

BAB 2

TINJAUAN PUSTAKA

2.1 Peramalan atau Forecasting

Peramalan adalah kegiatan yang umum dilakukan seseorang untuk memprediksi sesuatu hal yang akan terjadi di masa mendatang, guna untuk mendapatkan keuntungan dari tindakan yang bisa dilakukan pada saat ini. Peramalan diperlukan dalam banyak sekali bidang, mulai dari perencanaan pembangunan kota dalam beberapa tahun kedepan harus melihat perkembangan penduduk kota tersebut, pengaturan stok penjualan dari suatu toko bisa melihat hari apa dia berjualan, hingga ketika kita akan melakukan kegiatan untuk hari esok yang mungkin harus memperhatikan apa kemungkinan cuaca untuk besok [1].

Suatu hal dapat di prediksi dengan lebih akurat dan mudah dibanding hal lain contohnya, waktu terbit matahari untuk besok dapat di prediksi dengan lebih mudah dibandingkan nomor undian yang akan keluar untuk suatu acara tertentu [1]. Tingkat keberhasilan suatu prediksi dipengaruhi oleh beberapa hal yaitu :

1. Seberapa mengerti kita tentang faktor yang mempengaruhinya
2. Berapa banyak data yang tersedia
3. Apakah cara yang kita lakukan dapat mempengaruhi hal yang ingin kita prediksi

Peramalan dapat sangat bergantung dengan banyak hal mulai dari rentan waktu, keterkaitan antara faktor dan objek prediksi, pola dari data, dan masih banyak lagi. Peramalan bisa sangat mudah dilakukan seperti mengambil data paling baru untuk dijadikan hasil prediksi yang biasa disebut *naïve method* hingga penggunaan rumus statistika dan komputasi dengan *machine learning*. Bahkan bisa jadi objek yang akan kita prediksi tidak memiliki data sama sekali, seperti penjualan bulan pertama pada produk yang baru rilis, yang menjadikan peramalan bisa sangat mudah dilakukan ataupun sangat sulit dilakukan [1].

2.1.1 Time Series Forecasting

Time series adalah sekumpulan data dalam suatu waktu, tersusun dan diambil secara teratur [13]. Setiap data yang ada pada suatu time series haruslah tercatat

secara tersusun, dan secara kontinu. Hal ini bisa sangat sulit karena pada kenyataannya ada beberapa hal yang menjadikan suatu *time series* tidak sempurna yaitu [13] :

1. Hilangnya data pada salah satu siklus pada *time series* menjadi permasalahan yang paling umum terjadi, hal ini dapat diatasi dengan melihat apakah data tersebut memang diperlukan, dan jika memang diperlukan dapat melakukan manipulasi pengisian seperti pengambilan pada data sebelumnya, pengisian nilai secara manual, dan masih banyak lagi.
2. Data yang terpencil, maksudnya data yang nilainya sangat berbeda antara data pada umumnya, bisa lebih besar atau lebih kecil dari seharusnya dengan jarak nilai data yang cukup jauh. Hal ini memerlukan suatu metode yang dapat menggunakan data - data terpencil ini atau kita bisa saja mengabaikan data yang seperti ini.
3. Pengambilan data yang dilakukan secara tidak berkala, menjadikan susunan pada data tidak sesuai seperti yang diharapkan, kita dapat mengatasi hal ini dengan membuat asumsi bagaimana kita menggunakan data *time series* tersebut.

Data yang bersifat *time series* dapat dipecah dan digunakan untuk mengenal pola sesuai dengan kebutuhan yang kita inginkan [1]. Ada beberapa hal yang merupakan komponen pada suatu *time series* yaitu [1], [13] :

1. Tren, atau *Trend* adalah perubahan data *time series* secara umum dalam periode observasi yang dilakukan, tanpa melihat *seasonality* dan *irreguralities*. Tren biasanya terlihat dalam bentuk naik atau turunnya suatu nilai pada grafik data dalam periode yang cukup lama.
2. Musiman atau *Seasonal* adalah pola spesifik yang ada pada *time series* yang berhubungan dengan faktor yang bersifat waktu seperti cuaca, tanggal pada suatu kalender, atau acara perayaan.
3. Siklus atau *Cycle* adalah pola yang terjadi pada suatu *time series* tapi tidak berhubungan dengan faktor waktu seperti musiman. Umumnya siklus

adalah pola yang lebih panjang dari pada musiman, serta memerlukan observasi dari data yang cukup banyak.

4. Sisa atau *Residual* biasa dikenal dengan *error* merupakan nilai yang kita dapatkan dari perbedaan antara nilai yang diprediksi y dengan nilai aktual x .

Time series forecasting sendiri merupakan proses untuk memprediksi nilai yang akan muncul sesuai siklus *time series* dengan menggunakan data sebelumnya sebagai acuan, atau data lain yang mungkin bisa mempengaruhi nilai prediksi.

2.1.1.1 Pengisian Data Kosong pada Dataset

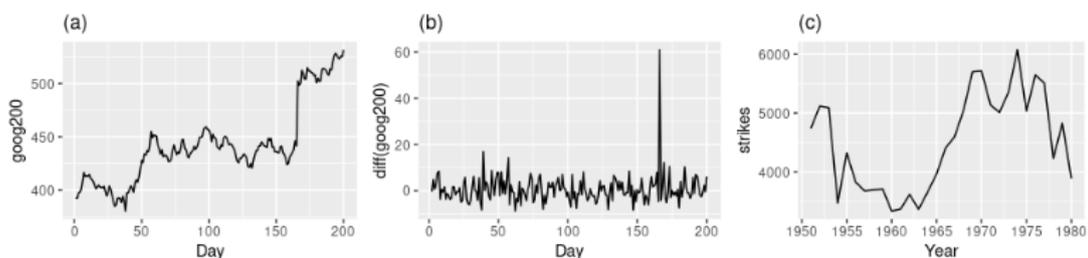
Seperti yang sudah dijelaskan sebelumnya, terdapat beberapa kekosongan yang mungkin terjadi pada data yang akan digunakan untuk melakukan prediksi, sehingga diperlukan tahapan tambahan untuk dapat mengurangi *error* yang dihasilkan dari kekosongan data tersebut, kurang lebih seperti berikut [14] :

1. Penghapusan / *Deletion*, menghapus data yang memang bernilai kosong, sehingga data tidak digunakan untuk pelatihan ataupun pengujian model, hal ini dapat dilakukan jika data yang akan dihapus memang tidak diperlukan, namun perlu diperhatikan bahwa data yang dihapus jangan lebih dari 10 persen keseluruhan dataset karena akan menghasilkan ketidakakuratan model prediksi.
2. Berdasarkan data terdekat / *Neighbor based*, yaitu menggunakan data tetangga yang memiliki nilai, dapat digunakan data setelah atau sebelumnya, metode ini dapat digunakan jika data yang diisi tidak memiliki pola tertentu, atau bersifat acak, serta duplikasi pada data tidak menjadi masalah yang terlalu penting.
3. Pengisian Statistika / *Statistical based*, pengisian data kosong dengan pendekatan statistik seperti *mean*, *median*, atau probabilitas lain dari keseluruhan dataset. Metode ini dapat digunakan jika dataset tidak memiliki banyak variasi pada data, sehingga data yang di isi tidak timpang atau berbeda jauh dengan data sekitarnya.

2.1.2 Stasioneritas

Stasioneritas pada *Forecasting* biasanya mengacu kepada data yang digunakan. Suatu *time series* bisa dikatakan stasioner jika dari data tersebut tidak bergantung pada faktor waktu, maupun nilai sebelumnya [15]. Bisa dikatakan *time series* yang memiliki tren dan musiman tidaklah stasioner. Untuk mencapai tingkat ini tentu sangat sulit, sehingga stasioneritas bisa didapatkan jika data *time series* memiliki rata - rata dan standar deviasi yang mendekati sama dalam setiap waktu [16].

Untuk menguji stasioneritas pada suatu dataset biasanya dapat dilihat dari plot grafik pada dataset yang telah disajikan. Jika grafik memiliki trend, faktor musiman, atau tidak memiliki kesamaan pada nilai statistiknya pada setiap jumlah data yang digunakan seperti rerata dan standar deviasi yang berubah maka dataset tersebut tidak stasioner, dan dapat dilihat dari Gambar 2.1 dimana dataset yang stasioner umumnya terdistribusi secara teratur, dan tidak memiliki trend, seperti pada contoh b dalam Gambar 2.1 dibawah.



Gambar 2.1 Contoh Dataset Stasioner dan Tidak Stasioner

Sedangkan untuk melakukan uji stasioneritas secara statistik, biasanya dilakukan pengujian uji akar unit yang salah satunya bernama *Augmented Dickey Fuller* (ADF) [17]. Nilai ADF yang baik adalah nilai yang kecil, lebih kecil daripada *critical value* yang dimiliki data. Untuk menguji ADF dilakukan perhitungan yang cukup sulit secara manual, oleh karena itu perlu memahami uji akar unit yang lain yaitu pengujian *Dickey Fuller* (DF) yang biasana digambarkan sebagai pengujian model *Autoregression (1)* atau model AR(1) yang dinyatakan sebagai berikut :

$$y_t = \rho y_{t-1} + \varepsilon_t \quad 2.1$$

Dimana y_t adalah suatu nilai pada t waktu, ρ adalah koefisien yang menentukan akar unit, y_{t-1} adalah suatu nilai pada waktu $t-1$, dan ε_t adalah error pada perhitungan ke t yang biasanya diasumsikan bernilai sama dengan nol (0). Pengujian *Dickey Fuller* memiliki hipotesis jika nilai absolut dari konstanta akar unit $|\rho|$ lebih besar atau sama dengan 1, maka serangkaian data tersebut memiliki akar unit, yang menjadikannya tidak stasioner. Untuk menguji konstanta akar unit, dilakukan pengurangan y_{t-1} pada setiap sisi, maka akan membentuk persamaan seperti dibawah

$$y_t - y_{t-1} = \rho y_{t-1} + \varepsilon_t - y_{t-1} \quad 2.2$$

$$\Delta y_t = (\rho - 1)y_{t-1} + \varepsilon_t \quad 2.3$$

$$\Delta y_t = \delta y_{t-1} + \varepsilon_t \quad 2.4$$

Dari persamaan diatas, jika ρ sama dengan 1, dimana menentukan bahwa data memiliki akar unit, maka nilai δ akan menjadi 0, dan menjadikan prediksi nilai hanya ditentukan oleh error sehingga data tidak bisa diprediksi atau biasa disebut *random walk*. Pengujian *Dickey Fuller* ini adalah dasar dari pengujian *Augmented Dickey Fuller* yang menggunakan lebih banyak parameter sehingga sulit untuk dihitung dengan perhitungan manual sebagai berikut :

$$\Delta y_t = c + \beta_t + \rho y_{t-1} + \delta_1 \Delta y_{t-1} + \dots + \delta_{p-1} \Delta y_{t-p+1} + \varepsilon_t \quad 2.5$$

c disini adalah konstanta pada dataset, dan β_t adalah konstanta trend yang sedang berlangsung pada waktu ke t .

2.1.3 ARIMA dan ARIMAX

ARIMA merupakan singkatan dari Autoregressive Integrated Moving Average, yang merupakan salah satu model statistika yang paling sering digunakan untuk memprediksi data time series [18]. ARIMA sendiri pertama kali diperkenalkan oleh Box dan Jenkins pada tahun 1976, dan biasa disebut Box-Jenkins model [18]. ARIMA biasanya digunakan pada data yang bersifat linear atau data yang memiliki suatu pola tertentu, agar data dapat diprediksi dengan baik.

Menurut Frank (2003) ARIMA dapat dibagi menjadi 4 bagian yaitu *Autoregressive (AR)*, *Moving Average (MA)*, *Autoregressive Moving Average*

(ARMA), dan *Autoregressive Integrated Moving Average* (ARIMA) sebagai berikut [19] :

1. *Autoregressive* (AR)

Model AR adalah model prediksi yang melihat linearitas atau kombinasi linear dari nilai nilai sebelumnya. AR biasa dinotasikan dengan (p) biasa dihitung dengan cara berikut :

$$\hat{y}_t = c + \theta_1 y_{t-1} + \theta_2 y_{t-2} + \dots + \theta_p y_{t-p} + \varepsilon_t \quad 2.6$$

\hat{y}_t adalah nilai yang diprediksi pada t waktu, c merupakan konstanta model, y_{t-p} adalah nilai y_t dari waktu sebelumnya ($t-p$) sedangkan θ adalah parameter estimasi AR, dan ε_t adalah *error* pada prediksi ke t .

2. *Moving Average* (MA)

Model MA adalah model prediksi yang melihat error pada masa lampau sebagai acuan prediksinya. Model MA biasa dinotasikan dengan (q) dan biasa dihitung dengan cara berikut :

$$\hat{y}_t = c + \beta_1 \varepsilon_{t-1} + \beta_2 \varepsilon_{t-2} + \dots + \beta_q \varepsilon_{t-q} + \varepsilon_t \quad 2.7$$

β disini merupakan parameter estimasi dari model MA, dan ε_{t-q} adalah error dari nilai sebelumnya ($t-q$)

3. *Autoregressive Moving Average* (ARMA)

Model ARMA merupakan gabungan dari model AR dan model MA, ARMA model haruslah memiliki stasioneritas pada data. Model ARMA(1,1) atau gabungan antara AR(1) dan MA(1) dapat dituliskan sebagai berikut :

$$\hat{y}_t = c + \theta_1 y_{t-1} + \beta_1 \varepsilon_{t-1} + \varepsilon_t \quad 2.8$$

4. *Autoregressive Integrated Moving Average* (ARIMA)

Sedangkan ARIMA memiliki model yang sama dengan ARMA, namun ARIMA melakukan diferensiasi terlebih dahulu sebelum memasukan ke perhitungan ARMA (3) karena data yang digunakan belum stasioner. proses diferensiasi ini dapat dilihat pada perhitungan berikut :

$$\Delta y_t = y_t - y_{t-1} \quad 2.9$$

Δy_t adalah nilai perbedaan antara data satu dengan data yang lain, y_t adalah nilai data yang akan di deferensiasi, dan y_{t-1} adalah nilai data sebelumnya..

ARIMAX sendiri adalah perluasan dari model ARIMA dengan menggunakan faktor eksogen independent yang dianggap mempengaruhi peramalan [17]. Faktor eksogen yang digunakan haruslah bersifat stasioner, sehingga perlu dilakukan differensiasi jika data yang digunakan belum stasioner. Untuk perhitungan ARIMAX(1,1,1) dengan satu parameter eksogen dapat digambarkan dengan perhitungan berikut [20]:

$$\hat{y}_t = c + \gamma x_t + \theta_1 y_{t-1} + \beta_1 \varepsilon_{t-1} + \varepsilon_t \quad 2.10$$

dimana γ konstanta variable eksogen, dan x_t adalah nilai dari parameter eksogen pada waktu ke t.

2.1.4 Evaluasi Model

Untuk melakukan evaluasi terhadap model prediksi yang telah dibuat, dapat menggunakan *Akaike Information Criterion* (AIC) dengan rumus sebagai berikut [21]:

$$AIC = N \log \left(\frac{\sum_{i=1}^N \varepsilon^2}{N} \right) + 2K \quad 2.11$$

K disini adalah parameter yang digunakan oleh model ditambah 1, contoh ARMA(1,1) akan bernilai $k = 3$ karena menggunakan 2 parameter masing - masing satu AR dan satu MA, ditambah dengan 1, sedangkan N disini adalah nilai yang diobservasi, sedangkan ε disini adalah *error* yang didapat dari hasil prediksi menggunakan model, pada observasi ke i.

Sama seperti ADF, semakin kecil nilai dari AIC maka akan semakin baik, yang menandakan bahwa parameter yang digunakan untuk prediksi sedikit, serta memiliki nilai *error* yang cukup kecil.

2.1.5 Evaluasi Peramalan

Model peramalan yang telah dibuat haruslah di evaluasi untuk menentukan tingkat akurasi dari model tersebut, dan untuk mengukur apakah model dapat memprediksi data secara akurat atau tidak. Ada beberapa teknik evaluasi peramalan

yang cukup banyak digunakan, namun yang paling sering digunakan adalah *Root Mean Squared Error* (RMSE) dan *Mean Error Absolute Percentage Error* (MAPE).

1. Root Mean Squared Error (RMSE)

RMSE merupakan salah satu perhitungan yang cukup sering digunakan untuk mengevaluasi peramalan dengan mengukur perbedaan nilai prediksi dengan nilai asli pada data. Semakin kecil nilai RMSE menandakan model yang digunakan untuk melakukan prediksi semakin baik [22].

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y} - y)^2 \quad 2.12$$

$$RMSE = \sqrt{MSE} \quad 2.13$$

MSE = Mean Square Error
 \hat{y} = Nilai Prediksi Model
 y = Nilai Asli

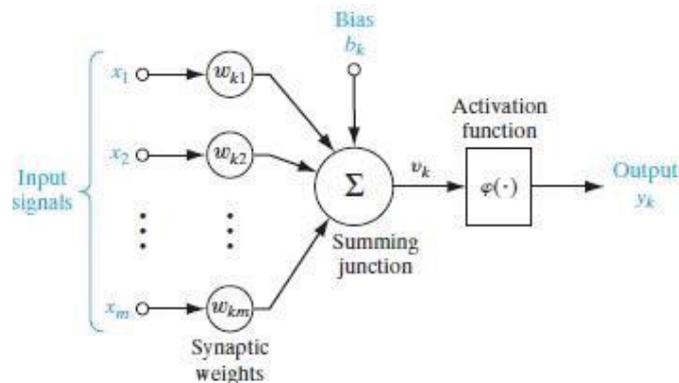
2. Mean Error Absolute Percentage Error (MAPE)

MAPE adalah perhitungan kesalahan rata - rata dari model prediksi secara mutlak, yang di tampilkan dalam bentuk persentase. MAPE pada suatu model prediksi dikatakan sangat baik jika memiliki nilai 10% dan dikatakan buruk jika memiliki nilai diatas 50% [23].

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y - \hat{y}}{y} \right| \quad 2.14$$

2.2 Artificial Neural Network (ANN)

ANN merupakan teknik komputasi yang meniru jaringan otak manusia untuk memecahkan masalah yang dihadapinya [24]. Otak manusia memiliki *neuron* dengan jumkah yang sangat banyak dan saling berhubungan, dengan fungsi untuk memproses informasi dan menentukan keluaran dari informasi yang didapatkan untuk menentukan aksi kerja pada manusia. ANN juga menggunakan *neuron* namun dengan jumlah yang lebih sedikit, namun masih tetap digunakan sebagai pemroses informasi karena memiliki prinsip kerja yang sama [24]. Setiap *neuron* pada ANN dapat diilustrasikan dengan Gambar 2.1 dibawah ini



Gambar 2.2 Diagram Neuron pada ANN

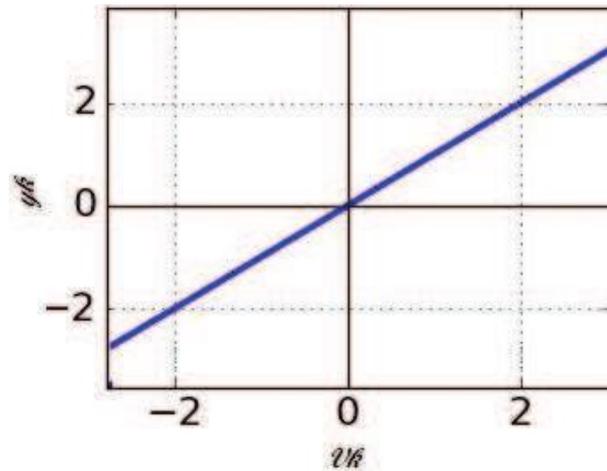
Setiap masukan dari *neuron* melalui sebuah jembatan dengan *weight* atau beban yang harus dilalui ($w_{k1} - w_{km}$), yang mengubah nilai dari input itu sesuai dengan pengetahuan dari *neuron* dengan cara mengkalikan nilai masukan dengan *weight* nya. Setiap masukan yang telah diubah berdasarkan *weight* akan dijumlahkan dengan bersamaan dengan *bias* yang juga telah ditentukan pada proses pelatihan, dan akhirnya nilai dari penjumlahan masukan bersama *bias* akan melewati sebuah *activation function* untuk menghasilkan keluaran dari *neuron* yang biasanya merupakan nilai yang jaraknya telah ditentukan sebelumnya. *bias* ini bisa meningkatkan atau menurunkan jumlah dari masukan yang akan untuk dapat mengaktifkan *activation function* [24].

2.2.1 Activation Function

Activation Function adalah penentu apakah keluaran dari suatu *neuron* akan diteruskan ke *neuron* yang lainnya [24]. Sederhananya, *Activation Function* adalah penentu seperti *yes or no* sebuah masukan akan diteruskan atau tidak, dengan ketentuan jarak nilai keluaran tergantung dari fungsi yang digunakan bisa berupa jarak antara $0 \sim 1$, atau $-1 \sim 1$ dan sebagainya. Secara garis besar, *Activation Function* dapat dibagi menjadi *Linear* dan *Non-Linear Activation Function* [24]

1. Linear Activation Function

Fungsi ini secara sederhana akan mengeluarkan keluaran (y_k) dengan sama nilai masukan (z_k), yang dapat dilihat pada Gambar 2.2 dibawah ini :



Gambar 2.3 Linear Activation Function

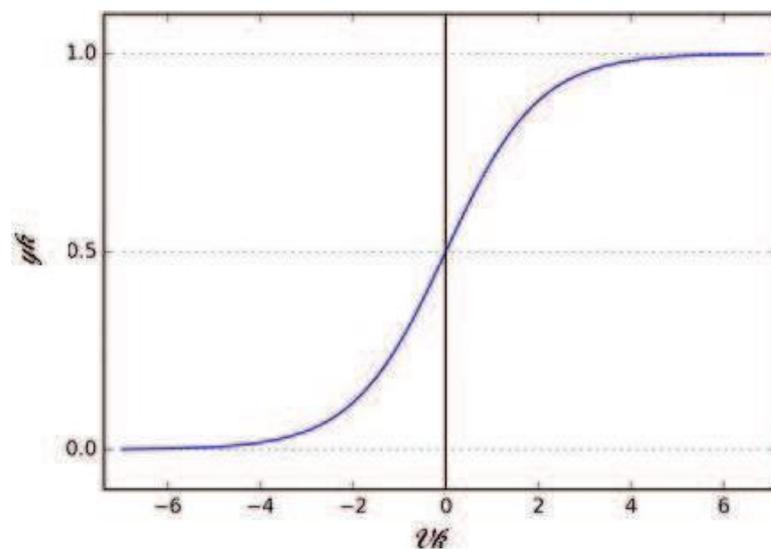
Persamaan yang mendeskripsikan *Linear Activation Function* adalah sebagai berikut:

$$y_k = z_k \quad 2.15$$

2. *Non-Linear Activation Function*

Fungsi ini merupakan fungsi yang paling banyak digunakan untuk model ANN karena lebih merepresentasikan kepada keadaan yang sebenarnya. Salah satu *Non-Linear Activation Function* yang paling sering digunakan adalah *Sigmoid Activation Function* yang dapat digambarkan pada Gambar 2.3 sebagai berikut [24]

:



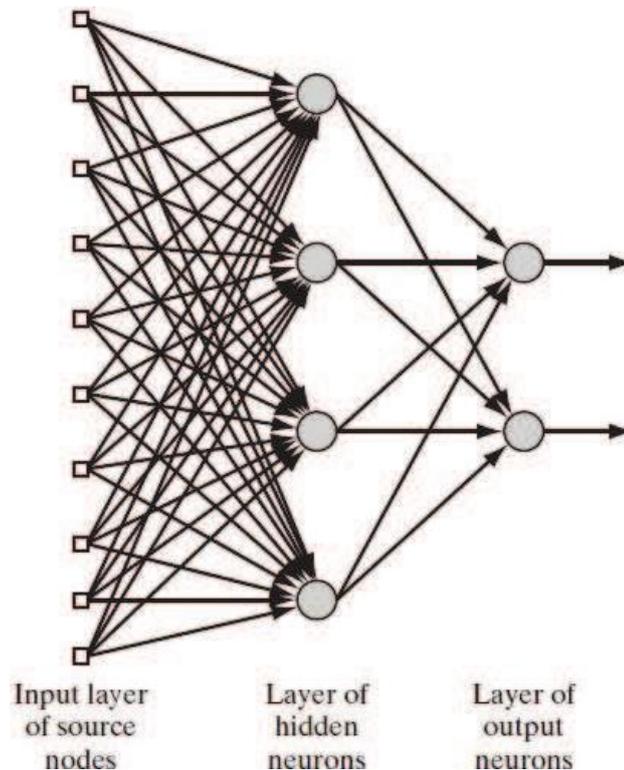
Gambar 2.4 Sigmoid Activation Function

Fungsi ini menentukan keluaran dari jarak 0 - 1 dengan ketentuan bahwa nilai keluaran akan melalui sebuah persamaan berikut untuk menghasilkan keluarannya :

$$y_k = \frac{1}{1 + e^{zk}} \quad 2.16$$

2.2.2 Model Neural Network

Model *Neural Network* yang umum diketahui dan digunakan adalah *Multilayer Perception* (MLP) yang merupakan gabungan dari beberapa *layer neuron*. Biasanya MLP dibentuk dalam bentuk minimal 3 *layer* yaitu *input layer*, *hidden layer*, dan *output layer* [24]. Data mengalir dari *input layer* ke *output layer* secara maju, biasa juga disebut *feed forward neural network*. Untuk jumlah yang ada di setiap *layer* tidaklah ditentukan, dan dapat dilihat contoh model MLP pada Gambar 2.4 dibawah :



Gambar 2.5 Multilayer Perception (MLP)

2.2.3 Pelatihan Neural Network

Sebelum *neural network* dapat digunakan, kita harus melakukan proses pelatihan terlebih dahulu, atau biasa disebut *training*. Pada tahapan ini kita menginisiasi terlebih dahulu setiap *neuron* dan jumlah *layer* yang akan kita gunakan, lalu menghubungkannya masing masing dengan nilai *weight* serta *bias* yang kita tentukan. Tujuan dari pelatihan ini adalah menyesuaikan *weight* dan *bias* yang ada pada model ANN yang kita buat [24].

2.2.3.1 Jenis Pelatihan Neural Network

Untuk jenis pelatihan *neural network*, dapat dibagi menjadi dua yaitu:

1. *Supervised Learning*

Pelatihan ini dilakukan dengan membandingkan nilai keluaran model dengan nilai keluaran sebelumnya, dan melakukan pengaturan yang *weight* dan *bias* sesuai pada model untuk dapat memprediksi keluaran dengan baik. Metode pelatihan ini yang paling sering digunakan untuk proses pelatihan *Neural Network*. Jenis ini adalah yang akan digunakan untuk penelitian ini.

2. *Unsupervised Learning*

Tipe pelatihan ini dilakukan jika kita melatih sebuah *network* tetapi tidak menetapkan kebutuhan atau target yang ditetapkan, melainkan lebih antara kompetisi antara *neuron* yang ada untuk menghasilkan hasil yang terbaik. Model pelatihan ini tidak terlalu sering dilakukan karena bersifat sensitif terhadap kasus yang dimiliki.

Peneliti akan menggunakan jenis pelatihan *Supervised Learning*, dengan menggunakan dataset yang telah di beri hasil output sehingga pelatihan dapat dilakukan dengan cara *Backpropagation* dengan memperbaiki nilai *weight* dari setiap koneksi *neuron* untuk mendapatkan model terbaik.

2.2.3.2 Data Normalization

Data *Normalization* atau normalisasi pada data adalah langkah yang digunakan untuk membagi data menjadi interval tertentu sehingga setiap data memiliki kontribusi yang setimpal pada model yang sedang dibangun, dan dapat

meningkatkan proses pelatihan serta akurasi dari hasil pelatihan. Biasanya normalisasi dilakukan untuk mendapatkan data dari nilai 0 ke 1 dengan metode *min-max* sebagai berikut [25] :

$$\hat{x} = \frac{x - \min}{\max - \min} \quad 2.17$$

\hat{x} = Nilai normalisasi

x = Nilai asli data

\min = nilai minimum dari seluruh dataset

\max = nilai maksimum dari seluruh dataset

2.2.3.3 Forward Phase

Forward phase atau fase maju merupakan alur data dimasukan ke sebuah arsitektur *neural network* secara maju, atau dari *neuron input* hingga *output*, untuk menghasilkan prediksi yang diinginkan. Fase ini merupakan fase umum, dan telah dijelaskan sebelumnya pada sub bab 2.2, namun secara matematis rumus yang digunakan pada fase ini adalah sebagai berikut [26] :

$$\hat{Z}_j = \beta_j + \sum_{i=1}^n x_i \cdot w_i \quad 2.18$$

\hat{Z}_j = Keluaran semu pada *neuron* ke j

β_j = Nilai *bias* pada *neuron* ke j

n = Jumlah masukan ke *neuron* j

x_i = Nilai *input* dari *neuron* i

w_i = nilai bobot dari *input neuron* i

Keluaran semu pada *neuron* kemudian dikalikan dengan *activation function* yang digunakan oleh *neuron* tersebut dengan perhitungan sebagai berikut :

$$Z_j = \varphi_j(\hat{Z}_j) \quad 2.19$$

φ_j = Fungsi aktivasi yang digunakan pada *neuron* j

2.2.3.4 Backpropagation

Backpropagation adalah tahapan umum yang akan dialami sebuah *neural network* pada masa pelatihan untuk dapat memperbaiki bobot dan *bias* yang dimiliki oleh *neuron* didalamnya. Pelatihan biasanya dilakukan secara berulang hingga nilai *error* mengecil, atau hingga iterasi yang dilakukan telah tercapai [27]. Pada tahapan ini ada yang dikenal dengan *learning rate* atau tingkat pembelajaran sebuah *neural network*, biasanya merupakan nilai dibawah 1, dan semakin besar nilai *learning rate*, maka akan semakin cepat proses *backpropagation* namun akan

beresiko menghasilkan model yang tidak akurat, berbeda jika *learning rate* semakin kecil, maka nilai akurasi model bisa semakin tinggi namun perlu banyak iterasi untuk mencapai hasil terbaik. Untuk memperbaiki bobot pada tahapan *backpropagation* ini biasanya menggunakan sebuah optimasi, yang pada penelitian ini menggunakan optimasi *gradient descent*.

Untuk melakukan *backpropagation* pertama harus mencari informasi *error* yang dihasilkan dari data yang telah kita lakukan *forward phase* dengan perhitungan sebagai berikut:

$$\hat{\varepsilon} = y - \hat{y} \quad 2.20$$

$\hat{\varepsilon}$ = Error semu yang dihasilkan keluaran
 y = Nilai asli keluaran yang diharapkan
 \hat{y} = Nilai prediksi model

Selanjutnya cari nilai *error* dari setiap masukan data dengan perhitungan dibawah ini [26] :

$$\delta_{in_j} = \sum_{k=1}^n \delta_k w_{jk} \quad 2.21$$

δ_{in_j} = Jumlah *error hidden layer*
 w_{jk} = Nilai bobot dari j ke k

Selanjutnya adalah mencari nilai *error* dengan cara membalikan nilai keluaran dengan turunan *activation function* yang digunakan dalam hal ini adalah *sigmoid*, dengan perhitungan sebagai berikut [26]:

$$\delta_j = \delta_{in_j} \cdot f'(\hat{Z}_j) = \delta_{in_j} \cdot Z_j(1 - Z_j) \quad 2.22$$

δ_j = Nilai error dari keluaran *neuron j*

Selanjutnya adalah mencari koreksi bobot yang seharusnya diterapkan pada setiap sambungan *neuron* dengan perhitungan berikut [26]:

$$\Delta w_{kj} = \alpha \cdot \delta_k \cdot Z_j \quad 2.23$$

Δw_{kj} = Selisih bobot antara *neuron k* dan *neuron j*
 α = Konstanta pelatihan atau *learning rate*

Sedangkan untuk mencari koreksi bobot bias digunakan perhitungan sebagai berikut :

$$\Delta w_{bj} = \alpha \cdot \delta_k \quad 2.24$$

Terakhir, koreksi bobot yang bersangkutan dengan perhitungan :

$$w_{kj} = \tilde{w}_{kj} + \Delta w_{kj} \quad 2.25$$

\tilde{w}_{kj} = Bobot lama antara *neuron k* dan *neuron j*

2.2.3.3 Data Denormalization

Data Denormalization atau denormalisasi pada data adalah kebalikan dari langkah normalisasi, yaitu membalikan data kepada dari nilai 0 ke 1 menjadi nilai aslinya. Proses ini hanya dilakukan jika data memang di normalisasi sebelumnya, dan dapat dihitung dengan cara dibawah [25] :

$$x = \hat{x}(\max - \min) + \min \quad 2.26$$

2.3 ARIMAX ANN

Zhang (2003) menyadari kekurangan dari suatu model prediksi hanya dapat memprediksi suatu jenis dataset tertentu yaitu data *linear* atau data yang memiliki pola, dan data *non-linear* atau data yang tidak memiliki pola. Maka Zhang mulai mengembangkan metode prediksi baru untuk mengatasi kedua jenis data tadi agar dapat di modelkan dengan lebih sempurna [28]. Model yang dikembangkan oleh Zhang adalah kombinasi antara metode yang dapat memprediksi sebuah model *linear* seperti ARIMAX dan sebuah model *non-linear* seperti ANN. Perhitungan model yang dibuat oleh Zhang dapat dilihat sebagai berikut [28] :

$$\hat{y}_t = L_t + N_t \quad 2.26$$

Konsep sederhananya adalah dengan menambahkan hasil perhitungan dari suatu model *linear* dengan model *non-linear*. Pada rumus diatas, L_t adalah hasil prediksi dari model *linear*, sedangkan N_t adalah hasil prediksi dari model *non-linear*. Model ini menggunakan sebuah model *linear* untuk memprediksi nilai yang akan datang, lalu menguranginya dengan nilai asli, yang dapat di notasikan sebagai berikut :

$$e_t = y_t - \hat{L}_t \quad 2.27$$

Dimana e_t merupakan *error* atau residu dari nilai asli (y_t) dikurangi nilai prediksi model *linear* (\hat{L}_t) yang nantinya akan dimasukkan ke model *non-linear* untuk menghasilkan prediksinya (N_t)

2.4 Harga Saham

Saham adalah surat berharga yang diterbitkan dari suatu perusahaan sebagai bukti bahwa seseorang atau sebuah organisasi berkepemilikan di perusahaan tersebut. Harga saham adalah harga yang ditetapkan oleh perusahaan dan investor untuk memperdagangkan kepemilikan suatu perusahaan [5]. Biasanya harga saham ini dapat dilihat pada pasar bursa yang memperjual belikan saham sekumpulan perusahaan.

2.4.1 Indeks Saham

Index saham merupakan gabungan dari berbagai harga saham yang menunjukkan indikator naik atau turunnya harga pada saham yang berada dalam kumpulan saham tersebut [29]. Index saham dapat digunakan sebagai indikator ketika seorang investor akan melakukan transaksi saham, namun Index ini bersifat terlalu umum dan memiliki banyak faktor eksternal maupun faktor internal yang mempengaruhinya [29]. Beberapa kondisi eksternal seperti perubahan nilai kurs dapat berpengaruh kepada suatu index saham, namun ada beberapa kondisi eksternal yang tidak berdampak pada seluruh sektor [29]. Index saham juga bukanlah sesuatu yang dapat digunakan untuk transaksi sehingga nilai dari index saham terkadang tidak sesuai dengan nilai saham dari suatu perusahaan yang dipegang.

2.4.2 Faktor yang Mempengaruhi Harga Saham

Secara umum harga saham dipengaruhi dengan cukup banyak faktor seperti faktor internal perusahaan seperti pergantian kepemimpinan, produk baru yang dikeluarkan, dan kebijakan dari perusahaan itu, faktor eksternal seperti perubahan kebijakan pemerintah, nilai tukar kurs, dan juga faktor opini publik terhadap prospek perusahaan itu sendiri [5].

Harga saham yang dipilih peneliti adalah saham PT Astra Indonesia karena merupakan salah satu saham yang cukup besar dampaknya ke Indeks Harga Saham Gabungan (IHSG) karena merupakan saham terbesar di bidang otomotif. Saham ini dipilih juga karena memiliki keselarasan faktor eksternal yang cukup tinggi dengan Dow Jones Industrial Average (DJIA) yang merupakan indeks internasional yang menjadi acuan dari berbagai saham yang bergerak dibidang ekspor dan impor [5].

Alasan lain memilih saham PT Astra dibanding memilih IHSG sebagai objek prediksi karena setelah melihat beberapa literatur, dampak yang diberikan dari suatu parameter eksternal tidak sama terhadap semua sektor [5], dan prediksi saham lebih masuk akal dan lebih memiliki nilai bisnis jika dibandingkan prediksi umum seperti memprediksi indeks IHSG.

2.5 Python Programming

Python adalah bahasa pemrograman tingkat tinggi yang mudah diimplementasikan dan digunakan untuk membangun sebuah program dalam berbagai sektor teknologi. Diciptakan oleh Guido van Rossum dan pengembangan dari bahasa pemrograman ABC pada 1989, bahasa pemrograman python berkembang sangat pesat dan karena komunitas serta antusias pengembangan yang cukup tinggi, Python 3 hadir pada tahun 2008 yang merupakan versi terbaru dan masih digunakan hingga saat ini [30]. Perkembangan bahasa pemrograman ini dikarenakan oleh mudahnya setiap orang yang ingin menggunakannya, karena menggunakan bahasa inggris yang mudah dimengerti dan diinterpretasikan kedalam bahasa sehari - hari, jumlah *library* yang sangat banyak karena pengembangannya dilakukan secara *open source*, serta kecintaan komunitas yang sangat besar karena telah menghasilkan banyak sekali produk yang berguna baik secara *online* maupun *offline* [30]

2.5.1 Kelebihan Bahasa Pemrograman Python

Ada banyak kelebihan yang ada pada bahasa pemrograman Python, mulai dari orang awam yang tidak familiar dengan pemrograman hingga para ahli bisa mendapatkan hal untuk di eksplor dan dikembangkan. Kelebihan dari bahasa pemrograman ini adalah [30] :

1. Mudah dibaca dan digunakan

Bahasa pemrograman Python mudah sekali dibaca karena sedikit sekali menggunakan simbol - simbol seperti *bracket* ({) , titik koma (;) ataupun simbol lain yang ada pada bahasa pemrograman lain. Python menggunakan indentasi yang lebih manusiawi sehingga dapat dibaca dan digunakan dengan lebih mudah daripada bahasa pemrograman lain.

2. Menggunakan bahasa Inggris

Definisi berbagai macam fitur yang ada pada python juga menggunakan bahasa inggris yang merupakan bahasa internasional, sehingga dapat dengan mudah di aplikasikan dan disampaikan oleh programmer kepada komputer.

3. Sudah ada di beberapa sistem

Python terkadang sudah ada di sistem yang akan kita gunakan, contohnya seperti Mac OS atau Ubuntu sudah memiliki python didalamnya, dan kita hanya perlu menginstall *interpreter* atau penerjemah dari bahasa ini ke bahasa mesin pada sistem kita.

4. Dapat bersinergi dengan bahasa pemrograman lain

Python dapat dengan mudah dikombinasikan dengan bahasa pemrograman lain untuk menyelesaikan permasalahan yang lebih kompleks, seperti bahasa pemrograman JavaScript untuk permasalahan pengembangan website, dan C++ untuk permasalahan pengembangan game

5. Kaya akan *Library*

Library merupakan program yang ditulis oleh orang lain dengan bahasa pemrograman yang sama, sehingga kita bisa menggunakan program tersebut didalam sistem yang kita bangun. Python memiliki banyak sekali *library* yang sangat membantu membangun berbagai macam sistem terutama di sektor *machine learning*.

2.5.2 Kekurangan Bahasa Pemrograman Python

Ada beberapa hal yang harus diperhatikan ketika menggunakan bahasa pemrograman python yaitu [30] :

1. Tidak terlalu cepat

Kecepatan bahasa pemrograman python tidaklah secepat bahasa pemrograman lain yang memiliki konsep *compiler*, karena perlu diterjemahkan terlebih dahulu dengan menggunakan *interpreter* sebelum digunakan. Pengembang dan juga pengguna bahasa pemrograman python sadar akan hal ini dan terus mengembangkan kualitas *interpreter* agar dapat mempercepat prosesnya, yang diharapkan dapat menyaingi bahasa *compiler* seperti C, C++ , ataupun bahasa pemrograman lain yang akan muncul di masa mendatang

2. Keterbatasan dengan penggunaan *mobile*

Zaman yang berkembang, dan banyaknya orang yang beralih ke pengembangan secara *mobile* adalah tantangan lain dari bahasa pemrograman python. Python di masa yang akan datang mungkin bisa berkembang dan bisa digunakan untuk pengembangan aplikasi *mobile*, tetapi untuk saat ini pengguna bahasa pemrograman python harus bisa puas dengan penggunaannya dalam desktop ataupun laptop.

3. Keterbatasan desain

Bahasa pemrograman python memiliki keterbatasan desain untuk keindahan pada kodenya. Contohnya adalah penggunaan indentasi yang bisa dibilang sangat penting untuk bahasa pemrograman python, tapi tidak penting untuk bahasa pemrograman lain. Contoh lainnya adalah penggunaan suatu konstanta untuk menginisiasi variable seperti *dollar sign* (\$) pada PHP, dan *var* pada JavaScript

2.5.3 Anaconda

Anaconda adalah *distribution* atau sebuah *software* yang berisi kumpulan *library* yang sudah tersusun bersamaan dengan *tools* lainnya, guna untuk memudahkan pengembangan program dengan Python. Dengan menginstall Anaconda, kita tidak perlu lagi menginstall Python dalam sistem kita karena sudah terintegrasi dengan Anaconda ini. Kelebihan menggunakan Anaconda yang lain adalah fitur yang menghubungkan *developer* dengan *library* yang digunakan, karena Anaconda akan melakukan pengecekan pada *environment* jika ada versi dependensi dari *library* yang tidak sesuai, sehingga program masih dapat berjalan

dengan baik karena Anaconda akan menyesuaikan setiap versi dependensi agar *library* dapat berjalan dengan baik .

2.5.4 Jupyter Notebook

Jupyter Notebook adalah alat yang bersifat *open-source, browser-based* yang berfungsi sebagai ruangan virtual untuk melakukan berbagai macam eksperimen terkait *data science*. Jupyter Notebook dapat menjalankan kode yang kita tulis, serta memberikan penampilan kode yang mudah dibaca, karena inti dari nama “notebook” disini berarti bisa mendokumentasikan kode dengan baik. Dengan menggunakan Jupyter Notebook, kita bisa membagikan hasil riset kita terkait *data science* dengan lebih mudah, dan lebih luas karena memiliki struktur dan langkah yang dapat dilacak karena kode dieksekusi berdasarkan *cell* yang aktif. Dengan ditulisnya penelitian kita pada Jupyter Notebook, akan memudahkan banyak sekali orang untuk melanjutkan penelitian yang kita lakukan, karena mereka dapat mereplikasi setiap hal yang kita lakukan mulai dari awal, hingga akhir [31].

2.5.5 Tensorflow

Tensorflow adalah *library open-source* yang dikembangkan oleh Google dan sering digunakan untuk pengembangan dan penerapan algoritma *machine learning* termasuk *neural network*. Tensorflow memudahkan pengembang untuk berfokus pada algoritma dasar terkait proses pengerjaan program yang dibuat, dan tidak perlu memikirkan seluk beluk cara algoritma itu dapat diimplementasikan, yang tentunya memudahkan proses pengembangan program. Tensorflow memungkinkan pengembang untuk memodifikasi sebuah model dengan mudah hanya dengan mengubah parameter yang digunakan untuk model tersebut, sehingga pengembang bisa menghasilkan berbagai macam hasil dari suatu model yang dibuat [32].

2.5.6 Pandas

Pandas adalah *library open-source* yang dapat digunakan dalam bahasa pemrograman Python untuk mempermudah dalam mengolah dan menganalisis data secara lebih terstruktur. Pandas adalah *library* utama yang digunakan untuk memproses data pada penelitian ini, karena data yang di ubah menjadi *Dataframe*

dan juga *Series* oleh Pandas dapat lebih mudah diproses. Pandas dapat melakukan proses pemilihan, pengelompokan, penggabungan, pengurutan, agregasi, pembersihan, pembentukan ulang pada data yang kita gunakan. Pandas juga memudahkan pengembang untuk berinteraksi dengan file eksternal berformat csv [33].

2.5.7 Numpy

Numpy adalah Python *library* yang berorientasi pada data angka dalam jumlah yang cukup banyak. Numpy bisa dibilang merupakan *library* fundamental yang digunakan oleh para *data scientist* untuk memanipulasi data mereka. Numpy juga merupakan *library* yang digunakan sebagai dasar untuk membangun library lain contohnya adalah Pandas yang dijelaskan sebelumnya. Numpy memungkinkan pengembang untuk dapat melakukan berbagai komputasi angka dengan lebih cepat karena walau ditulis dalam bahasa pemrograman Python, *method* yang memerlukan kecepatan untuk proses komputasinya biasanya menggunakan C, dan C++ [34].

2.5.8 Matplotlib

Matplotlib adalah *library* yang berguna untuk merepresentasikan sekumpulan data yang berhubungan dalam bentuk graph. Matplotlib terinspirasi dari MATLAB untuk kelebihanannya dalam merepresentasikan sebuah graph, dan dibuatlah dalam bentuk *library* Python agar dapat memudahkan Python *developer* untuk merepresentasikan hasil programnya hanya dengan beberapa baris kode, dan menyesuaikan tampilannya dengan baik berkat pengembangan API Matplotlib [35].

2.5.9 Scikit Learn

Scikit Learn adalah *library* yang sering digunakan oleh pengembang yang bergerak dibidang *machine learning* untuk membuat klasifikasi, regresi, pengelompokan (*clustering*), dll pada suatu dataset. *Library* ini dibangun diatas Numpy sehingga memiliki kecepatan yang tinggi untuk operasi array dan aljabra [36]. Pada penelitian ini Scikit Learn digunakan untuk validasi model dengan menerapkan perhitungan MAPE, dan MSE.

2.5.10 pmdarima

pmdarima adalah *library* yang dapat digunakan oleh pengembang untuk membuat model *time series forecasting* dengan menggunakan model ARIMA. pmdarima sendiri merupakan implementasi *library* dengan bahasa pemrograman Python, yang terinspirasi dari bahasa pemrograman R, yang biasanya digunakan untuk melakukan *time series forecasting*. pmdarima ini memudahkan pengembang untuk menggunakan model ARIMA, karena ada fungsi `auto_arma` yang menghasilkan model ARIMA terbaik dari suatu dataset untuk dapat langsung digunakan, serta dapat melibatkan *exogenous* data yang berguna untuk penelitian ini.

2.6 Studi Literatur

Tabel 2.1 Review Literatur Pertama

Review Literatur Pertama [8]	
Judul Jurnal (url online) - ISBN	Peramalan Indeks Harga Saham Gabungan (IHSG) Menggunakan ARIMAX dengan Variabel Eksogen Covid 19 (https://prosiding.unimus.ac.id/index.php/edusaintek/article/view/567/570) - 9786025614354
Penulis	Cita Meliana, Rochdi Wasono, M. Al Haris, Zamni Haquel Alfiyani, Eka Yuni Kartika Sari
Volume / Halaman	- / 258-267
Tahun Penerbitan	2020
Research Problem	Pandemi covid 19 menjadikan nilai IHSG tidak menentu
Research Question	Bagaimana cara menerapkan metode ARIMAX dengan variable eksogen covid 19 dalam prediksi harga IHSG?.
Research Objective	Mengetahui karakter IHSG sebelum dan selama pandemi covid 19 sebagai penentu harga penutupan saham.

Objek Penelitian	Indeks Harga Saham Gabungan (IHSG) pada tanggal 2 Januari 2019 sampai 30 Juli 2020 dengan jumlah data sebanyak 388.
Kontribusi Penulis	Menerapkan metode ARIMAX dengan variabel eksogen covid 19 untuk memprediksi harga saham, menentukan model ARIMA yang digunakan untuk data IHSG
Metodologi yang Digunakan	Penelitian kuantitatif dengan menerapkan metode ARIMAX pada IHSG untuk memprediksi nilainya dengan parameter eksternal covid 19.
Hasil dan Kesimpulan	Variabel covid 19 sebagai parameter eksternal data kurang efektif, sehingga diperlukan saham dengan variable eksternal yang lebih sensitif pengaruhnya terhadap variable eksternal dalam waktu dekat atau jauh.
Tanggapan terhadap Penelitian	Penerapan perhitungan menggunakan metode ARIMAX terbilang kurang lengkap karena variabel eksternal yang digunakan bersifat dummy atau 0 dan 1 saja, serta parameter yang digunakan bersifat telalu umum dan baru terhadap IHSG sehingga hasil prediksi terkadang tidak relevan. Pengujian akurasi hasil prediksi juga tidak dilakukan sehingga belum bisa membuktikan apakah ARIMAX dapat memprediksi IHSG dengan parameter eksternal covid 19.

Tabel 2.2 Review Literatur Kedua

Review Literatur Kedua [9]	
Judul Jurnal (url online) - ISSN	Forecasting index and stock returns by considering the effect of Indonesia pre-presidential election 2019

	using ARIMAX and VARX approaches (https://iopscience.iop.org/article/10.1088/1742-6596/1277/1/012053/meta) ,
Penulis	Ulyah S M
Volume / Halaman	1277 / -
Tahun Penerbitan	2019
Research Problem	Ingin mengetahui apakah pemilu presiden di Indonesia dapat mempengaruhi prediksi harga saham
Research Question	Apakah metode ARIMAX dengan parameter eksternal pemilihan presiden dapat memprediksi harga penutupan saham dengan lebih baik?
Research Objective	Membuktikan pemilu presiden dapat mempengaruhi harga saham, serta menemukan model yang tepat.
Objek Penelitian	Harga penutupan saham IHSG dan SRTG dari 18 desember 2015, hingga 19 desember 2018.
Kontribusi Penulis	Membandingkan metode ARIMAX dan VARX, membuktikan bahwa parameter eksternal yang digunakan harus berkaitan dengan saham yang diuji.
Metodologi yang Digunakan	Menentukan model ARIMA yang dapat digunakan dengan baik, lalu menguji data yang digunakan, setelah melalui proses ARIMA, dilakukan uji dengan variabel eksternal pemilihan presiden 2019 sehingga hasil prediksi ARIMAX didapatkan. Model VARX juga dikembangkan dengan beberapa parameter eksternal, lalu membandingkan model ARIMAX dan VARX untuk melihat mana yang lebih akurat prediksinya.
Hasil dan Kesimpulan	Menghasilkan model optimal untuk ARIMA dengan data IHSG. VARX dan ARIMAX tidak dapat memprediksi harga saham karena parameter yang

	digunakan bersifat tidak nyata atau <i>dummy</i> sehingga dapat dikembangkan dengan menerapkan variable yang lebih realistis dan berhubungan dengan saham.
Tanggapan terhadap Penelitian	Penelitian menggunakan data eksternal yang kurang relevan dan juga tidak nyata sehingga prediksi yang dihasilkan tidak dapat akurat. Pengembangan dapat dilakukan dengan menentukan variabel apa yang benar-benar berpengaruh dengan harga saham, dan menentukan harga saham apa yang sensitif terhadap variabel tersebut sebelum dilakukan pembuatan model dan prediksi harganya.

Tabel 2.3 Review Literatur Ketiga

Review Literatur Ketiga [10]	
Judul Jurnal (url online) - DOI	Performance Comparison between ARIMAX, ANN and ARIMAX-ANN Hybridization in Sales Forecasting for Furniture Industry (https://hrcak.srce.hr/214107) - doi:10.5552
Penulis	Melih Yucesan, Muhammet Gul, Erkan Celik
Volume / Halaman	69 / 357 - 370
Tahun Penerbitan	2018
Research Problem	Prediksi penjualan pada industri furnitur di turki masih belum menemukan model yang tepat.
Research Question	Apa metode peramalan yang paling tepat antara ARIMAX, ANN, dan ARIMAX-ANN untuk memprediksi penjualan pada industri furnitur
Research Objective	Menghasilkan sebuah model terbaik untuk memprediksi penjualan secara bulanan pada industri furniture di turki

Objek Penelitian	Data penjualan bulanan pada salah satu pabrik furniture yang cukup besar di daerah turki.
Kontribusi Penulis	Membuktikan bahwa metode ARIMAX-ANN dapat lebih baik untuk memprediksi penjualan pada industri furnitur di turki dibanding metode ARIMAX atau ANN saja.
Metodologi yang Digunakan	Membandingkan performansi antara metode ARIMAX, ANN, dan ARIMAX-ANN dengan menggunakan <i>Consumer Confident Index</i> (CCI), dan <i>Producer Price Index</i> (PPI) sebagai parameter eksternal untuk melihat hasil dari setiap metode prediksi.
Hasil dan Kesimpulan	Metode ARIMAX dan ANN menghasilkan error yang lebih tinggi jika dibandingkan metode ARIMAX-ANN. ARIMAX dapat memprediksi data linear dengan lebih baik daripada non linear sedangkan ANN dapat memprediksi data nonlinear dengan lebih baik dari pada data linear. Metode ARIMAX-ANN dapat menghilangkan kelemahan antara kedua metode sebelumnya, dan tergantung model yang digunakan dapat menghasilkan hasil prediksi dengan error lebih rendah
Tanggapan terhadap Penelitian	Penelitian telah berhasil menentukan metode mana yang lebih baik untuk memprediksi penjualan pada industri furnitur di turki, dan membuka peluang untuk menerapkan metode ARIMAX-ANN kepada kasus yang lain.

Tabel 2.4 Review Literatur Keempat

Review Literatur Keempat [12]	
Judul Jurnal - Journal	Improved ARIMAX modal based on ANN and SVM approaches for forecasting rice yield using weather variables - Indian Journal of Agricultural Sciences 88 (12) page 101 - 106
Penulis	Wasi Alam, Mrinmoy Ray, Rajeev Ranjan Kumar, Kanchan Sinha, Santosha Rathod, K N Singh
Tahun Penerbitan	2018
Research Problem	Model prediksi apa yang dapat digunakan untuk memprediksi hasil beras tiap tahunnya dengan melibatkan parameter eksternal cuaca
Research Question	Bagaimana penerapan terbaik untuk memprediksi penghasilan beras dengan metode ARIMAX, terutama model <i>hybrid</i> ARIMAX ANN atau ARIMAX SVM
Research Objective	Mengetahui penerapan model apa yang terbaik untuk memprediksi penghasilan beras dengan parameter eksternal cuaca antara ARIMAX ANN dan ARIMAX SVM
Objek Penelitian	Data beras yang dihasilkan di Uttar Pradesh, India dari tahun 1975 hingga 2013, serta data cuaca yang berkaitan dengan rentan waktu tersebut.
Kontribusi Penulis	Menerapkan metode yang tepat terutama pada penggabungan metode ARIMAX ANN dan SVM, juga memberikan beberapa referensi parameter ANN yang dapat digunakan untuk membuat model prediksi.
Metodologi yang Digunakan	Metodologi kuantitatif dengan melakukan uji terhadap parameter yang mungkin digunakan untuk

	membangun model ARIMAX, ANN dan SVM yang bersangkutan, Serta melihat hasil evaluasi prediksinya dengan MAPE
Hasil dan Kesimpulan	Proses prediksi yang melibatkan parameter eksogen yang cukup sensitif memang sebaiknya menggunakan metode ARIMAX daripada ARIMA, lalu perlu dilakukan penggabungan dengan metode lain karena ARIMAX hanya memprediksi data bersifat linear, sehingga dibutuhkan metode lain untuk memprediksi bagian non linear dari data seperti ANN dan SVM.
Tanggapan terhadap Penelitian	Penelitian telah berhasil memberikan gambaran yang sangat baik tentang cara menerapkan metode ARIMAX bersamaan dengan metode ANN ataupun SVM, dengan menggunakan parameter yang sesuai maka hasil prediksi yang cukup akurat dapat diraih.

Tabel 2.5 Review Literatur Kelima

Review Literatur Kelima [5]	
Judul Jurnal (url online)	Pengaruh Tingkat Suku Bunga, Yield Obligasi Pemerintah Indonesia, Nilai Tukar, Indeks Bursa Efek New York, Harga Komoditas Emas, Harga Komoditas Minyak Mentah, Dan COVID-19 Terhadap Perubahan Harga Saham Di Bursa Efek Indonesia Periode 2013 Sampai Dengan 2020
Penulis	Muhammad Anis Fikri
Tahun Penerbitan	2021
Research Problem	Perubahan harga saham yang tidak menentu karena berbagai macam faktor.
Research Question	Saham apa saja yang bersifat sensitif dalam jangka pendek maupun panjang terhadap parameter eksternal yang telah dirumuskan

Research Objective	Mengetahui saham apa saja yang bersifat sensitif perubahannya terhadap parameter eksternal yang telah dirumuskan
Objek Penelitian	Saham dari PT. Pembangunan Jaya Ancol, PT. Astra Internasional Indonesia, PT. Indofood Sukses Makmur, PT. Ramayana Lestari Sentosa, PT. Kalbe Farma dengan data eksternal suku bunga, obligai, nilai tukar rupiah, index saham internasional, harga komoditas emas, harga komoditas minyak mentah, dan covid 19 dari tahun 2013 hingga 2020
Kontribusi Penulis	Membuktikan bahwa ada beberapa saham yang bersifat sensitif terhadap perubahan harga penutupannya dengan data eksternal seperti suku bunga, obligai, nilai tukar rupiah, index saham internasional, harga komoditas emas, harga komoditas minyak mentah, dan covid 19.
Metodologi yang Digunakan	Melihat dan membandingkan dari setiap harga saham dengan parameter eksternal yang ada, lalu melakukan uji VECM untuk menentukan apakah parameter tersebut berpengaruh terhadap harga saham yang diuji.
Hasil dan Kesimpulan	Ada beberapa harga saham yang sensitif dan akan berubah terkait beberapa parameter eksternal yang ada, adapun yang tidak berpengaruh terhadap sebagian parameter eksternal yang digunakan. Setiap harga saham memiliki parameter eksternal yang berbeda untuk mempengaruhi perubahan harganya.
Tanggapan terhadap Penelitian	Penelitian telah berhasil menemukan parameter eksternal apa saja yang berpengaruh terhadap saham - saham tertentu. Penelitian ini dapat dilanjutkan

	dengan menerapkan prediksi pada harga saham terkait, dan menggunakan parameter yang berpengaruh terhadap saham tersebut.
--	--