

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Penjadwalan Matakuliah**

Penjadwalan matakuliah adalah proses perencanaan dan pengaturan jadwal perkuliahan untuk suatu program studi atau kurikulum di sebuah institusi pendidikan, seperti universitas atau perguruan tinggi (Wati & Rochman, 2013). Tujuan utama dari penjadwalan matakuliah adalah untuk memastikan bahwa semua matakuliah yang harus diajarkan dalam program studi tertentu dapat diselenggarakan dalam periode tertentu dengan cara yang efisien dan memenuhi kebutuhan mahasiswa (Anugrah, 2018). Beberapa aspek yang harus dipertimbangkan dalam penjadwalan matakuliah meliputi:

1. Ketersediaan Sumber Daya

Menentukan ketersediaan dosen, ruang kuliah, fasilitas, dan peralatan yang diperlukan untuk menjalankan matakuliah.

2. Prasyarat dan Kelangsungan

Memastikan bahwa urutan pengambilan matakuliah sesuai dengan prasyarat dan kelangsungan yang telah ditetapkan dalam kurikulum.

3. Kebutuhan Mahasiswa

Memahami preferensi dan kebutuhan mahasiswa, seperti jadwal kuliah yang memungkinkan mereka untuk mengambil matakuliah yang mereka inginkan tanpa tumpang tindih.

4. Batasan Waktu

Menghindari tumpang tindih waktu antara matakuliah yang diperlukan, sehingga mahasiswa dapat mengikuti semua kelas yang mereka perlukan.

## 5. Keseimbangan Beban Kerja Dosen

Memastikan bahwa beban kerja dosen terdistribusi secara merata dan sesuai dengan kapasitas mereka.

## 6. Pengoptimalan Sumber Daya

Mengoptimalkan penggunaan sumber daya, seperti ruang kuliah, sehingga tidak terjadi konflik jadwal yang tidak perlu.

## 2.2. *Artificial Intelligence*

Penelitian ini membangun perangkat lunak berbasis kecerdasan buatan atau *artificial intelligence* (AI). Menurut (Markiewicz & Zheng, 2020) AI adalah sistem yang membuat sebuah komputer memiliki kecerdasan seperti manusia. Dengan tujuan tertentu AI dapat memberikan sebuah masukan atau hasil yang mirip dengan kecerdasan manusia berdasarkan data yang sudah diberikan.

### 2.2.1. *Machine Learning*

*Machine learning* adalah sub bagian dari AI yang berfokus pada komputer untuk memberikan keputusan dari sebuah masalah tanpa membuat sebuah program yang kompleks. Menurut (Markiewicz & Zheng, 2020) *machine learning* memiliki 3 jenis yakni :

- *Supervised Learning*

*Supervised learning* dapat dilatih dari sebuah data yang berlabel atau terstruktur. Semakin besar data yang diberikan maka semakin baik *output* yang dikeluarkan. Pada *supervised learning*, sebuah sistem belajar dari label yang diberikan oleh manusia.

- *Unsupervised Learning*

*Unsupervised learning* dilatih dari sebuah data yang tidak ter label dan terstruktur. Di mana algoritma tersebut akan mencari data dan mencoba membuat data tersebut menjadi terstruktur. Salah satu contoh penerapan *unsupervised learning* adalah *clustering*.

- *Reinforcement Learning*

*Reinforcement learning* belajar dan memungkinkan melakukan pengoptimalan pada algoritmanya sendiri. Cara belajar algoritma *reinforcement learning* mirip seperti *supervised learning* namun memiliki perbedaan, di mana algoritma tersebut akan mengambil keputusan berdasarkan *reward* yang sudah didapat sebelumnya.

### 2.2.2. *Deep Learning*

*Deep learning* adalah suatu cara untuk mengembangkan komputer untuk belajar menggunakan *neural network* seperti pada jaringan syaraf manusia. Algoritma *deep learning* terdiri dari sekumpulan jaringan yang terhubung dinamakan neuron (Pratama et al., 2023). *Deep learning* juga merupakan salah satu algoritma yang paling dekat dengan bagaimana cara otak manusia memproses sebuah informasi melalui jaringan syaraf dan memberikan berbagai macam hasil. *Deep learning* dapat mengembangkan sebuah model dengan berbagai macam data seperti gambar, teks, video, suara. Untuk menghasilkan performa latih yang maksimal *deep learning* memanfaatkan keuntungan dari sebuah GPU. Pada saat ini penggunaan *deep learning* memiliki kemajuan yang sangat pesat seperti pengenalan suara, gambar, mesin penerjemah dan beberapa sistem cerdas yang dapat bekerja lebih baik dari pada manusia (Markiewicz & Zheng, 2020).

### 2.3. Jaringan Syaraf Tiruan atau Artificial Neural Network

*Neural network* adalah suatu cara untuk memprogram komputer yang dapat beroperasi mirip dengan otak manusia. Tujuan utama dari *neural network* ini adalah untuk melakukan fungsi yang dapat dilakukan oleh otak manusia seperti pemecahan masalah dan dapat belajar (Suwindra et al., 2021). Menurut (Merchant, 2019) *neural network* memiliki beberapa tipe antara lain:

- *Feedforward Neural Network*  
*Feedforward Neural Network* adalah salah satu tipe *neural network* yang sederhana, di mana data dari *input* akan bergerak satu arah saja. Di mana setelah data melewati *input nodes* dan keluar melalui *output nodes*, dan *feedforward NN* tidak bisa memberikan *feedback* sehingga ketika model mengeluarkan *output* yang salah maka tidak bisa memperbaiki sendiri.
- *Radial Basis Function Neural Network*  
*Radial Bias Function* menghitung jarak dari suatu titik terhadap titik pusat. RBF memiliki dua layer, di mana pada layer pertama fitur akan dikombinasikan dengan *Radial Bias Function* pada *inner* layer dan kemudian keluaran dari fitur akan di pertimbangkan saat menghitung keluaran yang sama pada fungsi berikutnya.
- *Recurrent Neural Network (RNN)*  
Prinsip kerja *RNN* dengan menyimpan *output* layer dan memberikannya lagi ke *input* untuk membantu prediksi dari keluaran layer tersebut. *RNN* menggunakan algoritma *backpropagation* untuk memeriksa ulang dan memastikan 99.9% keluarannya benar.
- *Convolutional Neural Network (CNN)*  
*CNN* mirip dengan *feedforward neural network*, di mana sebuah neuron memiliki bobot dan bias yang dapat digunakan untuk belajar. Banyak orang yang menyukai *CNN* untuk *image processing*.
- *Modular Neural Network (MNN)*  
*MNN* adalah sekumpulan jaringan yang bekerja secara independen sembari berkontribusi terhadap *output*. Setiap jaringan memiliki satu set *input* yang unik dibandingkan dengan jaringan yang lain yang membangun dan menjalankan *sub-tasks*. Keuntungan dari *MNN* adalah mereka memecah proses komputasi ke dalam proses yang kecil di mana dapat melakukan fungsi komputasi yang lebih mudah dan cepat.

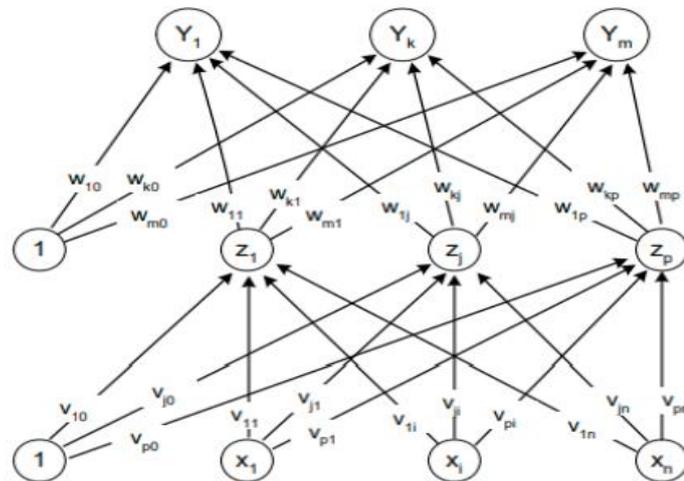
## **2.4. Algoritma Backpropagation**

*Backpropagation* algoritma pembelajaran yang terwarisi dan biasanya digunakan oleh perceptron dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya (Zulfiani & Fauzi, 2023). Metode ini merupakan metode yang sangat baik dalam menangani masalah pengenalan pola-pola kompleks (Zamsuri, 2020).

Metode *Backpropagation* yang digunakan dalam jaringan saraf tiruan dan yang paling sering digunakan dalam berbagai bidang aplikasi, seperti pengenalan pola, peramalan dan optimisasi (Masrizal & Hadiansa, 2019). Hal ini dimungkinkan karena metode ini menggunakan pembelajaran yang terbimbing. Pola masukan dan target diberikan sebagai sepasang data. Bobot-bobot awal dilatih dengan melalui tahap maju untuk mendapatkan galat keluaran yang selanjutnya galat ini digunakan dengan tahap mundur untuk memperoleh nilai bobot yang sesuai agar dapat memperkecil nilai galat sehingga target keluaran yang dikehendaki tercapai (Sunandi, 2019).

### **2.4.1. Arsitektur Algoritma Backpropagation**

Backpropagation memiliki beberapa unit yang ada dalam satu atau lebih Layer tersembunyi (Jatmika et al., 2020). Gambar 2.1 adalah arsitektur backpropagation dengan  $n$  buah masukan atau ditambah sebuah padanan lainnya, sebuah layer tersembunyi yang terdiri dari  $p$  unit atau ditambah sebuah padanan lainnya, serta  $m$  buah unit keluaran.



**Gambar 2. 1** Arsitektur *Backpropagation*

Pada Gambar 2.1  $v_{ji}$  merupakan bobot garis dari unit masukan  $x_i$  ke unit layar tersembunyi  $z_j$  atau  $v_{j0}$  merupakan bobot garis yang menghubungkan padanan lainnya di unit masukan ke unit layar tersembunyi  $z_j$ .  $w_{kj}$  merupakan bobot dari unit layar tersembunyi  $z_j$  ke unit keluaran  $y_k$  atau  $w_{k0}$  merupakan bobot dari bias di layar tersembunyi ke unit keluaran  $z_k$ .

Pada metode backpropagation, jaringan saraf tiruan perlu belajar terlebih dahulu dengan memasukkan sekumpulan contoh pelatihan yang disebut set pelatihan. Pelatihan kemudian dilangsungkan dengan maksud membuat jaringan saraf tiruan beradaptasi terhadap karakteristik dari contoh-contoh pada set pelatihan dengan cara melakukan pengubahan bobot-bobot yang ada dalam jaringan.

Cara kerja jaringan backpropagation adalah sebagai berikut, mula-mula jaringan diinisialisasi dengan bobot yang diset dengan bilangan acak. Lalu contoh-contoh pelatihan dimasukkan ke dalam jaringan. Contoh pelatihan terdiri dari pasangan vektor input dan vektor output target. Keluaran dari jaringan dibandingkan dengan vektor output target untuk mengetahui apakah output jaringan sudah sesuai dengan harapan. Error yang timbul akibat perbedaan antara output aktual dengan output target tersebut kemudian dihitung dan digunakan untuk meng-update bobot-bobot yang relevan dengan jalan mempropagasikan kembali error. Setiap perubahan bobot yang terjadi diharapkan dapat mengurangi besar error.

Backpropagation adalah metode penurunan gradien untuk meminimalkan kuadrat error keluaran (Br Sitepu, 2021). Terdapat tiga tahap yang harus dilakukan dalam pelatihan jaringan saraf tiruan yaitu tahap perambatan maju (forward propagation), tahap perambatan balik (backward propagation), dan perubahan bobot dan bias (Pradipika et al., 2017). Berikut adalah algoritma pelatihan perambatan balik:

1. Inisialisasi bobot dan bias. Bobot dan bias dapat di-inisialisasi dengan sembarang angka (acak) dan biasanya terletak antara 0, 1 ataupun -1.
2. Selama kondisi berhenti bernilai salah, kerjakan
3. Tahap 1 : Perambatan Maju (*Forward propagation*)
  - a. Setiap unit masukan ( $x_i, I = 1, 2, 3, \dots, n$ ) menerima sinyal  $x_i$  dan meneruskan sinyal tersebut ke semua unit pada lapisan tersembunyi
  - b. Setiap unit tersembunyi ( $z_j, j=1, 2, 3, \dots, n$ ) menjumlahkan bobot sinyal masukan dengan persamaan berikut:

$$z\_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij}$$

#### **Persamaan 2 1 Menjumlahkan Bobot Sinyal Unit Tersembunyi**

Keterangan:

$z\_in_j$  : Sinyal input untuk unit tersembunyi ke-j.

$v_{0j}$  : Bias untuk unit tersembunyi ke-j.

$x_i$  : Input dari unit masukan ke-i.

$v_{ij}$  : Bobot yang menghubungkan unit masukan ke-i dengan unit tersembunyi ke-j.

Dan menerapkan fungsi aktivasi untuk menghitung sinyal keluarannya:

$$z_j = f(z\_in_j)$$

#### **Persamaan 2 2 Menghitung Sinyal Keluaran**

Keterangan:

$z_j$ : Sinyal keluaran dari unit tersembunyi ke-j setelah diterapkan fungsi aktivasi.

$f$ : Fungsi aktivasi yang dapat berupa sigmoid, ReLU, tanh, atau fungsi lainnya.

- c. Setiap unit keluaran ( $y_{in_k} = 1, 2, 3, \dots, n$ ) menjumlahkan bobot sinyal masukan dengan persamaan berikut:

$$y_{in_k} = w_{0k} + \sum_{i=1}^n z_i w_{jk}$$

### Persamaan 2 3 Menghitung Bobot Sinyal Setiap Unit Keluaran

Keterangan:

$y_{in_k}$  : Sinyal input untuk unit keluaran ke-k.

$w_{0k}$  : Bias untuk unit keluaran ke-k.

$z_i$  : Sinyal keluaran dari unit tersembunyi ke-i.

$w_{jk}$  : Bobot yang menghubungkan unit tersembunyi ke-i dengan unit

Dan menerapkan fungsi aktivasi untuk menghitung sinyal keluarannya:

$$y_k = f(y_{in_k})$$

### Persamaan 2 4 Menghitung Sinyal Keluaran

Keterangan:

$y_k$  : Sinyal keluaran dari unit keluaran ke-k setelah diterapkan fungsi aktivasi

$f$  : Fungsi aktivasi yang sama seperti pada langkah sebelumnya.

Kemudian mengirimkan sinyal tersebut ke semua unit keluaran

#### 4. Tahap 2 : Perambatan Balik (*Backpropagation*)

- a. Setiap unit keluaran ( $y_k, k=1,2,3, \dots, n$ ) menerima pola target yang sesuai dengan pola masukan pelatihan, kemudian hitung galat dengan persamaan berikut :

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

### Persamaan 2 5 Hitung Galat

Keterangan:

$\delta_k$  : Galat unit keluaran ke-k.

$t_k$  : Nilai target yang sesuai dengan unit keluaran ke-k.

$y_k$  : Sinyal keluaran dari unit keluaran ke-k.

$f'$  : Turunan dari fungsi aktivasi yang diterapkan pada sinyal input  $y_{in_k}$

Kemudian menghitung koreksi bobot dengan persamaan berikut:

$$\Delta w_{jk} = \alpha \delta_k x_j$$

#### Persamaan 2 6 Hitung Koreksi Bobot

Keterangan:

$\alpha$  : Laju pembelajaran (learning rate), yang merupakan parameter yang mengatur seberapa besar perubahan bobot akan dilakukan pada setiap iterasi.

$x_j$  : Sinyal keluaran dari unit tersembunyi ke-j.

Menghitung pula koreksi bias dengan persamaan berikut:

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

#### Persamaan 2 7 Hitung Koreksi Bias

Keterangan:

$\Delta w_{0k}$  : Koreksi bias untuk unit keluaran ke-k.

$\alpha$  : Laju pembelajaran.

$\delta_k$  : Galat unit keluaran ke-k.

- b. Setiap unit tersembunyi ( $z_j$ ,  $j=1,2,3 \dots n$ ) menjumlahkan delta masukannya:

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk}$$

#### Persamaan 2 8 Menjumlahkan Delta

Keterangan:

$\delta_{in_j}$  : Delta masukan untuk unit tersembunyi ke-j.

$\delta_k$  : Galat unit keluaran ke-k.

$w_{jk}$  : Bobot yang menghubungkan unit keluaran ke-k dengan unit tersembunyi ke-j.

Untuk menghitung informasi error atau galat, kalikan nilai ini dengan turunan dari fungsinya:

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

**Persamaan 2 9 Hitung Informasi Error**

Keterangan:

$\delta_j$  : Informasi error untuk unit tersembunyi ke-j.

$f'$  : Turunan dari fungsi aktivasi yang diterapkan pada sinyal input  $z_{in_j}$

Kemudian menghitung koreksi bobot dengan persamaan sebagai berikut:

$$\Delta v_{ij} = \alpha \delta_i x_j$$

**Persamaan 2 10 Hitung Koreksi Bobot**

Keterangan:

$\Delta v_{ij}$  : Koreksi bobot untuk bobot yang menghubungkan unit masukan ke-i dengan unit tersembunyi ke-j.

$\alpha$  : Laju pembelajaran.

$\delta_i$  : Informasi error untuk unit tersembunyi ke-j.

$x_j$  : Sinyal keluaran dari unit masukan ke-j.

5. Tahap 3 : Perubahan Bobot dan Bias

- a. Setiap unit keluaran ( $y_k, k=1, 2, 3, \dots, n$ ) dilakukan perubahan bobot ( $j=0, 1, 2, \dots, n$ ) dengan persamaan sebagai berikut:

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

**Persamaan 2 11 Hitung Perubahan Bobot Setiap Unit Keluaran**

Keterangan:

$w_{jk}(\text{baru})$  : Bobot baru untuk koneksi dari unit tersembunyi ke-j ke unit keluaran ke-k.

$w_{jk}(\text{lama})$  : Bobot lama sebelum diperbarui.

$\Delta w_{jk}$  : Koreksi bobot yang dihitung pada tahap sebelumnya.

Setiap unit tersembunyi ( $z_j, j=1, 2, 3, \dots, n$ ) dilakukan perubahan bobot ( $i=0, 1, 2, \dots, n$ ) dengan persamaan sebagai berikut:

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

**Persamaan 2 12 Hitung Perubahan Bobot Setiap Unit tersembunyanya**

Keterangan:

$v_{ij}(baru)$  : Bobot baru untuk koneksi dari unit masukan ke-i ke unit tersembunyi ke-j.

$v_{ij}(lama)$  : Bobot lama sebelum diperbarui.

$\Delta v_{ij}$  : Koreksi bobot yang dihitung pada tahap sebelumnya.

b. Tes kondisi berhenti.

Terdapat tiga tahap dalam jaringan saraf tiruan:

- Tahap pelatihan jaringan
- Tahap pengujian jaringan
- Tahap prediksi

## 2.5. Fungsi Aktivasi

Activation Function adalah sebuah fungsi aktivasi yang digunakan pada neural network untuk menghitung seluruh dari jumlah bobot dan bias, yang dapat digunakan untuk memutuskan apakah neuron bisa di aktifkan atau tidak (Emanuella, 2022). Fungsi aktivasi yang digunakan pada penelitian kali ini adalah:

- *Sigmoid Function*

Fungsi Sigmoid bersifat nonlinier, berada pada rentang  $[0,1]$ , dan kurvanya terlihat seperti bentuk S. Meskipun monotonik, dan outputnya sangat curam, perubahan kecil pada input akan membawa perubahan besar pada output ketika inputnya mendekati 0, namun di sisi mana pun, outputnya cenderung ke 0 atau 1, dan responsnya sangat kecil terhadap perubahan tersebut. masukannya, artinya gradien pada daerah ini akan menjadi sangat kecil, bahkan mendekati 0, hal ini disebut dengan “gradien hilang”. Dalam hal ini, jaringan akan menolak untuk belajar lebih jauh, hampir tidak ada sinyal yang mengalir melalui neuron ke bobotnya dan secara rekursif ke datanya. Jadi, orang harus ekstra hati-hati saat menginisialisasi bobot neuron sigmoid untuk mencegah saturasi. Untungnya, ada cara untuk mengatasi masalah ini dan sigmoid masih sangat populer dalam masalah klasifikasi, terutama pada lapisan keluaran klasifikasi biner, yang hasilnya adalah 0 atau 1, karena nilai

fungsi sigmoid terletak di antara 0 dan 1 saja, jadi, hasilnya dapat diprediksi dengan mudah menjadi 1 jika nilainya lebih besar dari 0,5 dan 0 sebaliknya. Fungsi sigmoid ini dapat dilihat menggunakan persamaan berikut:

$$f(x_i) = \frac{e^{x_1}}{1 + e^{x_1}} = \frac{1}{1 + e^{-x_1}}, \quad f'(x_i) = \frac{e^{-x_1}}{(1 + e^{x_1})^2}$$

#### **Persamaan 2 13 Fungsi Aktivasi Sigmoid**

Fungsi  $f(x_i)$  diatas, adalah nilai output dari fungsi sigmoid untuk input  $x_1$  sedangkan  $f'(x_i)$  adalah turunan dari fungsi sigmoid terhadap input  $x_1$ . Turunan ini digunakan dalam proses perhitungan gradien saat melakukan pembelajaran dalam jaringan saraf tiruan menggunakan metode backpropagation.

## **2.6. Software yang Digunakan untuk Pengembangan Model**

### **2.6.1. Matlab**

Matlab adalah lingkungan komputasi numerik dan bahasa pemrograman yang digunakan secara luas dalam berbagai bidang, mulai dari ilmu teknik dan sains hingga keuangan dan analisis data (Maharani et al., 2020). Matlab memungkinkan pengguna untuk melakukan berbagai operasi matematika, analisis data, visualisasi, dan pemodelan dengan mudah. Lingkungan Matlab mencakup berbagai fungsi dan alat yang kuat untuk memecahkan masalah matematika, mengimplementasikan algoritma, serta menganalisis dan memvisualisasikan data. Matlab adalah software yang dibuat oleh the mathwork inc, software ini sangat bermanfaat dalam menyelesaikan berbagai permasalahan numerik, perangkat lunak ini menawarkan kemudahan dan kesederhanaan dalam menyelesaikan permasalahan yang berhubungan dengan vector dan, matriks. Memperoleh inversi matriks dan menyelesaikan persamaan linier merupakan contoh permasalahan yang dapat dipecahkan dengan cara yang sangat singkat dan mudah sekali (Hutabarat et al., 2021).

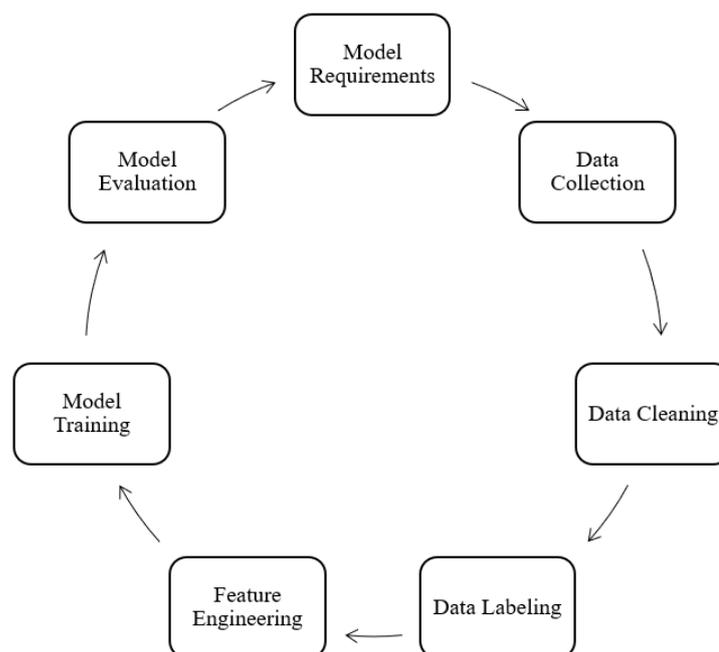
### 2.6.2. Microsoft Excel

Excel merupakan perangkat lunak spreadsheet yang umum digunakan dalam pemrosesan data. Dalam konteks preprosesing data, Excel mampu melakukan sejumlah fungsi krusial. Ini termasuk kemampuan membersihkan data dari nilai-nilai anomali, mengatasi kekosongan data, serta melakukan transformasi data dengan mengubah format dan mengintegrasikan kolom. Fitur filtering, pengurutan, dan fungsi statistik dalam Excel memungkinkan analisis yang sederhana, sedangkan kemampuan visualisasinya dapat menghasilkan grafik dasar (Khadka, 2019).

## 2.7. Metode Pengembangan Model

### 2.7.1. Machine Learning Life Cycle

Berikut adalah metode pengembangan perangkat lunak menggunakan *machine learning life cycle*



**Gambar 2. 2** Machine Learning Life Cycle

Pada tahapan pengembangan perangkat lunak menggunakan *machine learning life cycle* memiliki beberapa tahapan antara lain (Gärtler dkk., 2021) :

### 1. *Model Requirements*

Pada fase *model requirements* dilakukan pemilihan model, pengumpulan data, dan jumlah data yang akan diambil kemudian akan diimplementasikan ke dalam arsitektur model. Setelah seluruh kebutuhan model sudah sesuai dan terpenuhi kemudian akan diimplementasikan ke tahap-tahap selanjutnya.

### 2. *Data Collection*

Pada fase *data collection* akan dilakukan pengumpulan data terkait kasus penelitian yang kemudian akan dilakukan pelatihan untuk model.

### 3. *Data Cleaning*

*Data cleaning* digunakan untuk memastikan kualitas dari sampel seperti akurasi kelengkapan, konsistensi, keunikan dan integritas. Pada *data cleaning* sangat diperlukan untuk standarisasi dari dataset yang dimiliki agar sesuai dengan Bahasa pemrograman atau software yang digunakan.

### 4. *Data Labeling*

*Data labeling* adalah proses pelabelan semua data yang ada di dalam kumpulan dataset yang akan dijadikan sebagai keluaran atau output yang menjadi variable predictor di dalam penelitian ini.

### 5. *Feature Engineering*

*Feature engineering* adalah proses untuk mengekstrak, seleksi, mengubah dan memilih fitur dari data mentah ke dalam bentuk yang sesuai untuk arsitektur model

### 6. *Model Training*

*Model training* mencakup aktivitas untuk melatih dan memilih arsitektur model kemudian menyesuaikan *learning rate* pada data yang sudah dikumpulkan, dibersihkan dan dilabelkan.

### 7. *Model Evaluation*

*Model evaluation* pada tahapan ini akan mengkonfirmasi bahwa suatu model dapat memenuhi persyaratan dalam sebuah kasus. Pada kasus kali ini akan menggunakan teknik *MSE* di mana model akan di evaluasi berdasarkan kesalahan yang model perbuat dalam melakukan prediksi.

## 2.8. Penelitian Terkait

Ada beberapa penelitian terkait dengan penggunaan teknik convolutional neural network yang dapat dilihat pada tabel 2.2.

**Tabel 2. 1** Penelitian Terdahulu

<b>Judul, Penulis dan Tahun</b>	<b>Jumlah dan Atribut Dataset</b>	<b>Algoritma</b>	<b>Preprocessing</b>	<b>Feature Selection</b>	<b>Validasi</b>	<b>Open Source Dataset</b>	<b>Akurasi / Matrix Score / Hasil</b>
Penerapan Jaringan Syaraf Tiruan dengan Algoritma Backpropagation dalam Memprediksi Hasil Panen Gabah Padi (Maiyuriska, 2022)	Tidak disebutkan	Jaringan Syaraf Tiruan dan Metode Backpropagation	Normalisasi	Tidak disebutkan	MSE	-	Arsitektur 4-15-1 = 89.40%
							Arsitektur 4-20-1 = 91.30%
							Arsitektur 4-24-1 = 88.50%
							Arsitektur 4-26-1 = 92.90%
							Arsitektur 4-28-1 = 86.77%

<b>Judul, Penulis dan Tahun</b>	<b>Jumlah dan Atribut Dataset</b>	<b>Algoritma</b>	<b>Preprocessing</b>	<b>Feature Selection</b>	<b>Validasi</b>	<b>Open Source Dataset</b>	<b>Akurasi / Matrix Score / Hasil</b>
Penggunaan Jaringan Syaraf Tiruan – Backpropagation dalam Memprediksi Jumlah Mahasiswa Baru (Suahati et al., 2022)	Tidak disebutkan	Jaringan Syaraf Tiruan dan Metode Backpropagation	Normalisasi	Tidak disebutkan	MSE	-	Hasil penelitian menunjukkan bahwa arsitektur jaringan terbaik menggunakan pola data input 12, hidden neuron 8, dan output 1, dengan fungsi aktivasi logsig-purelin, dan fungsi percepatan traingda yang mencapai target MSE 0.0001 pada iterasi ke-94. Kemudian dilakukan proses prediksi jumlah mahasiswa baru yang menunjukkan bahwa

<b>Judul, Penulis dan Tahun</b>	<b>Jumlah dan Atribut Dataset</b>	<b>Algoritma</b>	<b>Preprocessing</b>	<b>Feature Selection</b>	<b>Validasi</b>	<b>Open Source Dataset</b>	<b>Akurasi / Matrix Score / Hasil</b>
							jumlah mahasiswa baru pada tahun 2021 diperkirakan sebanyak 115 mahasiswa.
Prediksi Beban Listrik Dengan Menggunakan Jaringan Syaraf Tiruan Metode Backpropagation (Rohman, 2022)	Jumlah data tidak disebutkan dan jumlah atribut data sebanyak 7 atribut	Jaringan Syaraf Tiruan dan Metode Backpropagation	Normalisasi	Tidak disebutkan	MSE dan MAPE	-	Berdasarkan Hasil pelatihan Jaringan Saraf Tiruan metode bacpropagation menunjukkan bahwa dengan menggunakan fungsi trainlm didapatkan pelatihan terbaik dengan nilai MSE yang di capai 0,0037377. Dalam prediksi beban listrik

Judul, Penulis dan Tahun	Jumlah dan Atribut Dataset	Algoritma	Preprocessing	Feature Selection	Validasi	Open Source Dataset	Akurasi / Matrix Score / Hasil
							menggunakan Jaringan Syaraf Tiruan, nilai MAPE prediksi tidak melebihi batas toleransi deviasi prediksi yang ditetapkan oleh PLN, yaitu $\pm 5\%$ .
IMPLEMENTASI ALGORITMA BACKPROPAGATION UNTUK MEMPREDIKSI PENJUALAN KERIPIK (STUDI KASUS: RUMAH	Jumlah data tidak disebutkan dan jumlah atribut data sebanyak 13 atribut	Jaringan Syaraf Tiruan dan Metode Backpropagation	Normalisasi	Tidak disebutkan	MSE dan MAPE	-	Model JST memiliki pola 12-20-1 dengan 12 lapisan masukan, 20 lapisan tersembunyi, dan 1 lapisan keluaran. Hasil prediksi menunjukkan bahwa untuk penjualan keripik singkong pada tahun 2022, nilai MSE adalah 0.14181

<b>Judul, Penulis dan Tahun</b>	<b>Jumlah dan Atribut Dataset</b>	<b>Algoritma</b>	<b>Preprocessing</b>	<b>Feature Selection</b>	<b>Validasi</b>	<b>Open Source Dataset</b>	<b>Akurasi / Matrix Score / Hasil</b>
INDUSTRI KERIPIK AGE ENAK) (Fitri et al., 2022)							dan MAPE sebesar 28%. Hal ini menunjukkan bahwa perhitungan JST Backpropagation dinyatakan layak atau cukup baik. Sementara itu, untuk penjualan keripik pisang pada tahun 2022, nilai MSE adalah 0.22071 dan MAPE sebesar 21%, juga dengan kesimpulan bahwa perhitungan JST Backpropagation layak.
Prediksi Kunjungan Wisata Kota	Jumlah data tidak	Jaringan Syaraf	Normalisasi	Tidak disebutkan	MSE	-	Prediksi jumlah kunjungan wisata di Kota

Judul, Penulis dan Tahun	Jumlah dan Atribut Dataset	Algoritma	Preprocessing	Feature Selection	Validasi	Open Source Dataset	Akurasi / Matrix Score / Hasil
Payakumbuh Menggunakan Metode Jaringan Syaraf Tiruan Backpropagation (Aulya, 2022)	disebutkan dan jumlah atribut data sebanyak 5 atribut	Tiruan dan Metode Backpropagation					Payakumbuh dengan menggunakan jaringan saraf tiruan algoritma backpropagation mendapatkan tingkat kesalahan yang sangat kecil, dengan nilai error MSE sebesar 0.00991. Jaringan syaraf tiruan ini memiliki pola 4-27-1. Proses prediksi melibatkan variabel prediktor seperti total kunjungan, inflansi, iklim, dan masa libur.