



BAB II KAJIAN PUSTAKA

2.1 Penelitian Terdahulu

Terdapat beberapa penelitian sejenis yang diangkat menjadi topik oleh beberapa peneliti terdahulu. Hal tersebut perlu dipelajari dan dikaji kembali dengan tujuan agar dapat dijadikan sebagai bahan panduan dalam melakukan penelitian yang serupa. Beberapa penelitian terdahulu yang berhubungan dengan prediksi jumlah pelanggan maupun dengan metode *Backpropagation* antara lain, sebagai berikut :

Table 2. 1 Penelitian Terdahulu

No	Peneliti (Tahun)	Judul Penelitian	Hasil Penelitian
1.	Vivi Auladina (2021)	Penerapan Algoritma <i>Backpropagation</i> dalam Memprediksi Jumlah Pengguna Kereta Api di Pulau	Hasil penelitian yang memprediksi jumlah pengguna kereta api di pulau Sumatera dengan Metode <i>Backpropagation</i>



		Sumatera	mendapatkan akurasi kebenaran yang diperoleh 92%
2.	Muhammad Syafiq (2020)	Prediksi Jumlah Penjualan Produk di PT. Ramayana Pematangsiantar Menggunakan Metode JST Backpropagation	Hasil penelitian dalam memprediksi jumlah penjualan produk di Ramayana yaitu arsitektur yang terbaik adalah 3-35-1 dengan tingkat akurasi 92%
3.	Rika Setiana (2022)	<i>Backpropagation Algorithm in Predicting Amount of West Siantar Population Growth</i>	Hasil penelitian dalam memprediksi pertumbuhan penduduk di Kota Pematang Siantar adalah arsitektur terbaik 4-19-1 dengan akurasi 100, angka MSE



			0.00010031375 dan nilai epoch 124777
4.	Ahmad Yusuf Naufal (2023)	Implementasi Backpropagation ANN dan Algoritma Genetika Terhadap Estimasi Pendapatan Agen Ekspedisi Pengiriman Barang	Hasil penelitian dalam estimasi pendapatan agen JNE Express adalah metode Backpropagation ANN menghasilkan arsitektur terbaiknya 4-10- 1, epoch 487 dan MSE 0.150807 sedangkan etode algoritma genetika fitness terbaiknya 10209,02 dan nilai MSE 0.000098 sehingga diprediksi pendapatan bulan Oktober



			2021 Rp. 25.011.603,00, November 2021 sebesar Rp. 20.920.021,00 dan Desember 2021 sebesar Rp. 50.553.019,00
5	Fery Catur Wijaya (2023)	Penerapan Metode JST <i>Backpropagation</i> pada Peramalan Produksi <i>Pastry</i> di Hyfresh Blitar	Dalam penelitian ini digunakan tiga arsitektur JST dengan konfigurasi yang berbeda yaitu 1- 5-1, 1-10-1, dan 1-15-1. Hasil pengujian menunjukkan bahwa ketiga arsitektur JST yang digunakan dalam peramalan penjualan menunjukkan tingkat akurasi yang baik,



			dengan nilai RMSE yang <i>relative</i> kecil yakni masing-masing 0.134923, 0.07458, 0.32818. hasil ini mengindikasikan bahwa arsitektur 2 yaitu 1-10-1 memiliki tingkat akurasi yang paling baik
--	--	--	--

2.2 Data Mining

Data mining adalah sebuah istilah yang sering digunakan sebagai suatu cara untuk menguraikan serta mencari penemuan berupa pengetahuan didalam *database*. Menurut Lila Setiyani dkk, (2020) *data mining* ialah suatu pencarian dan analisa pada sebuah pengumpulan data (*database*) untuk menemukan suatu pola yang menarik dengan tujuan mengekstrak informasi dan pengetahuan yang akurat dan potensial, juga dapat dipahami dan berguna untuk



pengambilan sebuah keputusan. *Data mining* adalah suatu kegiatan untuk menemukan, menggali informasi dari *data history* untuk menentukan keteraturan, pola, maupun hubungan dari sejumlah data berukuran besar guna mendapatkan informasi tersembunyi yang awalnya tidak diketahui.

Menurut MacLennan, Tang, dan Crivat (2011) dalam Lila Setiyani dkk, (2020), berdasarkan fungsinya *data mining* dapat dikelompokkan sebagai berikut :

1. Deskripsi

Deskripsi merupakan cara untuk memperlihatkan pola dan kecenderungan yang terdapat pada data yang dimiliki.

2. Estimasi

Estimasi hampir serupa dengan klasifikasi namun yang membedakan ialah *variable* target estimasi lebih cenderung statistik numerik daripada statistik kategori. Model yang diterapkan menggunakan *data record* lengkap yang akan disediakan nilai *variable* target sebagai nilai prediksi.

3. Prediksi



Prediksi ialah suatu proses yang menganggap atau menerka sebuah nilai yang belum diketahui kelasnya juga memperkirakan nilai untuk masa mendatang berdasarkan data masa lalu.

4. Klasifikasi

Hampir sama dengan estimasi namun yang membedakan ialah terdapat variable kategori contohnya tinggi, sedang, dan rendah.

5. *Clustering*

Adalah proses penggabungan data, pengamatan, memperhatikan dan membentuk kelas-kelas pada objek yang memiliki tingkat kesamaan ataupun kemiripan.

6. Asosiasi

Bertujuan menemukan perlengkapan yang terdapat dalam satu waktu. Dalam dunia bisnis disebut analisis keranjang belanja.

Dalam penelitian ini penulis menerapkan *data mining* menggunakan proses prediksi, sehingga perlu diketahui teknik-teknik prediksi dalam *data mining*. Teknik prediksi dalam data mining ialah metode untuk memprediksi hasil dimasa depan berdasarkan data historis.

Menurut Wiyata Mandala & Putri (2018) macam teknik prediksi dalam *data mining* :

1. Jaringan Saraf Tiruan (JST)



membantu menyelesaikan masalah pada kondisi variable naik turun (fluktuasi) yang tak linier menggunakan perubahan bobot sinapsisnya.

2. Tetangga Terdekat (*Nearest Neighbor*)

Ialah pendekatan untuk mencari kasus dengan menghitung kedekatan antara kasus baru dengan kasus lama berdasarkan pada pencocokan bobot dari sejumlah fitur.

3. Pohon Keputusan (*Decision Tree*)

Adalah model prediktif yang berbentuk layaknya pohon, dengan setiap node mewakili sebuah keputusan atau pertanyaan mengenai data.

2.3 Prediksi

Menurut Dewi, S.P. dkk (2022) prediksi ialah salah satu metode yang digunakan untuk memprediksi atau meramalkan suatu hal yang belum pernah terjadi. Prediksi ialah suatu kegiatan meramalkan secara terstruktur mengenai sesuatu yang mungkin terjadi dimasa yang akan datang yang bersumber dari informasi masa lalu dan sekarang yang dimiliki agar kesalahan dapat diperkecil.

Menurut Cahyati (2024) langkah-langkah dalam prediksi dapat dilakukan dengan menguraikan permasalahan yang muncul diikuti dengan pengumpulan informasi untuk membantu memecahkan permasalahan dan menganalisis



data untuk memastikan teknik prediksi yang sesuai. Setelahnya membangun model yang akan digunakan dalam proses prediksi dan mengujinya untuk mengukur seberapa besar tingkat akurasi yang dihasilkan. Terdapat dua metode pendekatan yang dapat digunakan dalam teknik prediksi yaitu pendekatan kualitatif dan kuantitatif. Metode pendekatan kualitatif bersifat lebih subjektif karena hanya berdasarkan pengetahuan dan pengalaman yang telah dialami penulis. Sedangkan metode pendekatan kuantitatif bersifat lebih objektif karena berdasarkan dari data historis periode sebelumnya. Dalam penelitian ini digunakan metode pendekatan kuantitatif karena menggunakan data historis jumlah pelanggan Agen Indah Logistik Cargo Ngoro Jombang 24 April 2025 – 31 Mei 2025.

Metode yang disusun dalam proses prediksi kuantitatif dibagi menjadi dua kategori antara lain :

- 1) Metode *Time Series* merupakan metode prediksi yang ber sumber dari data masa lalu yang digunakan untuk membuat perkiraan dimasa yang akan datang. Metode ini dapat digunakan pada kondisi dimana tidak dapat dijelaskan faktor apa saja yang mampu menyebabkan terjadinya keadaan yang akan diprediksi.
- 2) Metode Kausal (sebab-akibat) adalah metode yang digunakan ketika terdapat variable yang menyebabkan terjadinya keadaan yang akan diprediksi. Metode kausal



terdiri dari dua variable yaitu variable terikat (*dependent variable*) dan variable bebas (*independent variable*)

2.4 Jaringan Saraf Tiruan (Artificial Neural Network)

Menurut Cahyati (2024) Jaringan Saraf Tiruan (*Artificial Neural Network*) awal mula dipelopori oleh seorang matematikawan Walter Pitts dan 10tatistic Warren S. McCulloch pada tahun 1943 yang selanjutnya dikenal dengan sebutan *McCulloch-Pitts Neuron*. Neuron merupakan simulasi operasi logika dengan dua ataupun lebih input data yang menghasilkan satu keluaran.

Menurut Rifky dkk, (2024) Jaringan Saraf Tiruan adalah sebuah model komputasi yang terinspirasi dari cara kerja jaringan saraf biologis di dalam otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia. JST merupakan sistem adaptif yang mampu menyesuaikan strukturnya untuk mengatasi masalah dengan menggunakan informasi dari luar maupun dari dalam jaringan tersebut dan sering disebut sebagai jaringan adaptif. Jaringan Saraf Tiruan dapat dijelaskan sebagai struktur yang terdiri dari unit-unit pemrosesan sederhana yang saling terhubung, disebut neuron buatan, yang mampu melakukan komputasi secara parallel dalam skala besar untuk memproses data dan mewakili pengetahuan.



Jaringan Saraf Tiruan digunakan untuk memproses informasi yang dibentuk menyerupai cara kerja otak manusia dalam menyelesaikan masalah dengan melakukan proses menggali ilmu menggunakan perubahan bobot sinapsisnya (Sutikno, 2016). Jaringan Saraf Tiruan mampu mendeteksi kegiatan berdasarkan data masa lampau kemudian memberikan keputusan terhadap data yang baru. Jaringan Saraf Tiruan merupakan salah satu metode prediksi yang akan digunakan dalam penelitian ini, karena dapat menyelesaikan permasalahan yang kompleks.

Hubungan jaringan saraf biologis dan jaringan saraf tiruan dapat ditampilkan melalui tabel berikut :

Tabel 2.4 Hubungan Jaringan Saraf Biologis dan JST

Jaringan Saraf Biologis	Jaringan Saraf Tiruan
Dendrit (menerima)	Masukan (<i>Input</i>)
Neuron (mengirimkan)	Simpul (<i>Node</i>)
Sinapsis (penghubung)	Bobot (<i>Weight</i>)
Akson (mengembalikan)	Keluaran (<i>Output</i>)

Menurut Effendi (2018) jaringan saraf tiruan diasumsikan sebagai berikut :

1. Proses pembuatan informasi terjadi pada banyak elemen-elemen sederhana yang disebut neuron.



2. Sinyal-sinyal dikirim antar neuron melalui *connections-link* (Sinapsis).
3. Setiap sinapsis memiliki bobot tertentu tergantung tipe NN
4. Setiap neuron menggunakan fungsi aktivasi (tidak linier) yang merupakan penjumlahan sinyal-sinyal keluaran (T. Sutojo, AF, 2010)

Menurut Sari, A.W.P (2019) umumnya karakteristik jaringan saraf tiruan ditentukan oleh beberapa komponen diantaranya :

1. Arsitektur (pola hubung antar neuron)
2. Penentuan bobot koneksi menggunakan metode pelatihan(*training*), pembelajaran (*learning*), dan algoritma.
3. Fungsi aktivasi digunakan untuk menentukan keluaran neuron

1.1.1 Arsitektur Jaringan Saraf Tiruan

Berdasarkan informasi yang diperoleh dari analytixlabs.co.in yang diakses pada Agustus 2023 arsitektur jaringan merupakan salah satu karakteristik dari



JST yang membahas pola hubungan antar neuron. Neuron-neuron terkumpul pada lapisan-lapisan. Lapisan pada jaringan saraf tiruan meliputi tiga *layer* inti antara lain *input layer* (lapisan masukan), *hidden layer* (lapisan tersembunyi) serta *output layer* (lapisan keluaran).

Input layer merupakan tempat data mentah diterima, pada *hidden layer* data diproses sejumlah neuron nonlinier dan fungsi aktivasi, dalam setiap jaringan saraf simpul dan neuron melaksanakan tanggung jawabnya berdasarkan fungsi aktivasi, bobot, dan bias spesifiknya. Pada kebanyakan kasus fungsi aktivasi diterapkan pada lapisan tersembunyi, sedangkan pada *output layer* menghasilkan akhir dari proses perhitungan, prediksi maupun klasifikasi berdasarkan data masukan dan hasil yang diproses lewat *hidden layer*.

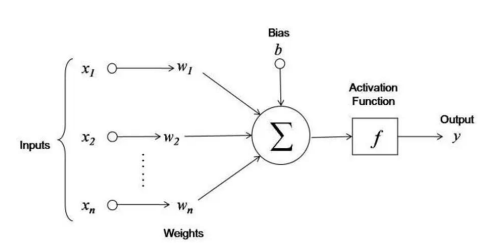
1.1.2 Fungsi Aktivasi

Node adalah tempat input sinyal (nilai input masukan). Dalam jaringan saraf tiruan fungsi aktivasi bertugas menerima masukan yang didapat oleh neuron kemudian memprosesnya untuk mendapat keluaran yang diinginkan. Neuron berisi adder dan fungsi aktivasi. Fungsi ini menentukan apakah neuron harus diaktifkan atau tidak,



berdasarkan dari input yang diterima. Fungsi aktivasi digunakan untuk membatasi keluaran dari neuron agar sesuai dengan batasan sinyal atau nilai keluaran yang ditetapkan (AnalitixLabs, 2023).

Peran fungsi aktivasi dalam jaringan saraf dapat dilihat dari gambar dibawah ini



Gambar 2.3.2.1 peran fungsi aktivasi

Menurut Riadi (2021) dari gambar tersebut ditampilkan bahwa y menerima masukan dari neuron x_1, x_2, x_n , dengan bobo hubungan masing-masing adalah w_1, w_2 , dan w_n . Ketiga impuls neuron yang ada dijumlahkan menjadi

$$\text{Net} : x_1w_1 + x_2w_2 + x_nw_n$$

Besar impuls yang diterima oleh y mengikuti fungsi aktivasi $y = f(\text{net})$. Jika nilai fungsi cukup kuat, maka sinyal akan diteruskan. Nilai fungsi aktivasi (keluaran model jaringan) dapat pula dipakai sebagai dasar untuk mengubah bobot.

Keterangan :



Menurut Eko Prasetyo (2014) secara umum, terdapat empat jenis fungsi aktivasi yang sering digunakan pada berbagai metode JST :

1. Fungsi Aktivasi Linear

Fungsi aktivasi ini biasa digunakan untuk keluaran JST dimana nilai keluarannya diskret. Jika v merupakan nilai gabungan dari semua *vector* oleh adder, maka sinyal keluaran y dihasilkan dengan memberikan nilai v apa adanya agar menjadi nilai keluaran. Tidak ada perhitungan apapun yang dilakukan pada nilai v yang didapat, sehingga sering kali dijumpai metode JST yang menggunakan fungsi aktivasi ini dianggap tidak menggunakan fungsi aktivasi.

Nilai y sebagai nilai output, dituliskan dalam persamaan berikut :

$$y = \text{sign}(v) = v$$

2. Fungsi Aktivasi Step

Jika v adalah nilai gabungan dari semua *vector* oleh adder, maka keluaran y didapatkan dengan memberikan ambang batas pada nilai v berdasarkan nilai T yang diberikan.

Nilai y dituliskan dalam persamaan berikut :

$$y = \text{sign}(v) = \begin{cases} 1 \\ -1 \end{cases}$$



1 jika $v \geq T$

-1 jika $v < T$

Bentuk diatas disebut ambang batas bipolar ada juga yang disebut ambang batas biner

$$y = \text{sign}(v) = \begin{cases} 1 \\ 0 \end{cases}$$

1 jika $v \geq T$

0 jika $v < T$

3. Fungsi Aktivasi Sigmoid Biner

Digunakan untuk JST dengan nilai *output* *kontinu* biasanya menggunakan fungsi kurva sigmoid dengan interval nilai *output* mulai 0 sampai 1.

Nilai y dituliskan sebagai berikut :

$$y = \text{sign}(v) = \frac{1}{1+e^{-av}}$$

4. Fungsi Aktivasi Sigmoid Bipolar

Fungsi ini hampir sama dengan sigmoid biner, namun batas nilai *output* yang diberikan hanya -1 sampai 1. Fungsi aktivasi ini baik digunakan untuk proses klasifikasi karena lebih kuat dalam menangani data-data yang banyak didominasi oleh nilai 0.

Nilai y dituliskan sebagai berikut :

$$y = \text{sign}(v) = \frac{2}{1+e^{-av}} - 1$$



nilai parameter α yang digunakan umumnya 1
oleh karena itu penulisan berubah menjadi :

$$y = \text{sign}(v) = \frac{2}{1+e^{-v}} - 1$$

2.5 Backpropagation

Menurut Sitepu (2021) *Backpropagation* adalah metode penurunan gradien untuk meminimalkan kuadrat *error* pengeluaran. *Backpropagation* merupakan suatu metode dari Jaringan Saraf Tiruan dengan memanfaatkan jaringan multilayer yang memiliki kemampuan untuk mengurangi *error*, sehingga menghasilkan tingkat akurasi yang tinggi. Modifikasi hubungan dalam Jaringan Saraf Tiruan dilakukan dengan tahapan terbalik dari lapisan *output* ke lapisan *input* kemudian ke lapisan tersembunyi (*hidden layer*) oleh sebab itulah metode ini disebut *backpropagation* atau propagasi balik.

Algoritma ini merupakan suatu teknik pembelajaran (*supervised learning*) atau pelatihan yang digunakan pada jaringan *multilayer* untuk mengubah bobot-bobot yang terkoneksi dengan neuron yang terdapat pada layar tersembunyi, kemudian terhubung pula dengan setiap unit yang ada pada layar keluaran. Bobot hubungan dalam jaringan diubah pada setiap tahapan proses dengan tujuan meminimalkan nilai *Mean Square Error* (MSE) antara nilai



prediksi jaringan dengan nilai sebenarnya. Untuk mengubah nilai bobot pada arah mundur dalam algoritma *backpropagation* dilakukan dengan hitungan *error* keluaran. *Error* tersebut didapatkan setelah dilakukan fase *forward propagation* (propagasi maju). Pada tahapan propagasi maju, fungsi aktivasi sigmoid berperan mengaktifasi neuron sehingga mampu menurunkan gradien serta meminimalkan penjumlahan *error* kuadrat dalam *output* JST Puspitaningrum (2006) dalam Naufal (2022)

Menurut Chopra (2019) jika nilai *output* mencapai *error* tertentu maka nilai tersebut diterima. Namun jika nilai *ouput* melebihi *error*, maka unit *neuron* akan dikembalikan pada layar *input* untuk diproses kembali pada nilai bobot dan biasanya. Proses tersebut akan terus berulang hingga seluruh kendala terpenuhi dan telah didapatkan nilai keluaran yang sesuai.

Algoritma Backpropagation

Menurut Hasun, S. (2013), berikut merupakan langkah-langkah kerja *Backpropagation* secara rinci:

Langkah 1 : Inisialisasi bobot-bobot, konstanta laju *training* (α) toleransi *error* atau nilai bobot (jika menggunakan nilai bobot



sebagai kondisi berhenti) atau set maksimal *epoch* (bila *epoch* banyak digunakan sebagai kondisi berhenti)

Langkah 2 ; Apabila kondisi belum dicapai, maka lakukan langkah ketiga hingga langkah kesepuluh

Langkah 3 : Untuk setiap pasangan pola *training*, lakukan langkah ke empat hingga kesembilan.

Secara umum algoritma *backpropagation* terbagi dalam tiga tahapan, di antaranya :

1. *Feedforward Propagation* (Propagasi Maju)

Langkah 4 : Masing-masing *input*

($X_i, i = 1, 2, 3, \dots, n$) menerima sinyal X_i , kemudian melanjutkan ke semua unit pada lapisan tersembunyi.

Langkah 5 : Pada masing-masing lapisan tersembunyi (dari unit ke 1 hingga unit ke n) dikalikan dengan bobotnya dan dijumlahkan serta ditambahkan biasnya

$$z_{net_j} = v_{0j} + \sum_{i=1}^n X_i V_{ij}$$
$$z_j = f(z_{net_j}) = \frac{1}{1 + e^{-z_{net_j}}}$$



Langkah 6 : Masing-masing unit *output*

$(Y_k, k = 1, 2, 3, \dots, m)$ dikalikan dengan bobot dan dijumlahkan serta ditambahkan biasanya

$$y_{net_j} = w_{0j} + \sum_{j=1}^p z_j k_j$$
$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}}$$

Pada tahapan ini data *input* dikirimkan melalui jaringan kemudian menghasilkan *output*. Setelah itu *output* yang dihasilkan akan dibandingkan dengan target yang akan dicapai. Selisih antara target dan *output* merupakan sebuah kesalahan yang terjadi dalam proses. Apabila proses tersebut kesalahan lebih kecil dari batas maksimum kesalahan yang ditentukan, maka iterasi dihentikan. Namun apabila kesalahan lebih besar dari batas maksimum kesalahan yang telah ditentukan, maka bobot setiap garis dalam jaringan akan diatur ulang untuk mengurangi kesalahan yang terjadi. Selama proses tersebut setiap neuron menghitung *output* berdasarkan bobot yang ada dan fungsi aktivasi yang digunakan.

2. *Backward Propagation* (Propagasi Mundur)

Langkah 7 : Pada masing-masing unit *output*



$(Y_k, k = 1, 2, 3, \dots, m)$ menerima target sesuai input training, selanjutnya menghitung nilai *error*. *Output* dikirim ke lapisan bawahnya dan digunakan untuk menghitung besarnya koreksi bobot dan bias (Δw_{jk} dan Δw_{0k}) antara lapisan tersembunyi dengan lapisan *output*.

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

f' = merupakan turunan dari fungsi aktivasi

Untuk menghitung koreksi bobot menggunakan persamaan :

$$\Delta w_{jk} = \alpha \delta_k z_j;$$

$$k = 1, 2, 3, \dots, m ; j = 0, 1, \dots, n$$

untuk menghitung koreksi bias menggunakan persamaan :

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

Langkah 8 : Pada masing-masing unit tersembunyi ($Z_j, j = 1, 2, 3, \dots, p$) dilakukan perhitungan informasi error lapisan tersembunyi, kemudian digunakan untuk menghitung besar koreksi bobot dan bias ΔV_{ji} dan ΔV_{0j} antara lapisan *input* dan lapisan tersembunyi.



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

Untuk menghitung informasi error, dengan mengalikan nilai dengan turunan dari fungsi aktivasinya, yaitu dengan persamaan :

$$\begin{aligned}\delta_j &= \delta_{net} f'(\delta_{netj}) \\ &= \delta_{netj} z_j (1 - z_j)\end{aligned}$$

Hitung koreksi bobot menggunakan persamaan :

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

Hitung koreksi bias menggunakan persamaan :

$$\Delta V_{0j} = \alpha \delta_j$$

Dalam tahapan ini setelah *output* dihasilkan, kesalahan dihitung dengan membandingkan *output* yang dihasilkan dengan *output* yang diinginkan. Kemudian gradien dan fungsi kerugian menggunakan aturan rantai untuk menentukan bagaimana bobot harus disesuaikan. Bobot diperbarui berdasarkan gradien untuk meminimalkan kesalahan.

3. Perubahan Bobot

Langkah 9 : Pada masing-masing unit *output* ($Y_k, k = 1, 2, 3, \dots, m$) dilakukan perubahan pada bobot dan bias ($j = 0, 1, 2, 3, \dots, p$) sehingga menghasilkan bobot dan bias baru :

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



Pada masing-masing unit tersembunyi ($z_j, j = 1, 2, 3 \dots p$)

dilakukan perubahan bobot dan bias ($i = 0, 1, 2, \dots n$) :

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

Langkah 10 : Tes kondisi berhenti.

Keterangan rumus :

X_i : Unit *input*

v_{ij} : Bobot unit *input* terhadap unit tersembunyi

z_j : Keluaran pada unit tersembunyi

z_{net_j} : Faktor keluaran pada unit tersembunyi

Y_k : Keluaran pada unit *output*

w_{jk} : Bobot unit tersembunyi terhadap unit *output*

y_{net_k} : Faktor keluaran pada unit *output*

V_{0j} : Bobot bias pada unit tersembunyi

w_{0k} : Bobot bias pada unit *output*

δ_j : Faktor kesalahan pada lapisan tersembunyi

δ_k : Faktor kesalahan pada lapisan *output*

α : Laju pembelajaran (Learning rate)

Δw_{jk} : Suku perubahan bobot

Menurut Kusumadewi (2004) dalam Naufal (2022), setelah terhitungnya faktor *error*, maka bobot pada semua garis akan diubah secara bersama-sama.



Perubahan bobot suatu garis berdasarkan atas faktor *error* dari unit di layar atasnya. Penyesuaian bobot koneksi dilakukan untuk meminimalisir *error* yang terjadi. Untuk meminimumkan kesalahan atau *error* dapat diukur dengan *Mean Square Error* (MSE), *Root Mean Square Error* (RMSE) dan *Mean Absolute Percentage Error* (MAPE) supaya didapatkan suatu gambaran

a. *Mean Square Error* (MSE)

Menurut Sara (2019) *Mean Square Error* (MSE) ialah metode lain untuk mengevaluasi model prediksi. Masing-masing sisa atau kesalahan dikuadratkan. MSE merupakan cara untuk mengukur kesalahan prediksi keseluruhan. MSE adalah rata-rata selisih kuadrat antara nilai prediksi dan nilai yang diamati (aktual). Kekurangan MSE yaitu bahwa MSE cenderung menonjolkan deviasi besar karena adanya pengkuadratan. Rumus untuk menghitung MSE sebagai berikut :

$$MSE = \frac{\sum_{t=1}^n (X_t - F_t)^2}{n}$$

b. *Root Mean Square Error* (RMSE)



$$RMSE = \sqrt{\frac{\sum (aktual - prediksi)^2}{n}}$$

Menurut Wijaya (2023) *Root Mean Square Error* (RMSE) merupakan penjumlahan kuadrat *error* atau selisih antara nilai aktual dengan nilai prediksi, lalu membagi jumlah tersebut dengan banyaknya waktu data prediksi dan selanjutnya menarik akarnya. Jika nilai RMSE semakin kecil maka estimasi model atau *variable* tersebut semakin baik. RMSE adalah akar kuadrat dari MSE, biasanya digunakan untuk mengukur deviasi standar kesalahan.

c. *Mean Absolute Percentage Error (MAPE)*

Menurut Hauriza,B, dkk (2021) *Mean Absolute Percentage Error* (MAPE) adalah perhitungan yang menggunakan kesalahan absolut pada setiap periode dibagi dengan nilai yang diamati yang terbukti untuk periode tersebut. Lalu, hitung rata-rata persentase tetap tersebut. Pendekatan ini berguna ketika ukuran atau ukuran *variable* prediksi signifikan dalam mengevaluasi kekuatan prediksi. MAPE menunjukkan seberapa besar kesalahan dalam memprediksi dibandingkan dengan nilai sebenarnya.

$$MAPE = \left(\frac{1}{n} \right) \sum_{t=1}^n \left| \frac{X_t - F_t}{X_t} \right|$$

Keterangan ;



- n = jumlah sampel
 X_t = nilai aktual indeks pada 26th statistic
 F_t = nilai prediksi indeks pada 26th statistic

Kemampuan prediksi sangat baik jika memiliki nilai MAPE kurang dari 10% dan memiliki prediksi yang baik jika nilai MAPE kurang dari 20%.

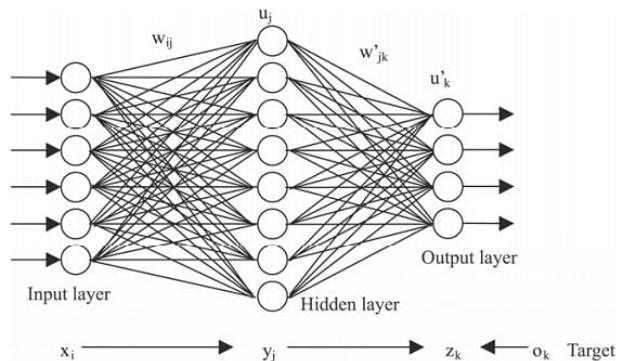
2.6 Rapidminer

Menurut Cahyati (2024) *Rapidminer* berkembang sejak tahun 2001, sebelumnya dikenal dengan nama YALE (*Yet Another Learning Environment*). *Software* ini dikembangkan oleh Ralf Klinkenberg, Ingo Mierswa, serta Simon Fischer pada *Unit Artificial Intelligence* dari *Technical University of Dortmund*. *Rapidminer* adalah platform perangkat lunak yang kuat untuk ilmu data dan pembelajaran mesin yang bersifat terbuka (*open source*). *Rapidminer* memberikan berbagai solusi untuk melakukan analisis terhadap *data mining*, *text mining*, dan analisis prediksi karena memiliki beberapa alat yang digunakan untuk mempersiapkan data, memodelkan, mengevaluasi dan mengimplementasi, sehingga memungkinkan pengguna untuk membangun dan menguji berbagai model walaupun tanpa pengalaman pemrograman (Marwati, 2024).



Karena kemudahan akses yang diberikan oleh *Rapidminer*, maka sering dijumpai beberapa penelitian yang menggunakan *software Rapidminer* salah satunya adalah penelitian yang dilakukan oleh Fingki Marwati dan Rizky Fauzi (2024). Dalam penelitiannya *Rapidminer* digunakan untuk memprediksi penyakit diabetes melitus dengan menggunakan metode *Backpropagation*.

Algoritma *Backpropagation* dalam *Rapidminer* :



Gambar 2. 1 Algoritma *Backpropagation*

Algoritma *Backpropagation* merupakan algoritma yang sederhana yang dapat bekerja dengan baik bahkan dengan data yang kompleks. *Backpropagation* memiliki sifat komputasi yang baik apalagi bila data yang tersaji dalam skala besar. Algoritma pembelajaran *Backpropagation* sudah dijelaskan pada sub bab 2.4.1.



2.7 Kerangka Pemikiran

Kerangka pemikiran merupakan dasar pemikiran yang dijadikan acuan pada sebuah penelitian, kerangka pemikiran dapat berupa bagan atau deskripsi yang merangkum garis besar dalam penelitian.

Masalah		
Memprediksi jumlah pelanggan PT. Indah Logistik Cargo menggunakan metode JST <i>Backpropagation</i>		
Tujuan		
Mengetahui tingkat akurasi metode <i>Backpropagation</i> dalam memprediksi jumlah pelanggan PT. Indah Logistik Cargo		
Penelitian		
Software	Data	Metode
<i>Rapidminer</i>	Jumlah pelanggan per hari Agen Indah Logistik Cargo Ngoro Jombang dari 24 April 2025 – 31 Mei 2025	<i>Backpropagation</i>
Hasil yang ingin dicapai		
Tingkat akurasi tersebut diharapkan mampu memprediksi jumlah pelanggan Indah Logistik Cargo		
Manfaat		



Dapat digunakan sebagai informasi tambahan dan
acuan dalam pengambilan keputusan