

BAB II TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Penelitian terkait prediksi menggunakan algoritme *artificial neural network backpropagation* sudah banyak dilakukan dengan berbagai parameter dan model arsitektur. Dalam penulisan penelitian ini terdapat beberapa penelitian terdahulu tentang *backpropagation*. Berikut adalah penelitian yang terkait dengan *backpropagation*:

Penelitian yang berjudul "*Penerapan Algoritma Backpropagation Dalam Memprediksi Jumlah Pengguna Kereta Api Di Pulau Sumatera*". Penelitian ini bertujuan untuk meramalkan jumlah pengguna kereta api di Pulau Sumatera, menggunakan informasi yang berasal dari laporan tahunan PT. Kereta Api Indonesia (PT. KAI) pada tahun 2015. Setelah menerapkan lima variasi arsitektur model yang berbeda, berhasil diidentifikasi bahwa model paling optimal memiliki *Mean Squared Error* (MSE) sebesar 0,0215923, dengan tingkat akurasi mencapai 92%. [9].

Penelitian lain yang berjudul "*Implementasi Metode Backpropagation untuk Prediksi Harga Batu Bara*". Dalam penelitian ini, dilakukan estimasi nilai batu bara dengan memanfaatkan pendekatan *backpropagation*. Data yang dimanfaatkan meliputi rentang waktu Januari 2009 hingga September 2017, fokusnya adalah pada merek Gunung Bayan I. Setelah serangkaian uji coba, ditemukan bahwa angka *Mean Squared Error* (MSE) terkecil adalah 0,00205284, sementara angka MSE tertinggi mencapai 0,01778293. [10].

Penelitian dengan judul "*Peramalan Jumlah Penjualan Menggunakan Jaringan Syaraf Tiruan Backpropagation Pada Perusahaan Air Minum Dalam Kemasan*". Penelitian ini melakukan prediksi jumlah penjualan air minum dalam kemasan menggunakan ANN *backpropagation* pada periode 2019 mendapatkan nilai MSE terbaik sebesar 0,00043743 dan nilai MAPE terbaik sebesar 6,88% [11].

Penelitian lain dengan judul "*Prediksi IHSG dengan Backpropagation Neural Network*" menerapkan ANN *Backpropagation* dalam memprediksi Indeks Harga Saham Gabungan (IHSG). Penelitian ini menghasilkan proyeksi yang memuaskan, dengan angka *Mean Squared Error* (MSE) mencapai 320,49865083640924 [12].

Penelitian selanjutnya dengan judul "*Prediksi Distribusi Air Perusahaan Daerah Air Minum (PDAM) Tirta Dharma Kota Pasuruan Menggunakan Metode Jaringan Syaraf Tiruan Backpropagation*". Penelitian ini menggunakan metode jaringan saraf tiruan *backpropagation* untuk meramalkan pola distribusi air PDAM Tirta Dharma di wilayah Kota Pasuruan. Dalam uji coba, metode ini menghasilkan tingkat akurasi tertinggi, mencapai 100% [13].

Penelitian lainnya yang berjudul "*Jaringan Syaraf Tiruan Backpropagation dalam Memprediksi Jumlah Tilang di Kejaksaan Negeri Simalungun*". Penelitian ini bertujuan untuk meramalkan jumlah tilang di Kejaksaan Negeri Simalungun dengan data yang digunakan mulai tahun 2016 sampai 2018. Dengan menggunakan Jaringan Syaraf Tiruan *Backpropagation* diperoleh nilai MSE terbaik 0,03458058 dan mendapatkan tingkat akurasi sebesar 75% [14].

Selanjutnya penelitian dengan judul "*Algoritma Backpropagation Neural Network dalam Memprediksi Harga Komoditi Tanaman Karet*". Penelitian ini berfokus pada meramalkan harga komoditas tanaman karet di Provinsi Kalimantan Timur dalam jangka waktu 2014 hingga 2018. Metode yang digunakan adalah algoritma *artificial neural network backpropagation*. Hasil paling optimal diperoleh melalui pengaturan perbandingan data latihan dan pengujian sebesar 4:1, dengan model arsitektur 5-10-10-10-1, penerapan fungsi aktivasi *trainln*, penggunaan *learning rate* sebesar 0,5, toleransi *error* sebesar 0,01, serta pelatihan dengan jumlah *epoch* 1.000. Akhirnya, angka *Mean Squared Error* (MSE) tercatat sebesar 0.00015464 [15].

Selain itu, terdapat penelitian dengan judul "*Prediksi Nilai Cryptocurrency Bitcoin menggunakan Algoritme Extreme Learning Machine (ELM)*". Dalam

penelitian ini, metode *Extreme Learning Machine* (ELM) diterapkan untuk meramalkan nilai *Bitcoin*. Hasilnya menunjukkan bahwa terdapat nilai *Mean Absolute Percentage Error* (MAPE) terkecil dengan tingkat sebesar 2.657%. Hal ini dicapai melalui penggunaan rentang bobot fungsi aktivasi sigmoid biner optimal yaitu [-1.8, 1.8], pengalokasian 80% data sebagai data latihan, pemanfaatan empat unit tersembunyi, dan penggunaan dua fitur sebagai variabel masukan [5].

Tabel 2. 1 Penelitian Sebelumnya

No.	Peneliti	Judul	Studi Kasus	Metode	Hasil
1	Vivi Auladina, Jaya Tata Hardinata, dan M. Fauzan (2021)	Penerapan Algoritma <i>Backpropagation</i> Dalam Memprediksi Jumlah Pengguna Kereta Api Di Pulau Sumatera	Penelitian ini dilaksanakan dengan tujuan meramalkan jumlah penumpang kereta api di Pulau Sumatera. Sumber data yang digunakan diperoleh dari laporan tahunan PT. Kereta Api Indonesia (PT. KAI) pada tahun 2015.	<i>Artificial Neural Network Backpropagation</i>	Proses pelatihan melibatkan penerapan lima variasi arsitektur model. Dalam hal ini, ditemukan bahwa arsitektur yang paling optimal adalah 3-3-1, dengan tiga unit pada lapisan masukan (<i>input layer</i>), tiga unit pada lapisan tersembunyi (<i>hidden layer</i>), dan satu unit pada lapisan keluaran (<i>output layer</i>). Arsitektur ini berhasil mencapai akurasi sekitar 92% dan <i>Mean Squared Error</i> (MSE) sebesar 0,0215923.
2	Sony Irwanda, Jaya Tata Hardinata, dan Irfan Sudahri Damanik (2019)	Jaringan Syaraf Tiruan <i>Backpropagation</i> dalam Memprediksi Jumlah Tilang di Kejaksaan Negeri Simalungun	Penelitian ini bertujuan untuk meramalkan jumlah tilang yang dilaporkan di Kejaksaan Negeri Simalungun, dengan menggunakan data yang mencakup periode tahun 2016 hingga 2018.	<i>Artificial Neural Network Backpropagation</i>	Dari hasil pengujian, arsitektur model yang optimal dalam penelitian ini adalah 12-6-1, dengan penerapan <i>learning rate</i> sebesar 0.05. Didapatkan nilai <i>Mean Squared Error</i> (MSE) sebesar 0.03458058, dan penelitian ini memasukkan batas toleransi minimum sebesar 0.001, sehingga tingkat akurasi yang tercapai mencapai 75%.
3	Nur Fitrianiingsih Hasan, Kusrini, dan Hanif Al Fatta (2019)	Peramalan Jumlah Penjualan Menggunakan Jaringan Syaraf Tiruan <i>Backpropagation</i> Pada Perusahaan Air Minum Dalam Kemasan	Penelitian ini bertujuan untuk melakukan estimasi penjualan Air Minum Dalam Kemasan (AMDK) di perusahaan ABC di Kota Jayapura. Data yang dianalisis merupakan catatan penjualan produk AMDK berukuran 600ml dari tahun 2017 hingga 2018. Dalam kerangka penelitian ini, data penjualan tahun 2017 digunakan sebagai data latihan, sementara data tahun 2018 digunakan sebagai data	<i>Artificial Neural Network Backpropagation</i>	Hasil dari penelitian ini diperoleh arsitektur model Jaringan Saraf Tiruan (JST) dengan 12 lapisan masukan (<i>input layer</i>), 10 lapisan tersembunyi (<i>hidden layer</i>), dan satu lapisan keluaran (<i>output layer</i>). Selain itu, hasil pengujian menunjukkan nilai <i>Mean Squared Error</i> (MSE) sebesar 0,00043743 dan nilai <i>Mean Absolute Percentage Error</i> (MAPE) sebesar 6,88%.

No.	Peneliti	Judul	Studi Kasus	Metode	Hasil
			pengujian. Untuk melatih dan menguji jaringan, perangkat lunak MATLAB R2018a digunakan.		
4	Andy Santoso dan Seng Hansun (2019)	Prediksi IHSG dengan <i>Backpropagation Neural Network</i>	Penelitian ini dilakukan untuk memprediksi Indeks Harga Saham Gabungan (IHSG).	<i>Artificial Neural Network Backpropagation</i>	Dari hasil penelitian ini terungkap bahwa pendekatan <i>backpropagation</i> mampu melakukan perkiraan dengan hasil yang memadai, hal ini terlihat dari nilai <i>Mean Squared Error</i> (MSE) pada pengujian sebesar 320,49865083640924. Proses ini dilakukan dengan penerapan <i>learning rate</i> sebesar 0,3 dan dengan melakukan 3000 kali <i>epoch</i> .
5	Dwi Agustina, Moh. Hafiyusholeh, Aris Fanani, dan Dono Prasetijo (2023)	Prediksi Distribusi Air Perusahaan Daerah Air Minum (PDAM) Tirta Dharma Kota Pasuruan Menggunakan Metode Jaringan Syaraf Tiruan <i>Backpropagation</i>	Penelitian ini melakukan prediksi terhadap jumlah dari distribusi air per bulan. Data yang digunakan bersumber dari PDAM Tirta Dharma kota Pasuruan dari tahun 2019 sampai dengan tahun 2021.	<i>Artificial Neural Network Backpropagation</i>	Dari hasil pengujian, terlihat bahwa penggunaan ANN <i>backpropagation</i> mampu memberikan tingkat akurasi yang signifikan dalam meramalkan pola distribusi air dari PDAM Tirta Dharma di kota Pasuruan. Hasil yang diperoleh menunjukkan akurasi tertinggi sebesar 100%, yang diperoleh melalui model arsitektur 4-2-1. Parameter-parameter yang optimal adalah <i>learning rate</i> sebesar 0.1, target <i>error</i> 0.001, dan batas maksimum <i>epoch</i> 1000.
6	Miracle Fachrunnisa Almas, Budi Darma Setiawan, dan Sutrisno (2018)	Implementasi Metode <i>Backpropagation</i> untuk Prediksi Harga Batu Bara	Dalam penelitian ini, dilakukan estimasi nilai harga batu bara dengan memprediksi data harga yang tercatat dari bulan Januari 2009 hingga September 2017, dengan fokus pada merek Gunung Bayan I.	<i>Artificial Neural Network Backpropagation</i>	Hasil dari penelitian ini mengindikasikan bahwa struktur terbaik yang diperoleh adalah 10-10-1, mencakup 10 unit pada lapisan masukan (<i>input</i>), 10 unit pada lapisan tersembunyi (<i>hidden</i>), dan satu unit

No.	Peneliti	Judul	Studi Kasus	Metode	Hasil
					pada lapisan keluaran (<i>output</i>). Penggunaan <i>learning rate</i> teroptimal adalah 0,1, dan dilakukan pengujian pada 500 iterasi terbaik. Pada pengujian akhir berdasarkan jumlah iterasi, sistem berhasil meraih nilai <i>mean error</i> terendah sebesar 0,00205284, dengan nilai <i>mean error</i> maksimum mencapai 0,01778293.
7	Julius Rinaldi Simanungkalit, Havaluddin Havaluddin, Herman Santoso Pakpahan, Novianti Puspitasari, dan Masna Wati (2020)	Algoritma <i>Backpropagation Neural Network</i> dalam Memprediksi Harga Komoditi Tanaman Karet	Dalam penelitian ini, dilakukan estimasi nilai harga komoditas tanaman karet di Provinsi Kalimantan Timur dengan memanfaatkan data dari tahun 2014 hingga 2018. Penggunaan parameter-parameter yang beragam juga turut diterapkan dalam proses ini.	<i>Artificial Neural Network Backpropagation</i>	Hasil optimal diperoleh melalui pengujian dengan perbandingan data latihan dan pengujian sebesar 4:1. Struktur model yang paling baik adalah 5-10-10-10-1, dengan masing-masing lapisan terdiri dari lima unit pada <i>input</i> , 10 unit pada tiga lapisan tersembunyi, dan satu unit pada <i>output</i> . Fungsi aktivasi yang diterapkan adalah <i>trainln</i> , <i>learning rate</i> pada 0.5, serta toleransi <i>error</i> sebesar 0.01, dengan proses pelatihan berlangsung selama 1000 kali <i>epoch</i> . Akhirnya, hasil nilai <i>Mean Squared Error</i> (MSE) yang diperoleh adalah 0.00015464.
8	Rahmat Faizal, Budi Darma Setiawan, dan Imam Cholissodin (2019)	Prediksi Nilai <i>Cryptocurrency Bitcoin</i> menggunakan Algoritme <i>Extreme Learning Machine</i> (ELM)	Penelitian ini dilakukan untuk memprediksi nilai <i>cryptocurrency bitcoin</i> .	<i>Extreme Learning Machine</i> (ELM)	Hasil yang tercapai dalam penelitian ini adalah nilai <i>Mean Absolute Percentage Error</i> (MAPE) terendah dengan rata-rata 2.657%. Hasil ini diperoleh dengan menerapkan rentang bobot fungsi aktivasi sigmoid biner optimal yaitu [-1.8, 1.8], memanfaatkan 80% data sebagai data latihan dan 20% sebagai data

No.	Peneliti	Judul	Studi Kasus	Metode	Hasil
					pengujian. Pengaturan ini juga melibatkan empat unit tersembunyi dan dua fitur sebagai masukan dalam model.

Berdasarkan Tabel 2.1 dapat diambil kesimpulan bahwa *cryptocurrency Bitcoin* mengalami fluktuasi setiap harinya. Sehingga diperlukan prediksi terhadap harga *Bitcoin* untuk mengurangi risiko dalam penggunaan *Bitcoin* baik untuk investasi, transaksi, maupun *trading*. Metode *artificial neural network backpropagation* mampu diterapkan untuk melakukan prediksi dengan kesalahan yang terbukti rendah atau mendekati nilai aktual. Untuk mendapatkan model optimal dengan nilai *Mean Squared Error* (MSE) yang minimal, proses perancangan model akan mengusung nilai target *error* 0.001 dan *learning rate* 0.001. Sumber utama yang dijadikan panduan dalam penelitian ini adalah penelitian nomor tiga dan enam pada tabel, mengingat metodenya serupa. Perbedaan utamanya terletak pada studi kasus yang diujikan. Referensi lain juga dimanfaatkan sebagai dukungan dalam penelitian ini.

2.2. Landasan Teori

Dalam penelitian ini, beberapa dasar teori diadopsi dari berbagai referensi sebagai pendukung. Beberapa konsep teoritis yang menjadi dasar bagi penelitian ini antara lain:

2.2.1. Cryptocurrency

Cryptocurrency adalah aset digital yang dikelola oleh jaringan, yang pada umumnya menggunakan buku besar yang didistribusikan teknologi dan enkripsi untuk mencatat, mengontrol, dan memverifikasi transaksi dan pembuatan unit mata uang baru, yaitu koin atau token. Seperti mata uang fisik, misalnya, Dolar AS (USD), koin digital juga digunakan sebagai alat tukar. Namun *cryptocurrency* tidak didukung oleh bank sentral atau lembaga pemerintah [16].

Cryptocurrency adalah bentuk mata uang digital yang dibangun dengan basis teknologi *blockchain*, dimana teknologi tersebut tidak membutuhkan keterlibatan pihak ketiga sebagai perantara. Oleh karena itu, setiap transaksi yang dilakukan menjadi lebih transparan. *Cryptocurrency* menerapkan teknologi *blockchain* untuk menghubungkan setiap data yang ada dan setiap data berada di tangan orang di lingkungan pengguna sistem *cryptocurrency* [17].

2.2.2. Bitcoin

Bitcoin adalah sebuah bentuk mata uang digital yang mengaplikasikan prinsip kriptografi dan beroperasi melalui mekanisme *blockchain* serta *proof-of-work* dalam sistem terdistribusi dan terdesentralisasi. Karena itu, *Bitcoin* sering diidentifikasi sebagai *cryptocurrency*. Diluncurkan awal mula pada pertengahan tahun 2008 dan mulai beroperasi pada awal tahun 2009, *Bitcoin* merupakan *cryptocurrency* pertama di dunia yang diusulkan oleh Satoshi Nakamoto [18]. *Bitcoin* merupakan sebuah mata uang digital yang menerapkan teknologi *peer-to-peer*, yang berarti tidak membutuhkan otoritas pusat atau bank sentral seperti halnya mata uang konvensional [19]. *Bitcoin* tidak dikelola oleh bank sentral, tetapi dikelola langsung oleh pengguna *Bitcoin* yang identitasnya tidak ditampilkan. Dalam situasi ini, menjadikan *Bitcoin* dipandang masyarakat untuk menyimpan

asetnya karena dikelola dengan cara yang terdesentralisasi, bersifat rahasia dan prosesnya sederhana [5].

2.2.3. Prediksi

Prediksi merupakan usaha menerapkan pendekatan ilmiah untuk memproyeksikan situasi yang mungkin terjadi di masa yang akan datang, berdasarkan pemanfaatan informasi relevan dari masa sebelumnya (sejarah) [20].

Berdasarkan penelitian [21]. Metode yang diterapkan dalam melakukan prediksi bisa dikategorikan ke dalam dua jenis, yaitu prediksi kualitatif dan prediksi kuantitatif.

1. Prediksi kualitatif

Penggunaan prediksi kualitatif terjadi ketika tidak ada riwayat data yang tersedia, dan didasarkan pada informasi kualitatif untuk meramalkan kejadian di masa depan. Namun, metode ini memiliki tingkat akurasi yang sangat subjektif.

2. Prediksi kuantitatif

Prediksi kuantitatif terbagi menjadi dua metode, yaitu prediksi *casual* dan *time series*. Prediksi *casual* melibatkan faktor-faktor yang terkait dengan variabel yang ingin diprediksi, seperti analisis regresi. Sementara itu, prediksi *time series* merupakan pendekatan kuantitatif yang menganalisis data historis yang telah terakumulasi secara beraturan. Teknik-teknik yang relevan digunakan dalam proses ini, dan hasilnya digunakan untuk meramalkan nilai di periode mendatang.

Penelitian yang dilakukan oleh peneliti melibatkan penggunaan data numerik dari masa sebelumnya, dan tingkat akurasi prediksi ditentukan melalui perbandingan antara hasil prediksi dengan realitas yang sebenarnya. Oleh karena itu, pendekatan yang diterapkan dalam penelitian ini adalah prediksi kuantitatif.

2.2.4. Machine Learning

Machine Learning, yang merupakan cabang dari Kecerdasan Buatan (*Artificial Intelligence/AI*), merujuk pada domain ilmu yang melibatkan

perancangan serta pengembangan algoritma yang membolehkan komputer untuk mengembangkan tindakan berdasarkan pada data empiris, termasuk data yang dihasilkan dari sensor dan basis data [22]. *Machine Learning* merupakan strategi yang umumnya digunakan untuk menggantikan atau meniru kemampuan manusia dalam mengatasi masalah atau mencapai tingkat otomatisasi. Prinsip *Machine Learning* mencoba memodelkan proses belajar dan generalisasi yang ada pada manusia. Di dalamnya, terdapat tahap pelatihan, pembelajaran, atau *training*. Karenanya, data pelatihan diperlukan agar mesin dapat mengambil pembelajaran dari data tersebut [23].

Machine Learning memiliki tiga jenis metode pembelajaran, yaitu sebagai berikut

1. *Supervised Learning*

Supervised Learning merupakan metode pembelajaran terarah atau terawasi. Sehingga diperlukan adanya guru yang mengajar atau mengarahkan dan siswa yang diajar, di mana manusia berperan sebagai guru, dan mesin berperan sebagai siswa [24]. *Supervised Learning* memiliki mekanisme pembelajaran yang menciptakan hubungan mendasar antara data yang diamati (data *input*) dan variabel target (variabel dependen atau label) untuk diprediksi [25]. Teknik ini dimulai jika permasalahan mempunyai variabel *input* (x) dan variabel *output* (y) dan pola data dipelajari oleh *artificial neural network* menggunakan sebuah algoritme. Tujuan dari *supervised learning* adalah agar sistem dapat memprediksi *output* ketika ada nilai *input* baru [26].

2. *Unsupervised Learning*

Unsupervised Learning adalah pendekatan pembelajaran yang tidak mengandalkan pengawasan, sehingga tidak melibatkan peran guru manusia dalam mengajar [24]. Pendekatan ini dirancang untuk mengidentifikasi pola yang tersembunyi dalam sekumpulan data yang tidak memiliki label atau informasi *output* yang tidak diketahui [25]. Tujuan *unsupervised learning* adalah mengembangkan model untuk menggambarkan struktur atau distribusi yang mendasari data, serta untuk mendalami pemahaman tentang data tersebut [26].

3. *Reinforcement Learning*

Reinforcement Learning, atau yang juga dikenal sebagai semi *supervised learning*, memiliki kesamaan dengan *supervised learning* dalam banyak hal, tetapi perbedaannya terletak pada bagaimana data diberi label. Dalam *supervised learning*, seorang guru manusia membuat pasangan *input-output*, sedangkan dalam semi *supervised learning*, tidak selalu diperlukan pembuatan pasangan *input-output* secara eksplisit oleh manusia. *Input-output* pada semi *supervised learning* dapat diperoleh secara otomatis [24].

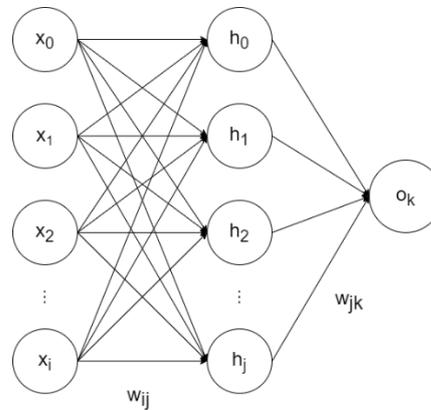
2.2.5. Deep Learning

Deep Learning adalah bagian dari *machine learning* yang merupakan pengembangan dari *neural network multiple layer*, yang merujuk pada algoritma yang terinspirasi dari struktur otak manusia. Struktur yang dimaksud dikenal sebagai *Artificial Neural Networks* atau disingkat ANN. Pada dasarnya, ia merupakan jaringan saraf yang memiliki tiga atau lebih lapisan ANN [22].

2.2.6. Artificial Neural Network

Artificial Neural Network (ANN) atau biasa disebut Jaringan Saraf Tiruan (JST) adalah sistem pemrosesan informasi yang terinspirasi dari sistem saraf biologis manusia, yang mirip dengan pemrosesan informasi di otak manusia [27]. Istilah "tiruan" digunakan karena jaringan ini mampu meniru perilaku jaringan saraf biologis manusia dengan mengadopsi konsep tersebut dalam program komputer yang dapat melakukan berbagai proses komputasi selama tahap pembelajaran [28]. *Artificial neural network* dapat memecahkan masalah dengan belajar dari contoh pelatihan yang disediakan. Proses pembelajaran menampilkan semua *input* dan *output* secara bersamaan [10]. Jaringan saraf tiruan terdiri dari sejumlah unit yang saling terhubung. Neuron-neuron ini memiliki untuk mengirimkan informasi yang diterimanya ke neuron lain dalam jaringan [12].

Berdasarkan penelitian [26], *Artificial Neural Network* terdiri dari tiga lapisan yaitu *input layer* (lapisan masukan), *hidden layer* (lapisan tersembunyi), *output layer* (lapisan keluaran) yang ditunjukkan pada Gambar 2.1.



Gambar 2. 1 Topologi *Artificial Neural Network*

Gambar 2.1 Menunjukkan topologi dari *artificial neural network* yang terdiri dari tiga lapisan sebagai berikut:

1. *Input Layer* (x) juga dikenal sebagai lapisan masukan, bertugas untuk mengakomodasi data dari lingkungan eksternal (dunia luar) ke dalam jaringan. Sekumpulan variabel *input* ini disebut sebagai lapisan *input*.
2. *Hidden Layer* (h) yang dikenal lapisan tersembunyi tidak memiliki hubungan dengan dunia luar. Lapisan ini berfungsi untuk melakukan perhitungan dan mentransfer informasi dari lapisan *input* ke lapisan *output*.
3. *Output Layer* (o) yang juga dikenal sebagai lapisan keluaran, merujuk pada kumpulan *output nodes* yang berperan dalam komputasi dan mentransfer informasi dari jaringan ke luarannya.

2.2.7. *Backpropagation*

Rumelhart, Hinton, dan Williams memperkenalkan *artificial neural network backpropagation* pada tahun 1986, yang kemudian dikembangkan lebih lanjut pada tahun 1988 oleh Rumelhart dan Mc Clelland [29]. *Backpropagation* adalah salah satu jenis teknik pembelajaran dalam *artificial neural network* yang termasuk dalam kategori *supervised learning* [30]. Tidak ada komputasi di lapisan *input*, tetapi di lapisan *input* mengirimkan sinyal *input* ke lapisan tersembunyi. Pada lapisan tersembunyi dan lapisan keluaran terjadi proses komputasi pada bobot dan

bias serta dihitung besarnya *output* dari lapisan tersembunyi dan lapisan keluaran berdasarkan fungsi aktivasi tertentu [15].

Fungsi aktivasi yang dapat memenuhi karakteristik jaringan *backpropagation* adalah sebagai berikut [31]:

1. Fungsi Sigmoid Biner

Fungsi sigmoid biner menghasilkan *output* dalam rentang antara nol hingga satu. Inilah mengapa fungsi ini umumnya diterapkan dalam *artificial neural network* yang membutuhkan keluaran yang berkisar antara nol dan satu. Perhitungan fungsi sigmoid biner menggunakan persamaan (2.1), dan turunannya dapat dihitung menggunakan persamaan (2.2).

$$y = f(x) = \frac{1}{1 + e^{-x}} \quad (2.1)$$

$$\text{Dengan: } f'(x) = f(x)[1 - f(x)] \quad (2.2)$$

Keterangan:

$f(x)$: fungsi aktivasi sigmoid biner

e : eksponensial

x : data ke- x

$f'(x)$: turunan (derivatif) dari fungsi sigmoid biner

2. Fungsi Sigmoid Bipolar

Fungsi sigmoid bipolar mirip dengan fungsi sigmoid biner, perbedaannya terletak pada kisaran *output* yang dihasilkan oleh fungsi sigmoid bipolar, yaitu dari negatif satu hingga satu. Fungsi sigmoid bipolar dihitung dengan menggunakan persamaan (2.3) dan derivatifnya dinyatakan oleh persamaan (2.4).

$$y = f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (2.3)$$

$$\text{Dengan: } f'(x) = \frac{1}{2}[1 + f(x)][1 - f(x)] \quad (2.4)$$

Keterangan:

$f(x)$: fungsi aktivasi sigmoid bipolar

e : eksponensial

x : data ke- x

$f'(x)$: turunan (derivatif) dari fungsi sigmoid bipolar

Algoritme *artificial neural network backpropagation* memanfaatkan kesalahan *output* untuk melakukan penyesuaian pada bobot-bobotnya dengan arah mundur (*backward*). Untuk menghitung kesalahan ini, langkah awal adalah melalui tahap perambatan maju (*feedforward*) [32]. *Backpropagation* dalam *artificial neural network* adalah teknik yang digunakan untuk melatih model prediksi dengan tingkat akurasi yang signifikan. Dalam konteks ini, *backpropagation* mampu menganalisis data historis dan menghasilkan prediksi yang akurat [8]. Pendekatan *backpropagation* diterapkan secara luas dalam berbagai bidang aplikasi, seperti pengenalan pola, prediksi, dan optimisasi [7].

Berdasarkan [32] algoritme *backpropagation* memiliki beberapa kelebihan yaitu:

1. *Backpropagation* dapat digunakan untuk memecahkan masalah yang berkaitan dengan identifikasi, prediksi, pengenalan pola, dll.
2. Memiliki kemampuan belajar adaptif dan toleransi kesalahan untuk menciptakan sistem yang andal dan bekerja konsisten.
3. Melatih jaringan untuk menemukan keseimbangan selama pelatihan sehingga merespons dengan benar pola *input* yang mirip dengan pola yang digunakan dalam pelatihan.

Adapun tahapan dari algoritme *artificial neural network backpropagation* sebagai berikut [33]:

1. Tahap *feedforward* yaitu di mana nilai untuk data pelatihan dimasukkan sehingga didapatkan nilai *output*.
2. Tahap *backpropagation* yaitu proses propagasi balik dari nilai *error* yang diperoleh.
3. Meminimalkan nilai *error* yang didapat dengan menyesuaikan kembali bobot koneksi.

Berdasarkan penelitian [34], proses pelatihan algoritme *backpropagation* secara lebih detail dilakukan melalui tahapan berikut ini:

1. Menginisiasi bobot (menetapkan nilai acak kecil).
2. Selama kondisi berhenti salah, lakukan tahap *feedforward* dan *backpropagation* pada setiap data *training*.

Tahap *Feedforward*:

1. Tiap unit *input* ($X_i, i = 1, 2, \dots, n$); menerima sinyal *input* x_i lalu menyebarkannya ke seluruh lapisan tersembunyi (*hidden layer*).
2. Pada tiap unit tersembunyi ($Z_j, j = 1, 2, \dots, p$); menjumlahkan setiap sinyal *input* yang sudah memiliki bobot (termasuk biasanya) menggunakan persamaan (2.5).

$$Z_{in_jk} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2.5)$$

Kemudian menggunakan fungsi aktivasi yang telah ditentukan $z_j = f(z_{in_j})$ untuk menghitung sinyal *output* dari unit tersembunyi. Sinyal *output* ini kemudian dikirimkan ke seluruh lapisan *output*.

3. Tiap unit *output* ($Y_k, k = 1, 2, \dots, m$); menjumlahkan bobot sinyal *input* menggunakan persamaan (2.6).

$$y_{in_k} = w_{k0} + \sum_{i=1}^p z_i w_{kj} \quad (2.6)$$

Lalu menggunakan fungsi aktivasi yang telah ditentukan $y = f(y_{in_k})$ untuk menghitung sinyal *output* dari unit *output* bersangkutan. Sinyal *output* ini selanjutnya dikirim ke seluruh unit *output*.

Tahap *Backpropagation*:

1. Tiap unit *output* ($Y_k, k = 1, 2, \dots, m$); menerima suatu pola target yang mengacu terhadap pola *input* pelatihan, untuk menghitung kesalahan (*error*) antara target dengan *output* yang dihasilkan jaringan menggunakan persamaan (2.7).

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad (2.7)$$

Faktor δ_k digunakan untuk menghitung koneksi *error* (Δw_{kj}) yang nanti akan dipakai untuk memperbaiki w_{kj} menggunakan persamaan (2.8)

$$\Delta w_{kj} = \alpha \delta_k z_j \quad (2.8)$$

Selain itu juga menghitung koreksi bias yang nanti akan digunakan untuk memperbaiki w_{k0} menggunakan persamaan (2.9).

$$\Delta w_{k0} = \alpha \delta_k \quad (2.9)$$

Kemudian mengirimkan δ_k ke lapisan bawah.

2. Tiap unit tersembunyi ($Z_j, j = 1, \dots, p$); menjumlahkan *input* delta (dari unit-unit yang berada pada lapisan atasannya) yang sudah berbobot menggunakan persamaan (2.10).

$$\delta_{in_j} = \sum_{i=1}^m \delta_k w_{kj} \quad (2.10)$$

Kemudian hasilnya dikalikan dengan turunan dari fungsi aktivasi yang digunakan untuk menghitung informasi kesalahan *error* menggunakan persamaan (2.11).

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \quad (2.11)$$

Kemudian menghitung koreksi bobot menggunakan persamaan (2.12).

$$\Delta v_{ji} = \alpha \delta_j x_i \quad (2.12)$$

Kemudian menghitung koreksi bias menggunakan persamaan (2.13).

$$\Delta v_{0j} = \alpha \delta_j \quad (2.13)$$

3. Tiap unit *output* ($Y_k, k = 1, 2, \dots, m$), memperbaiki bias dan bobot-bobotnya ($j = 0, \dots, p$) menggunakan persamaan (2.14).

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (2.14)$$

4. Menguji kondisi berhenti jika *error* ditemukan. Pembelajaran jaringan dapat dihentikan ketika kondisi berhenti terpenuhi.

Keterangan:

z_{in_j} : sinyal input yang diterima oleh unit tersembunyi dari unit input.

v_{j0} : bobot bias pada lapisan tersembunyi, $j = 1, 2, \dots, p$.

v_{ji} : bobot dari unit input ke- i menuju unit tersembunyi ke- j .

z_j : sinyal output yang telah diaktivasi pada unit tersembunyi.

y_{in_k} : sinyal *input* yang diterima oleh unit *output* dari unit tersembunyi.

w_{kj} : bobot dari unit tersembunyi ke- j menuju unit *output* ke- k .

w_{k0} : bobot bias pada lapisan output, $k = 1, 2, \dots, m$.

y_k : sinyal output yang telah diaktivasi pada unit output.

t_k : target yang ingin dicapai.

α : laju percepatan.

2.2.8. Normalisasi Data

Normalisasi data merupakan proses transformasi nilai atribut pada data asli sehingga menjadi rentang tertentu sebelum digunakan dengan tetap mempertahankan informasi yang terdapat dalam data aslinya [31]. Pada *neural network*, normalisasi terhadap data *input* dapat meningkatkan performa jaringan dan mengurangi tingkat *error* pada proses *training* [35]. Berikut beberapa jenis normalisasi berdasarkan penelitian [36].

1. *Min-Max Normalization*

Min-max normalization adalah suatu teknik normalisasi yang melakukan transformasi linier dari data asli guna mencapai keseimbangan nilai perbandingan antara data sebelum dan sesudah proses normalisasi.

Rumus yang digunakan menggunakan persamaan (2.15):

$$x' = \frac{(x - \text{minValue})(\text{maxRange} - \text{minRange})}{(\text{maxValue} - \text{minValue})} + \text{minRange} \quad (2.15)$$

Dengan x' = nilai data hasil normalisasi, x = nilai data aktual, minValue = nilai minimal data aktual, maxValue = nilai maksimal data aktual, minRange = nilai *range* minimal yang akan digunakan dan maxRange = nilai *range* maksimal yang akan digunakan.

2. *Z-score Normalization*

Z-score normalization merupakan suatu metode normalisasi yang bergantung pada nilai rata-rata (*mean*) dan standar deviasi dari data. Pendekatan ini menjadi sangat bermanfaat ketika rentang nilai minimum dan maksimum dari data sebenarnya tidak diketahui. Metode ini dapat dihitung menggunakan persamaan (2.16):

$$\text{nilaibaru} = \frac{\text{nilailama} - \text{mean}}{\text{stdev}} \quad (2.16)$$

Di mana *mean* = nilai rata-rata dan *stdev* = standar deviasi dari data.

3. *Decimal Scaling Normalization*

Decimal scaling normalization adalah suatu teknik normalisasi yang menggeser nilai desimal data menuju arah yang diinginkan. Rumus yang digunakan menggunakan persamaan (2.17):

$$\text{nilaibaru} = \frac{\text{nilailama}}{10^i} \quad (2.17)$$

Metode normalisasi yang diterapkan pada penelitian ini adalah *min-max normalization* dengan mengaplikasikan fungsi sigmoid biner yang sering digunakan dalam *artificial neural network*, di mana nilai keluaran antara nol sampai satu. Hasil transformasi data ini menyesuaikan rentang nilai yang lebih terbatas, antara 0,1 hingga 0,9, seperti yang diperlihatkan dalam persamaan (2.18).

$$x' = \frac{0.8(x - \text{min})}{\text{max} - \text{min}} + 0.1 \quad (2.18)$$

Keterangan:

x' = nilai hasil normalisasi

x = nilai aktual (*real*) pada data yang akan dinormalisasi

min = nilai terkecil pada dataset

max = nilai terbesar pada dataset

Contoh penggunaan persamaan (2.18) sebagai berikut:

1. Misalkan terdapat dataset angka [15; 30; 45; 60; 75].

2. Nilai terkecilnya (*min*) adalah 15.
3. Nilai terbesarnya (*max*) adalah 75.
4. Normalisasi dengan menggunakan persamaan (3.1)

a. Untuk data 15:

$$\text{Nilai hasil normalisasi} = \frac{0.8(15-15)}{75-15} + 0.1 = 0,1$$

b. Untuk data 30:

$$\text{Nilai hasil normalisasi} = \frac{0.8(30-15)}{75-15} + 0.1 = 0,3$$

c. Untuk data 45:

$$\text{Nilai hasil normalisasi} = \frac{0.8(45-15)}{75-15} + 0.1 = 0,5$$

d. Untuk data 60:

$$\text{Nilai hasil normalisasi} = \frac{0.8(60-15)}{75-15} + 0.1 = 0,7$$

e. Untuk data 75:

$$\text{Nilai hasil normalisasi} = \frac{0.8(75-15)}{75-15} + 0.1 = 0,9$$

5. Hasil dari normalisasi menggunakan persamaan (3.1) pada dataset menjadi [0,1; 0,3; 0,5; 0,7; 0,9].

2.2.9. Mean Square Error

Mean Square Error (MSE) adalah sebuah metrik performa yang sering diterapkan dalam algoritme *backpropagation*. Fungsi ini menghitung rata-rata dari perbedaan kuadrat antara nilai prediksi yang dihasilkan oleh model dan nilai aktual yang sebenarnya [37].

Mean Square Error dapat dihitung menggunakan persamaan (2.19).

$$MSE = \frac{1}{n} \sum (Y_t - Y'_t)^2 \quad (2.19)$$

Di mana Y_t adalah nilai aktual periode t , Y'_t adalah nilai hasil prediksi periode t dan n adalah jumlah periode yang dilakukan.