

BAB II

KAJIAN LITERATUR

2.1 Tinjauan Pustaka

Kerangka konsep penelitian ini dimulai dari pengenalan prediksi harga saham. Selanjutnya, akan membahas mengenai *attention mechanism* (AM) dilanjutkan dengan membahas mengenai model *deep learning* TCN dan diakhir akan membahas model LSTM yang akan digunakan dalam memprediksi harga saham.

2.1.1 Prediksi Harga Saham

Prediksi harga saham adalah proses memperkirakan harga saham di masa yang akan datang pada suatu perusahaan atau instrumen keuangan lain berdasarkan pola dan informasi masa lalu maupun masa sekarang yang dimiliki. Prediksi tersebut menjadi fokus karena memungkinkan para pelaku pasar saham untuk memaksimalkan pendapatan [16]. Hingga saat ini, tingkat keakuratan prediksi harga saham terus ditingkatkan dengan berbagai usulan algoritma dan teknik [17]. Hal ini dikarenakan harga saham tidak hanya dipengaruhi oleh faktor internal melainkan juga faktor eksternal yang tidak terbatas seperti pada kondisi politik, ekonomi global, laporan keuangan, kinerja perusahaan, dan lain sebagainya. Oleh karena itu, terdapat dua pendekatan utama untuk melakukan prediksi harga saham, yaitu: [18, 19]

1. Analisis secara teknis yang dilakukan dengan menggunakan harga historis saham yaitu *open price*, *closing price*, *highest price*, *lowest price*, *trading volume*, dan *adjusted close price* untuk memperkirakan harga saham di masa yang akan datang.
2. Analisis secara kualitatif yang berdasarkan faktor eksternal yang mempengaruhi, seperti situasi pasar, faktor politik, faktor ekonomi, informasi baik melalui media sosial ataupun para analis ekonomi.

Pendekatan di atas sebelumnya, telah dijalankan secara tradisional yang didasarkan pada asumsi teoretis dan implementasi sederhana sehingga tidak dapat secara efektif menangkap sifat *nonlinear* dari urutan dalam data skala besar. Selain itu, dibandingkan dengan model sebelumnya, model *nonlinear* telah menghasilkan prediksi yang lebih akurat [7, 20] Namun, kemampuan untuk menangkap ketergantungan model dalam jangka panjang cenderung rendah, Dalam tahun-tahun terakhir ini, metode DL telah menjadi pendekatan utama prediksi harga saham. Beberapa di antaranya, RNN telah menunjukkan keunggulan dalam tugas pemodelan urutan, LSTM mampu menangkap ketergantungan jangka panjang

dan secara efektif mampu memecahkan masalah hilangnya gradien dalam RNN tradisional. Tidak berhenti disitu saja, model terbaru hasil dari penggabungan arsitektur RNN dan CNN yaitu model TCN. Model ini menggunakan konvolusi kausal untuk memiliki ukuran sejarah yang efektif dan sangat panjang serta berkinerja baik dalam pemodelan urutan [12].

2.1.2 Mekanisme Atensi

Attention Mechanism (mekanisme atensi) adalah mekanisme yang mampu memodelkan ketergantungan antara *output* target dan urutan *input*. Mekanisme ini telah banyak diusulkan dan berhasil diterapkan dalam banyak tugas penelitian seperti teks gambar, klasifikasi dokumen, pengenalan suara, dan lainnya. Mekanisme perhatian membantu model berbasis data untuk lebih fokus pada segmen data yang informatif dan mengabaikan segmen yang kurang berkontribusi pada hasil akhir. Hasilnya, pengetahuan yang didapatkan diinterpretasikan sehingga meningkatkan potensial kinerja algoritma [21].

Adapun mekanisme atensi yang digunakan berbeda dengan mekanisme atensi yang digunakan oleh Paolo Janssen. Mekanisme atensi yang digunakan bersumber dari Bahdanau et al yang juga menjadi referensi dari Paolo Janssen sendiri. Berikut ini langkah-langkah dan rumus-rumus mekanisme atensi seperti berikut [12]:

1. Hitung *alignment scores*

$$e_{t,i} = a(s_{t-1}, h_i) \dots \dots \dots (2.1)$$

2. Hitung weights

$$\alpha_{t,1} = softmax(e_{t,i}) \dots \dots \dots (2.2)$$

3. Hitung *context vector*

$$c_t = \sum_{i=1}^T \alpha_{t,1} h_i \dots \dots \dots (2.3)$$

Dimana:

$e_{t,i}$ = *alignment scores*

a = *alignment model*

s_{t-1} = *previous decoder output*

h_i = *hidden states*

$\alpha_{t,1}$ = *weights*

c_t = *context vector*

T = *weighted sum of all*

2.1.3 Long Short-Term Memory

Long Short-Term Memory (LSTM) pertama kali diusulkan oleh Hochreiter dan Schmidhuber pada tahun 1997, dan kemudian menjadi sangat populer khususnya untuk mengatasi masalah prediksi deret waktu. LSTM adalah RNN yang dimodifikasi yang bekerja dengan baik pada berbagai macam masalah, dan digunakan secara luas sekarang. LSTM menangani masalah mencari tahu bagaimana mengumpulkan kembali data selama periode waktu tertentu, dengan menghadirkan *gate unit* dan *memory cells* dalam desain jaringan saraf. Perilaku *default* LSTM adalah mengingat informasi untuk jangka waktu yang lama, bukan sesuatu yang dipelajari melalui perjuangan.

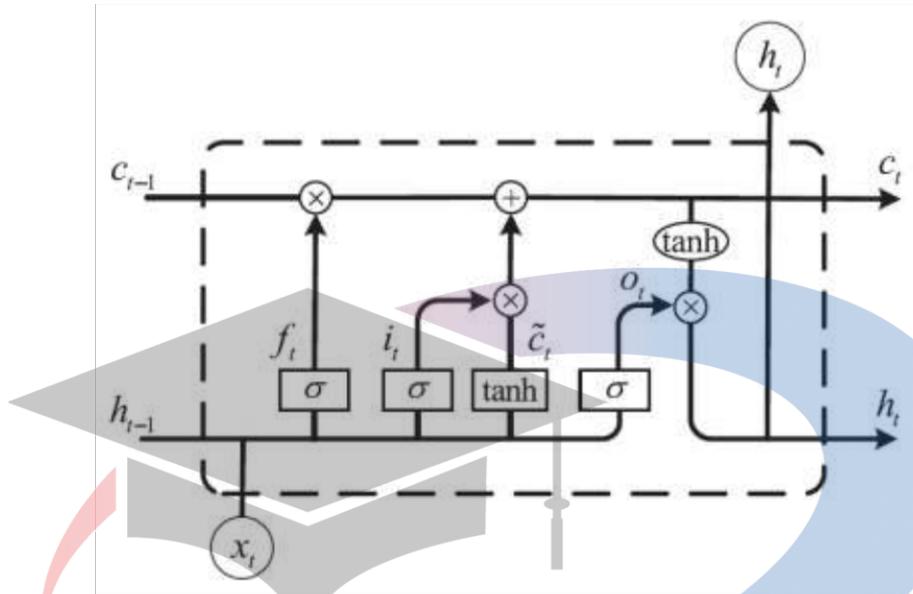
LSTM dapat mengingat deret waktu karena setiap unit sel di LSTM berisi tiga gerbang logika berdasarkan lapisan jaringan saraf sigmoid, yaitu *input gate*, *output gate*, dan *forgetting gate* yang melaluinya data dapat dilewatkan atau diproses secara selektif. Setiap lapisan jaringan sigmoid dapat menghasilkan angka dari 0 hingga 1 untuk menggambarkan jumlah segmen data yang dapat melewati setiap unit. Jika ditampilkan sebagai 0, berarti tidak ada data yang boleh lewat, dan jika 1, berarti semua data bisa lewat. *Forgetting gate* menampilkan data dari 0 hingga 1, di mana 1 berarti "sepenuhnya dicadangkan" dan 0 berarti "sepenuhnya diabaikan". *Storage gate* terdiri dari lapisan sigmoid (*input gate*) dan lapisan tanh, dan untuk memilih data baru yang dapat disimpan di dalam sel. Lapisan sigmoid memilih nilai yang perlu disesuaikan, dan lapisan tanh menghasilkan vektor nilai kandidat baru, yang kemudian ditambahkan ke keadaan unit sel. *Input gate* merangkum nilai status unit sel, nilai yang difilter, dan nilai tambah untuk menghasilkan nilai baru. Adapun kelebihan LSTM dijabarkan seperti berikut ini:

1. Mampu mempelajari dan mengingat ketergantungan jangka panjang dalam data
2. Mampu menangani urutan input dengan panjang variabel
3. Mampu memproses data secara end-to-end tanpa perlu pra-pemrosesan atau rekayasa fitur sehingga lebih mudah dan efisien

Sedangkan kelemahan yang dimiliki oleh LSTM yaitu:

1. Jaringan lebih sulit dilatih dan disempurnakan daripada model DL yang lain karena arsitekturnya yang kompleks dan kebutuhan untuk memilih dan menyetel hyperparameter jaringan dengan hati-hati.
2. Jaringan dapat intensif secara komputasi terutama pada kumpulan data besar yang dapat membuatnya lebih sulit untuk digunakan dalam aplikasi *real-time*
3. Jaringan yang sulit untuk diinterpretasikan dan dipahami sehingga sulit untuk memecahkan masalah dan menjelaskan hasil prediksi kepada orang lain

Model jaringan saraf LSTM merupakan jenis dari *Recurrent Neural Networks* (RNN). Struktur jaringan LSTM menggunakan mekanisme *gate control* yang terdiri dari *forget gate*, *input gate*, *cell state*, dan *output gate*. Model jaringan saraf LSTM memiliki fungsi memori jangka panjang yang dapat dilihat pada gambar 2.1 [12].



Gambar 2.1 Jaringan Saraf Long Short-Term Memory (LSTM) [12]

Langkah-langkah algoritma LSTM sebagai berikut [12]:

1. Hitung fungsi aktivasi *forget gate* dengan rumus:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \dots \dots \dots (2.4)$$

2. Hitung *input gate* menggunakan rumus berikut:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \dots \dots \dots (2.5)$$

3. Hitung *cell state* \tilde{c}_t sebagai vektor kandidat baru menggunakan tanh layer dengan rumus berikut:

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \dots \dots \dots (2.6)$$

4. Berdasarkan kalkulasi di atas, diperoleh nilai perubahan *cell state* pada waktu t dan dirumuskan sebagai berikut

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \dots \dots \dots (2.7)$$

5. Nilai *output gate* diperoleh dengan rumus sebagai berikut:

$$O_t = \sigma(W_o [h_{t-1}, x_t] + b_o) \dots \dots \dots (2.8)$$

6. Lapisan *hidden layer* berikutnya dapat diperoleh dengan rumus berikut:

$$h_t = O_t * \tanh(c_t) \dots \dots \dots (2.9)$$

Dimana:

x_t = masukan dari *node* saat ini

C_t = mewakili vektor nilai kandidat di proses perhitungan

σ = mewakili fungsi aktivasi sigmoid *feedforward* dari lapisan jaringan

tanh = mewakili fungsi aktivasi dari lapisan jaringan *feedforward*.

W_i, W_f, W_o, W_c = mewakili *weight matrices* pada neuron

b_i, b_f, b_o, b_c = *bias terms*

* = perkalian titik

. = perkalian biasa

2.1.4 Temporal Convolutional Network

Temporal Convolutional Network (TCN) adalah model hasil gabungan dari arsitektur RNN dan CNN. TCN didasarkan pada dua prinsip, yaitu: konvolusi dalam arsitektur bersifat kausal sehingga tidak ada kebocoran informasi dari masa depan ke masa lalu dan arsitektur yang mampu memetakan urutan dengan panjang ke *output* dengan panjang yang sama [22]. TCN menggunakan konvolusi melebar untuk mencapai bidang reseptif yang besar dan fleksibel (*Dilated convolutions*) serta memanfaatkan koneksi sisa untuk memungkinkan jaringan mempelajari dan memodifikasi pemetaan identitas, mencegah ledakan, dan masalah gradien yang menghilang (*Residual connections*) [15]. Adapun kelebihan dari TCN ini yaitu [22]:

1. Tidak ada kebocoran informasi dari masa depan ke masa lalu
2. Arsitektur yang mampu memetakan urutan dengan panjang ke *output* dengan panjang yang sama
3. Dapat dengan mudah diparalelkan sehingga membuatnya lebih efisien dan lebih cepat untuk dilatih daripada model lain yang lebih sulit diparalelkan
4. Mampu menyesuaikan receptive field size dengan jumlah lapisan, dilation factors, dan ukuran filter sehingga memungkinkan kontrol ukuran memori model untuk kebutuhan domain yang berbeda
5. Jumlah kernel bergantung pada jumlah layer bukan pada panjang input

Sedangkan kelemahan dari TCN yaitu [22]:

1. Memerlukan lebih banyak penyimpanan data daripada RNN
2. Transfer domain yang tidak berfungsi dengan baik

2.1.4.1 Causal Convolutions

TCN tidak bisa menggunakan informasi dari masa depan untuk melakukan proses konvolusi sehingga untuk menghindari hal itu digunakan *causal convolution* yang artinya hanya dapat menggunakan data saat ini dan data yang sebelumnya. Untuk menjaga ukuran lapisan berikutnya sama dengan lapisan masukan, maka akan digunakan teknik *zero padding of length* yang didefinisikan sebagai berikut [15]:

$$P = k - 1 \dots\dots\dots(2.10)$$

Dimana:

P : jumlah *padding*

k : ukuran kernel

2.1.4.2 Dilated Convolutions

Proses konvolusi pada data input yang nantinya menghasilkan data output, hasilnya tergantung dari kernel yang digunakan dan data sebelumnya. Misalkan panjang data input adalah 5 dengan ukuran kernel adalah 3, maka data ke-5 akan sangat bergantung pada data ke-3 dan ke-4. Untuk itu untuk menghindari ketergantungan pada data model mampu menangkap keterikatan dengan data jangka panjang maka digunakanlah konsep *dilated convolutions*, yaitu menambahkan layer lagi di dalam jaringan dan melebarkan jangkauan dari kernel pada layer berikutnya dengan melewati beberapa data. Artinya, data output harus terhubung dengan data input dari masa lalu yang sangat jauh. *Dilated convolution* dapat di definisikan sebagai berikut [15]:

$$F(s) = (x \cdot_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot x_{s-di} \dots\dots\dots(2.11)$$

Dimana:

F = *dilated convolution operation*

s = *sequence*

f = *filter*

d = nilai dari dilatasi,

k = nilai dari kernel

x = nilai input

s - di = menunjukkan arah masa lalu

Pelebaran yang lebih besar memungkinkan *output* dari level yang lebih tinggi mewakili rentang *input* yang lebih luas untuk meningkatkan jangkauan bidang reseptif

jaringan yang nampak pada gambar 2.2 terkait representasi visual dari konvolusi yang melebar. Untuk menggunakan faktor pelebaran yang meningkat secara eksponensial, diperlukan perhitungan mengenai *total receptive field* dapat menggunakan rumus [15]:

$$RF = 2^{L-1} * k * b \dots\dots\dots(2.12)$$

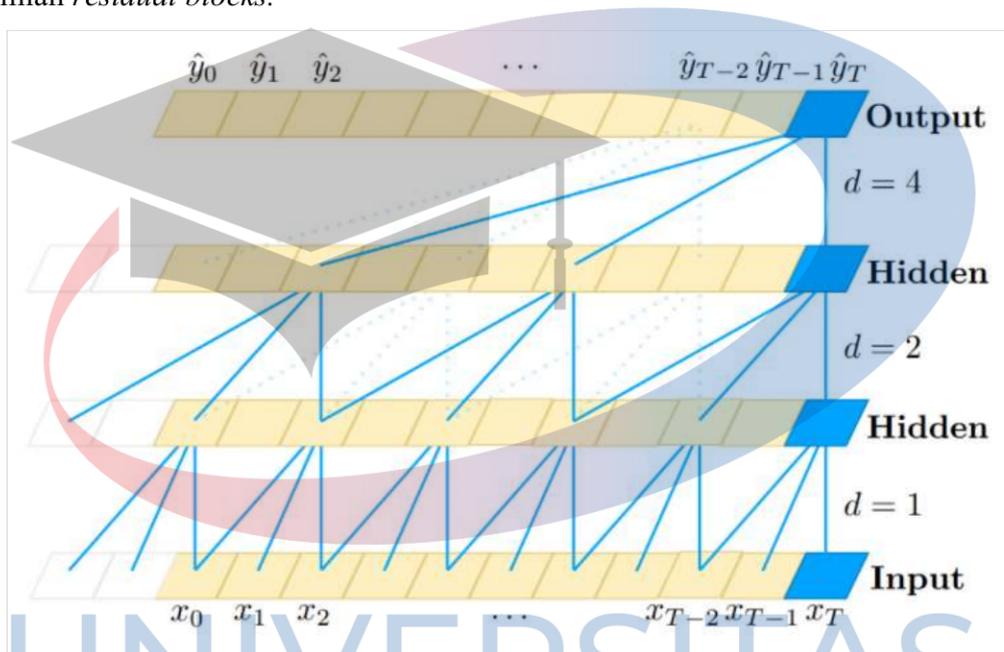
Dimana:

RF = *Receptive field*

L = Panjang layers

k = Ukuran kernel

b = jumlah *residual blocks*.



Gambar 2.2 Representasi Visual dari Konvolusi Dilatasi [15]

Dimana:

Faktor dilatasi d = 1, 2, 4 dan ukuran kernel k = 3

2.1.4.3 Residual Connections

Setiap layer akan saling terhubung dengan layer lainnya, sedangkan jika menggunakan *residual connection* data dapat langsung disalurkan tanpa harus melewati berbagai layer. *Residual connection* juga disebut koneksi pintas yang melakukan pemetaan identitas pada *input* dan menambahkan hasilnya ke *output* lapisan lain. Secara formal, dapat dirumuskan sebagai berikut [15]:

$$o = x + f(x) \dots\dots\dots(2.13)$$

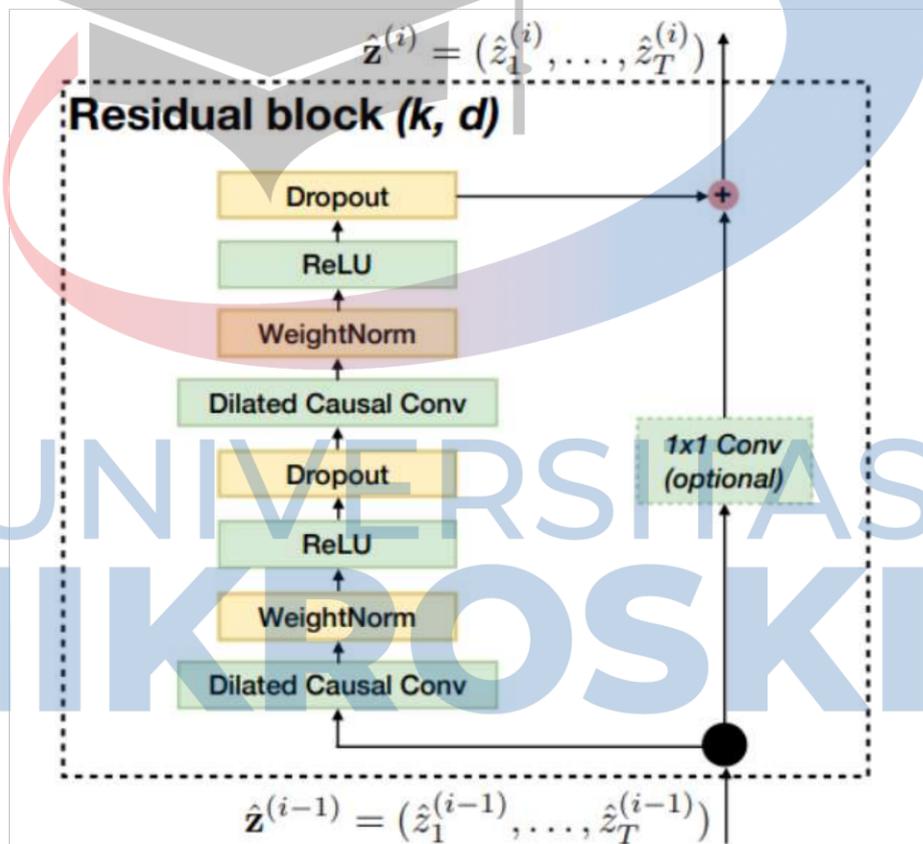
Dimana:

$o = output$

$x = original\ input$

$f = transformasi\ pada\ input\ x$

Menambahkan *input* asli x ke *output* dari serangkaian transformasi memungkinkan lapisan untuk mempelajari modifikasi pada pemetaan identitas bukannya modifikasi pada seluruh transformasi. Ini membantu menstabilkan jaringan yang lebih besar dan lebih dalam dengan mencegah masalah gradien yang meledak dan hilang. Ini juga terdiri dari blok residual yang mengandung konvolusi kausal melebar dan non-linier menggunakan fungsi *Rectified Linear Unit* (ReLU). Tidak hanya itu, untuk memastikan bahwa penambahan elemen menerima tensor dengan bentuk yang sama, konvolusi 1x1 tambahan digunakan pada sambungan residual. Representasi visual *residual block* dapat dilihat pada gambar 2.3 [15].



Gambar 2.3 Representasi Visual dari *Residual Block* [15]

Dimana:

Konvolusi 1x1 ditambahkan ketika Input dan Output Memiliki Dimensi yang Berbeda

2.1.5 Normalisasi

Normalisasi adalah langkah pra-pemrosesan yang penting untuk prediksi harga saham karena dapat membantu membuat data lebih dapat diterima oleh algoritma pembelajaran mesin. Ini penting karena banyak algoritma pembelajaran mesin menganggap bahwa data terdistribusi secara normal dan memiliki rentang atau distribusi nilai tertentu. Jika data tidak memenuhi asumsi ini, kinerja algoritma pembelajaran mesin dapat terpengaruh secara negatif. Normalisasi dapat membantu memperbaiki masalah ini dengan menskalakan data agar memiliki rentang atau distribusi nilai tertentu, yang dapat mempermudah algoritme pembelajaran mesin untuk belajar dari data dan membuat prediksi yang lebih akurat. Selain itu, normalisasi dapat membantu menghilangkan tren dan musiman dari data, yang dapat menjadi penting untuk tugas seperti prediksi harga saham di mana tujuannya adalah untuk mengidentifikasi pola dan hubungan yang mendasari data. Adapun jenis dari normalisasi yang dapat digunakan dapat prediksi harga saham adalah sebagai berikut [23]:

1. *MinMaxScaler* adalah jenis normalisasi yang menskalakan data ke rentang tetap yaitu antara 0 dan 1. Normalisasi ini sering digunakan saat data berisi rentang nilai yang besar yang dapat dirumuskan sebagai berikut

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \dots\dots\dots(2.14)$$

Dimana:

x' = nilai transformasi baru

x = nilai data

x_{max} = nilai terbesar diantara seluruh rangkaian data

x_{min} = nilai terkecil diantara seluruh rangkaian data

2. *Z-Score* adalah normalisasi yang menskalakan data sehingga memiliki rata-rata 0 dan standar deviasi 1. Normalisasi ini sering digunakan saat data terdistribusi normal yang dapat dirumuskan seperti berikut ini

$$Z = \frac{(x - \bar{x})}{stddev(x)} \dots\dots\dots(2.15)$$

Dimana:

x = *original data*

\bar{x} = *mean of the data*

$stddev(x)$ = *standard deviation of the data*

$z = z\text{-score}$

3. *Log Normalization* adalah normalisasi yang menerapkan logaritma natural pada data untuk mengurangi efek kemiringan pada data dan membuatnya lebih simetri yang dapat dirumuskan sebagai berikut:

$$x_{\log} = \ln(x) \dots \dots \dots (2.16)$$

Dimana:

$x_{\log} = \log\text{-normalized data}$

$x = \text{original data}$

4. *Maximum absolute scaling* adalah normalisasi dengan penskalaan absolut maksimum yang menskala ulang setiap fitur antara -1 dan 1 dengan membagi setiap pengamatan dengan nilai absolut maksimumnya.

$$x_{\text{scaled}} = x / \max(\text{abs}(x)) \dots \dots \dots (2.17)$$

Dimana:

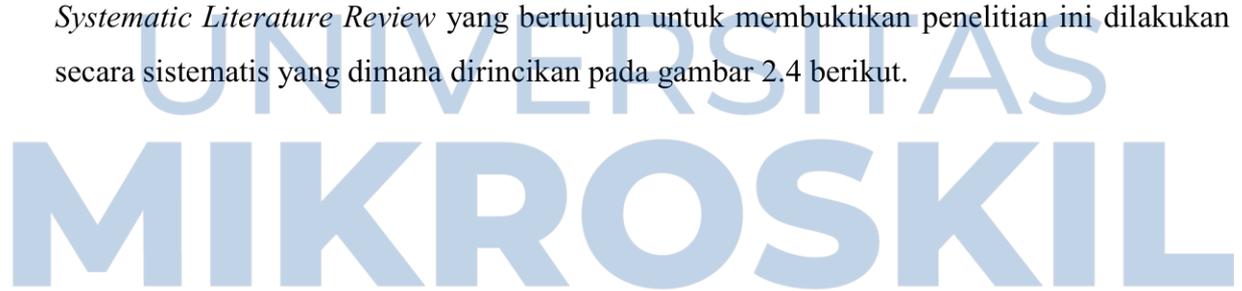
$x_{\text{scaled}} = \text{scaled data}$

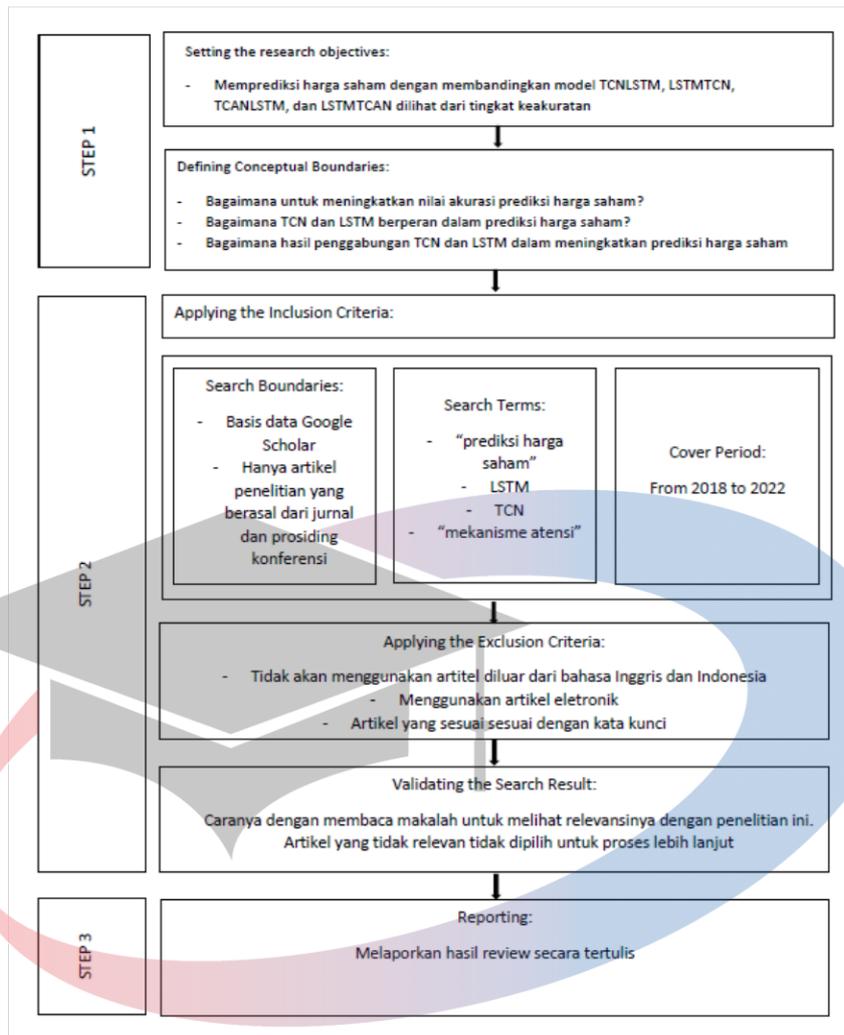
$x = \text{original data}$

$\text{abs} = \text{absolute}$

2.2 Penelitian Terdahulu

Adapun sebelum memulai mencari penelitian terdahulu ini, dilakukan tahapan *Systematic Literature Review* yang bertujuan untuk membuktikan penelitian ini dilakukan secara sistematis yang dimana dirincikan pada gambar 2.4 berikut.





Gambar 2.4 *Systematic Literature Review*

Setelah melewati tahapan tersebut, maka didapatkan beberapa penelitian yang dijadikan sebagai penelitian terdahulu yang dijabarkan secara berurutan. Penelitian pendahuluan pertama ini mengacu pada prediksi harga saham yang menggunakan 2 model yaitu RNN dan LSTM. *Dataset* yang digunakan diambil dari *Yahoo Finance* pada saham GREE dari tanggal 01 Januari 2017 hingga 02 Januari 2020 yang terdiri dari 728 *record* data sebagai data *training* dan dari tanggal 03 Januari 2020 hingga 16 Januari 2020 yang terdiri dari 10 *record* data sebagai data *testing*. Hasilnya LSTM memiliki tingkat akurasi yang lebih baik daripada RNN dilihat dari perhitungan *Mean Squared Error* (MSE) dan *Mean Absolute Error* (MAE). LSTM mendapatkan MSE = 1,8129 dan MAE = 2,7005 sedangkan RNN mendapatkan MSE = 5,2461 dan MAE = 9,9591 [10]. Akan tetapi, hal ini tentu sangat dipengaruhi oleh banyaknya data pengolahan karena seperti yang dikemukakan oleh peneliti lain terdapat keterbatasan model LSTM yang hanya menggunakan ketergantungan pada data masa depan dan tidak secara efisien mempertimbangkan ketergantungan pada data

backward [11]. Selain itu, untuk menutupi keterbatasan di atas peneliti lain membuktikan bahwa dengan menggabungkan 2 model yaitu TCN dan LSTM pada *realistic network traffic* mampu meningkatkan tingkat akurasi dengan sangat baik yang diukur dari *Root Mean Squared Logarithmic Error* (RMSLE) memperoleh nilai 0.0360. Jumlah record data yang digunakan sebanyak 9528 yang diambil dari *Website Spanish Wikipedia* dengan pembagian data *training* dan data *testing* 9:1 [12].

Di tahun yang sama, terdapat penelitian yang menggunakan model yang sama namun dengan urutan penggabungan model yang berbeda yaitu LSTM-TCN. Prediksi terhadap *Cellular Network Traffic* ini menggunakan data lalu lintas kota besar di China yang berisi 500 *base station cell* dan rentang waktu dua bulan, yang terdiri dari 1464 sampel waktu. Hasil prediksi yang dilakukan dibagi menjadi 2 yaitu prediksi untuk 3 hari dan prediksi untuk 7 hari, dimana kedua hasil ini menunjukkan tingkat akurasi yang baik dibanding dengan *Support Vector Regression model* (SVR), *Seasonal Auto-Regressive Integrated Moving Average* (SARIMA), MLP, LSTM, GRU, dan TCN. Hal ini dapat dilihat dari hasil prediksi 3 hari dengan MAE = 0.4489 dan RMSE = 0.6458. Dilanjutkan dengan hasil prediksi 7 hari dengan MAE = 0,4306 dan RMSE = 0,6199 [13].

Tidak berhenti disitu saja, model yang sama LSTM-TCN juga digunakan dalam prediksi oksigen terlarut dalam lingkungan *aquaculture* oleh peneliti lain. *Dataset* yang digunakan adalah data kualitas air yang dikumpulkan dari *industrial recirculating aquaculture workshop* dengan menggunakan sensor kualitas air *multi-parameter* yang diambil dari rentang tanggal 01 Januari 2019 hingga 09 Juli 2019 setiap 10 menit. Data kualitas air yang dikumpulkan meliputi 6 karakteristik yaitu suhu air, pH, kekeruhan, ammonia nitrogen, dan ketinggian air yang akan digunakan sebagai data *input*. Data yang berhasil dikumpul berisi 27.077 sampel data dimana 24.370 data dipakai sebagai data *training* dan sisanya 2.707 data dipakai sebagai data *testing*. Hasilnya, LSTM-TCN pada penelitian ini memiliki tingkat akurasi yang baik yang diukur dan didapatkan MAE = 0,236, MAPE = 0,031, RMSE = 0,342, dan $R^2 = 0,94$ [14].

Pada penelitian selanjutnya, peneliti lain mengusulkan arsitektur model *attention mechanism* yang mampu meningkatkan akurasi prediksi harga saham dari model TCN. Untuk membuktikan hal tersebut, maka dilakukan perbandingan ke-4 model yaitu TCN standar, *Attention Temporal Convolutional Network* (ATCN), *Hierarchical Attention Temporal Convolutional Network* (HATCN), dan *Temporal Attention Convolutional Network* (TCAN). Data yang tersedia berasal dari *Yahoo Finance* mulai dari tahun 1973 sampai 02 Oktober 2021 yang mencakup variabel dasar seperti *open price*, *closing price*,

highest price, lowest price, dividends, stock splits dan *trading volume*. Data tersebut dibagi menjadi 2 dimana data dari tahun 2000 hingga 2014 digunakan sebagai data *training* dan data dari tahun 2015 hingga 02 Oktober 2021 digunakan sebagai data *testing*. Adapun saham yang digunakan adalah saham Apple Inc (AAPL), Heineken (HEIA), dan PostNL (PNL). Hal ini dikarenakan, ketiga saham ini memiliki perbedaan dari sisi sektor, ukuran kapitalisasi pasar, tingkat internasional, dan tingkat keberhasilan sehingga pengujian ini menjadi ideal untuk menguji kemampuan arsitektur model. Hasil yang didapatkan yaitu TCAN menjadi model terbaik rata-rata MSE yaitu 0.00082 yang mampu mengeneralisasi dengan baik dengan jumlah lapisan yang rendah [15]. Sehingga jika diringkas artikel yang digunakan maka akan dihasilkan tabel 2.1 berikut:

Tabel 2.1 Rangkuman Penelitian Terdahulu

| Study | Study Purpose | Dataset | Model | Findings |
|---|---|--|---|--|
| D. Liu, A. Chen, dan J. Wu (2020) | Prediksi harga saham dengan membandingkan 2 model | Saham GREE dari rentang 01 Januari 2017 hingga 02 Januari 2020 | <i>Recurrent Neural Networks</i> (RNN) dan <i>Long Short-Terms Memory</i> (LSTM), | Hasilnya LSTM unggul dibanding RNN dengan MSE dan MAE secara berturut-turut pada LSTM adalah 1,8129 dan 2,7005 dibandingkan dengan RNN mendapatkan MSE= 5,2461 dan MAE = 9,9591. |
| J. Bi, X. Zhang, H. Yuan, J. Zhang dan M. Zhou (2021) | Prediksi <i>realistic network traffic</i> | Website Spanish Wikipedia | <i>Temporal Convolutional Network</i> (TCN), <i>Long</i> | Hasil RMSLE yang didapatkan adalah 0,0360 |

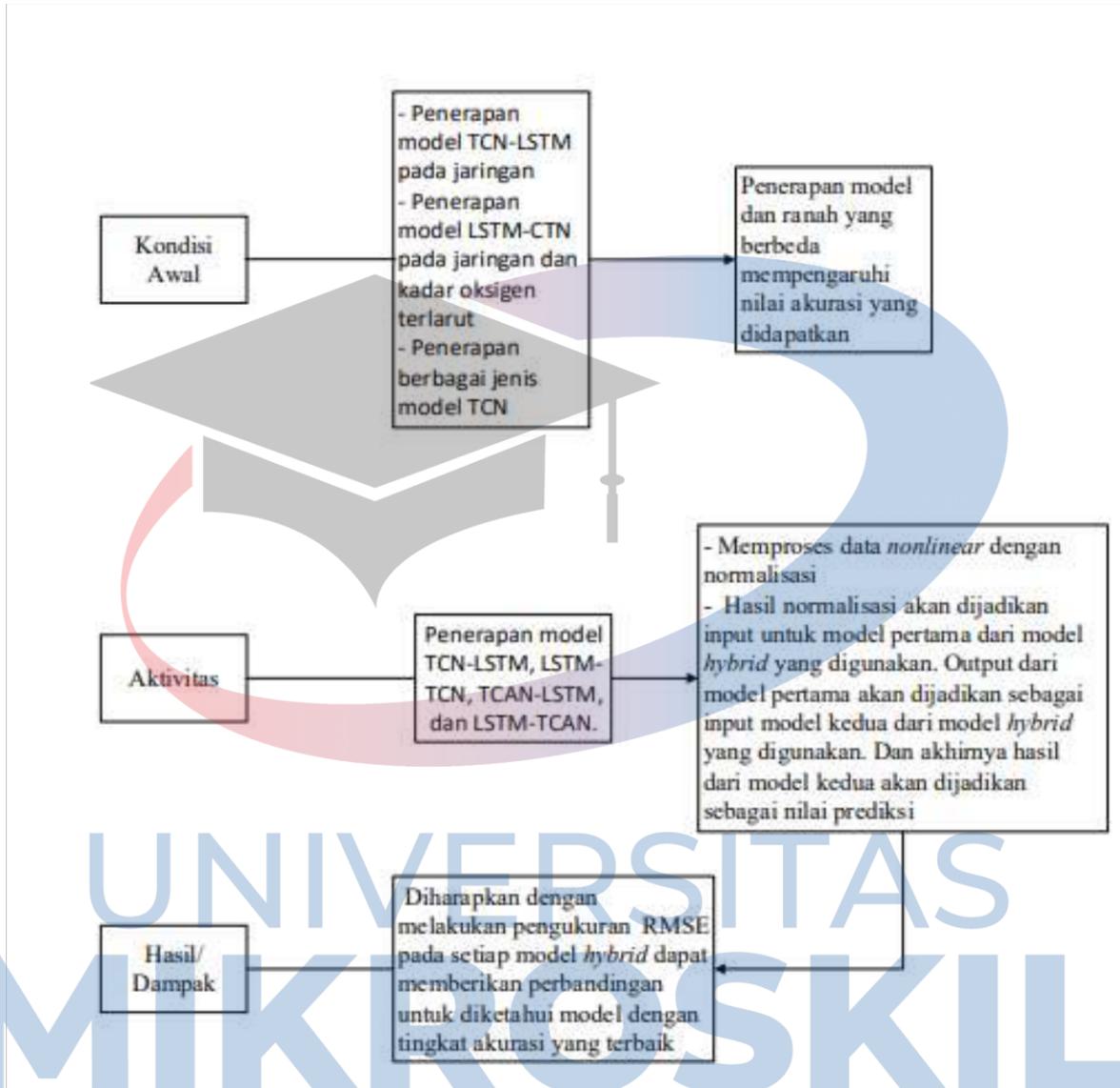
| | | | | |
|---|---|--|---|---|
| | dengan model <i>hybrid</i> | | <i>Short-Terms Memory</i> (LSTM), dan filter <i>Savitzky-Golay</i> (SG) | |
| X. Xu, S. Gao dan Z. Jiang (2021) | Prediksi <i>cellular network traffic</i> dengan model <i>hybrid</i> | Data lalu lintas kota besar China yang berisi 500 <i>base station cell</i> dan rentang waktu dua bulan. | <i>Long Short-Terms Memory</i> (LSTM) dan <i>Temporal Convolutional Network</i> (TCN) | Hasil pengujian pada 3 hari data menghasilkan MAE 0,04489 dan RMSE 0,6458 dan 7 hari dengan MAE 0,4306 dan RMSE 0,6199. |
| W. Li, Y. Wei, D. An, Y. Jiao dan Q. Wei (2022) | Prediksi oksigen yang terlarut dalam lingkungan <i>aquaculture</i> dengan model <i>hybrid</i> | Data kualitas air yang dikumpulkan oleh <i>industrial recirculating aquaculture workshop</i> dari rentang tanggal 01 Januari 2019 hingga 09 Juli 2019 setiap 10 menit. | <i>Long Short-Terms Memory</i> (LSTM) dan <i>Temporal Convolutional Network</i> (TCN) | Hasil yang didapatkan adalah MAE 0,236, MAPE 0,031, RMSE 0,342 dan R^2 0,94. |
| J. Paolo (2022) | Prediksi harga saham dengan membandingkan 4 model TCN | Saham Apple Inc (AAPL), Heineken (HEIA), dan PostNL (PNL) dengan rentang | TCN standar, <i>Attention Temporal Convolutional Network</i> (ATCN), | Hasil yang didapatkan yaitu TCAN menjadi model terbaik rata-rata |

| | | | | |
|--|--|--|--|-----------------------|
| | | 01 Januari 2000 hingga 02 Oktober 2021 | <i>Hierarchical Attention Temporal Convolutional Network (HATCN), dan Temporal Attention Convolutional Network (TCAN).</i> | MSE yaitu 0.00082. |
|--|--|--|--|-----------------------|

Dari penelitian di atas, dapat dilihat bahwa peneliti terakhir masih belum mencoba melihat keterbatasan model TCN seperti pada penelitian lain dimana TCN lemah dalam adaptasi dalam pembelajaran transfer dan jumlah informasi sejarah yang membatasi perkembangannya [24]. Namun, peneliti sebelumnya pula telah membuktikan bahwa penggabungan model TCN-LSTM dapat membantu meningkatkan akurasi dibandingkan prediksi yang dilakukan hanya dengan masing-masing TCN maupun LSTM [12]. Hingga saat ini, belum terdapat penelitian yang melakukan kombinasi model TCN-LSTM ataupun LSTM-TCN pada prediksi harga saham dan diperlukan pula penelitian terkait dengan kombinasi model TCAN-LSTM atau LSTM – TCAN pada prediksi harga saham. Oleh karena itu, pada penelitian ini akan dilakukan pengujian terhadap model gabungan TCN-LSTM, LSTM-TCN, TCAN-LSTM, dan LSTM-TCAN yang kemudian akan dibandingkan keakuratan dilihat dari pengukuran RMSE. *Dataset* yang digunakan sama seperti *dataset* yang digunakan oleh peneliti terakhir di atas dimana berasal dari *Yahoo Finance* dengan membuang celah hari yang disebabkan oleh akhir pekan dan hari libur. Rentang waktu yang digunakan dari tanggal 01 Januari 2000 hingga 02 Oktober 2021, pada saham Apple Inc (AAPL), Heineken (HEIA), dan PostNL (PNL) [15]. Diharapkan dengan penelitian ini, mampu mengatasi karakteristik data harga saham yang fluktuatif, kompleks dan *chaos* sehingga tingkat akurasi prediksi yang didapatkan lebih tinggi.

2.3 Kerangka Konseptual

Kerangka ini menggambarkan bagaimana masalah penelitian yang dapat diselesaikan melalui solusi yang diusulkan dan memberi dampak yang dapat menyelesaikan permasalahan penelitian. Kerangka ini dapat dilihat pada gambar 2.5.



Gambar 2.5 Kerangka Konseptual

Pada gambar 2.5 di atas, kondisi awal dimulai dari adanya penelitian yang menerapkan model TCN-LSTM pada jaringan. Dengan ranah yang sama yaitu jaringan, dilakukan penelitian dengan membalikkan urutan model yang digunakan sehingga menerapkan LSTM-TCN. Ada pula penelitian yang menggunakan LSTM-TCN guna untuk memprediksi kadar oksigen. Dilanjutkan dengan adanya penelitian terkait penerapan 4 jenis model TCN yaitu TCAN, TCN, HATCN, dan ATCN. Urutan yang berbeda pada model yang

digunakan ternyata berpengaruh pada tingkat keakuratan yang dihasilkan. Hal ini terbukti dari penelitian terdahulu yang sudah dijabarkan sebelumnya [13] [14].

Berdasarkan kondisi awal tersebut, maka penelitian yang akan dilakukan adalah menggunakan model TCN-LSTM, LSTM-TCN, TCAN-LSTM, dan LSTM-TCAN. Data yang didapatkan akan melewati proses normalisasi kemudian hasil dari normalisasi akan digunakan sebagai *input* model pertama gabungan. *Output* dari model pertama tersebut akan menjadi *input* model kedua gabungan. Lalu *output* dari model kedua gabungan akan menjadi nilai hasil prediksi harga saham. Keempat model tersebut masih perlu diuji dalam prediksi harga saham dan dibandingkan untuk mendapatkan tingkat akurasi yang terbaik melalui perhitungan RMSE.

