

## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1. Tinjauan Pustaka

Pertanian *greenhouse* menggunakan teknologi jaringan sensor nirkabel untuk menghemat air. Integrasi jaringan sensor nirkabel atau *wireless sensor network* (*WSN*) di bidang pertanian telah memberikan inovasi baru untuk produksi tanaman yang dapat diterapkan khususnya pada *greenhouse* (Kochar dan Kumar, 2019). Parameter lingkungan seperti suhu, tingkat kelembapan pada ruang *greenhouse*, tingkat kelembapan tanah, tingkat level air, dan intensitas cahaya dipantau menggunakan *WSN* kemudian diproses menggunakan model algoritma *neural network* atau jaringan saraf tiruan dengan melakukan uji model dan validasi pada pertanian *greenhouse*. Dengan bantuan *WSN* pada *IoT* memberikan informasi parameter lingkungan (Agrawal dan Hussain, 2015).

*Internet of Things* (*IoT*) berintegasi dengan *WSN* untuk mendukung pemantauan parameter pada *greenhouse*. *IoT* berguna untuk mengambil data waktu secara nyata dan memantau parameter lingkungan pertumbuhan tanaman pada *greenhouse*. *WSN* mampu melakukan pengendalian dengan memantau tanaman jarak jauh yang terhubung internet memberikan data seperti jumlah daun, tinggi batang tanaman, berat kering dan klorofil tanaman. *Node* atau simpul *WSN* yang terdiri dari suhu, kelembapan tanah, kelembapan lingkungan, intensitas cahaya dan aktuator terhubung pada perangkat *microcontroller* dan *wifi* mengirim data melalui *cloud* ke *gateway* atau *router* untuk ditransfer ke *server cloud*. *WSN* memberikan data dan melakukan sistem kontrol pada lingkungan untuk memberikan kondisi optimal pertumbuhan tanaman. *WSN* memberikan informasi sensor dan aktuator yang disimpan dalam *cloud* melalui *wifi* untuk berbagi informasi dan memantau dengan perangkat jarak jauh. Hasil informasi sensor dapat melakukan pengembangan pada tanaman (Haruna dkk, 2019).

Pertanian merupakan sektor yang penting untuk melakukan pengembangan pada tanaman. Proses pengembangan tanaman tidak selalu berjalan baik karena adanya perubahan kondisi pada iklim, tanah dan irigasi yang tidak pasti hampir

pada semua negara. Dari perubahan kondisi tersebut diperlukan *neural network backpropagation* dengan memprediksi data dengan mengenali pola data tanaman secara cerdas. Parameter tanaman yang diolah yaitu suhu, kelembaban, kecepatan angin, radiasi matahari dan level air pada tanah. *Neural network backpropagation* bekerja dengan cara mengintegrasikan input data di lapisan input sehingga menghasilkan hasil simulasi dan memberikan hasil prediksi yang baik ditunjukkan dengan kesalahan kuadrat rata-rata minimum dibawah ambang 0,01 (Putro dkk, 2018).

Model prediksi tanaman telah diaplikasikan pada tumbuhan selada dengan faktor internal dan eksternal yang mempengaruhi pertumbuhan pada tanaman tersebut. Faktor-faktor yang mempengaruhi pertumbuhan selada salah satunya adalah infeksi mikroba karena perbedaan suhu. Prediksi pertumbuhan tanaman selada telah dilakukan menggunakan perangkat lunak *Matrix Laboratory (MATLAB)* yang mampu memberikan gambaran antara pertumbuhan mikroba dengan pengaruh suhu yang berbeda berdasarkan parameter pertumbuhan tanaman selada. Perangkat lunak *MATLAB* memberikan hasil dengan akurasi untuk mengukur antara prediksi dan pengukuran nilai dengan kolerasi yang baik. *MATLAB* sebagai alat pendukung untuk melakukan prediksi dengan model *neural network backpropagation* (Duan dkk, 2015).

Algoritma *neural network backpropagation* dapat memprediksi buah dari tanaman lada dari tahap menanam pada *greenhouse* di siang atau malam hari dengan suhu 25/21°C di bawah pencahayaan seluruh ruang dan pencahayaan tambahan selama 14 jam. Tanaman lada berumur delapan minggu ditanam dengan jarak 25 cm antara tanaman dan pemisahan 55 cm antara baris. Tanaman diukur setiap hari sejak menabur dan munculnya bunga sebanyak 50% hingga akhir panen dengan mengukur ciri-ciri tanaman yaitu tinggi tanaman, lebar kanopi, jumlah buah per tanaman, hasil buah segar per tanaman (berat masing-masing tanaman buah), dan hasil buah kering per tanaman. Prediksi dengan menerapkan *neural network* menghasilkan arsitektur 8 *neuron input layer*, 10 *neuron hidden layer* dan 1 *output layer* yang mencapai akurasi tertinggi ( $R^2 = 0.97$ ) (Gholipoor dan Nadali, 2019).

Model *neural network backpropagation* telah diterapkan pada prediksi hasil gandum dengan data input dari tahun 1997-2007 yaitu cahaya matahari, embun beku, curah hujan, dan suhu. Data tersebut dibagi menjadi tiga proses yaitu pelatihan, validasi dan pengujian. Penerapan model *backpropagation* pada hasil gandum didapatkan model arsitektur sederhana yang telah terbukti akurat dengan melakukan percobaan data. Data yang relatif kecil memiliki keakuratan sekitar 65% dengan sampel data prediksi hasil gandum sebesar 924 data. Data yang relatif besar memiliki keakuratan sekitar 98% dari sampel data prediksi hasil gandum sebesar 2772 data. Dari variasi jumlah data yang berbeda, *neural network backpropagation* mampu melakukan prediksi dengan dengan akurasi tinggi (Kadir dkk, 2015).

Algoritma *neural network backpropagation* telah digunakan secara luas untuk fungsi perkiraan dan prediksi. Salah satu keunggulan *neural network backpropagation* yang paling signifikan dibandingkan model prediksi lain yaitu prediksi dapat mendekati tingkat akurasi yang tinggi didukung informasi dari data. Algoritma *neural network backpropagation* dapat memproses data *time series* dengan melakukan pemilihan jumlah pengamatan berdasarkan data masa lalu. Algoritma *neural network backpropagation* dengan *time series* merupakan metode sederhana yang jelas untuk penentuan parameter dengan menguji banyak jaringan dari beragam jumlah *input* dan *unit* tersembunyi. Algoritma *neural network backpropagation* mampu memperkirakan masing-masing generalisasi kesalahan dan memilih jaringan dengan generalisasi kesalahan terendah (Khashei dan Bijari, 2010).

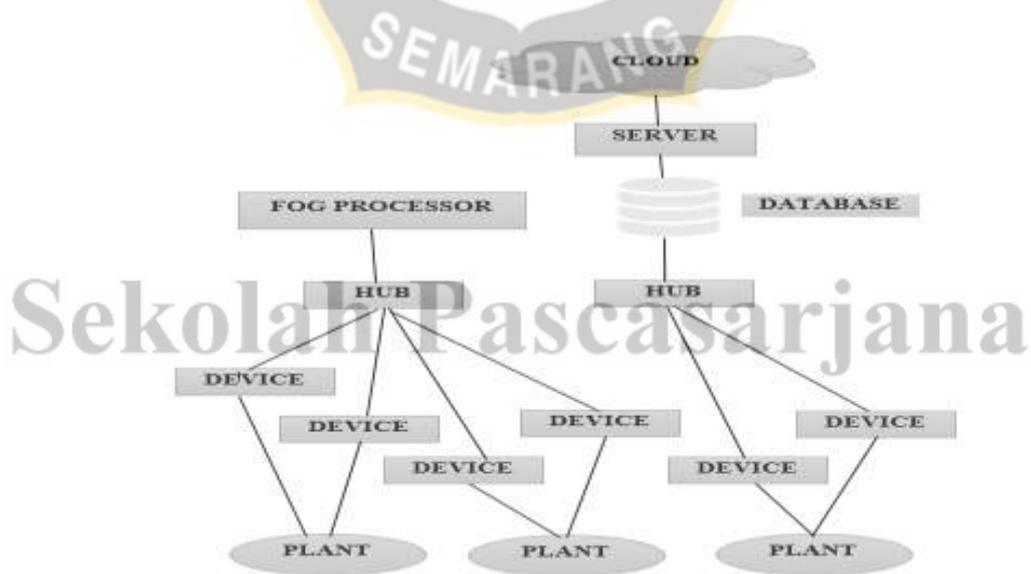
## **2.2. Dasar Teori**

### **2.2.1. Internet of Things (IoT)**

*Internet of Things (IoT)* merupakan suatu revolusi teknologi yang mewakili masa depan. *IoT* memberikan inovasi komunikasi dan komputasi secara dinamis dengan menggunakan teknologi sensor nirkabel yang berguna dalam mengidentifikasi, memantau, mengendalikan setiap objek secara otomatis (Madakam dkk., 2015). *IoT* digunakan untuk mendukung optimalisasi produksi

yang sehat dan terjangkau pada *greenhouse*. Sensor- sensor pada *IoT* digunakan untuk memberikan informasi tentang suhu, tekanan, kelembapan dan tingkat cahaya pada tanaman. Parameter lingkungan ini dipantau oleh sensor dan dikendalikan baik dengan sistem kontrol atau pengendalian secara manual. Semua ini terhubung menggunakan *server cloud IoT* mengakses data dan memberikan solusi hemat biaya bagi para petani (Kanchana, 2018).

*Internet of Things (IoT)* telah terbukti memberikan layanan yang efektif karena menghilangkan masalah spasial dan temporal. Komponen utama *IoT* meliputi jaringan sensor yang langsung kontak dengan objek, *gateway* untuk mengirim data sensor, dan *cloud* untuk merekam, menyimpan, dan memproses data-data tersebut (Suryono dkk., 2019). Arsitektur sistem *IoT* memiliki komponen yang digunakan untuk jaringan, basis data, dan lain-lain. Bagan organisasi sistem *IoT* dapat dilihat pada Gambar 2.1.



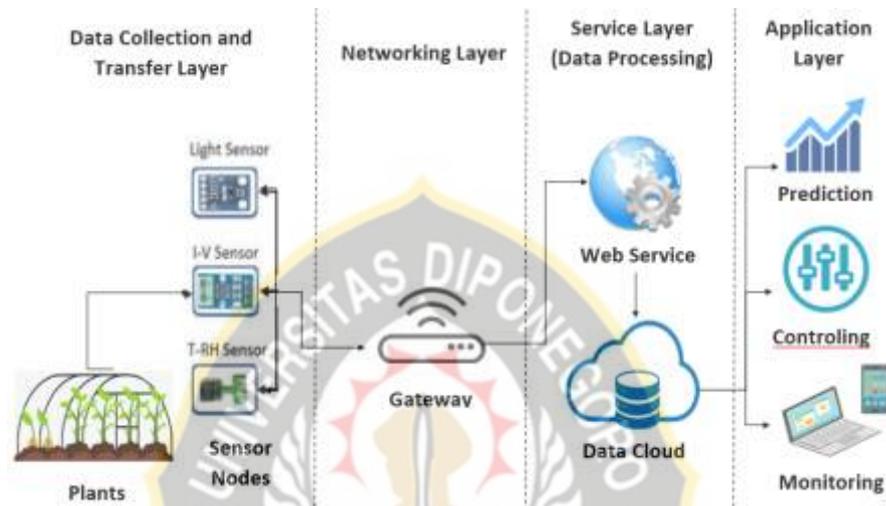
Gambar 2.1 Bagan Organisasi Sistem IoT (Serpanos dan Wolf, 2018)

Penjelasan pada Gambar 2.1 bagan organisasi sistem *IoT* adalah:

1. *Plant* sebagai sistem fisik yang berinteraksi dengan sistem *IoT*.
2. *Device* sebagai seperangkat alat yang membentuk struktur jaringan dengan menggunakan *node* yang menghubungkan sensor, aktuator, prosesor, dan

- memori. Setiap *node* memiliki antarmuka jaringan dan sebuah simpul yang memungkinkan dalam menjalankan protokol internet.
3. *Hub* menyediakan konektivitas tingkat pertama antara *node* dan seluruh jaringan dengan menjalankan *IP*.
  4. *Fog processor* melakukan operasi pada *set node* dan hub lokal. Beberapa penyimpanan pada *server* lebih dekat dengan *node* dalam mengurangi *latency* atau jeda waktu yang dibutuhkan. Namun, perangkat *fog* mungkin tidak memiliki kekuatan komputasi sebanyak *server cloud*. Perangkat *fog* juga memperkenalkan masalah manajemen sistem.
  5. *Server cloud* menyediakan layanan komputasi untuk sistem *IoT* dengan *database* yang menyimpan data dan hasil komputasi. *Cloud* juga dapat menyediakan berbagai layanan yang menghubungkan antara *node* dan pengguna (Serpanos dan Wolf, 2018).

*Internet of Things (IoT)* pada pertanian memiliki empat lapisan dalam arsitektur *IoT* yaitu lapisan pengumpulan data dan lapisan transfer, lapisan jaringan, lapisan layanan dan lapisan aplikasi. Lapisan pengumpulan data mampu menampung *node* sensor atau jaringan sensor nirkabel pada setiap sensor yang berguna untuk mengontrol penanaman. Sensor-sensor yang digunakan adalah sensor suhu, sensor kelembapan, sensor intensitas cahaya. Penanaman tumbuhan tersebut dipantau menggunakan kamera, aktuator, alat penyiraman otomatis dan lain-lain. Integrasi antara jaringan sensor nirkabel dengan *wifi* atau internet mampu memberikan data berupa informasi pertumbuhan tanaman karena *gateway* jaringan sensor nirkabel dapat mentransmisikan data melalui *gateway* internet. Data pertumbuhan tanaman dikumpulkan dalam lapisan layanan kemudian dilakukan pemrosesan data seperti visualisasi data, analisis data, penyimpanan dan perlindungan data. Setelah semua proses dilakukan lapisan aplikasi yang dijalankan oleh pengguna akhir dapat memantau dan mengendalikan berbagai proses di pertanian dan juga membuat keputusan penting berdasarkan prediksi, *tren* pasar dan departemen pertanian lokal (Ponraj dan Vigneswaran, 2019). Arsitektur *IoT* pertanian dapat dilihat pada gambar 2.2.



Gambar 2.2 Arsitektur *Internet of Things* untuk Pertanian (Ponraj dan Vigneswaran, 2019).

### 2.2.2. *Greenhouse*

*Greenhouse* atau rumah kaca adalah salah satu teknik pertanian yang efektif menyesuaikan iklim dengan kebutuhan tanaman. Teknologi penggunaan sistem sensor nirkabel yang terhubung dengan internet diterapkan pada rumah kaca dapat mengatasi kondisi lingkungan yang menyebabkan pertumbuhan tanaman menjadi terhambat. Perangkat keras dan perangkat lunak digunakan untuk mengumpulkan data dalam kapasitas besar dan algoritma perangkat lunak yang dapat mengantisipasi sistem komunikasi yang berlebihan. Sifat kimia dan fisik lingkungan sekitar tanaman sangat mempengaruhi pertumbuhan tanaman. Pertumbuhan tanaman di industri pertanian sangat kuat dipengaruhi oleh faktor lingkungan yang berkontribusi besar terhadap produktivitas (Suryono dkk., 2018).

Fasilitas *greenhouse* atau rumah kaca adalah alat penelitian tanaman yang memiliki keunggulan mudah mencapai pertumbuhan tanaman secara optimal. Lingkungan *greenhouse* relatif lebih seragam dibandingkan lingkungan luar yang disebabkan oleh iklim mikro seperti temperatur, kelembapan yang relatif, karbon yang tidak seragam dengan konsentrasi dioksida dan iradiasi. *Greenhouse* telah terbukti memperkenalkan dampak gangguan pada pertumbuhan, produksi dan kualitas tanaman. Dampak gangguan pada pertumbuhan seperti suhu dan iradiasi

dapat dianggap sebagai dua faktor utama yang berkaitan dengan iklim. Agar skema relokasi tanaman berhasil, diperlukan pemantauan dan waktu untuk pertumbuhan tanaman di masing-masing iklim mikro selama tahap pertumbuhan yang sebanding (Ma dkk., 2019). Untuk membangun dan melakukan pemantauan tanaman berkualitas tinggi pada *greenhouse* yang memberikan lingkungan optimal dari kondisi luar perlu dilakukan hal-hal seperti pemanas *greenhouse*, konservasi energi, pendingin *greenhouse*, mengontrol kelembapan, sirkulasi udara, kontrol lingkungan, tambahan pencahayaan (Wartok, 2000).

Pengukuran pertumbuhan tanaman pada lingkungan *greenhouse* harus dilakukan secara terus-menerus agar data pengukuran pertumbuhan tanaman lebih akurat. Untuk melakukan pemantauan dibutuhkan sensor dengan lokasi titik yang tepat dalam pengambilan sampel tanaman, adanya standar pengukuran tanaman, dan adanya metode frekuensi kalibrasi sensor. Sensor yang sesuai akan mengukur dan memberikan data pada pengguna sensor terkait parameter dari metadata (menggambarkan konten dan konteks file data yang tersimpan). Pada *greenhouse* diperlukan parameter yang mempengaruhi pemantauan pertumbuhan tanaman diantaranya (Both dkk., 2015):

1. Intensitas cahaya yang diukur tanaman dengan menghitung beberapa banyak total cahaya yang disediakan masing-masing sumber dan melihat perubahan dalam jumlah UV atau rasio cahaya yang berdampak pada pertumbuhan tanaman.
2. Suhu yaitu masuk dan keluarnya udara pada *greenhouse*, memerlukan pengoperasian sistem pemanas atau pendingin *greenhouse* yang akan berpengaruh pada pertumbuhan tanaman.
3. Gas (termasuk uap air) yaitu uap air mendidih yang mengembun ke permukaan dan permukaan itu lebih dingin daripada suhu titik embun yang akan berpengaruh pada pertumbuhan tanaman dengan jumlah uap air yang ada di udara area *greenhouse*.
4. Kelembapan yaitu komponen terbesar tanaman sebagai media transportasi untuk memindahkan nutrisi dari tanah ke tanaman. Kelembapan diukur untuk melihat efek pada pertumbuhan tanaman.

5. Nutrisi yaitu bahan makanan yang ada pada tanaman dilihat dari komposisi, dan dosis air dari lingkungan pertumbuhan tanaman (Both dkk., 2015).
6. Biomassa berat tanaman (kering atau basah), panjang batang, jumlah daun, kandungan klorofil, tinggi dan lebar tanaman merupakan parameter pertumbuhan tanaman (Zaidi, 1999).

### 2.2.3. *Neural Network Backpropagation*

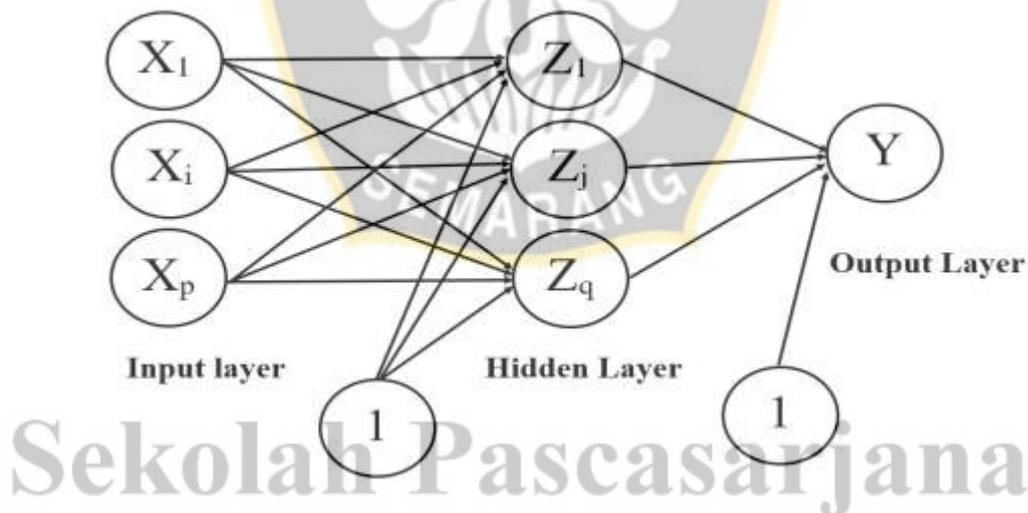
*Neural network backpropagation* atau Jaringan Saraf Tiruan (JST) adalah pemrosesan sistem informasi yang memiliki kinerja yang sama dengan jaringan saraf biologis. Algoritma *neural network backpropagation* telah dikembangkan sebagai generalisasi model matematika dari kognisi manusia atau saraf biologi berdasarkan:

1. Pemrosesan informasi terjadi pada banyak elemen yang disebut *neuron*.
2. Signal diantara *neuron* akan melewati melalui tautan koneksi.
3. Setiap tautan koneksi memiliki bobot yang terhubung dalam jaringan saraf tipikal kemudian mengalikan sinyal yang ditransmisikan.
4. Setiap *neuron* menerapkan fungsi aktivasi ke *input* bersih (jumlah sinyal *input* tertimbang) untuk menentukan sinyal *output* (Fausett, 1994).

Tujuan dari jaringan saraf tiruan adalah melatih jaringan untuk mencapai keseimbangan dengan kemampuan jaringan dalam merespons (menghafal). Jaringan memiliki kemampuan untuk memberikan tanggapan yang masuk akal terhadap *input* yang serupa tetapi tidak identik dengan yang digunakan dalam pelatihan (generalisasi). Pelatihan jaringan dilakukan dengan pembelajaran yang diawasi (*supervise*) dan pembelajaran tanpa pengawasan (*unsupervise*). Pembelajaran yang diawasi (*supervise*) melakukan pembelajaran berdasarkan nilai-nilai bobot tertentu dengan hasil yang dapat ditentukan. Pembelajaran tanpa pengawasan (*unsupervise*) melakukan pembelajaran dengan kemampuan beradaptasi sendiri yang memberikan keuntungan dalam pengelompokan data. Kumpulan data yang diambil dari pertumbuhan tanaman yang berbeda dapat

memberikan karakteristik dan model pembelajaran yang berbeda (Gosh dan Koley, 2014).

Pada algoritma *neural network backpropagation* salah satu pengembangan jaringan saraf tiruan salah satunya adalah *backpropagation* yang berguna untuk melatih jaringan sehingga diperoleh bobot-bobot optimal dalam meminimalkan *error*. Jaringan yang digunakan pada model *backpropagation* adalah jaringan *multilayer* karena memiliki kemampuan yang lebih baik dari *single layer* dalam mempelajari pola *input* dengan adanya *hidden layer* sebagai tambahan *layer* pada jaringan. Unit-unit jaringan saraf tiruan *backpropagation* terdiri dari unit *input*, unit *hidden*, dan unit *output* yang terhubung seperti pada Gambar 2.3 (Warsito, 2009).



Gambar 2.3 *Neural Network Backpropagation* (Warsito, 2009)

Pada Gambar 2.3 Algoritma *neural network backpropagation* menjelaskan bagaimana *input layer* menerima data dari vektor *input* ( $X_1$ ,  $X_i$ , dan  $X_p$ ) kemudian mengirimkannya ke *hidden layer* ( $Z_1$ ,  $Z_j$ , dan  $Z_q$ ). Selanjutnya unit dalam *hidden layer* ( $Z_1$ ,  $Z_j$ , dan  $Z_q$ ) menerima bobot masukan dan mentransfer bobot ini ke *output layer* ( $Y$ ) dengan sebuah fungsi transfer. Informasi disebarkan kedalam jaringan dengan setiap manipulasi *input* dan hasilnya dihitung disetiap unit proses. Setiap unit dalam *output layer* menghitung bobot dan kesalahan (*error*). Penyebaran balik dari *output layer* kembali ke *hidden layer* untuk melihat nilai *error* dan bobot yang

dihitung kemudian dikembalikan ke *hidden layer* untuk memperbaharui bobot (Warsito, 2009).

Pengembangan model algoritma *neural network backpropagation* memerlukan pembuatan data untuk pelatihan/pengujian model, pemilihan konfigurasi optimal dan validasi model pada data. JST saling berhubungan oleh struktur yang disebut *perceptrons* dan terdiri dari lapisan *input*, lapisan *output* dan lapisan *hidden* yang mengubah *input* menjadi sesuatu yang dapat dimanfaatkan oleh lapisan *output* (Abrougui dkk., 2019). Variabel independen untuk pembangunan model *neural network* dipilih sedemikian rupa untuk memungkinkan *neural network* berfungsi berdasarkan variabel *input*. Saat memilih topologi jaringan dan metode pembelajaran, harus memperhitungkan perkiraan dan kemampuan berdasarkan ukuran kualitas jaringan. Dengan model jaringan tersebut akan diseleksi berdasarkan penentuan parameter kualitas jaringan terbaik (Niedbala, 2019).

Tahap-tahap pembelajaran algoritma *neural network backpropagation* yaitu sebagai berikut (Effendi, 2013):

1. Normalisasi data dengan persamaan :

$$x_i \text{ normalisasi} = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (2.1)$$

Keterangan:

$x_i \text{ normalisasi}$  : hasil normalisasi pada data  $x$  ke  $i$

$x_i$  : data  $x$  ke  $i$

$\min(x)$  : data minimum pada  $x$

$\max(x)$  : data maksimum pada  $x$

2. Inisialisasi parameter jumlah *neuron input*, *neuron hidden*, *neuron output layer*, laju pembelajaran ( $\alpha$ ), jumlah epoch dan toleransi *error*.
3. Inisialisasi bobot dan bias dengan persamaan berikut :

$$\text{bobot } V = (n_{\text{input}} + 1) * n_{\text{hidden}} \quad (2.2)$$

$$\text{bobot } W = (n_{\text{hidden}} + 1) * n_{\text{output}} \quad (2.3)$$

Keterangan:

$n_{input}$  : Jumlah *neuron* pada *input layer*

$n_{hidden}$  : Jumlah *neuron* pada *hidden layer*

$n_{output}$  : Jumlah *neuron* pada *output layer*

#### 4. Perambatan maju

Perhitungan nilai *neuron* pada *hidden layer* (nilai Z) diantaranya perhitungan nilai  $Z_i$  ( $Z_1, Z_2, Z_3, \dots, Z_n$ ) dengan menggunakan persamaan berikut ini :

$$z_{in_j} = v_{oj} + \sum_{i=0}^n x_i v_{ij} \quad (2.4)$$

$$z_j = f(z_{in_j}) \quad (2.5)$$

Keterangan:

$v_{oj}$  : Bias pada *neuron hidden layer* ke-j

$x_i$  : *Neuron input* ke-i

$v_{ij}$  : Bobot yang menghubungkan *neuron input* ke-i dan *neuron hidden layer* ke-j

$z_{in_j}$  : Sinyal dari *input layer* ke *neuron hidden layer* ke-j

$z_j$  : *Neuron hidden layer* ke-j

$f(z_{in_j})$  : Fungsi aktivasi terhadap nilai  $z_{in_j}$

Sekolah Pascasarjana

Perhitungan nilai *neuron* pada *output layer* (nilai Y) diantaranya perhitungan nilai  $Y_i$  ( $Y_1, Y_2, Y_3, \dots, Y_n$ ) dengan menggunakan persamaan berikut ini:

$$y_{in_k} = w_{ok} + \sum_{j=0}^n z_j w_{jk} \quad (2.6)$$

$$y_k = f(y_{in_k}) \quad (2.7)$$

Keterangan:

$w_{ok}$  : Bias pada *neuron output* ke-k

$z_j$  : *Neuron hidden layer* ke-j

$w_{jk}$  : Bobot yang menghubungkan *neuron hidden layer* ke-j dan *neuron output layer* ke-k

$y_{in_k}$  : Sinyal dari *hidden layer* ke *neuron output layer* ke-k

$y_k$  : *Neuron output layer* ke-k

$f(y_{in_k})$  : Fungsi aktivasi terhadap nilai  $y_{in_k}$

#### 5. Perambatan mundur

Perhitungan perambatan mundur dilakukan dari *output layer* ke *hidden layer* untuk memperbarui bobot yang menghubungkan *output layer* dan *hidden layer* (bobot  $W$ ). Persamaan yang digunakan pada tahap ini yaitu sebagai berikut :

$$\delta_k = (t_k - y_k)f'(y_{in_k}) \quad (2.8)$$

$$= (t_k - y_k)y_k(1 - y_k)$$

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

$$\Delta w_{0k} = \alpha \delta_k \quad (2.10)$$

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (2.11)$$

$$w_{0k}(\text{baru}) = w_{0k}(\text{lama}) + \Delta w_{0k} \quad (2.12)$$

Keterangan:

$\delta_k$  : Faktor koreksi dari *neuron output* ke-k

$t_k$  : Target *output* pada *neuron output* ke-k

$y_k$  : *Neuron output* ke-k

$\alpha$  : Laju pembelajaran

$\Delta w_{jk}$  : Koreksi bobot yang menghubungkan *neuron output* ke-k dan *neuron hidden layer* ke-j

$\Delta w_{0k}$  : Koreksi bias pada *neuron output* ke-k

$w_{jk}(\text{baru})$  : Bobot terbaru yang menghubungkan *neuron output layer* ke-k dan *neuron hidden layer* ke-j

$w_{jk}(\text{lama})$  : Bobot lama yang menghubungkan *neuron output layer* ke-k dan *neuron hidden layer* ke-j

Perhitungan perambatan mundur dilakukan dari *hidden layer* ke *input layer* untuk memperbarui bobot  $V$  dengan menggunakan persamaan berikut ini:

$$\delta_{in_j} = \sum_{k=1}^n \delta_k w_{jk} \quad (2.13)$$

$$\begin{aligned} \delta_j &= \delta_{in_j} f'(z_{in_j}) \\ &= \delta_{in_j} z_j (1 - z_j) \end{aligned} \quad (2.14)$$

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (2.15)$$

$$\Delta v_{0j} = \alpha \delta_j \quad (2.16)$$

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (2.17)$$

$$v_{0j}(\text{baru}) = v_{0j}(\text{lama}) + \Delta v_{0j} \quad (2.18)$$

Keterangan:

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

$\delta_j$  : Faktor koreksi dari *neuron hidden layer* ke- $j$

$\alpha$  : Laju pembelajaran

$\Delta v_{ij}$  : Koreksi bobot yang menghubungkan *neuron hidden* ke- $j$  dan *neuron input layer* ke- $i$

$\Delta v_{0j}$  : Koreksi bias pada *neuron hidden* ke- $j$

$v_{ij}(\text{baru})$  : Bobot terbaru yang menghubungkan *neuron hidden layer* ke- $j$  dan *neuron input layer* ke- $i$

$v_{ij}(\text{lama})$  : Bobot lama yang menghubungkan *neuron hidden layer* ke- $j$  dan *neuron input layer* ke- $i$

6. Denormalisasi data dengan persamaan :

$$\begin{aligned} y(\text{denormalisasi}) &= (y \times \text{max total} - y \times \text{min total}) \\ &\quad + \text{min total} \end{aligned} \quad (2.19)$$

#### 2.2.4. Akurasi Prediksi

Prediksi variabel memiliki sifat kompleks yang memunculkan variasi acak (*random*) dalam memprediksi nilai-nilai yang akan datang secara benar. Akurasi atau ketepatan merupakan hal penting pada prediksi yang dapat meminimalkan kesalahan (*error*). *Error* berguna dalam menentukan perbedaan antara nilai sebenarnya (*actual*) dengan nilai prediksi untuk jangka waktu tertentu, sehingga dalam melakukan prediksi selalu ada beberapa sisa *error* yang dapat ditentukan apakah kesalahan memiliki dampak atau tidak (Stevenson, 2009).

Pengukuran akurasi prediksi menggunakan *Mean Square Error (MSE)*, dimana akurasi prediksi akan tinggi apabila nilai *MSE* semakin kecil. Pada *actual* menunjukkan nilai data yang sebenarnya dan *forecasted* merupakan nilai hasil prediksi, sedangkan *M* merupakan jumlah data sebenarnya. *MSE* dapat dihitung dengan menjumlahkan kuadrat semua *error* prediksi pada setiap periode dan membaginya dengan jumlah periode prediksi. Berikut persamaan *MSE* (Bunnoon, 2011) :

$$MSE = \sum_{i=1}^M \frac{(Actual(i) - Forecasted(i))^2}{M} \quad (2.20)$$

#### 2.2.5. Data Time Series

Data *time series* merupakan data yang dikumpulkan dengan urutan waktu masa lalu. Waktu yang digunakan seperti jam, hari, minggu, bulan, dan tahun. Data *time series* membutuhkan analisis dalam mengidentifikasi sifat dari urutan waktu dengan menggambarkan satu atau lebih plot data dalam format grafis. Pola yang ditampilkan memiliki beberapa variasi seperti pola irreguler, pola tren, pola variasi musiman, pola siklus. Data *time series* bertujuan untuk mengetahui bentuk pola data dengan peramalan model data kuantitatif (Montgomery dkk., 2015).

Data time series pada *neural network* propagasi maju yang dipengaruhi dari data masa lalu sebagai *input* dan data masa kini sebagai target. Model *neural network* untuk *time series* ditunjukkan dengan target berdasarkan data masa kini ( $X_t$ ) yang dipengaruhi data masa lalu ( $X_{t-1}, X_{t-2}, \dots, X_{t-p}$ ). *Neural network time series* dilakukan pada perambatan maju yang dipengaruhi nilai bobot-bobot atau

parameter ( $w_{bn}$ ,  $w_{in}$ ,  $v_{bo}$ ,  $v_{no}$ ) dan fungsi aktivasi yang digunakan pada tiap unit proses ( $\psi_n, \psi_0$ ). Berikut model persamaan *neural network untuk time series* (Yuliandar dkk, 2012):

$$X_t = \psi_0 \{ v_{bo} + \sum_{n=1}^H v_{no} \psi_n (w_{bn} + \sum_{i=1}^p w_{in} X_{t-i}) \} \quad (2.21)$$

Keterangan:

$w_{bn}$  : Bobot bias pada *hidden layer*

$w_{in}$  : Neuron *input layer* ke *hidden layer*

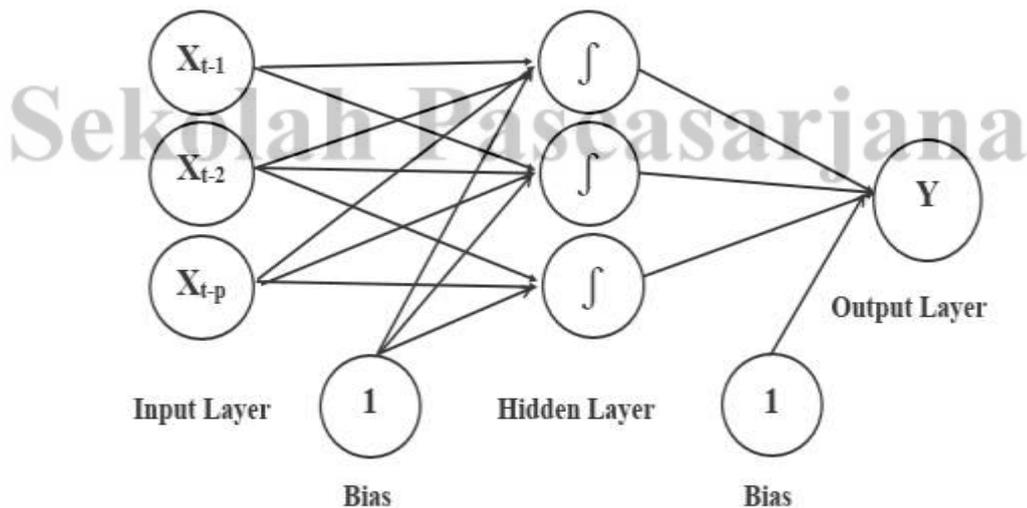
$v_{bo}$  : Bobot bias pada *output layer*

$v_{no}$  : Neuron *hidden layer* ke *output layer*

$\psi_n$  : Aktivasi neuron *output layer*

$\psi_0$  : Aktivasi neuron *hidden layer*

Pada arsitektur model *neural network time series* yang digunakan terdiri dari *input layer*, *hidden layer* dan *output layer*. Arsitektur *neural network time series* ditunjukkan pada Gambar 2.4.



Gambar 2.4 *Neural Network Time Series* (Warsito, 2006)