMA)

Triple Exponential Moving Average (TEMA) adalah jenis indikator EMA yang memberikan pengurangan fluktuasi harga kecil dan menyaring volatilitas. Itu dapat dihitung sebagai berikut:

$$(3 * EMA - 3 * EMA(EMA)) + EMA(EMA(EMA))$$

- WMA (Weighted Moving Average)

Weighted Moving Average (WMA) adalah jenis lain dari indikator rata-rata bergerak yang sama dengan rata-rata bergerak eksponensial. Satu-satunya perbedaan adalah pentingnya harga penutupan menurun secara linier ketika bergerak kembali ke

masa lalu. Di sisi lain, signifikansi harga penutupan saham menurun secara eksponensial di EMA. Persamaan 16 menunjukkan bagaimana WMA dihitung.

$$WMA(M, n) = \frac{\text{Sum of Weighted Averages}}{\text{Sum of Weight}}$$

- HMA (Hull Moving Average)

Hull Moving Average (HMA) adalah jenis indikator rata-rata bergerak yang mengurangi lag yang terkait dengan SMA. EMA dan WMA mencoba pengurangan lag dengan lebih menekankan pada data terbaru. HMA meningkatkan pengurangan lag ini dan mendapatkan hasil yang lebih baik jika dibandingkan dengan EMA dan WMA. Persamaan 17 menunjukkan perhitungan HMA.

$$HMA(M, n) = WMA\left(2 * WMA\left(\frac{n}{2}\right) - WMA(n)\right), \text{sqrt}(n)$$

- Williams %R

Williams %R adalah indikator teknis berbasis momentum yang juga menentukan kondisi overbought dan oversold untuk harga saham. Ini berskala antara -100 dan 0 nilai. Logika yang sesuai untuk Williams %R persis sama dengan RSI. Jika nilainya di bawah -80, diartikan bahwa harga saham berada di wilayah oversold. Sebaliknya, jika nilainya di atas -20, harga saham dianggap berada di wilayah overbought. Persamaan 13 menunjukkan bagaimana nilai Williams %R dihitung.

$$R = \frac{\max(\text{high}) - \text{close}}{\max(\text{high}) - \min(\text{low})} * (-100)$$

- Moving Average Convergence and Divergence (MACD)

Moving Average Convergence and Divergence (MACD) adalah indikator teknis yang menunjukkan tren harga saham. Jika garis MACD melintasi garis sinyal ke arah atas, diprediksi harga saham akan meningkat. Sebaliknya, jika garis MACD memotong garis sinyal ke arah bawah, diinterpretasikan bahwa harga saham akan turun. Persamaan 22 dan Persamaan 23 menunjukkan perhitungan MACD dan Garis Sinyal.

$$\text{MACD Line: } (12 \text{ Day EMA} - 26 \text{ Day EMA})$$

$$\text{Signal Line: } 9 \text{ Day EMA of MACD Line}$$

- Relative Strength Indeks (RSI)

Relative Strength Index (RSI) adalah indikator analisis teknikal tipe osilator yang menunjukkan kekuatan historis dan kelemahan harga saham. Saat harga saham



berubah, nilai RSI beresilasi antara 0 dan 100 yang menunjukkan apakah harga saham berada di wilayah overbought atau oversold. Penggunaan indikator RSI yang paling umum dan interpretasinya berfungsi sebagai berikut: Jika nilainya di atas 70, saham dianggap berada di wilayah overbought. Sedangkan jika nilainya di bawah 30, maka saham diasumsikan berada di wilayah oversold.

$$RSI = 100 - \frac{100}{1 + \frac{\text{average gain}}{\text{average loss}}}$$

- Rate of Change (ROC)

Rate of Change adalah indikator teknis yang menggambarkan kecepatan perubahan harga selama periode waktu tertentu. Persamaan 26 menunjukkan perhitungan rumus.

$$RoC = \frac{(\text{Latest close} - \text{Previous close})}{\text{Previous Close}} * 100$$

- Commodity Channel Indeks (CCI)

Commodity Channel Index (CCI) adalah indikator yang membandingkan harga saat ini dan harga rata-rata selama periode waktu tertentu. Ini beresilasi sebagian besar (% 75) antara -100 dan 100 nilai. %25 periode waktu, indikator melewati nilai rentangnya.

$$CCI = \frac{\text{Typical Price} - 20 \text{ Period SMA of TP}}{0.015 * \text{Mean Deviation}}$$

$$\text{Typical Price} = \frac{\text{High} + \text{Low} + \text{Close}}{3}$$

- Chande Momentum Oscillator Indikator (CMO)

Chande Momentum Oscillator (CMO) adalah jenis indikator momentum yang mirip dengan RSI dan Stochastic Oscillator. Ini beresilasi antara -100 dan 100. Jika nilai indikator lebih dari 50, itu ditafsirkan bahwa harga saham berada di wilayah overbought. Jika nilainya di bawah -50, biasanya harga saham dianggap berada di wilayah oversold. Rumus indikator diilustrasikan dalam Persamaan 21. Su adalah jumlah momentum hari naik dan Sd adalah jumlah momentum hari turun.

$$CMO = 100 * \frac{(S_u - S_d)}{(S_u + S_d)}$$

- Percentage Price Oscillator (PPO)

Persentase Harga Oscillator (PPO) mirip dengan MACD. Perhitungan PPO dan Signal Line PPO diilustrasikan pada Persamaan 24 dan Persamaan 25.

$$PPO = \frac{(12 \text{ Day EMA} - 26 \text{ day EMA})}{26 \text{ Day EMA}} * 100$$

*Signal Line: 9 Day EMA of PPO*

- Average Directional Indeks (ADX)  
Average directional index (ADX) adalah indikator analisis teknikal yang dapat digunakan untuk menentukan kekuatan sebuah trend. ADX dapat menunjukkan kapan trend dimulai, seberapa besar kekuatan trend, dampai waktu trend mulai melemah. Karena itu, indikator ADX dapat memberikan kemudahan bagi trader dalam menganalisis kecenderungan pasar.
- True Strength Index (TSI)  
True Strength Index (TSI) adalah sebuah osilator momentum yang memiliki rentang antara -100 dan +100, memiliki nilai dasar 0. Momentum menjadi positif apabila osilator tersebut sedang positif (mengarah ke pandangan pasar yang bullish) dan sebaliknya. Indeks ini dikembangkan oleh William Blau dan terdiri dari 2 garis: garis indeks dan pergerakan rata-rata eksponensial dari TSI yang disebut sebagai garis sinyal. Trader dapat mencari 5 tipe kondisi sebagai berikut: overbought, oversold, garis tengah, persilangan, divergen dan persilangan garis sinyal. Indikator ini seringkali digunakan bersamaan dengan sinyal-sinyal lainnya.
- The Ease of Movement Indicator (EMV)  
The Ease of Movement Indicator (EMV) adalah osilator berbasis volume yang dirancang untuk mengukur kemudahan (atau pergerakan) pergerakan harga suatu sekuritas. EMV adalah osilator terpusat, artinya nilai dapat berfluktuasi di atas dan di bawah nol.

## 2.3 Convolutional Neural Network

Model yang digunakan pada penelitian kali ini adalah Convolutional neural network (CNN). CNN adalah salah satu bagian dari model deep learning yang menggunakan operasi konvolusi dalam perhitungannya. CNN terdiri dari tiga lapisan, yaitu convolutional layer, pooling layer, dan fully-connected layer. CNN banyak digunakan pada kasus identifikasi gambar karena CNN memiliki keunggulan dalam

mengekstrak low-level feature sampai high-level feature pada gambar (Hua, Mou, & Zhu, 2018).

### 2.3.1 Convolution Layer

Konvolusi adalah operasi matematika dari dua fungsi yang menghasilkan fungsi baru yang merupakan modifikasi satu fungsi oleh fungsi lainnya (Behl, Bhatia, & Putri, 2014). Pada kasus identifikasi gambar, masukan berupa matriks yang merupakan konversi gambar menjadi satuan warna dalam ukuran pixel. Matriks akan dianggap sebagai fungsi diskrit pada convolution layer dengan pendefinisian sebagai berikut.

$$f(i, j) = H_{i,j}, \text{ untuk } i = 1, 2, \dots, m \text{ dan } j = 1, 2, \dots, n \quad (2.1)$$

dengan

$H_{i,j}$ : elemen baris ke- $i$  dan kolom ke- $j$  pada matriks  $H$

$m$ : banyaknya baris pada matriks  $H$

$n$ : banyaknya kolom pada matriks ke  $H$

Konvolusi yang terjadi pada *convolutional layer*, konvolusi diskrit 2-D, menghasilkan matriks baru yang merupakan penjumlahan dari perkalian masing-masing elemen dari matriks filter dan submatriks dengan ukuran yang sama. Kemudian, matriks filter tidak berubah, tetapi submatriks masukan yang digunakan bergeser tergantung pada ukuran lompatan atau stride.

Stride adalah parameter yang menentukan berapa jumlah pergeseran filter. Jika nilai stride adalah 1, maka *filter* akan bergeser sebanyak 1 pixel secara horizontal lalu vertikal. Semakin kecil stride maka akan semakin detail informasi yang kita dapatkan dari sebuah input, namun akan membutuhkan komputasi yang lebih jika dibandingkan dengan stride yang besar.

Terkadang kita memerlukan dimensi *output* dengan ukuran tertentu sehingga kita perlu menambahkan ukuran matriks. Dalam hal ini digunakan *padding*. *Padding* adalah parameter yang menentukan jumlah pixels (yang biasanya berisi nilai 0) yang akan ditambahkan di setiap sisi luar dari input. Tujuan dari penggunaan padding adalah karena dimensi output dari *convolution layer* selalu lebih kecil dari inputnya (kecuali penggunaan 1x1 filter dengan stride 1). *Output* ini akan digunakan kembali sebagai input dari *convolution layer* selanjutnya, sehingga makin banyak informasi yang terbuang. Dengan menggunakan padding, kita dapat mengatur dimensi output agar tetap sama seperti dimensi input atau setidaknya tidak berkurang secara drastis.

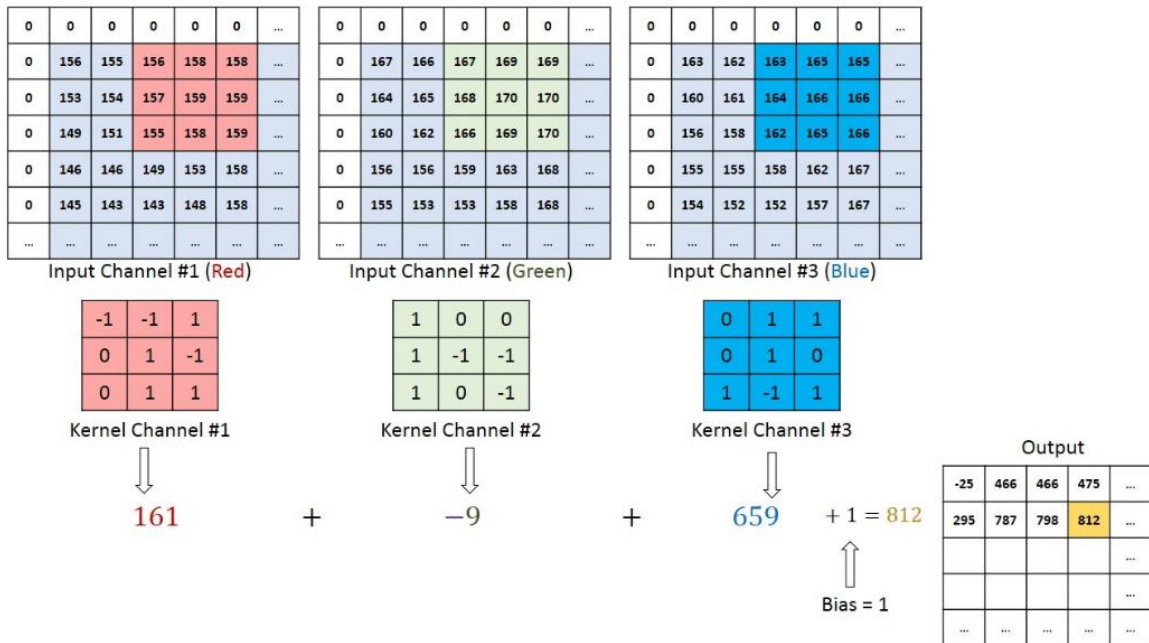
Sehingga kita bisa menggunakan *convolution layer* yang lebih dalam sehingga lebih banyak features yang berhasil diekstrak. Selain itu juga dapat meningkatkan performa dari model karena *convolution layer* akan berfokus pada informasi yang sebenarnya yaitu yang berada diantara *padding* tersebut.

Secara matematis dapat ditulis sebagai,

$$H_{i,j} = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} F_{i+m,j+n} \cdot G_{m+1,n+1}$$

$F$  adalah matriks input yang berukuran  $a \times a$ ,  $G$  adalah matriks *filter* yang berukuran  $k \times k$ , dan  $H$  adalah matriks *output* dari hasil konvolusi yang berukuran  $(a - k + 1) \times (a - k + 1)$ .

Dengan operasi konvolusi demikian, fitur-fitur dapat diekstrak dari gambar dengan menggunakan matriks filter tertentu. Nilai dari matriks filter ditentukan oleh model sebagai variabel yang dicari sehingga semakin besar ukuran dan banyak filter semakin besar juga parameter yang akan dioptimalkan oleh model nantinya.



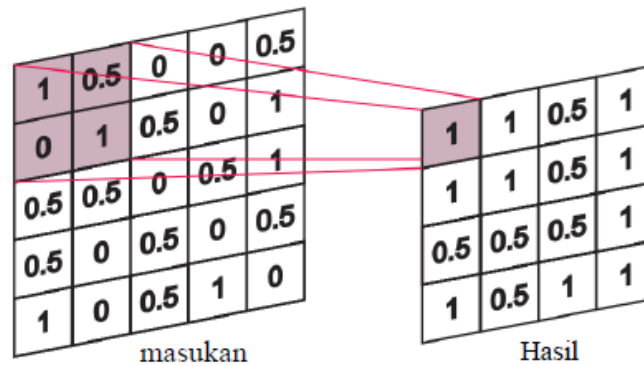
Gambar 2. Ilustrasi convolution layer

### 2.3.2 Pooling Layer

*Pooling* adalah suatu operasi submatriks yang digunakan untuk menurunkan ukuran *feature map* dengan menurunkan dimensinya. Tujuan dari penggunaan *pooling layer* adalah mengurangi dimensi dari *feature map* (*downsampling*), sehingga

mempercepat komputasi karena parameter yang harus diupdate semakin sedikit dan mengatasi overfitting.

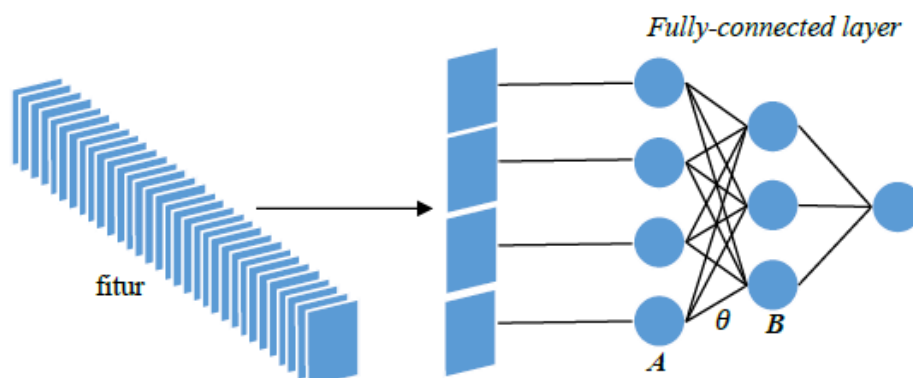
Pooling yang umum digunakan adalah *max-pooling* dan *average-pooling* (Saeedan, Weber, Goesele, & Roth, 2018). *Max-pooling* menggunakan operasi maksimum untuk setiap submatriksnya, sedangkan *average-pooling* menggunakan rata-rata dari setiap elemen submatriks.



Gambar 3. Ilustrasi pooling layer

### 2.3.3 Fully-connected Layer

*Fully-connected layer* digunakan untuk mengklasifikasi dari *convolutional neural network*. Layer ini ditempatkan setelah serangkaian *convolution layer* dan *max-pooling layer*. Setelah menerima hasil fitur-fitur yang telah diekstrak, kemudian *feature map* yang berbentuk dua dimensi tersebut dilakukan *flatten*, yaitu membentuk ulang fitur menjadi sebuah vektor agar dapat digunakan sebagai input dari *fully-connected layer*. *Fully-connected layer* menghasilkan keluaran yang merupakan hasil pembobotan dari semua elemen masukan (Liu, 2018), seperti ilustrasi gambar di bawah.

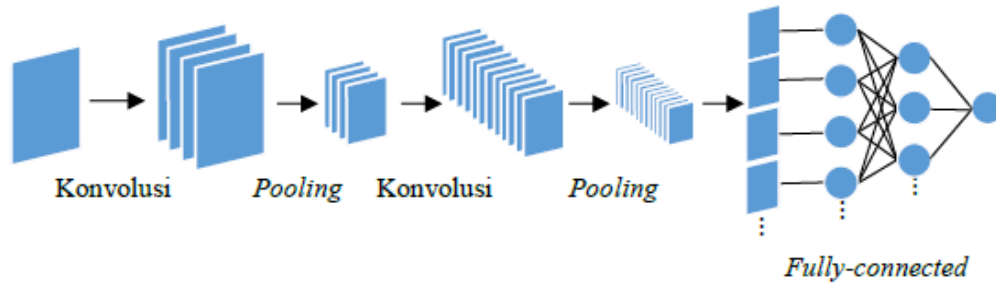


Gambar 4. Ilustrasi fully-connected layer

*Fully-Connected Layer* adalah lapisan di mana semua neuron dari lapisan sebelumnya terhubung semua dengan neuron di lapisan selanjutnya. Jadi, implementasinya seperti halnya jaringan syaraf tiruan biasa.

#### 2.3.4 Optimisasi Model CNN

Dari penjelasan sebelumnya, CNN dapat diilustrasikan seperti Gambar II.8. Penentuan banyak dan ukuran masing-masing lapisan dapat ditentukan oleh pengguna berdasarkan kebutuhan. Pada bagian ini, akan dijelaskan cara model menentukan parameter-parameter yang sesuai agar membuat model menjadi bagus, kriteria bagus pada model akan dijelaskan pada Subbab III.4.



Semakin panjang *convolutional neural network* semakin banyak juga parameter yang harus dioptimasi oleh model. Parameter dari CNN dihasilkan dari *convolution layer* dan *fully-connected layer*. Parameter dari *convolution layer* adalah matriks *filter* yang bergantung pada banyaknya dan ukurannya. Banyaknya parameter dalam satu *convolution layer* dapat dinyatakan dalam persamaan,

$$n_{cl} = (\text{banyak input}) \times (\text{ukuran filter})^2 \times (\text{banyak filter})$$

Parameter dari *fully connected layer* adalah matriks bobot yang bergantung pada banyaknya neuron dari input dan output. Banyak parameter dalam satu *fully-connected layer* dapat dinyatakan dalam persamaan

$$n_{cl} = (\text{banyak input}) \times (\text{banyak output})$$

Output dari *convolutional neural network* dihasilkan dari fungsi peubah banyak dengan

$$F: (I, \alpha_1, \alpha_2, \dots, \beta_1, \beta_2, \dots) \mapsto y_{prediksi}$$

Dengan

$I$  : input

$\alpha_i \in \mathbb{R}^{n_{cli}}$  : vektor parameter dari *convolution layer* ke- $i$  dengan banyaknya parameter  $n_{cli}$

$\beta_i \in \mathbb{R}^{n_{fcli}}$  : vektor parameter dari *fully-connected layer* ke- $i$  dengan banyaknya parameter  $n_{fcli}$

Untuk memperoleh model yang bagus, haruslah galat dari  $y_{prediksi}$  terhadap  $y_{asli}$  kecil. Sehingga, dapat dibuat suatu masalah optimisasi untuk meminimalkan galat atau fungsi yang merepresentasikan besarnya kerugian, *cost*. Fungsi kerugian dapat berupa rata-rata kuadrat galat, *binary crossentropy*, dan lain-lain.

$$\mathcal{L}: \mathbb{R}^2 \rightarrow \mathbb{R}$$

$$\text{minimalkan : } \mathcal{L}(y_{asli}, y_{prediksi}) \quad (2.8)$$

dengan,

$\mathcal{L}$ : fungsi kerugian.

### 2.3.5 Forward Propagation dan Backward Propagation pada CNN

Untuk melakukan pelatihan pada model, dilakukan tahapan untuk menyelesaikan permasalahan optimisasi dengan meminimalkan *loss function*. *Forward Propagation* menghitung nilai prediksi dengan melakukan kalkulasi sesuai dengan model yang telah dibuat, sedangkan *Backward Propagation* memberikan umpan balik kepada model berupa turunan parsial dari fungsi kerugian untuk masing-masing parameter. gradien dari suatu fungsi terhadap variabelnya ( $\nabla F$ ) dapat diimplementasikan sebagai arah gerak suatu titik untuk naik paling cepat dan berlaku sebaliknya untuk turun paling cepat. Perambatan maju dan perambatan mundur dilakukan secara berulang untuk meminimalkan fungsi kerugian. Istilah satu kali epoch diartikan sebagai satu kali prosedur Perambatan maju dan perambatan mundur (Chollet, Francois 2018).

- Perambatan maju

Untuk meminimalkan fungsi objektif seperti pada persamaan (2.8), diperlukan  $y_{prediksi}$  yang diperoleh dari model CNN yang telah dibuat. Sebagai inisiasi, nilai dari parameter model dipilih dengan suatu distribusi agar diperoleh nilai  $y_{prediksi}$  yang pertama. Perambatan maju untuk epoch selanjutnya, dilakukan dengan parameter-parameter yang telah diperbaharui oleh proses perambatan mundur.

- Perambatan mundur

Perambatan mundur merupakan suatu proses dalam neural network yang berperan sebagai pembaharuan nilai-nilai parameter yang ditaksir. Pembaharuan nilai-nilai parameter menggunakan arah untuk turun tercepat dari fungsi kerugian di parameter sebelumnya atau berlawanan dengan arah gradiennya. Untuk mengoptimisasi parameternya digunakan *gradient descent*, yaitu:

$$\theta_{t+1} = \theta_t - \alpha \left. \frac{\partial \mathcal{L}}{\partial \theta} \right|_{\theta=\theta_t}$$

Keterangan :

$\theta_{t+1}$  : parameter yang telah diperbarui

$\theta_t$  : parameter sebelumnya

$\alpha$  : *learning rate*

$\mathcal{L}$  : *loss function*

Untuk mencari nilai dari  $\frac{\partial \mathcal{L}}{\partial w}$  dengan  $w$  adalah parameter, digunakan aturan rantai untuk peubah banyak. Berikut ini adalah rinciannya.

### 1. Output layer

Dicari turunan *loss function* terhadap  $\theta$  dengan  $A$  adalah hidden layer dan  $y$  adalah outputnya. Misalkan  $A, w \in \mathbb{R}^n$ , untuk  $n \in \mathbb{N}$ , maka

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial w} &= \left[ \frac{\partial \mathcal{L}}{\partial w_1} \quad \dots \quad \frac{\partial \mathcal{L}}{\partial w_n} \right]^T \\ &= \left[ \frac{\partial \mathcal{L}}{\partial y} \frac{\partial y}{\partial w_1} \quad \dots \quad \frac{\partial \mathcal{L}}{\partial y} \frac{\partial y}{\partial w_n} \right]^T \end{aligned}$$

Karena  $y = \sum_{j=1}^n A_j w_j$ , maka

$$\frac{\partial \mathcal{L}}{\partial w} = \left[ \frac{\partial \mathcal{L}}{\partial y} A_1 \quad \dots \quad \frac{\partial \mathcal{L}}{\partial y} A_n \right]^T = \frac{\partial \mathcal{L}}{\partial y} A$$

### 2. Fully-connected layer

Dicari turunan *loss function* terhadap  $w$  dengan  $A$  dan  $B$  adalah dua *fully-connected layer* yang berdekatan.

Misalkan  $A \in \mathbb{R}^n, B \in \mathbb{R}^m, w \in M_{m \times n}$ , untuk  $m, n \in \mathbb{N}$ , maka

$$\frac{\partial \mathcal{L}}{\partial w} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial w_{1,1}} & \dots & \frac{\partial \mathcal{L}}{\partial w_{1,n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathcal{L}}{\partial w_{m,1}} & \dots & \frac{\partial \mathcal{L}}{\partial w_{m,n}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial b_1} \frac{\partial b_1}{\partial w_{1,1}} & \dots & \frac{\partial \mathcal{L}}{\partial b_1} \frac{\partial b_1}{\partial w_{1,n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathcal{L}}{\partial b_m} \frac{\partial b_m}{\partial w_{m,1}} & \dots & \frac{\partial \mathcal{L}}{\partial b_m} \frac{\partial b_m}{\partial w_{m,n}} \end{bmatrix}$$

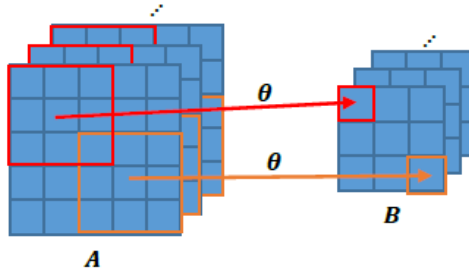
Karena  $b_i = \sum_{j=1}^n \theta_{i,j} A_j$

$$\frac{\partial \mathcal{L}}{\partial w} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial b_1} A_1 & \dots & \frac{\partial \mathcal{L}}{\partial b_1} A_n \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathcal{L}}{\partial b_m} A_1 & \dots & \frac{\partial \mathcal{L}}{\partial b_m} A_n \end{bmatrix} = \frac{\partial \mathcal{L}}{\partial B} A^T$$



### 3. Convolution Layer

Akan dicari turunan *loss function* terhadap parameter  $w$  dengan A dan B adalah input 2 dimensi dan hasil konvolusinya. Untuk mencari turunan *loss function* terhadap B diperlukan tahapan turunan pada *pooling layer* dan *flatten*.



Operasi konvolusi untuk input berupa matriks adalah sebagai berikut. Misalkan A berukuran  $v \times v \times t$  dan  $w$  berukuran  $s \times s \times t$ ,  $s$  adalah ukuran filter, maka

$$(B_i)_{(x,y)} = \sum_{c=1}^s \sum_{d=1}^s \sum_{e=1}^t (w_i)_{(c,d,e)} A_{(x+c-1, y+d-1, e)}$$

Keterangan

$w_i$  : filter ke- $i$  untuk suatu operasi konvolusi

$(\theta_i)_{(l,m,n)}$  : baris ke- $l$ , kolom ke- $m$ , dan lapisan ke- $n$  pada filter ke- $i$

$A_{(l,m,n)}$  : baris ke- $l$ , kolom ke- $m$ , dan lapisan ke- $n$  pada A

$(B_i)_{(x,y)}$  : baris ke- $x$  dan kolom ke- $y$  pada matriks  $B_i$

$B_i \in M_{(v-s+1) \times (v-s+1)}$  : matriks lapisan ke- $i$  pada lapisan matriks B

Dengan menggunakan aturan rantai diperoleh turunan *loss function* terhadap suatu elemen pada  $w_i$ , yaitu

$$\frac{\partial \mathcal{L}}{\partial (w_i)_{(l,m,n)}} = \sum_{p=1}^{v-s+1} \sum_{q=1}^{v-s+1} \frac{\partial \mathcal{L}}{\partial (B_i)_{(p,q)}} \frac{\partial (B_i)_{(p,q)}}{\partial (w_i)_{(l,m,n)}}$$

#### 2.3.6 Fungsi Aktivasi dan Bias

Pada subbab sebelumnya, telah dijelaskan rangkaian CNN yang dasar dan cara mengoptimalkannya. Untuk memperoleh model yang lebih baik, diperlukan beberapa

modifikasi atau penambahan dari CNN. Salah satu modifikasi atau penambahan dari CNN adalah fungsi aktivasi.

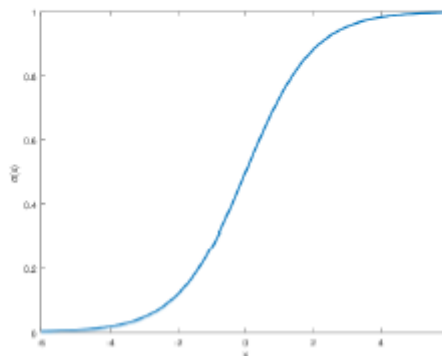
Fungsi aktivasi memberikan efek ketidaklinearan pada model atau menambah kompleksitas dari model (Sree, 2019). Model CNN pada bab sebelumnya hanya menggunakan penjumlahan dari perkalian masukan dengan bobot yang bersesuaian. Sehingga, model demikian hanya bersifat linear terhadap parameter.

Fungsi aktivasi adalah fungsi yang memetakan keluaran dari suatu neuron pada *neural network* untuk menambah efek ketidaklinearan pada model *neural network*. Beberapa fungsi aktivasi yang umum digunakan dalam model *deep neural network* adalah

- Sigmoid

Fungsi aktivasi sigmoid dalam matematika ditulis sebagai berikut.

$$f: \mathbb{R} \rightarrow (0,1)$$
$$f(x) = \frac{1}{1 + e^{-x}}$$

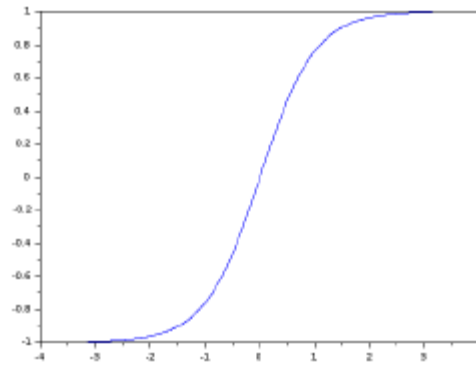


Sigmoid adalah fungsi terbatas bernilai real yang dapat diturunkan dan memiliki turunan bernilai positif di semua domainnya. Fungsi aktivasi sigmoid dapat digunakan pada *output layer* untuk model yang memprediksi keluaran berupa peluang dan bagus untuk model biner (Nwankpa, Ijomah, Gachagan, & Marshall, 2018).

- Hiperbolik tangen (Tanh)

Fungsi aktivasi tanh dalam matematika ditulis sebagai,

$$f: \mathbb{R} \rightarrow (-1,1)$$
$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



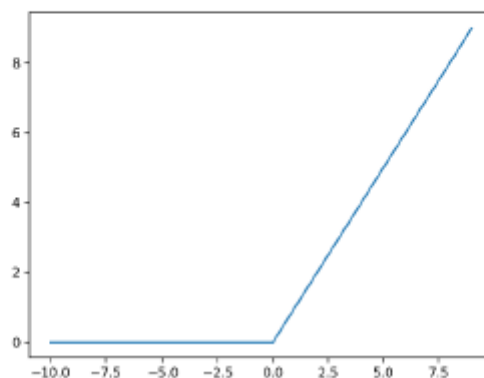
Fungsi aktivasi hiperbolik tangen memiliki bentuk yang sama dengan fungsi aktivasi sigmoid, tetapi memiliki *range* yang berbeda, yaitu  $(-1,1)$ . Selain itu, fungsi aktivasi hiperbolik tangen lebih curam di daerah  $x = 0$  sehingga turunan yang dihasilkan di sekitaran  $x = 0$  akan lebih besar yang akan berpengaruh pada bagian pembaharuan parameter.

- *Rectified Linear Unit* (ReLU)

Fungsi aktivasi ReLU secara matematika ditulis sebagai

$$f: \mathbb{R} \rightarrow \mathbb{R}^+$$

$$f(x) = \max(0, x)$$

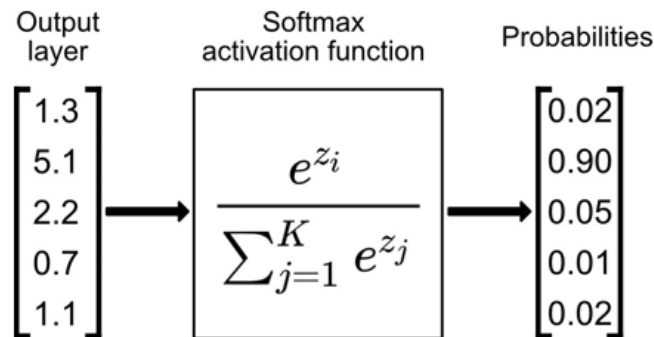


ReLU adalah fungsi aktivasi yang sering digunakan. ReLU memiliki kelebihan dalam mudah dalam komputasi, kecepatan mencapai konvergen, dan dapat mengatasi masalah *vanishing gradient* (Lin & Shen, 2018). Dengan sifat dari ReLU yang bisa membuat nilai suatu neuron menjadi 0, jika  $x < 0$ , membuat ReLU sebagai penyeleksi fitur-fitur yang akan diteruskan pada model *neural network* (Borisov, Haug, & Kasneci, 2019).

Pada penelitian ini, akan digunakan fungsi aktivasi ReLU di bagian konvolusi dan *fully-connected layer* agar dapat menyeleksi fitur-fitur. Pada *output layer*, akan digunakan fungsi aktivasi sigmoid agar membuat keluaran bernilai 0 sampai 1.

- Softmax

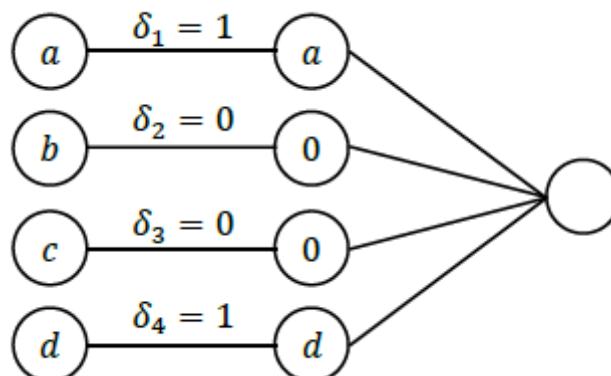
$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$



Fungsi aktivasi ini digunakan saat mencari nilai *output* dalam permasalahan klasifikasi. Fungsi aktivasi ini akan mengubah *output* menjadi suatu nilai probabilitas dengan penjumlahan dari semua kategorinya bernilai 1.

### 2.3.7 Dropout Layer

Salah satu kendala utama dari model *neural network* adalah *overfitting*, yaitu kondisi model dapat memprediksi dengan bagus untuk data *train* tetapi model kurang bagus dalam memprediksi data atau sampel baru. Untuk mengatasi hal tersebut, diberikan *dropout layer* (Srivastava, Geoffrey, Krizhevsky, Sutskever, & Salakhutdinov, 2014). Berikut adalah ilustrasi dari *dropout layer* pada suatu *multilayer perceptron*.



$\delta_i$  berdistribusi Bernoulli dengan suatu parameter  $p$  yang ditentukan sebagai *hyperparameter*. *Dropout layer* bisa mengatasi *overfitting* dengan mematikan beberapa neuron pada suatu lapisan, mengubah suatu neuron menjadi 0. *Overfitting* terjadi karena

pada suatu sampel data pelatihan terdapat fitur yang kontras yang hanya dimiliki oleh data sampel pelatihan. Sehingga, pembobotan untuk fitur yang kontras besar, sedangkan untuk fitur lainnya tidak. Fitur yang tidak kontras bisa menjadi suatu fitur yang umum (*general*). Dengan mematikan beberapa neuron, model tidak terpacu pada fitur yang dominan saat perbaharuan parameter sehingga dapat menghindari *overfitting* (Srivastava, Geoffrey, Krizhevsky, Sutskever, & Salakhutdinov, 2014).

## 2.4 Arsitektur Model CNN Yang Digunakan

Model CNN bagus dalam mengekstrak fitur-fitur yang ada pada gambar. Ukuran dan banyaknya *filter* berpengaruh pada seberapa dalam level fitur hasil ekstrak. Berikut adalah hasil ekstrak beberapa pasangan *convolution layer* dan *maxpooling layer* untuk menentukan banyaknya pasangan yang tepat.

Untuk memprediksi apakah saham perlu dilakukan “Hold”, “Sell”, atau “Buy”, digunakan model CNN sebagai berikut. Data yang digunakan terdiri dari *input* yang merupakan citra dari indikator saham dan rentang hari yang direpresentasikan dalam piksel berukuran  $15 \times 15$ , dan *output* label yang berupa keputusan yang dilakukan dengan label ‘Buy’, ‘Sell’, dan ‘Hold’.

Arsitektur dari model CNN ini terdiri dari *convolution layer* dengan filter berukuran  $2 \times 2$ , dengan jumlah *filter* 64, dan dilakukan *padding* agar ukuran setelah dilakukan konvolusi tetap sama yaitu tetap  $15 \times 15$ . Setelah itu menggunakan fungsi aktivasi ReLU dan kemudian dilakukan *max pooling* dengan ukuran  $2 \times 2$  sehingga dihasilkan matriks berukuran  $7 \times 7$ . Kemudian dilakukan konvolusi kembali dengan filter berukuran  $2 \times 2$  sejumlah 64 dan dilakukan *padding* agar ukuran hasil konvolusinya tetap sama yaitu  $7 \times 7$ . Kemudian menggunakan fungsi aktivasi ReLU dan dilakukan *max pooling* dengan ukuran  $2 \times 2$  sehingga matriks akhir berukuran  $3 \times 3$ . Setelah itu dilakukan *flatten* agar dihasilkan vektor berukuran 576. Setelah itu, proses berlanjut di *fully connected layer* yang mana prosesnya adalah mengklasifikasi seperti pada arsitektur *Multi Layer Perceptron*. Pada *fully-connected layer* ini digunakan 2 buah layer yang masing-masing berukuran 64 node dan 1 layer *output* dengan 3 node. Di antara 2 layer digunakan fungsi aktivasi ReLU dan pada *output* digunakan fungsi aktivasi *Softmax*. Untuk mengoptimisasinya digunakan *cross entropy* sebagai *loss function* dan *optimizer* ADAM.

Secara umum, model ini terdiri dari urutan arsitektur yang sama yaitu: Konvolusi – Aktivasi dengan ReLU – Max Pooling – Konvolusi – Aktivasi dengan

ReLU – Max Pooling – Dua Layer di *fully connected layer* – *Output Layer*. Hanya saja beberapa nilai *hyperparameter* dibuat berbeda. Beberapa model yang digunakan yaitu:

Layer (type)	Output Shape	Param #	Layer (type)	Output Shape	Param #
conv2d_55 (Conv2D)	(None, 14, 14, 64)	320	conv2d_57 (Conv2D)	(None, 14, 14, 64)	320
activation_76 (Activation)	(None, 14, 14, 64)	0	activation_78 (Activation)	(None, 14, 14, 64)	0
max_pooling2d_55 (MaxPooling2D)	(None, 7, 7, 64)	0	max_pooling2d_57 (MaxPooling2D)	(None, 7, 7, 64)	0
conv2d_56 (Conv2D)	(None, 6, 6, 64)	16448	conv2d_58 (Conv2D)	(None, 6, 6, 64)	16448
activation_77 (Activation)	(None, 6, 6, 64)	0	activation_79 (Activation)	(None, 6, 6, 64)	0
max_pooling2d_56 (MaxPooling2D)	(None, 3, 3, 64)	0	max_pooling2d_58 (MaxPooling2D)	(None, 3, 3, 64)	0
flatten_26 (Flatten)	(None, 576)	0	flatten_27 (Flatten)	(None, 576)	0
dense_70 (Dense)	(None, 64)	36928	dense_73 (Dense)	(None, 32)	18464
dense_71 (Dense)	(None, 64)	4160	dense_74 (Dense)	(None, 32)	1056
dense_72 (Dense)	(None, 3)	195	dense_75 (Dense)	(None, 3)	99
=====			=====		
Layer (type)	Output Shape	Param #	Layer (type)	Output Shape	Param #
conv2d_61 (Conv2D)	(None, 15, 15, 64)	640	conv2d_2 (Conv2D)	(None, 15, 15, 64)	320
activation_82 (Activation)	(None, 15, 15, 64)	0	activation_3 (Activation)	(None, 15, 15, 64)	0
max_pooling2d_61 (MaxPooling2D)	(None, 7, 7, 64)	0	max_pooling2d_2 (MaxPooling2D)	(None, 7, 7, 64)	0
conv2d_62 (Conv2D)	(None, 7, 7, 64)	36928	conv2d_3 (Conv2D)	(None, 7, 7, 64)	16448
activation_83 (Activation)	(None, 7, 7, 64)	0	activation_4 (Activation)	(None, 7, 7, 64)	0
max_pooling2d_62 (MaxPooling2D)	(None, 3, 3, 64)	0	max_pooling2d_3 (MaxPooling2D)	(None, 3, 3, 64)	0
flatten_29 (Flatten)	(None, 576)	0	flatten_1 (Flatten)	(None, 576)	0
dense_79 (Dense)	(None, 64)	36928	dense_3 (Dense)	(None, 64)	36928
dense_80 (Dense)	(None, 64)	4160	dense_4 (Dense)	(None, 64)	4160
dense_81 (Dense)	(None, 3)	195	dense_5 (Dense)	(None, 3)	195
=====			activation_5 (Activation)	(None, 3)	0

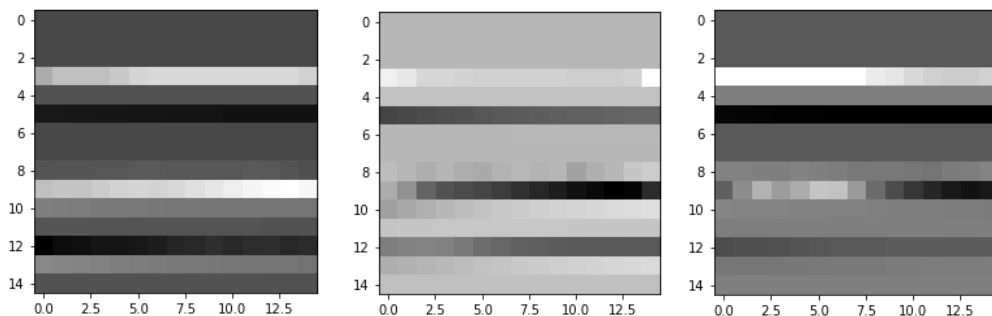
Gambar 5. Summary model CNN yang digunakan

## BAB III

### HASIL PENELITIAN

#### 3.1 Hasil Transformasi Data

Dalam penelitian ini, kami menggunakan pendekatan mengubah deret waktu keuangan 1-D menjadi representasi data seperti gambar 2-D agar dapat memanfaatkan kemampuan *Convolutional Neural Network* dalam untuk sistem *algorithmic trading*. Untuk menghasilkan representasi seperti itu, kami menggunakan 15 indikator teknis yang berbeda dengan berbagai pengaturan parameter masing-masing dengan rentang 15 hari yang disesuaikan untuk mewakili nilai di setiap kolom. Demikian pula, sumbu x terdiri dari rangkaian waktu data senilai 15 hari untuk setiap indikator teknis tertentu pada setiap baris. Baris diurutkan sedemikian rupa sehingga indikator serupa dikelompokkan bersama untuk memenuhi persyaratan lokalitas di sepanjang sumbu y. Hasilnya, gambar berukuran 15x15 pixel dihasilkan dan dimasukkan ke dalam CNN.



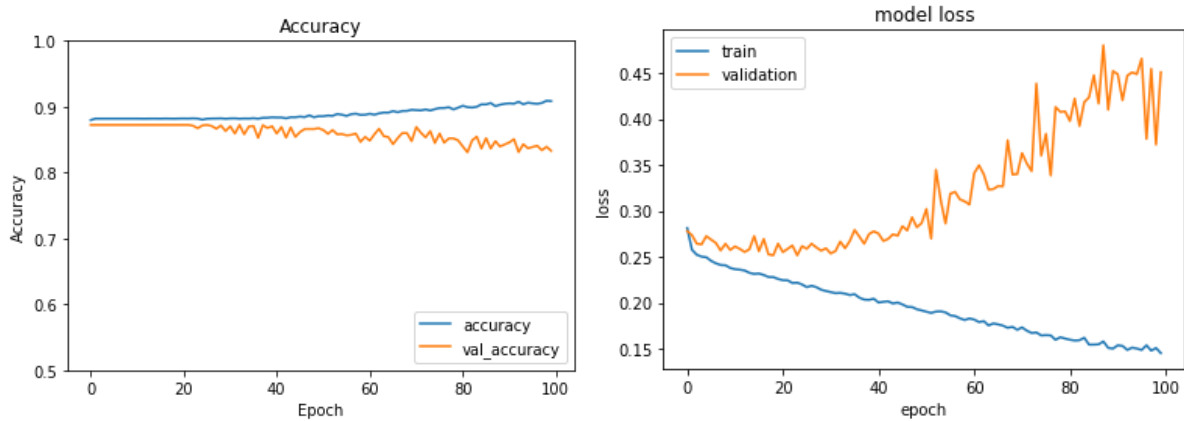
Gambar 6. Contoh gambar yang dihasilkan dari 15 indikator teknikal

*Output* dari model yaitu aksi yang dilakukan apakah akan ‘Buy’, ‘Sell’, atau ‘Hold’. Ketiga label ini direpresentasikan dalam bentuk *one hot encoding* dalam hal ini vektor dengan 3 elemen, yang mana untuk label tertentu, satu elemen diberi angka 1 dan 0 untuk label lainnya.

#### 3.2 Evaluasi Pengambilan Keputusan Jual Beli Saham

##### 3.2.1 Pengaruh emiten saham terhadap hasil evaluasi model

- Saham International Business Machines Corporation (IBM)

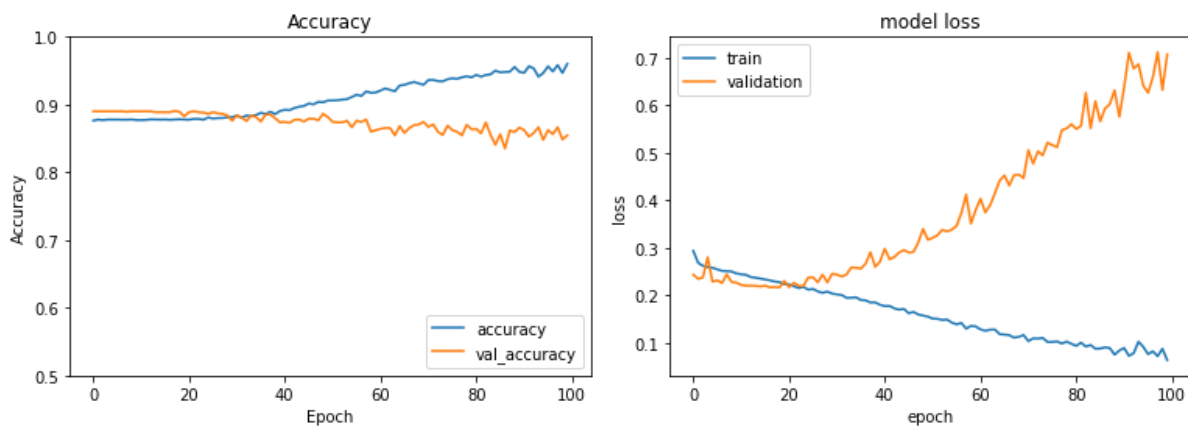


Gambar 7. Akurasi prediksi saham IBM

	Recall	Precision	F1 Score
<b>Action</b>			
<b>Hold</b>	0.221374	0.237705	0.229249
<b>Buy</b>	0.941049	0.886850	0.913146
<b>Sell</b>	0.032000	0.190476	0.054795

Gambar 8. Perbandingan recall, precision, dan F1 score pada setiap keputusan untuk saham IBM

- Saham Apple Inc. (AAPL)



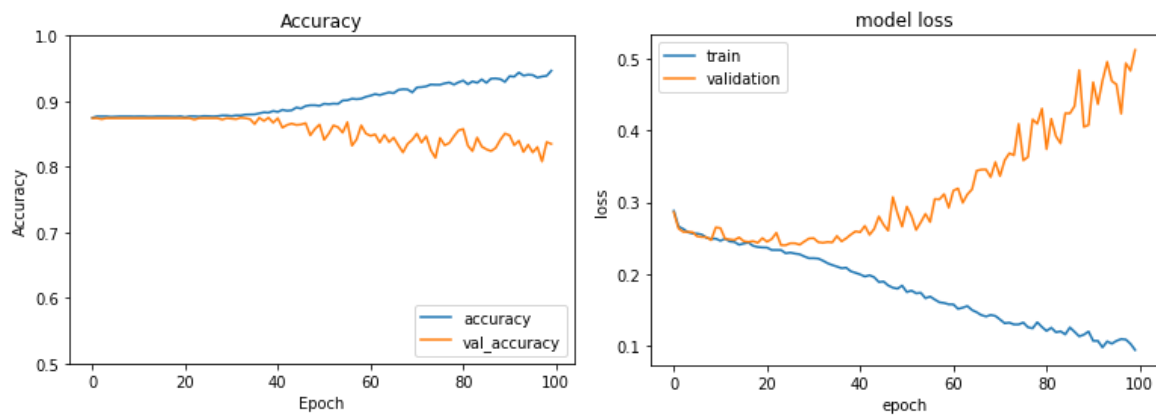
Gambar 9. AAPL



	Recall	Precision	F1 Score
<b>Action</b>			
<b>Hold</b>	0.016949	0.133333	0.030075
<b>Buy</b>	0.989177	0.889105	0.936475
<b>Sell</b>	0.008850	0.125000	0.016529

Gambar 10. AAPL

- Saham Microsoft Corporation (MSFT)

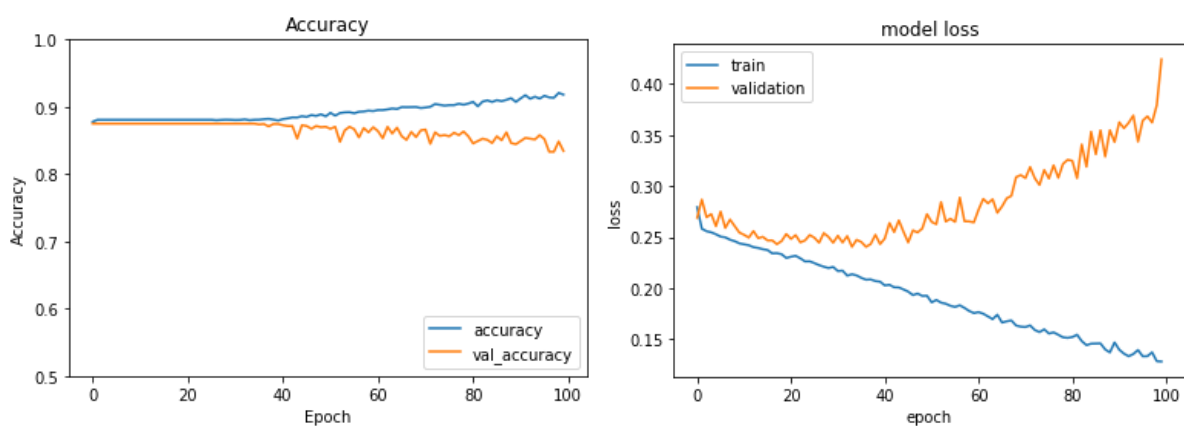


Gambar 11. MSFT

	Recall	Precision	F1 Score
<b>Action</b>			
<b>Hold</b>	0.027273	0.214286	0.048387
<b>Buy</b>	0.943325	0.876536	0.908705
<b>Sell</b>	0.126050	0.159574	0.140845

Gambar 12. MSFT

- Saham JP Morgan Chase & Co. (JPM)

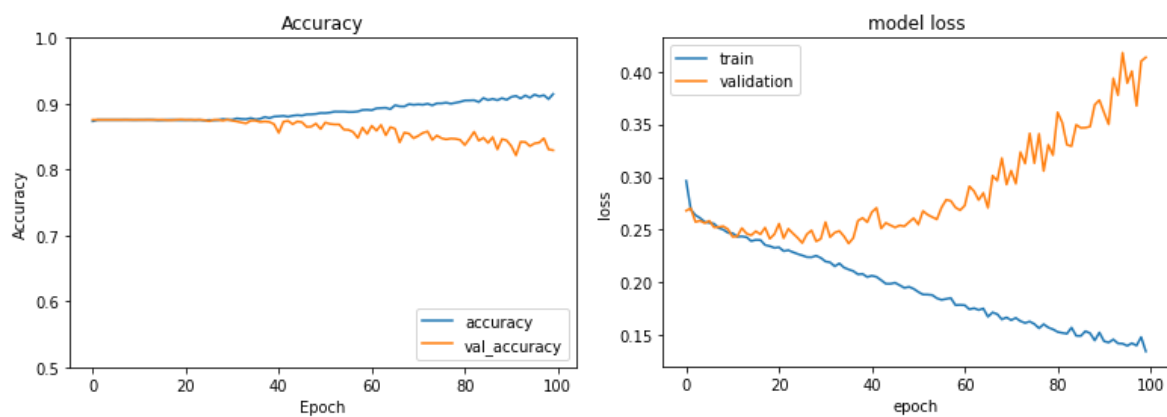


Gambar 13. JPM

	Recall	Precision	F1 Score
<b>Action</b>			
<b>Hold</b>	0.022901	0.142857	0.039474
<b>Buy</b>	0.978004	0.881528	0.927263
<b>Sell</b>	0.040000	0.161290	0.064103

Gambar 14. JPM

- Saham The Coca Cola Company (KO)

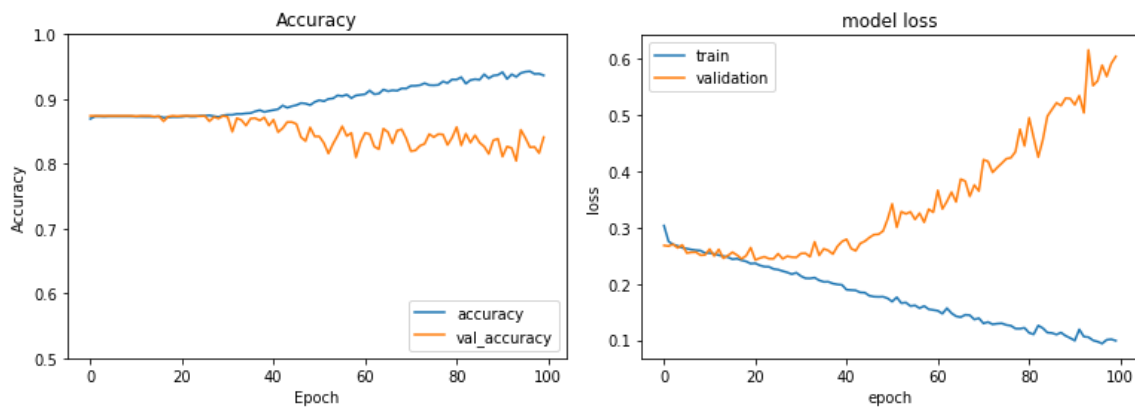


Gambar 15. KO

	Recall	Precision	F1 Score
<b>Action</b>			
<b>Hold</b>	0.200000	0.189055	0.194373
<b>Buy</b>	0.909535	0.887871	0.898572
<b>Sell</b>	0.083799	0.145631	0.106383

Gambar 16. KO

- Saham Honda Motor Co., Ltd. (HMC)

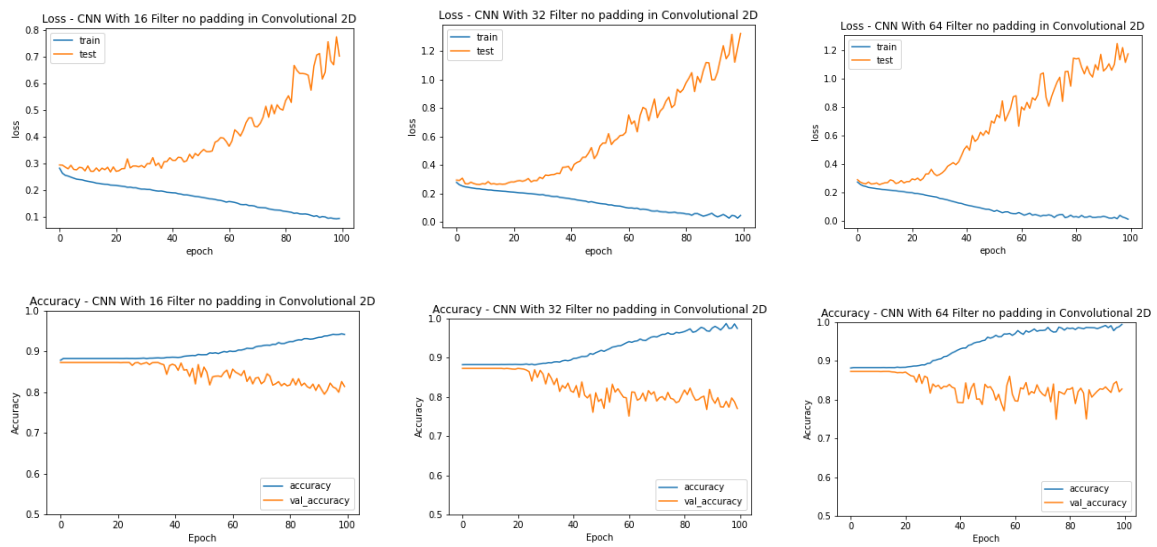


Gambar 17. HMC

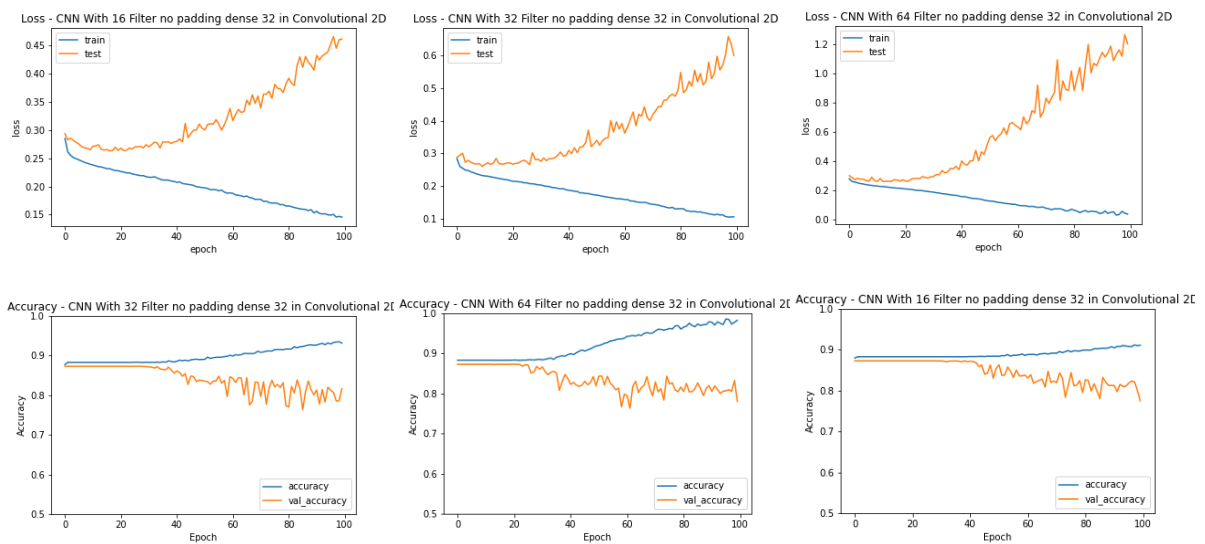
	Recall	Precision	F1 Score
<b>Action</b>			
<b>Hold</b>	0.148148	0.240964	0.183486
<b>Buy</b>	0.931980	0.885819	0.908313
<b>Sell</b>	0.150794	0.228916	0.181818

Gambar 18. HMC

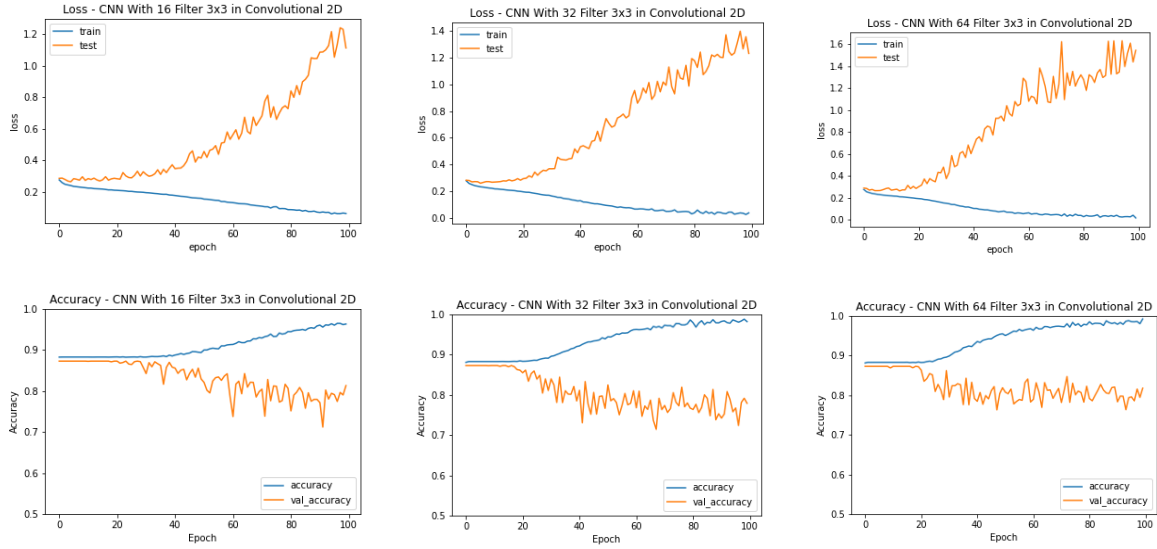
### 3.2.2 Pengaruh perubahan hyperparameter terhadap hasil evaluasi model



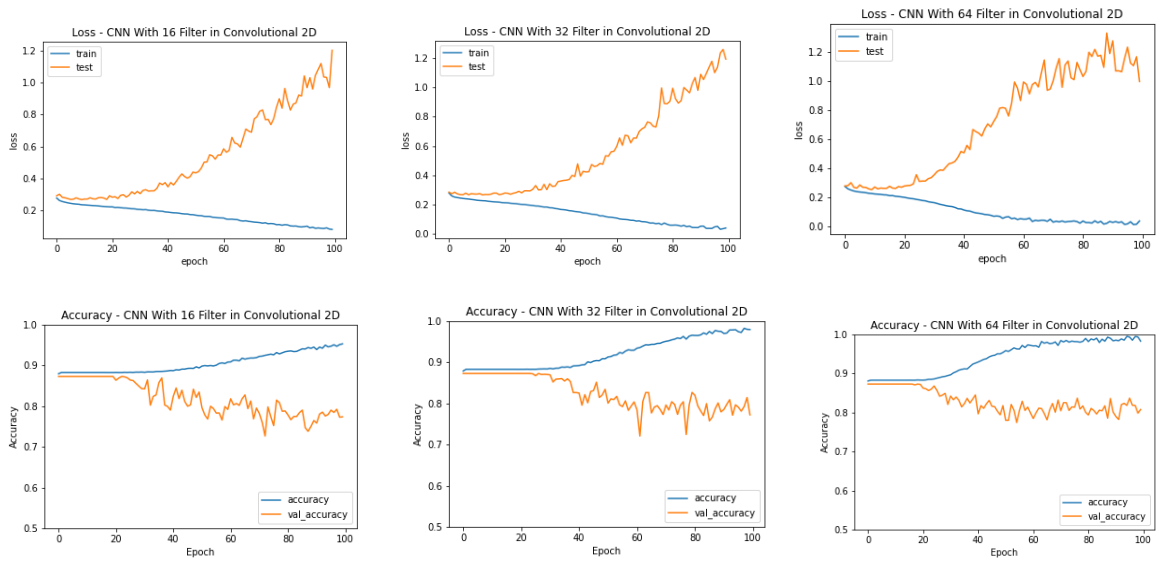
Gambar 19. Grafik loss dan akurasi model dengan filter berukuran  $2 \times 2$  dan tanpa padding dengan dense 64



Gambar 20. Grafik loss dan akurasi model dengan filter berukuran  $2 \times 2$  dan tanpa padding dengan dense 32



Gambar 21. Grafik loss dan akurasi model dengan filter berukuran  $3 \times 3$  dan padding, juga dense 64



Gambar 22. Grafik loss dan akurasi model dengan filter berukuran  $2 \times 2$ , dengan padding dan dense 64

## **BAB IV**

### **DISKUSI**

#### **4.1 Pola Yang Mirip Terlepas dari Input Data dan Hyperparameter**

Dengan menggunakan beberapa kemungkinan nilai *hyperparameter* dan input saham yang berbeda, diperoleh pola yang sama yaitu nilai yang tidak sinkron antara akurasi dan *loss* pada data train dan validation. Jadi, perubahan *hyperparameter* tidak terlalu berpengaruh pada kecenderungan *overfitting*, meskipun secara umum peningkatan nilai *hyperparameter* atau dengan kata lain semakin kompleks model maka nilai akurasi pada data train membesar dan nilai *loss* pada model mengecil dengan cepat.

#### **4.2 Terjadinya Overfitting**

Dari hasil yang diperoleh, akurasi dan perkembangan loss function pada saat training bagus, tetapi saat divalidasi menjadi kurang bagus. Hal ini terlihat dari nilai akurasi yang meningkat pada data train, tetapi pada data validation mengalami penurunan. Di sisi lain, nilai *loss function* selalu menurun untuk data train dan justru cenderung naik tetapi tidak monoton naik untuk data validation. Arsitektur model yang digunakan tidak terlalu kompleks, hanya dilakukan konvolusi, penggunaan fungsi aktivasi ReLU, dan *max pooling* masing-masing dua kali. Jika model terlalu kompleks, maka kemungkinan dihasilkan overfitting lebih besar.

#### **4.3 Hasil Rekomendasi yang Hanya Baik Untuk Rekomendasi ‘Buy’ Saja**

Jika dilihat dari precision, recall, dan F1 score-nya, hanya satu jenis prediksi saja yang memberikan hasil evaluasi yang baik, yaitu ‘Buy’. Hasil percobaan menggunakan beberapa emiten saham menunjukkan bahwa dari semua percobaan, untuk rekomendasi ‘Buy’ model mampu memberikan nilai precision, recall, dan F1 score lebih dari 80%. Sedangkan untuk rekomendasi ‘Hold’ dan ‘Sell’, model hanya mampu memberikan nilai precision, recall, dan F1 score kurang dari 30%. Jadi, jika model memberikan rekomendasi ‘Buy’, kita dapat mengikuti rekomendasi model. Sedangkan untuk rekomendasi ‘Hold’ dan ‘Sell’ sebaiknya tidak diikuti.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan penelitian ini, kita dapat menggunakan 15 indikator teknikal untuk dapat mengkonstruksi suatu gambar berukuran 15x15 pixel sedemikian sehingga gambar tersebut dapat digunakan untuk memberikan prediksi pergerakan harga saham dan rekomendasi keputusan jual beli saham dengan model Convolutional Neural Network. Berdasarkan hasil yang diperoleh, arsitektur yang penulis gunakan dapat memberikan hasil yang mirip dengan tidak terpengaruh oleh perubahan hyperparameter maupun input data. Namun, arsitektur yang digunakan masih mengalami overfitting sehingga performanya saat diuji tidak sebaik performa pada saat training. Selain itu, arsitektur ini hanya mampu memberikan hasil yang bagus untuk rekomendasi 'Buy' saja, sedangkan untuk rekomendasi 'Hold' dan 'Sell' memberikan hasil yang buruk. Untuk rekomendasi 'Buy' model mampu memberikan nilai precision, recall, dan F1 score lebih dari 80%. Sedangkan untuk rekomendasi 'Hold' dan 'Sell', model hanya mampu memberikan nilai precision, recall, dan F1 score kurang dari 30%. Jadi, jika model memberikan rekomendasi 'Buy', kita dapat mengikuti rekomendasi model. Sedangkan untuk rekomendasi 'Hold' dan 'Sell' sebaiknya tidak diikuti.

#### **5.2 Saran**

Untuk memastikan apakah performa model sudah baik atau belum, sebaiknya perlu dicoba model dengan arsitektur lainnya terutama model-model yang sudah umum digunakan sebagai benchmark, kemudian dibandingkan dengan model ini supaya terlihat apakah model ini mampu memberikan performa yang lebih baik dari benchmark atau tidak.

## REFERENSI

- Behl, A., Bhatia, A., & Putri, A. (2014): Convolution and Applications of Convolution. International Journal of Innovative Research in Technology Volume 1 Issue 6, 2122 - 2166.
- Berat Sezer, Omar, Ahmet Murat Ozbayoglu. (2018). Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. Applied Soft Computing, Volume 70, Pages 525-538.
- 2018,Borisov, V., Haug, J., & Kasneci, G. (2019): CancelOut: a Layer for Feature Selection in Deep Neural Network. International Conference on Artificial Neural Networks (pp. 72 - 83). Jerman: Springer International Publishing.
- Chollet, Francois. (2018): Deep Learning with Python, Manning Publications Co : US
- Genç, Ö. (2019, January 26): Notes on Artificial Intelligence, Machine Learning and Deep Learning for curious people. Diambil dari Towards Data Science: <https://towardsdatascience.com/notes-on-artificial-intelligence-ai-machine-learning-ml-and-deep-learning-dl-for-56e51a2071c2>. diakses pada 15 Juni 2020
- Gudelek, M. Ugur. (2017): A Deep Learning Based Stock Trading Model with 2-D CNN Trend Detection, IEEE 2017 Symposium Series on Computational Intelligence (SSCI),
- Hua, Y., Mou, L., & Zhu, X. X. (2018): LAHNET: A Convolutional Neural Network Fusing Low- and High-level features for Aerial Scene Classification. IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, 4728 - 4731.
- Kusuma, Rosdyana Mangir Irawan dkk. (2018): Using Deep Learning Neural Network and Candlestick Chart Representation to Predict Stock Market [Thesis]. Chungli (TW): Yuan Ze University. arXiv preprint arXiv: 1903.12258
- Lin, G., & Shen, W. (2018): Research on Convolutional Neural Network based on Improved Relu Piecewise Activation Function. International Congress of Information and Communication Technology (ICICT-2018) Volume 8, 977 - 984.
- Liu, Y. H. (2018): Feature Extraction and Image Recognition with Convolutional Neural. Journal of Physics Conference Volume 1087.



- Nwankpa, C. E., Ijomah, W., Gachagan, A., & Marshall, S. (2018): Activation Functions: Comparison of Trends in Practice and Research for Deep Learning. arXiv:1811.03378v1 [cs.LG] 8 Nov 2018.
- Saeedan, F., Weber, N., Goesele, M., & Roth, S. (2018): Detail-Preserving Pooling in Deep Networks. e IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 9108 - 9116.
- Scherer, D., Muller, A., & Behnke, S. (2010): Evaluation of Pooling Operations in. 20th International Conference on Artificial Neural Networks (ICANN).
- Sree, K. D. (2019): Data Analytics: Role of Activation function. International Journal of Innovative Technology and Exploring Engineering, 229 - 301.

## KONTRIBUSI

- Yayan Febriansyah (10118017):

Membuat kodingan, melakukan eksperimen dengan perubahan nilai hyperparameter, membuat abstrak, menambahkan manfaat penelitian, menulis metodologi bagian teori CNN, menulis hasil penelitian bagian pengaruh perubahan hyperparameter terhadap hasil evaluasi model, menambahkan bagian BAB IV Diskusi, menambahkan bagian arsitektur model.

- Arya Pratama Putra (10118104):

Melakukan percobaan menggunakan beberapa input data emiten saham berbeda berdasarkan kode yang dibuat oleh Yayan, membuat template makalah, mengerjakan makalah BAB I, BAB II, BAB III, BAB IV, dan BAB V.