

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Peramalan atau Prediksi**

Prediksi menurut Kamus Besar Bahasa Indonesia, prediksi adalah hasil dari kegiatan memprediksi atau meramal atau memperkirakan (Suryansyah, 2020). Peramalan atau prediksi adalah proses untuk memperkirakan berapa kebutuhan dimasa akan datang meliputi kebutuhan dalam ukuran kuantitas, kualitas, waktu dan lokasi yang dibutuhkan dalam rangka memenuhi kebutuhan. Peramalan permintaan atau prediksi merupakan suatu hal yang diharapkan dapat terealisasi untuk jangka waktu tertentu pada masa yang akan datang (Jumadi, 2021).

Peramalan yang baik mengandung unsur : ada dalam jangkauan waktu, akurat, handal dan bermanfaat, dapat dituliskan, mudah dipahami, dan pembiayaan yang terjangkau (Jumadi, 2021). Prediksi adalah perpaduan antara seni dan ilmu dalam memperkirakan keadaan dimasa yang akan datang, dengan memproyeksikan data-data masa lampau ke masa yang akan datang dengan menggunakan model matematika maupun perkiraan yang subjektif. Kegunaan atau manfaat dari prediksi adalah sebagai alat bantu untuk perencanaan yang efektif dan efisien, untuk menetapkan kebutuhan sumber daya pada masa yang akan datang dan untuk membuat keputusan yang tepat (Ambarwati, 2021).

Peramalan memiliki tujuan untuk mendapatkan informasi dimasa yang akan datang dengan probabilitas kejadian terbesar (Suryansyah, 2020). Peramalan adalah suatu hal yang paling penting bagi suatu perusahaan maupun organisasi bisnis yang mana peramalan berguna sebagai dasar acuan pengambilan keputusan manajemen ataupun melakukan perencanaan, baik perencanaan jangka pendek maupun perencanaan jangka panjang. Dalam peramalan diperlukan seminimal mungkin tingkat kesalahan (*error*) dan karenanya untuk meminimalisir suatu kesalahan tersebut akan lebih baik apabila peramalan dilaksanakan dalam satuan angka (Ambarwati, 2021).

Dalam jaringan syaraf tiruan peramalan atau prediksi memiliki tingkat *error* data yang cukup rendah dan cukup baik dalam proses generalisasi karena didukung

oleh data *training* yang cukup dan proses pembelajaran yang menyesuaikan bobot sehingga model ini mampu untuk meramalkan data *time series* untuk beberapa waktu kedepan (Windarto, 2020).

## 2.2 Tanaman Daun Teh

Teh sebagai bahan minuman (*Camellia Sinensis*) dibuat dari pucuk muda daun teh yang telah mengalami proses pengolahan tertentu seperti pelayuan, penggilingan, oksidasi enzimatis dan pengeringan. Manfaat yang dihasilkan dari minum teh adalah memberikan rasa segar, dapat memberi kesehatan badan dan terbukti tidak memberikan efek yang buruk bagi tubuh manusia. Bahkan teh memiliki senyawa yang berkhasiat bagi tubuh seperti pelifenol dan katekin yang berperan sebagai antioksidan, anti kanker, anti diabetes, anti penyakit jantung dan lain sebagainya (Hayati, 2022).

Teh Merupakan tanaman perkebunan penghasil bahan minuman yang memiliki peran strategis dalam perekonomian indonesia. Usaha budidaya tanaman teh selain dijadikan komoditas ekspor penghasil devisa negara, juga berdampak positif secara berantai terhadap berkembang industri lain, seperti penyerapan tenaga kerja, sumber pendapatan masyarakat tani dan pelaku industri teh, menanggulangi kemiskinan dan konsevasi lingkungan (Rukhman, 2015).

Banyaknya produksi teh dipengaruhi oleh sistem pengambilan. Seperti pengambilan yang kasar atau menggunakan mesin pemetik, jika dilakukan dengan cara ini maka hasilnya akan lebih banyak. Sebaliknya jika pengambilan dilakukan dengan cara yang halus atau manual dengan memetik setu persatu pucuk teh, maka hasil yang akan didapatkan akan sedikit dibandingkan dengan pengambilan kasar. Namun keduanya akan berimbang dalam pemasaran. Sebab jika dilakukan dengan pengambilan kasar kuliatas produk yang dihasilkan tergolong kualitas rendah, sedangkan jika pengambilan dilakukan dengan cara halus walaupun produksinya tergolong sedikit namun kualitasnya tergolong kualitas tinggi (Muljana, 2019).

Harga sendiri merupakan sejumlah nilai yang ditukarkan konsumen dengan manfaat memiliki atau menggunakan produk yang nilainya ditetapkan oleh pembeli dan penjual melalui tawar menawar atau ditetapka oleh penjual untuk satu harga

yang sama terhadap semua pembeli. Harga pasar sendiri adalah harga keseimbangan karena penjual dan pembeli mencapai kesepakatan harga setelah proses tawar-menawar (Hidayat, 2021).

Penetapan harga dan persaingan harga telah dinilai sebagai masalah utama yang dihadapi suatu perusahaan. Keputusan-keputusan mengenai harga dipengaruhi melalui berbagai faktor lingkungan eksternal dalam hal faktor internal. Keputusan harga disesuaikan dengan sasaran pasar, misalnya untuk bertahan hidup, memaksimalkan laba jangka pendek, pangsa pasar atau kepemimpinan mutu produk. Keputusan harga disesuaikan dengan strategi *marketing-mix*-nya, dimana manajemen harus mempertimbangkannya menjadi satu keseluruhan. Jika produk diposisikan atas dasar faktor-faktor bukan harga, maka keputusan-keputusan mengenai mutu, promosi, dan distribusi akan mempengaruhi harga, dan sebaliknya, jika harga merupakan sebuah faktor dalam penentuan posisi, maka harga akan sangat mempengaruhi keputusan lainnya (Hidayat, 2021).

Keputusan harga didasarkan pada pertimbangan suatu organisasi atau perusahaan. Dalam hal faktor eksternal, dapat dijelaskan sebagai berikut: pasar dan permintaan konsumen merupakan plafon harga (harga tertinggi). Konsumen akan membandingkan harga satu produk dengan manfaat yang dimiliki. Oleh karenanya, sebelum melakukan penetapan harga, terlebih dahulu harus memahami hubungan antara harga dan permintaan produk tersebut baik untuk jenis pasar yang berbeda maupun persepsi konsumennya, lalu dianalisis oleh metode-metode yang sesuai. Harga dan tawaran pesaing perlu diketahui untuk menentukan harga atau reaksi konsumen. Setelah keputusan harga diperlukan pula faktor-faktor eksternal lainnya seperti kondisi ekonomi, tingkat inflasi, biaya bunga, resesi, booming dan keputusan-keputusan pemerintah (Hidayat, 2021).

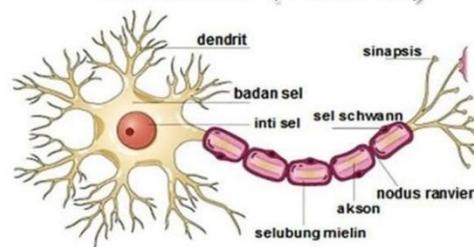
### **2.3 Jaringan Syaraf Tiruan**

Jaringan syaraf tiruan (atau yang biasa disebut *artificial neural network*) merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah buatan disini digunakan karena jaringan syaraf diimplementasikan dengan menggunakan

program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran (Solikhun, 2020).

Berdasarkan sejarah perkembangannya, jaringan syaraf tiruan diakui sebagai salah satu cabang ilmu pengetahuan sejak McCulloch-Pitts (1943) memperkenalkan teorinya dalam *A Logical Calculus of The Ideas Immanent in Nervous Activity*, Donald Hebb (1949) tentang *Organization of Behavior* dan Rosenblatt (1958) tentang *Perceptron* (Awangga, 2020).

### Sel Saraf (Neuron)



**Gambar 2.1** Susunan saraf pada manusia

(Al Haris, 2019)

Jaringan otak manusia tersusun tidak kurang dari  $10^{13}$  buah neuron dengan masing-masing terhubung oleh sekitar  $10^{15}$  buah sinapsis. Setiap sel saraf (neuron) memiliki satu inti sel dimana inti sel ini yang bertugas untuk melakukan pemrosesan informasi. Informasi yang datang diterima oleh dendrit. Selain menerima informasi, dendrit juga menyertai axon sebagai keluaran dari suatu pemrosesan informasi. Informasi hasil olahan ini menjadi masukan bagi neuron lain dimana antar dendrit kedua sel tersebut dipertemukan dengan synapsis. Informasi yang dikirimkan antara neuron ini berupa rangsangan yang dilewatkan melalui dendrit. Informasi yang datang dan diterima oleh dendrit dijumlahkan dan dikirim melalui axon ke dendrit akhir yang bersentuhan dengan dendrit dari neuron yang lain. Informasi ini diterima oleh neuron lain jika memenuhi batasan tertentu, yang sering dikenal dengan nama ambang (*threshold*). Pada kasus ini neuron tersebut dikatakan teraktivasi. Hubungan antara neuron terjadi secara adaptif, artinya struktur hubungan terjadi secara dinamis (Al Haris, 2019). Komponen jaringan syaraf tiruan itu sendiri adalah neuron atau sel saraf yang akan mentransformasikan informasi yang diterima

melalui sambungan keluarnya menuju neuron-neuron lain dan bobot atau pada jaringan syaraf tiruan merupakan hubungan antar neuron-neuron yang dikenal dengan nama bobot menggantikan fungsi sinapsis (Situmorang & Jannah, 2021).

Jaringan Syaraf Tiruan (JST) merupakan pemodelan pembelajaran mesin (*Machine Learning*), yang menggunakan pendekatan belajar secara terbimbing (*supervised*) dan pendekatan belajar secara tidak terbimbing (*unsupervised*). Kemampuan JST dalam mempelajari data, membuatnya mampu melaksanakan tugas-tugas seperti pengenalan pola, klasifikasi, prediksi, pengolahan sinyal dan tugas-tugas lainnya (Armansyah, 2021). Metode jaringan syaraf tiruan ini menggunakan elemen perhitungan non-linier dasar yang disebut neuron yang diorganisasikan sebagai jaringan yang saling terhubung sama halnya dengan jaringan saraf manusia (Windarto et al., 2018).

Secara singkat jaringan syaraf tiruan merupakan representasi dari jaringan biologis dalam bentuk matematika, dengan asumsi bahwa : 1) Pemrosesan informasi terjadi pada banyak elemen sederhana (neuron); 2) Sinyal dikirimkan diantara neuron; 3) Setiap penghubung antar neuron memiliki bobot yang unik untuk mengaktifkan neuron; 4) Fungsi aktivasi merupakan komponen yang digunakan untuk menentukan keluaran jaringan dari hasil perhitungan antara input dengan bobot jaringan, yang kemudian keluaran jaringan akan dibandingkan dengan nilai ambang batas (Armansyah, 2019).

Pada kasus ini, neuron tersebut dikatakan teraktivasi jika hubungan antar neuron terjadi secara adaptif, artinya struktur hubungan terjadi secara dinamis. Ada tiga elemen penting dalam JST , yaitu:

1. Arsitektur jaringan beserta pola hubungan antar neuron (Solikhun, 2020).
2. Algoritma pembelajaran yang digunakan untuk menemukan bobot-bobot jaringan (Solikhun, 2020).
3. Fungsi aktivasi yang digunakan JST terdiri dari sejumlah besar elemen pemroses sederhana yang sering disebut neuron, cell, atau node (Solikhun, 2020).

Proses pengolahan informasi pada JST terjadi pada neuron-neuron. Sinyal antara neuron-neuron diteruskan melalui link-link yang saling terhubung dan

memiliki bobot terisolasi. Kemudian setiap neuron menerapkan fungsi aktivasi terhadap input jaringan.

### 2.3.1 Konsep Dasar Jaringan Syaraf Tiruan

Jaringan saraf tiruan merupakan cabang ilmu dalam bidang ilmu komputer atau teknik informatika yang mengadopsi jaringan syaraf manusia secara biologis kedalam teknologi. Jaringan syaraf dalam ilmu komputer digambarkan melalui model matematika. Terdapat tiga istilah utama dalam jaringan syaraf tiruan (Reyendra, 2021). yaitu:

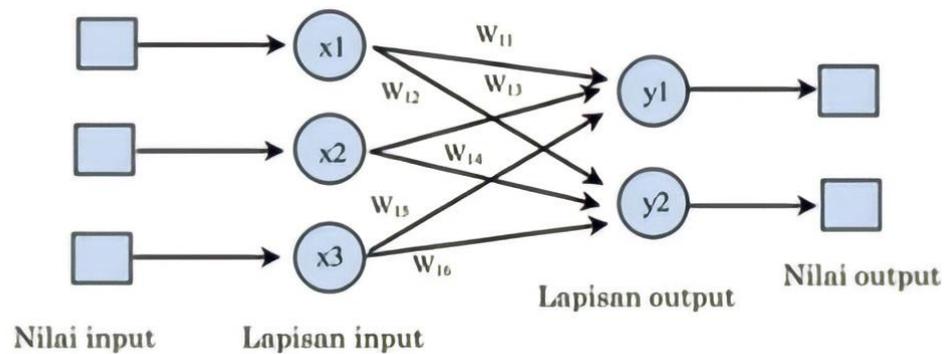
1. Neuron : unit pemroses informasi.
2. Bobot : indikator yang dimiliki penghubung neuron yang akan memperkuat atau memperlemah sinyal.
3. Fungsi aktivasi: fungsi matematika yang menentukan output neuron.
4. Batas ambang: variabel pembanding dalam menentukan besarnya output.
5. Arsitektur jaringan: pola hubungan antar neuron.
6. Metode training/learning/algorithm: metode untuk menentukan bobot penghitung (Reyendra, 2021).

### 2.3.2 Arsitektur Jaringan Syaraf Tiruan

Jaringan syaraf tiruan memiliki beberapa arsitektur jaringan yang sering digunakan dalam berbagai aplikasi yang berisi:

- a. Jaringan lapis tunggal (*single layer network*).

Jaringan dengan lapisan tunggal hanya memiliki satu lapisan input dan satu lapisan output dengan bobot-bobot terhubung. Setiap neuron yang terdapat pada lapisan input selalu terhubung kepada setiap neuron yang terdapat pada lapisan output. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi output tanpa harus melalui lapisan tersembunyi (Simarmata, 2020).

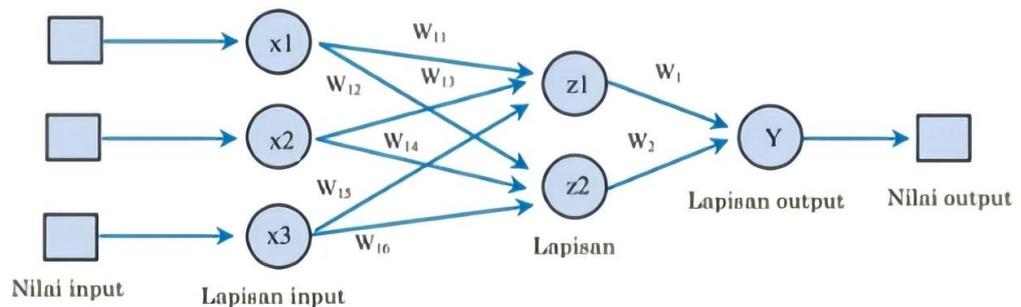


**Gambar 2.2** Arsitektur Lapisan *Single Layer*

(Simarmata, 2020)

b. Jaringan lapis jamak (*multilayer network*)

Jaringan dengan banyak lapisan (jamak) yang memiliki 3 lapisan yang terletak diantara lapisan input, lapisan output, dan lapisan tersembunyi (memiliki 1 atau lebih lapisan tersembunyi). Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit dari pada lapisan dengan lapisan tunggal dengan pembelajaran yang lebih rumit (Simarmata, 2020).

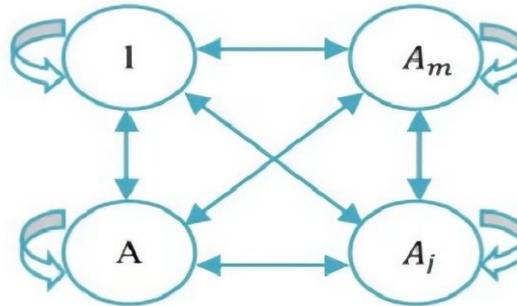


**Gambar 2.3** Arsitektur Lapisan *Multilayer*

(Simarmata, 2020)

c. Jaringan lapis kompetitif (*competitive layer network*)

Jaringan ini mirip dengan jaringan *single layer* ataupun jamak. Hanya saja, ada neuron output yang memberikan sinyal pada neuron input (sering disebut *feedback loop*). Sekumpulan neuron bersaing untuk mendapatkan hak menjadi aktif (Simarmata, 2020).



**Gambar 2.4** Arsitektur Lapisan *Competitive*  
(Simarmata, 2020)

### 2.3.3 Fungsi Aktivasi Jaringan Syaraf Tiruan

Dalam jaringan syaraf tiruan fungsi aktivasi digunakan untuk menentukan keluaran suatu neuron. Argumen fungsi aktivasi adalah net masukan (kombinasi linier masukan dan bobotnya) (Solikhun, 2020). Dalam beberapa fungsi aktivasi yang sering digunakan dalam jaringan syaraf tiruan, antara lain:

a. Fungsi Undak Biner (*Hard Limit*)

Jaringan dengan lapisan tunggal sering menggunakan fungsi undak (*step function*) untuk mengkonfersikan input dari suatu variabel yang bernilai kontinu ke suatu output biner (0 atau 1) (Al Haris, 2019), yang dirumuskan:

$$y = \begin{cases} 0, & \text{jika } x \leq 0 \\ 1, & \text{jika } x > 0 \end{cases}$$

b. Fungsi Undak Biner (*Threshold*)

Fungsi undak biner dengan menggunakan nilai ambang sering juga disebut dengan nama fungsi ambang (*threshold*). Fungsi undak biner (dengan nilai ambang  $\emptyset$ ) (Al Haris, 2019), dirumuskan sebagai:

$$y = \begin{cases} 0, & \text{jika } x < \emptyset \\ 1, & \text{jika } x \geq \emptyset \end{cases}$$

c. Fungsi Bipolar (*Symetric Hard Limit*)

Fungsi bipolar sebenarnya hampir sama dengan fungsi undak biner, hanya saja output yang dihasilkan 1, 0, -1 (Al Haris, 2019). Fungsi *Symetric Hard Limit* dirumuskan:

$$y = \begin{cases} 1, & \text{jika } x > 0 \\ 0, & \text{jika } x = 0 \\ -1, & \text{jika } x < 0 \end{cases}$$

d. Fungsi Bipolar (*Threshold*)

Fungsi bipolar sebenarnya hampir sama dengan fungsi undak biner dengan *Threshold*, hanya saja output yang dihasilkan berupa 1, 0, atau -1 (Al Haris, 2019). Fungsi *bipolar* (dengan nilai ambang  $\emptyset$ ) dirumuskan:

$$y = \begin{cases} 1, & \text{jika } x \geq \emptyset \\ -1, & \text{jika } x < \emptyset \end{cases}$$

e. Fungsi Linear (*Identitas*)

Fungsi linear memiliki nilai output yang sama dengan nilai inputnya (Al Haris, 2019), yang dirumuskan:

$$y = x$$

f. Fungsi Saturating Linear

Fungsi ini akan bernilai 0 jika input kurang dari  $-1/2$ , dan akan bernilai 1 jika inputnya lebih dari  $1/2$ . Jika nilai input terletak antara  $-1/2$  dengan  $1/2$ , maka nilai outputnya akan bernilai sama dengan nilai input ditambah  $1/2$  (Al Haris, 2019). Fungsi saturating linear dirumuskan:

$$y = \begin{cases} 1; & \text{jika } x \geq 0,5 \\ x + 0,5; & \text{jika } -0,5 \leq x \leq 0,5 \\ 0; & \text{jika } x \leq -0,5 \end{cases}$$

g. Fungsi *Symetric* Saturating Linear

Fungsi ini akan bernilai -1 jika input kurang dari -1, dan akan bernilai 1 jika inputnya lebih dari 1. Jika nilai input terletak antara -1 dengan 1, maka nilai outputnya akan bernilai sama dengan nilai input (Al Haris, 2019), yang dirumuskan:

$$y = \begin{cases} 1; & \text{jika } x \geq 1 \\ x; & \text{jika } -1 \leq x \leq 1 \\ 0; & \text{jika } x \leq -1 \end{cases}$$

h. Fungsi Sigmoid Biner

Fungsi ini digunakan untuk jaringan syaraf yang dilatih dengan menggunakan metode *backpropagation*. Fungsi sigmoid biner memiliki nilai pada range 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan untuk jaringan syaraf yang membutuhkan nilai output yang terletak pada interval 0 sampai 1. Namun, fungsi

ini bisa juga digunakan oleh jaringan syaraf yang nilai outputnya 0 atau 1 (Al Haris, 2019). Fungsi sigmoid biner dirumuskan:

$$y = f(x) = \frac{1}{1 + e^{-\sigma x}}$$

$$\text{dengan } f'(x) = \sigma f(x)[1 - f(x)]$$

#### i. Fungsi Sigmoid Bipolar

Fungsi ini hampir sama dengan fungsi sigmoid biner hanya saja output dari fungsi ini memiliki range antara 1 sampai -1 (Al-Haris, 2019), dengan rumus:

$$y = f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

$$\text{dengan } f'(x) = \frac{\sigma}{2} [1 + f(x)][1 - f(x)]$$

### 2.3.4 Kelebihan dan Kekurangan Jaringan Syaraf Tiruan

Namun begitu jaringan syaraf tiruan (JST) tetap memiliki kelebihan dan kekurangan. Adapun kelebihan jaringan syaraf tiruan ini adalah sebagai berikut:

1. JST dapat menjalankan tugas yang tidak dapat dilakukan oleh program linear.
2. JST dapat bekerja secara paralel.
3. JST dapat mempelajari pola data tanpa perlu diprogram ulang.
4. JST dapat diterapkan tanpa ada masalah yang mendasar (Arif, 2021).
5. JST dapat menyimpulkan pengetahuan walaupun tidak mempunyai kepastian.
6. JST dapat melakukan general dan mengekstrak dari pola.
7. JST dapat menimbulkan satu pola pengetahuan melewati batas belajar atau *self-organizing*.
8. Mempunyai *fault tolerances* atau noise dianggap gangguan.
9. Mampu memperhitungkan secara berurutan sehingga proses lebih cepat.
10. Mempunyai kelebihan belajar, melakukan suatu pekerjaan berdasarkan data yang diberikan.
11. JST dapat berdiri sendiri dari informasi yang didapat saat pembelajaran.
12. JST dapat melakukan suatu pekerjaan secara berurutan sehingga *hardware* yang dirancang dan diproduksi dapat menghasilkan keuntungan (Reyendra, 2021).

Dibalik kelebihan JST yang tertera diatas, terdapat pula kekurangan JST itu sendiri. Diantaranya:

1. Untuk memprediksi pola data, diperlukan proses pembelajaran (*training*) lebih dahulu, sehingga memerlukan lebih banyak data empiris.
2. Untuk dapat digunakan dalam sebuah *microprocessor*, dibutuhkan penyesuaian arsitekturnya supaya sesuai dengan *microprocessor*.
3. Membutuhkan waktu proses yang lama untuk model yang kompleks.
4. Model JST dapat memprediksi pola data, tetapi tidak dapat menjelaskan mekanismenya didalam krena tidak memiliki persamaan matematika yang eksplisit (Arif, 2021).

#### 2.4 Metode Backpropagation

*Backpropagation* ialah algoritma pendidikan yang terawasi serta umumnya digunakan untuk mengganti bobot- bobot yang tersambung dengan neuron-neuron yang terdapat pada susunan tersembunyinya. Algoritma *backpropagation* memakai *error* output untuk mengganti nilai bobot- bobotnya dalam arah mundur (*backward*) (Hutabarat et al., 2021). *Backpropagation* bersifat adaptive yaitu dapat menyesuaikan pada dataset dan *fault tolerance* (kesalahan *error* kecil) dalam menyelesaikan masalah pada sistem (Maiyuriska, 2022). *Backpropagation* melakukan pembelajaran terbimbing (*supervised learning*) yang digunakan pada jaringan multi-layer yang terdiri dari beberapa *hidden-layer* yang bertujuan untuk meminimalkan *error* terhadap jaringan yang menghasilkan keluaran (*output*)(Putra & Ulfa Walmi, 2020).

Jaringan *backpropagation* dikembangkan oleh Paul Werbos pada tahun 1974, kemudian dikembangkan lebih lanjut oleh David E. Rumelhart, Geoffrey E. Hinton dan Ronald J. William pada tahun 1986 (Nahari, 2017). Hampir 80% dari seluruh jaringan saraf tiruan yang ada dalam perkembangannya menggunakan jaringan ini karena mudah dalam proses belajarnya. Jaringan syaraf tiruan *backpropagation* memiliki kemampuan untuk menentukan hubungan antara sekelompok pola masukan dengan sekelompok pola keluaran yang diberikan dan menggunakan hubungan ini pada saat diberikan pola masukan baru (Apriyani, 2018).

### 2.4.1 Tahap-Tahap Algoritma *Backpropagation*

Algoritma *backpropagation* memiliki dua tahap utama yaitu perambatan maju dan perambatan mundur. Perambatan maju ialah proses yang dilakukan untuk mengetahui hasil output, sedangkan perambatan mundur adalah proses yang ditunjukkan untuk mengupdate/memperbarui nilai bobot  $v$  dan nilai bobot  $w$  (Wadi, 2021). *Backpropagation* merupakan algoritma yang bekerja dengan mengenali pola data berdasarkan histori data terdahulu sehingga mampu mengenali pola data untuk melakukan prediksi ataupun klasifikasi terhadap data yang akan datang (Wadi, 2021).

Untuk melakukan prediksi data menggunakan *backpropagation* harus dilakukan dua proses terlebih dahulu yaitu:

#### 1. Proses Pelatihan

Proses ini merupakan proses yang bertujuan untuk mengenali pola data. proses pelatihan ini untuk melatih bobot sehingga dapat mengenali pola data. hasil dari proses pelatihan ini adalah bobot  $w$ , bobot  $v$  dan bias yang terlatih. Bobot-bobot terlatih inilah yang merupakan representasi dari pengetahuan JST. Pada proses pelatihan ini memiliki dua perambatan yaitu perambatan maju dan perambatan mundur. Proses pelatihan ini dapat diartikan juga sebagai perulangan dari perambatan maju dan perambatan mundur untuk mendapatkan nilai bobot  $w$ , bobot  $v$  dan bias yang terlatih (Wadi, 2021).

#### 2. Proses Pengujian

Proses ini merupakan proses prediksi yang dilakukan pada data uji atau data yang akan datang. Pada proses pengujian ini hanya dilakukan perambatan maju saja (Wadi, 2021).

Algoritma *backpropagation* memiliki parameter yang digunakan dalam memprediksi suatu hal. Berikut adalah parameter-parameter yang harus ditentukan:

##### a. Jumlah neuron pada input *hidden layer* ( $n_{input}$ )

Parameter ini digunakan untuk menentukan berapa jumlah neuron yang terdapat pada *input layer* (Wadi, 2021).

##### b. Jumlah neuron pada *hidden layer* ( $n_{hidden}$ )

Parameter ini digunakan untuk menentukan berapa jumlah neuron yang terdapat pada *hidden layer* (Wadi, 2021).

c. Jumlah neuron *output layer* ( $n_{\text{output}}$ )

d. Parameter ini digunakan untuk menentukan berapa jumlah neuron yang terdapat pada *output layer* (Wadi, 2021).

e. Laju pembelajaran( $\alpha$ )

Parameter ini akan menentukan seberapa cepat laju pembelajaran JST *backpropagation*. Dengan kata lain, parameter ini akan menentukan seberapa cepat pelatihan pada JST *backpropagation* (Wadi, 2021).

f. Jumlah itersi

Parameter ini akan menentukan jumlah siklus dari proses pelatihan. Dengan kata lain, parameter ini akan menentukan berapa kali proses perambatan maju dan proses perambatan mundur akan dilakukan pada proses pelatihan (Wadi, 2021).

g. Toleransi *error*

Parameter ini dilakukan sebagai acuan *error* antara output JST dan target output JST. Parameter ini biasa digunakan sebagai pemicu/triger yang akan menghentikan proses pelatihan. Jadi ketika *error* antara output JST dengan target output JST lebih kecil atau sama dengan toleransi *error* maka proses pelatihan dapat dihentikan (Wadi, 2021).

#### 2.4.2 Arsitektur Algoritma *Backpropagation*

Algoritma *backpropagation* memiliki arsitektur tersendiri yang meliputi tiga lapisan (*layer*) utama serta bobot dan bias (Wadi, 2021) yang berisi:

1. Lapisan Masukan (*Input Layer*)

Layer ini berisi neuron-neuron yang menjadi masukan bagi JST *backpropagation*. Neuron pada layer ini dapat diubah sesuai dengan jumlah data yang sedang diolah, atau sesuai dengan jumlah parameter data yang akan diproses. Neuron–neuron yang terdapat pada layer ini biasa dikenal dengan istilah *input layer* atau yang disimbolkan dengan  $X_i$  ( $x_1, x_2, x_3, \dots, x_n$ ) (Wadi, 2021).

2. Lapisan Tersembunyi (*Hidden Layer*)

Layer ini merupakan bagian JST yang berada ditengah-tengah penghubung antara *input layer* dan *output layer*. Layer ini berisi neuron-neuron tersembunyi atau sering disebut *hidden layer* dimana jaringan dapat memiliki lebih dari satu *hidden layer* namun sifatnya tidak wajib ada. Hidden layer ini biasa di simbolkan dengan  $Z_i$  ( $z_1, z_2, z_3, \dots, z_n$ ) (Wadi, 2021).

### 3. Lapisan Keluaran (Output Layer)

Layer ini berisi neuron-neuron yang menjadi keluaran JST *backpropagation*. *Output layer* ini terletak di bagian akhir yang biasa disimbolkan dengan  $Y_i$  ( $y_1, y_2, y_3, \dots, y_n$ ) (Wadi, 2021).

### 4. Bobot V+Bias

Setiap neuron yang terdapat pada input layer dan *hidden layer* dihubungkan dengan bobot dan bias. Bobot yang menjadi penghubung antara input layer dengan hidden layer dinamakan bobot V. Bobot V merupakan komponen wajib yang menjadi penghubung setiap neuron yang terdapat pada input layer dan *hidden layer*. Sedangkan bias merupakan komponen yang dapat ditiadakan karena hanya memiliki fungsi mempercepat laju pembelajaran serta tidak merupakan komponen wajib (Wadi, 2021).

### 5. Bobot W+Bias

Setiap neuron yang terdapat pada *hidden layer* dan output layer dihubungkan dengan bobot dan bias. Bobot yang menjadi penghubung antara hidden layer dengan output layer dinamakan bobot W. Bobot W merupakan komponen wajib yang menjadi penghubung setiap neuron yang terdapat pada hidden layer dan output layer. Sedangkan bias merupakan komponen yang dapat ditiadakan karena hanya memiliki fungsi mempercepat laju pembelajaran (Wadi, 2021).

## 2.4.3 Pelatihan Algoritma *Backpropagation*

Pelatihan algoritma *backpropagation* memiliki 2 perambatan, yaitu perambatan maju (*forward propagation*) dan perambatan mundur (*backward propagation*) (Wadi, 2021).

### A. Perambatan maju (*forward propogation*)

Pada perambatan maju dilakukan perhitungan sinyal yang dikirim dari input layer menuju *hidden layer* (perhitungan nilai neuron pada *hidden layer*). Selain itu dilakukan perhitungan sinyal keluaran yang diteruskan dari *hidden layer* ke *output layer* (perhitungan nilai neuron pada *output layer*). Berikut adalah tahapan yang dilakukan pada proses perambatan maju (Wadi, 2021):

1. Perhitungan nilai neuron pada hidden layer (nilai Z)

Perhitungan  $Z_i$  ( $z_1, z_2, z_3, \dots, z_n$ ) dilakukan dengan persamaan (Wadi, 2021):

$$Z\_in_j = V_{0j} + \sum_{i=1}^n Z_i V_{ij}$$

$$Z_j = f(Z\_in_j)$$

Dimana:

$V_{0j}$  = bias pada neuron *hidden layer* ke-j

$X_i$  = neuron input layer ke i

$V_{ij}$  = bobot penghubung *input layer* ke-i dan neuron *hidden layer* ke-j

$Z\_in_j$  = sinyal informasi dari *input layer* ke neuron *hidden layer* ke-j

$Z_j$  = neuron *hidden layer* ke-j

$F(z\_in_j)$  = fungsi aktivasi terhadap nilai  $z\_in_j$ .

2. Perhitungan nilai neuron pada *output layer* (nilai Y)

Perhitungan  $Y_i$  ( $y_1, y_2, y_3, \dots, y_n$ ) dilakukan dengan persamaan (Wadi, 2021):

$$y\_in_k = W_{k0} + \sum_{j=1}^p Z_j W_{kj}$$

$$Y_k = f(Y\_in_k)$$

Dimana:

$W_{0k}$  = bias pada neuron *output layer* ke-k

$Z_j$  = neuron *hidden layer* ke-j

$W_{jk}$  = bobot penghubung neuron *hidden layer* ke-j dan *output layer* ke-k

$Y\_in_k$  = sinyal informasi dari *hidden layer* ke neuron *output layer* ke-k

$Y_k$  = neuron *output layer* ke-k

$F(y_{in_k})$  = fungsi aktivasi terhadap nilai  $y_{in_k}$ .

### B. Perambatan mundur (*backward propagation*)

Pada perambatan mundur dilakukan untuk memperbarui bobot yang menghubungkan antara *output layer* dan *hidden layer* (bobot  $W$ ). Dan untuk memperbarui bobot yang menghubungkan antara *hidden layer* dan *input layer* (bobot  $V$ ). Berikut adalah tahapan proses yang dilakukan pada perambatan mundur (Wadi, 2021):

$$\begin{aligned}\Delta_k &= (t_k - y_k) f'(y_{in_k}) \\ &= (t_k - y_k) y_k(1 - y_k)\end{aligned}$$

$$\Delta W_{jk} = \alpha \delta_k z_j$$

$$\Delta W_{0k} = \alpha \delta_k$$

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk}$$

$$W_{0k}(\text{baru}) = W_{0k}(\text{lama}) + \Delta W_{0k}$$

Dimana:

$\delta_k$  = faktor koneksi dari neuron *output layer* ke-k.

$t_k$  = target *output* pada neuron *output layer* ke-k.

$y_k$  = neuron *output layer* ke-k.

$\alpha$  = laju pembelajaran (*learning rate*).

$\Delta W_{jk}$  = koneksi bobot penghubung antara neuron *output layer* ke-k dan neuron *hidden layer* ke-j.

$\Delta W_{0k}$  = koneksi bias pada neuron *output layer* ke-k.

$W_{jk}(\text{baru})$  = bobot terbaru penghubung antara neuron *output layer* ke-k dan neuron *hidden layer* ke-j.

$W_{jk}(\text{lama})$  = bobot lama penghubung antara neuron *output layer* ke-k dan neuron *hidden layer* ke-j.

$W_{0k}(\text{baru})$  = bias baru yang terhubung ke neuron *output layer* ke-k.

$W_{0k}(\text{lama})$  = bias lama yang terhubung ke neuron *output layer* ke-k.

### C. Perhitungan perambatan mundur dari *hidden layer* ke *input layer*

Tahapan ini bertujuan untuk memperbarui bobot yang menghubungkan antara hidden layer dan input layer (bobot  $V$ ). Berikut adalah persamaan yang digunakan dalam tahapan ini (Wadi, 2021):

$$\begin{aligned}\delta_{in_j} &= V_{0j} + \sum_{k=1}^n \delta_k W_{jk} \\ \delta_j &= \delta_{in_j} f'(z_{in_j}) \\ &= \delta_{in_j} z_j (1 - z_j) \\ \Delta V_{ij} &= \alpha \delta_j x_i \\ \Delta V_{0j} &= \alpha \delta_j \\ V_{ij}(\text{baru}) &= v_{ij}(\text{lama}) + \Delta V_{ij} \\ V_{0j}(\text{baru}) &= v_{0j}(\text{lama}) + \Delta V_{0j}\end{aligned}$$

Dimana:

$\delta_{in_j}$  = sinyal faktor koreksi dari *output layer* ke neuron *hidden layer* ke-j

$\delta_j$  = faktor koreksi dari neuron *hidden layer* ke-j

$\alpha$  = laju pembelajaran (*learning rate*)

$\Delta V_{ij}$  = koreksi bobot yang menghubungkan antara neuron *hidden layer* ke-j dan neuron *input layer* ke-i

$\Delta V_{0j}$  = koreksi bias pada neuron *hidden layer* ke-j

$X_j$  = neuron *input layer* ke-i

$V_{ij}(\text{baru})$  = bobot terbaru yang menghubungkan antara neuron *hidden layer* ke-j dan neuron *input layer* ke-i

$V_{ij}(\text{lama})$  = bobot lama yang menghubungkan antara neuron *hidden layer* ke-j dan neuron *input layer* ke-i

$V_{0j}(\text{baru})$  = bias baru yang terhubung ke neuron *hidden layer* ke-j

$V_{0j}(\text{lama})$  = bias lama yang terhubung ke neuron *hidden layer* ke-j.

Perincian proses adalah :

0. Mengenalkan parameter-parameter terkait seperti bobot-bobot, konstanta laju pelatihan ( $\alpha$ ), nilai error atau nilai bobot (bila menggunakan nilai bobot sebagai

kondisi berhenti) atau set maksimal epoch (jika menggunakan banyaknya epoch sebagai kondisi berhenti) (Cynthia & Ismanto, 2017).

1. Jika belum sesuai maka proses dari langkah 2 hingga 9 maka akan terus berlanjut (Cynthia & Ismanto, 2017).
2. Cara pemasangan untuk setiap pola latihan maka akan dilakukan proses dari poin ke 3 hingga ke 8 (Cynthia & Ismanto, 2017).

### **Tahap I : Umpan Maju (*feedforward*)**

3. Untuk data masuk  $x_i$  (dari unit ke-1 sampai node ke-n pada lapisan input) akan mengirim tanda kepada lapisan hidden dari unit input (Cynthia & Ismanto, 2017).
4. Setiap bobot dan bias akan dikalikan dengan nilai yang terdapat pada lapisan tersembunyi (Cynthia & Ismanto, 2017) :

$$Z_{in_j} = V_{0j} + \sum_{i=1}^n Z_i V_{ij}$$

$$Z_j = f(Z_{in_j}) = \frac{1}{1 + e^{-x_{net^j}}}$$

5. Nilai pada data keluar ( $y_k$ ,  $k=1,2,3,\dots,m$ ) dikali bobot, ditambah bias (Cynthia & Ismanto, 2017)

$$y_{in_k} = W_{k0} + \sum_{j=1}^p Z_j W_{jk}$$

$$y_k = f(y_{in_k}) = \frac{1}{1 + e^{-y_{net^k}}}$$

### **Tahap II : Umpan Mundur (*backward*)**

6. Untuk nilai keluar ( $y_k$ ,  $k=1,2,3,\dots,m$ ) akan menerima rincian target  $t_k$  yang cocok dengan nilai input dan *output* pada proses dan dilanjut dengan menganalisis nilai error pada *layer output* ( $\delta_k$ ).  $\delta_k$  perhitungannya digunakan untuk memperbaiki nilai bias dan bobot ( $\Delta W_{jk}$  dan  $\Delta W_{0k}$ ) yang berada pada *layer hidden* dan *output* (Cynthia & Ismanto, 2017).

$$\delta_k = (t_k - y_k) f'(y_{in_k}) = (t_k - y_k) y_k(1-y_k)$$

Hitung suku perubahan bobot  $W_{jk}$  (yang akan digunakan untuk merubah bobot  $W_{jk}$ ) dengan laju pelatihan  $\alpha$ .

$$\Delta W_{jk} = \alpha \delta_k z_j ; k=1,2,3,\dots,m; j=0,1,\dots,p$$

Hitung perubahan bias

$$\Delta W_{0k} = \alpha \delta_k$$

7. Pada setiap unit di lapisan tersembunyi (dari unit ke-1 hingga ke-p;  $i=1\dots n; k=1\dots m$ ) dilakukan perhitungan informasi kesalahan lapisan tersembunyi ( $\delta_j$ ).  $\delta_j$  kemudian digunakan untuk menghitung besar koreksi bobot dan bias ( $\Delta V_{ji}$  dan  $\Delta V_{j0}$ ) antara lapisan input dan lapisan tersembunyi (Cynthia & Ismanto, 2017).

$$\delta_{inj} = \sum_{k=1}^m \delta_k w_{kj}$$

$$\delta_j = \delta_{inj} f'(\delta_{inj}) = \delta_{inj} z_j(1-z_j)$$

bobot pada  $V_{ji}$  dihitung (pada perbaikan bobot  $V_{ji}$ )

$$\Delta V_{ji} = \alpha \delta_j x_i$$

Bias berubah (untuk memperbaiki  $V_{j0}$ ).

$$\Delta V_{j0} = \alpha \delta_j$$

### Tahap III : pembaharuan Bobot dan Bias

8. Pada lapisan output ( $y_k, k=1,2,3,\dots,m$ ) nilai bobot dan bias akan diperbaharui ( $j = 0,1,2,\dots,p$ ) untuk mendapatkan nilai bias dan bobot yang baru (Cynthia & Ismanto, 2017):

$$W_{kj}(\text{baru}) = W_{kj}(\text{lama}) + \Delta W_{kj}$$

9. Demikian juga untuk setiap unit tersembunyi mulai dari unit ke-1 sampai dengan unit ke-p dilakukan pengupdatean bobot dan bias (Cynthia & Ismanto, 2017)
10. Tes kondisi berhenti apabila error ditemukan. Jika kondisi berhenti terpenuhi. Maka pelatihan jaringan dapat dihentikan (Cynthia & Ismanto, 2017).

Sedangkan untuk pengujian *backpropagation* sendiri berisi:

0. Inisialisasi bobot (hasil pelatihan)
1. Untuk setiap vektor input, kerjakan langkah 2 – 4

2. Untuk  $i=1, \dots, n$  : set aktivasi unit input X1
3. Untuk  $j=1, \dots, p$

$$Z_{in\ j} = V_{0j} + \sum_{i=1}^n Z_i V_{ij}$$

$$Z_j = f(Z_{inj})$$

4. Untuk  $k=1, \dots, p$ ;

$$y_{in\ j} = W_{0k} + \sum_{i=1}^n Z_i W_{jk}$$

$$y_j = f(y_{inj})$$

Sebelum diproses, data dinormalisasi terlebih dahulu. Normalisasi terhadap data dilakukan agar keluaran jaringan sesuai dengan fungsi aktivasi yang digunakan. Fungsi aktivasi yang peneliti gunakan dalam penelitian ini adalah fungsi aktivasi sigmoid. Fungsi sigmoid adalah fungsi asimtotik (tidak pernah mencapai 0 ataupun 1) (Wanto & Windarto, 2017), dengan rumus:

$$X' = \frac{0.8(x-a)}{b-a} + 0.1$$

Dimana:

$\alpha$  = nilai minimum

$b$  = nilai maksimum.

Lalu denormalisasi testing. Setelah dilakukan proses training dan testing pola-pola yang dilatih, maka akan diperoleh hasil bahwa pengujian terhadap pola-pola tersebut apakah telah benar/akurat atau sebaliknya. Untuk menghitung Rata-rata Error (RMSE) jaringan, dapat dilakukan dengan rumus (Cynthia & Ismanto, 2017):

$$RMSE = \sum_{i=1}^n \frac{(y_1 - y_n)^2}{n}$$

Dimana:

$Y_i$  = nilai aktual data (target)

$Y_n$  = nilai hasil prediksi (actuaal output)

$N$  = jumlah data yang diuji.

Sedangkan untuk proses denormalisasi atau mengembalikan kembali nilai hasil prediksi jaringan ke bentuk data semula (sebelum normalisasi) dapat menggunakan rumus sebagai berikut (Cynthia & Ismanto, 2017).

$$X_i = y_n(x_{\max} - x_{\min}) + x_{\min}$$

Dimana:

$X_i$  = nilai  $x$  yang akan dilakukan denormalisasi

$Y_n$  = nilai hasil prediksi (actual output) yang sesuai dengan  $x_i$

$X_{\max}$  = nilai maksimum pada barisan  $x$

$X_{\min}$  = nilai minimum pada barisan  $x$ .

#### 2.4.4 Kelebihan dan Kekurangan Algoritma Backpropagation

Beberapa kelebihan algoritma backpropagation adalah:

1. Dapat diaplikasikan pada penyelesaian suatu masalah berkaitan dengan identifikasi, prediksi, peramalan, pengenalan pola dan sebagainya (Windarto, 2020).
2. Kemampuannya untuk belajar (bersifat adaptif) dan kebal terhadap kesalahan (*Fault tolerance*) sehingga dapat mewujudkan sistem yang tahan kerusakan (*robust*) dan bekerja secara konsisten (Windarto, 2020).
3. Melatih jaringan untuk mendapat kesimpulan selama proses pelatihan sehingga dapat memberikan respon yang benar terhadap pola masukan yang serupa dengan pola yang dipakai selama pelatihan (Windarto, 2020).

Sedangkan kekurangan dari algoritma backpropagation adalah:

1. Membutuhkan waktu cukup lama dalam proses pembelajaran untuk mencapai konvergen (Windarto, 2020).
2. Parameter learning rate atau tingkat pembelajaran akan selalu berubah-ubah sesuai dengan kondisi error pada setiap iterasinya (Windarto, 2020).
3. Dalam menghitung perubahan bobot algoritma *backpropagation* dapat menyebabkan masalah lokal minimum sehingga tidak stabil (Windarto, 2020).

## 2.5 Jaringan Syaraf untuk Peramalan

Teknik peramalan banyak digunakan untuk proses perencanaan dan pengambilan keputusan, suatu ramalan mencoba memperkirakan apa yang akan terjadi dan apa yang akan menjadi kebutuhan. Dalam JST terdapat teknik peramalan yang sering digunakan yaitu *backpropagation*. Teknik ini biasanya digunakan pada jaringan multilayer dengan tujuan meminimalkan *error* pada keluaran yang dihasilkan oleh jaringan. Salah satu cara peramalan adalah dengan metode urutan waktu (*time series*) menggunakan data history (data waktu yang lampau), misalnya data permintaan, untuk membuat ramalan permintaan diwaktu mendatang (Cynthia & Ismanto, 2017).

Tujuan dari metode ini adalah untuk mengidentifikasi pola data history dan kemudian mengekstrapolasikan pola ini ke masa datang. Metode NN ini dilatih dengan seperangkat data untuk bisa mengenal dan mengidentifikasi pola data atau kurva. Proses pelatihan disebut tahap belajar (*Learning Proses*), yang merupakan bagian penting dalam metode ini. Pemilihan algoritma dan parameter yang bersesuaian dan penentuan berapa banyak perangkat data yang dibutuhkan sangat penting untuk menentukan akurasi dari peramalan yang dihasilkan (Cynthia & Ismanto, 2017).

## 2.6 Perhitungan Kinerja Error

Hasil peramalan yang akurat adalah peramalan yang bisa meminimalkan kesalahan meramal. Karena itu dalam mengukur kinerja error JST pada pengerjaan tugas akhir ini digunakan metode perhitungan *Mean Absolute Percentage Error* (MAPE) dan *Mean Squared Error* (MSE) untuk menghitung tingkat akurasi prediksi.

### 2.6.1 Mean Square Error (MSE)

*Mean Squared Error* (MSE) merupakan ukuran rata-rata kuadrat kesalahan antara nilai aktual dengan nilai hasil peramalan. MSE memberikan informasi seberapa dekat sekumpulan poin dengan garis regres (Wardhani, 2022). Mean Square Error (MSE) juga dapat disebut salah satu perhitungan yang digunakan untuk mengetahui nilai error dari sebuah metode yang nantinya akan menghitung

nilai hasil prediksi dengan cara mengurangkan nilai yang sebenarnya dengan nilai hasil prediksi. Selengkapnya dapat dilihat pada persamaan (Humam et al., 2019):

$$MSE = \frac{\sum(\text{Actual}-\text{Forecast})^2}{n} = \frac{\sum(\text{Error})^2}{n}$$

$$MSE = \frac{\sum(t-y_k)^2}{n}$$

Dimana:

- t = nilai actual
- $y_k$  = nilai hasil prediksi
- k = Indeks untuk unit output
- n = Jumlah data latih

### 2.6.2 Mean Absolute Percentage Error (MAPE)

*Mean Absolute Percentage Error* (MAPE) merupakan salah satu metode pengujian kesalahan hasil peramalan yang dihitung dengan cara membagi kesalahan absolute tiap priode dengan nilai observasi yang nyata pada priode tersebut dan merata-ratakan presentasi kesalahan absolute tersebut (Wardhani, 2022). Adapun persamaannya:

$$MAPE = \frac{\sum \frac{|\text{Actual} - \text{Forecast}|}{\text{Actual}} \times 100\%}{n}$$

$$MAPE = \frac{\sum_{i=1}^n \left| \frac{Y - Y'}{Y} \right| \times 100}{n}$$

Dimana:

- y = nilai aktual
- $y'$  = nilai prediksi
- n = jumlah data

## 2.7 Contoh Perhitungan Jaringan Syaraf Tiruan *Backpropagation*

Pada contoh perhitungan manual ini diambil dari jurnal Solikhun, M. Safii dan Agus Trisno yang berjudul jaringan saraf tiruan untuk memprediksi tingkat pemahaman siswa terhadap matapelajaran dengan menggunakan algoritma

*backpropagation*. Pada prediksi ini penulis menggunakan beberapa kriteria yaitu : Pengetahuan, ketrampilan/kemampuan, penilaian & beban kerja dan bimbingan & konseling sedangkan data dipresentasikan dalam bentuk numerik antara 0 sampai dengan 1 sebagai pengenalan pola dan keluaran yang merupakan prediksi pemahaman siswa yang diperoleh dari model arsitektur terbaik pada saat penentuan pola terbaik. Hal ini dikarenakan jaringan menggunakan fungsi aktivasi sigmoid biner (logsig) yang rangenya dari 0 sampai 1. Nilai-nilai yang digunakan diperoleh berdasarkan kategori dari masing-masing variabel selain juga untuk memudahkan mengingat dalam pendefinisianya (Solikhun d. , 2017).

**Tabel 2.1** Daftar Kriteria dalam Pemahaman Siswa terhadap Mata pelajaran  
(Solikhun d. , 2017)

No	Kriteria	Variabel	Rank Rata <sup>2</sup> nilai	Keterangan	Berat
1	Pengetahuan	A	5	Sangat Setuju	1
			4 – 4,99	Setuju	0,8
			3 – 3,99	Sedang	0,6
			2 – 2,99	Tidak Setuju	0,4
			0 – 1,99	Sangat Tidak Setuju	0,2
2	Keterampilan/ Kemampuan	B	5	Sangat Setuju	1
			4 – 4,99	Setuju	0,8
			3 – 3,99	Sedang	0,6
			2 – 2,99	Tidak Setuju	0,4
			0 – 1,99	Sangat Tidak Setuju	0,2
3	Penilaian dan Beban Kerja	C	5	Sangat Setuju	1
			4 – 4,99	Setuju	0,8

			3 – 3,99	Sedang	0,6
--	--	--	----------	--------	-----

			2 – 2,99	Tidak Setuju	0,4
			0 – 1,99	Sangat Tidak Setuju	0,2
4	Bimbingan dan Konseling	D	5	Sangat Setuju	1
			4 – 4,99	Setuju	0,8
			3 – 3,99	Sedang	0,6
			2 – 2,99	Tidak Setuju	0,4
			0 – 1,99	Sangat Tidak Setuju	0,2

Sampel Data baku 40 siswa yang telah diproses dan ditransformasikan adalah sebagai berikut:

**Tabel 2.2** Sampel dari data yang telah ditransformasikan  
(Solikhun d. , 2017)

NO	NIS	NAMA	VARIABEL INPUT				TARGET
			A	B	C	D	
			X1	X2	X3	X4	X5
1	1514	ADE IRMA CITA DEWI	0.8	0.6	0.6	0.8	1
2	1517	ANDI KUSUMA	0.8	1	0.8	0.8	1
3	1518	ANDREANSYAH	0.8	0.6	0.4	0.4	1
4	1519	ANJAS SULISTIAWAN	0.8	0.8	0.8	1	1
5	1520	APRIL FRANCIS TAMPUBOLON	0.6	0.6	0.8	0.6	1
6	1522	ARYA DWI SUMANTHA SIDAURU	0.8	0.8	0.8	0.8	1
7	1523	ARYA SUKMA JAYA WARDANA	0.6	0.8	0.6	0.8	1
8	1524	DEVI GUSTIADANI PARANGIN-ANGIN	0.8	0.6	0.6	0.8	1
9	1526	DICKY WAHYUDI	0.8	0.8	0.6	0.8	1
10	1527	DWI SHAFIRA BATUBARA	0.6	0.8	0.6	0.6	1
11	1528	EDI KURNIAWAN	0.8	0.8	0.6	0.8	1
12	1529	FAISAL	0.6	0.4	0.6	0.8	1
13	1530	FARADILA HAFIZAH PARINDURI	0.8	0.8	1	1	1
14	1532	ILLIAS	0.6	0.6	0.8	0.6	1
15	1533	INDAH DESWITA	0.8	1	0.8	0.8	1
16	1534	JENI ARISKA	0.8	0.8	0.8	0.8	1
17	1536	LISNA WATI	0.8	0.8	0.6	0.8	1
18	1537	MILO SANDIKA	0.8	0.6	0.8	1	1
19	1538	MONICA APRILLIA DAMANIK	0.8	0.8	0.8	0.8	1
20	1539	MUHAMMAD BAYU SHOPAN	0.8	0.6	0.8	0.6	1
21	1542	NIA BAZHLINA	0.8	0.8	0.6	0.8	1
22	1544	PANCA IRAWAN	0.6	0.6	0.8	0.6	1
23	1545	PUTRA PANDU KHAIRUN NAZRI	0.6	0.4	0.6	0.8	1
24	1546	PUTRI MAYANG SARI	0.6	0.8	0.8	0.8	1
25	1547	RAHMAD	0.6	0.6	0.8	0.8	1
26	1548	RAMA DONA	0.6	0.8	0.8	0.8	1

NO	NIS	NAMA	VARIABEL INPUT				TARGET
			A	B	C	D	
27	1549	RIDUAN SANTOSO	0.6	0.4	0.6	1	1
28	1550	RIKA WARDANI	0.8	0.6	0.8	1	1
29	1551	RIZKY SYAHPUTRA	0.6	0.6	0.6	0.8	1
30	1552	RONI SANJAYA HASUGIAN	0.6	0.6	0.4	0.6	1
31	1553	SAHRUL RAMADAN	0.8	0.6	0.6	0.8	1
32	1554	SARTIKA ANDRIANI	0.6	0.6	1	1	1
33	1555	SHENDI IKHWANDA	0.8	0.8	0.6	1	1
34	1556	SISKA	0.8	0.8	0.6	1	1
35	1557	SUGIARTO	0.6	0.6	0.6	1	1
36	1558	SULIS HARIANI	0.4	0.4	0.4	0.6	1
37	1559	SURYA RAMADHAN	0.8	0.6	0.6	0.8	1
38	1560	TRI WULAN DARI	0.8	0.8	0.8	0.6	1
39	1561	ULUL AMRI	0.8	0.8	1	0.8	1
40	1563	YUNITA RAMAYANI	1	0.4	1	1	1

Algoritma pelatihan backpropagation dengan menggunakan 4 input layer, 2 layer tersembunyi, 1 output layer dengan fungsi aktivasi sigmoid biner adalah sebagai berikut :

1. Tahap inisialisasi :

Tuliskan nilai input yang diberikan

$X_1 = 0.8$  ,  $X_2 = 0.6$  ,  $X_3 = 0.6$  ,  $X_4 = 0.8$

Target = 1, Learning rate( $\alpha$ )=0.1

Berikan nilai bobot (V) dari input ke lapisan tersembunyi dengan nilai acak.

**Tabel 2.3** Nilai Bobot dari Input ke Hidden Layer

(Solikhun d. , 2017)

	K <sub>1</sub>	K <sub>2</sub>
X <sub>1</sub>	0,2	-0,3
X <sub>2</sub>	0,4	0,1
X <sub>3</sub>	0,3	0,5
X <sub>4</sub>	0,5	-0,4

Berikan nilai bobot (W) dari lapisan tersembunyi ke output dengan nilai acak.

**Tabel 2.4** Nilai Bobot dari Hidden Layer ke Output

(Solikhun d. , 2017)

	L
K <sub>1</sub>	0,3
K <sub>2</sub>	-0,2

## 2. Tahap Aktivasi

Hitung keluaran tiap node (node tersembunyi dan node output)

$$\begin{aligned} K_1 &= X_1 \cdot V_{11} + X_2 \cdot V_{21} + X_3 \cdot V_{31} + X_4 \cdot V_{41} \\ &= 0,8 \cdot 0,2 + 0,6 \cdot 0,4 + 0,6 \cdot 0,3 + 0,8 \cdot 0,5 \\ &= 0,9800 \\ &= \text{Sigmoid} [0,9800] = 0,2729 \end{aligned}$$

$$\begin{aligned} K_2 &= X_1 \cdot V_{12} + X_2 \cdot V_{22} + X_3 \cdot V_{32} + X_4 \cdot V_{42} \\ &= 0,8 \cdot -0,3 + 0,6 \cdot 0,1 + 0,6 \cdot -0,5 + 0,8 \cdot -0,4 \\ &= -0,8000 \\ &= \text{Sigmoid} [-0,8000] = 0,6900 \end{aligned}$$

$$\begin{aligned} L &= K_1 \cdot W_{11} + K_2 \cdot W_{21} \\ &= 0,2729 \cdot -0,3 + 0,6900 \cdot -0,2 \\ &= -0,2199 \\ &= \text{Sigmoid} [-0,2199] = 0,5547 \end{aligned}$$

Hitung nilai error output dan hidden layer

Rumus mencari error output layer :  $= (1 -$

Rumus mencari error hidden layer :  $= (1 -$

$$\begin{aligned} \text{Err L} &= L \cdot (\alpha - L) \cdot (T - L) \\ &= 0,5547 \cdot (0,1 - 0,5547) \cdot (1 - 0,5547) \\ &= -0,1123 \end{aligned}$$

$$\begin{aligned} \text{Err K}_1 &= K_1 \cdot (\alpha - K_1) \cdot (\text{Err L} - W_{11}) \\ &= 0,2729 \cdot (0,1 - 0,2729) \cdot (-0,1123 - (-0,3)) \\ &= -0,3034 \end{aligned}$$

$$\begin{aligned} \text{Err K}_2 &= K_2 \cdot (\alpha - K_2) \cdot (\text{Err L} - W_{21}) \\ &= 0,6900 \cdot (0,1 - 0,6900) \cdot (-0,1123 - (-0,2)) \\ &= -0,0357 \end{aligned}$$

Modifikasi/hitung bobot baru

Rumus memodifikasi bobot baru :

$$W_{ij} = W_{ij} + I \cdot \text{Err}_j \cdot O_i$$

$$\begin{aligned} \Delta W_{11} &= W_{11} + \alpha \cdot \text{Err L} \cdot K_1 \\ &= -0,3 + 0,1 \cdot -0,1123 \cdot 0,2729 \end{aligned}$$

$$= -0,3031$$

$$\Delta W21 = W21 + \alpha \cdot \text{Err L. K2}$$

$$= -0,2 + 0,1 \cdot -0,1123 \cdot 0,6900$$

$$= -0,2078$$

$$\Delta V11 = V11 + \alpha \cdot \text{Err K1. j1}$$

$$= 0,2 + 0,1 \cdot -0,3034 \cdot 0,8$$

$$= 0,1993$$

$$\Delta V12 = V12 + \alpha \cdot \text{Err K2. j1}$$

$$= -0,3 + 0,1 \cdot -0,0357 \cdot 0,8$$

$$= -0,3029$$

$$\Delta V21 = V21 + \alpha \cdot \text{Err K1. J2}$$

$$= 0,4 + 0,1 \cdot -0,3034 \cdot 0,6$$

$$= 0,3665$$

$$\Delta V22 = V22 + \alpha \cdot \text{Err K2. J2}$$

$$= 0,1 + 0,1 \cdot -0,0357 \cdot 0,6$$

$$= 0,0979$$

$$\Delta V31 = V31 + \alpha \cdot \text{Err K1. J3}$$

$$= 0,3 + 0,1 \cdot -0,3034 \cdot 0,6$$

$$= 0,2995 \quad \Delta V32 = V32 + \alpha \cdot \text{Err K2. J3}$$

$$= -0,5 + 0,1 \cdot -0,0357 \cdot 0,6$$

$$= -0,5021$$

$$\Delta V41 = V41 + \alpha \cdot \text{Err K1. J4}$$

$$= 0,5 + 0,1 \cdot -0,3034 \cdot 0,8$$

$$= 0,4993$$

$$\Delta V42 = V42 + \alpha \cdot \text{Err K2. J4}$$

$$= -0,4 + 0,1 \cdot -0,0357 \cdot 0,8$$

$$= -0,4029$$

Tahap selanjutnya adalah membandingkan nilai error minimum dari hasil yang didapat. Dengan model arsitektur 4-5-1, data akan diprediksi untuk melihat seberapa akurat model ini dapat mengenali data.

**Tabel 2.5** Hasil Prediksi dengan Model 4-2-1  
(Solikhun d. , 2017)

Prediksi Model 4-2-1		Prediksi		Ket
No	NIS	Database	JST	Hasil
1	1542	Cukup Paham	0.0116	Benar
2	1544	Cukup Paham	0.0232	Benar
3	1545	Cukup Paham	0.0502	Benar
4	1546	Cukup Paham	0.0122	Benar
5	1547	Cukup Paham	0.0136	Benar
6	1548	Cukup Paham	0.0122	Benar
7	1549	Cukup Paham	0.0442	Benar
8	1550	Cukup Paham	0.0111	Benar
9	1551	Cukup Paham	0.0255	Benar
10	1552	Tidak Paham	0.1308	Salah
11	1553	Cukup Paham	0.0126	Benar
12	1554	Cukup Paham	0.0112	Benar
13	1555	Cukup Paham	0.0114	Benar
14	1556	Cukup Paham	0.0114	Benar
15	1557	Cukup Paham	0.0232	Benar
16	1558	Tidak Paham	0.4181	Salah
17	1559	Cukup Paham	0.0126	Benar
18	1560	Cukup Paham	0.0134	Benar
19	1561	Cukup Paham	0.0112	Benar
20	1563	Cukup Paham	0.0108	Benar

Dari hasil prediksi yang diperoleh didapat hasil bahwa JST dapat melakukan prediksi diatas 90 % tingkat akurasi kebenarannya (Solikhun d. , 2017).

## 2.8 Matlab

Matlab singkata dari *Matrix Laboratory* yang dikembangkan oleh *MathWorks Inc* dari bahasa C. Software ini didukung oleh sistem operas *Unix*, *Macintosh* dan *Windows*. Matlab mengintegrasikan komputasi matematika, visualisasi dan bahasa pemrograman untuk memberikan lingkungan fleksibel bagi komputasi teknis. Matlab membuat pengguna mudah dalam melakukan eksplorasi data, menciptakan algoritma, menciptakan beberapa perangkat grafik (GUI) dan lain sebagainya (Siahaan, 2020).

Matlab dikenal karena perhitungan vektor dan matriks dengan kecepatan tinggi. Matlab menawarkan solusi terhadap permasalahan secara matematik dan visual yang mana hal ini diperuntukan terutama untuk:

1. Komputasi numerik dan pengembangan algoritma.
2. komputasi simbolik (dengan fungsi-fungsi pustaka *Symbolic Math*).
3. Pemodelan, simulasi dan penciptaan prototipe.
4. Analisis data dan pemrosesan sinyal/ citra/video.

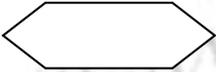
## 5. Visualisasi saintifik dan grafik rekayasa (Siahaan, 2020).

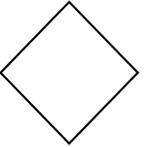
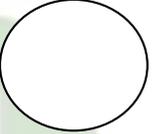
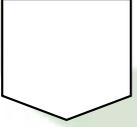
**2.9 Flowchart**

*Flowchart* adalah penyajian sistematis tentang proses dan logika yang dari penanganan informasi atau penggambaran grafis dari langkah-langkah dan pengaturan teknik suatu program. *Flowchart* membantu para ahli dan pengembang untuk memisahkan masalah menjadi fragmen yang lebih sederhana dan membantu dalam membedah opsi yang berbeda dalam pengoperasian (Kadang, 2021).

**Tabel 2.6** Simbol-simbol Flowchart

(Kadang, 2021)

Simbol	Nama	Fungsi
	Terminator	Permulaan/ akhir Program
	Garis Alir ( <i>Flow Line</i> )	Arah Aliran Program
	<i>Preparation</i>	Proses Inisialisasi/ Pemberian harga awal
	Proses	Proses Perhitungan

	<i>Input/Output Data</i>	Proses <i>input/output</i> data, parameter, informasi.
	<i>Predefined proses (sub program).</i>	Permulaan sub program/ proses menjalankan sub program
	<i>Decision</i>	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	<i>One Page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> yang berada pada satu halaman
	<i>Off Page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> yang berada pada halaman berbeda

## 2.10 Riset Terkait

Ada beberapa riset terlaik diantaranya:

**Tabel 2.7** Penelitian Terdahulu

No	PENELITI	JUDUL	KESIMPULAN
1.	Dwi Ari Suryaningrum, Dian Eka Ratnawati,	rediksi Pemilihan Jurusan Siswa Kelas 1 SMK Menggunakan Jaringan Syaraf Tiruan	Metode <i>backpropagation</i> dapat diterapkan untuk memprediksi Pemilihan Jurusan Siswa Kelas 1 SMK. Pada penelitian ini

	Budi Darma Setiawan	dengan Metode Backpropagation	mendapat nilai MSE nya berhenti pada epochs ke 9 dengan nilai 0.0001 (Furqan et al., 2021).
2.	Ayu Artika Fardhani, Desi Insani Natalia Simanjuntak, Anjar Wanto	Prediksi Harga Eceran Beras Di Pasar Tradisional Di 33 Kota Di Indonesia Menggunakan Algoritma Backpropagation	Penelitian memprediksi harga eceran beras dengan algoritma <i>backpropagation</i> dikarenakan banyaknya peminatan yang membuat tingginya harga ecer beras dengan menggunakan 5 arsitektur antara lain : 4-25-1, 4-35-1, 440-1, 4-42-1, 4-45-1 dengan learning rate 0,09. Dari ke 5 Arsitektur ini, arsitektur terbaiknya adalah 4-45-1 dengan tingkat akurasi sebesar 88%, epoch 12718 iterasi dan waktu 1 menit 14 detik yang terhitung memiliki tingkat akurasi yang baik (Fardhani et al., 2018).
3.	Ade Pujiyanto, Kusriani, Andi Sunyoto	Perancangan Sistem Pendukung Keputusan Untuk Prediksi Penerima Beasiswa Menggunakan Metode Neural Network Backpropagation	Backpropagation juga dapat membantu mengambil keputusan dalam menentukan penerima beasiswa dengan nilai akurasi sebesar 90% dan nilai error terkecil sebesar 0,000101 pada epoch ke 329 dengan jumlah 3000 data dengan pembagian data <i>training</i> 2.250 dan 750 data <i>testing</i> serta

			konfigurasi <i>learning rate</i> sebesar 0,2 dan momentum 0,2 (Pujiyanto et al., 2018).
4.	Hasdi Putra, Nabilah Ulfa Walmi	Penerapan Prediksi Produksi Padi Menggunakan Artificial Neural Network Algoritma Backpropagation	Penelitian ini dapat membantu memprediksi produksi padi guna untuk menunjang pembangunan nasional di sektor pertanian pada suatu negara atau wilayah dengan hasil pengujian sistem prediksi produksi padi yang terdiri dari 75 kali pengujian pada di 19 daerah di Sumatera Barat, diperoleh tingkat akurasi mencapai 88,14% atau dengan tingkat <i>error</i> yang relatif rendah yaitu 11,86% (Putra & Ulfa Walmi, 2020).
5.	Randi Maiyuriska	Penerapan Jaringan Syaraf Tiruan dengan Algoritma Backpropagation dalam Memprediksi Hasil Panen Gabah Padi	Hasil panen gabah padi disetiap daerah tiap tahunnya selalu berubah-ubah, termasuk di Kabupaten Pesisir Selatan. Penelitian ini bertujuan untuk membuat memprediksi dan untuk mengetahui hasil panen gabah padi di Kabupaten Pesisir Selatan agar dapat membantu petani dalam menanggulangi gagal panen jika terjadi kegagalan di kedepannya. Dari penelitian dihasilkan tingkat

			akurasi mencapai 92.9% atau tingkat <i>error</i> 7.1% dengan MSE = 0.00094783 (Maiyuriska, 2022).
6.	Dio Very Hutabarat, Solikhun, M. Fauzan, Agus Perdana Windarto, Fitri Rizki	Penerapan Algoritma Backpropagation dalam Memprediksi Hasil Panen Tanaman Sayuran	Penelitian ini membantu pemerintah dan masyarakat dalam meningkatkan penyediaan stok hasil panen untuk mencukupkan kebutuhan pangan bagi masyarakat Kabupaten Simalungun. Hasil dari penelitian ini diperoleh model arsitektur terbaik adalah model 2-1-1 dengan tingkat akurasi 75,0% dan epoch sebesar 1392 iterasi dalam waktu 00:07 detik (Hutabarat et al., 2021).
7.	Delima Sinaga, Solikhun, Iin Parlina	Jaringan Syaraf Tiruan untuk Memprediksi Penjualan Kelapa Sawit Menggunakan Algoritma Backpropagation	Penelitian ini membahas tentang prediksi penjualan kelapa sawit menggunakan Jaringan Syaraf Tiruan dengan metode <i>bacpropagation</i> dengan tingkat akurasi terbaik menggunakan arsitektur 12-21 yang akurasinya sebesar 92% dan tingkat akurasi terendah menggunakan arsitektur 12-6-1 dengan tingkat akurasi sebesar 58% (Sinaga et al., 2019).

8.	Willy R.A Situmorang, Miftahul Jannah	Implementasi Jaringan Saraf Tiruan Memprediksi Hasil Panen Padi Pada Desa Pagar Jati Dengan Metode Backpropagation.	Penelitian ini dilakukan untuk memprediksi jumlah hasil panen padi pada desa Pagar Jati. Penelitian ini menggunakan metode <i>backpropagation</i> yang di perlukan untuk meningkatkan target produktivitas padi (Situmorang & Jannah, 2021).
9.	Anjar Wanto, Agus Perdana Windarto	Analisis Prediksi Indeks Harga Konsumen Berdasarkan Kelompok Kesehatan Dengan Menggunakan Metode Backpropagation	Penelitian bermanfaat secara luas, baik bagi Pemerintah Daerah maupun pihak swasta sebagai salah satu bahan kajian dalam pengembangan bisnis, maupun bagi para akademisi sebagai bahan kajian/penelitian khususnya yang terkait dengan bidang ekonomi dan kebijakan publik. penelitian ini menggunakan metode jaringan saraf tiruan Backpropagation dengan menggunakan 8 model arsitektur, yakni : 12-5-1 yang nanti nya akan menghasilkan prediksi dengan tingkat akurasi sebesar 58%, 12-26-1 = 58%, 12-29-1 = 75%, 12-35-1 = 50% , 12-40-1 =

			<p>42%, 12-60-1 = 67%, 12-70-1 = 92% dan 12-75-1 = 50%. Sehingga diperoleh model arsitektur terbaik menggunakan model 12-70-1 yang menghasilkan tingkat akurasi sebesar 92%, MSE 0,3659742 dengan tingkat <i>error</i> yang digunakan 0,001 – 0,05. Dengan demikian, model ini cukup bagus untuk prediksi Indeks Harga Konsumen berdasarkan kelompok kesehatan (Wanto &amp; Windarto, 2017).</p>
10.	Armansyah	<p>Prototipe Jaringan Syaraf Tiruan Multilayer Perceptron Untuk Prediksi Mahasiswa Dropout</p>	<p>Penelitian ini bertujuan memprediksi mahasiswa yang berpotensi putus sekolah dengan pendekatan jaringan syaraf tiruan. Dengan mengamati 13 variabel yang mempengaruhi, dan 1 variabel keluaran yang akan dilatih dengan model <i>multi layer perceptron</i> yang diharapkan dapat menghasilkan kinerja prediksi dengan nilai 0, untuk mahasiswa berpotensi dropout, dan 1 untuk mahasiswa yang tetap melanjutkan hingga akhir, dengan menunjukkan hasil</p>

			komputasi tingkat akurasi setidaknya 96,90% dan tingkat <i>error</i> yang rendah (Armansyah, 2021).
--	--	--	---

