BAB II

LANDASAN TEORI

2.1. Studi Literatur

Penggunaan ANN dalam memprediksi cuaca telah banyak dilakukan dalam jangka waktu 5 tahun terakhir, beberapa contohnya dapat dilihat pada studi literatur yang ditampilkan pada tabel 1. Tabel 1 menampilkan berbagai penelitian dengan atribut yang berbeda-beda, serta dataset dan algoritma yang juga berbeda, demi mendapatkan tujuan yang sama, yaitu melakukan prakiraan cuaca.

Pada [6]–[9], peneliti menggunakan ANN untuk memprediksi unsur-unsur cuaca beberapa waktu ke depan, bukan intensitas curah hujannya. Baru pada [2], [6], [10]–[14], peneliti menggunakan beberapa unsur cuaca untuk memprediksi curah hujan. Meskipun begitu, pada penelitian prediksi curah hujan yang dilakukan adalah curah hujan bulanan dan curah hujan harian, sehingga prediksinya menjadi lebih mudah karena memiliki dinamika cuaca yang lebih konsisten. Berbeda dengan prediksi hujan per jam, yang memiliki dinamika atmofer yang lebih kompleks, sehingga lebih sulit untuk diprediksi.

Tabel 2.1 Penelitian terkait tentang prakiraan cuaca dengan menggunakan ANN

Judul, Penulis, Tahun	Jumlah Atribut	Algoritma	Pre-processing	Dataset	Akurasi
Backpropagation and Radial Basis FunctionMethods for Predicting Rainfall in Sukabumi CityUsing Artificial Neural Networks: A ComparativeAnalysis [10]	1 (Curah Hujan)	Backpropagation Neural Network	Normalisasi	https://sukabumikota.bps.go.id/	MSE BPNN < MSE RBFNN
Analisis Performansi Prakiraan Cuaca Menggunakan Algoritma Machine Learning [11]	7 (Suhu udara, tekanan, kelembaban, tutupan awan, kecepatan angin, presipitasi, klasifikasi data hujan)	Backpropagation	Akuisisi Data	BMKG Makassar	Akurasi: Hujan (68.33%) dan kemarau (87.57%). MSE: Hujan (0.2646) dan Kemarau (0.1248)

Prediksi Tinggi Curah Hujan dan Kecepatan Angin BerdasarkanData Cuaca Dengan Penerapan Algoritma Artiicual Neural Network [12]	6 (Curah Hujan, Penyinaran Matahari, Suhu Maksimum, Suhu rata-rata, Kelembaban, dan Kecepatan Angin)	Backpropagataion	Normalisasi	BMKG	RMSE: 0.079535
Weather Forecasting Using Artificial Neural Network [6]	3 (Kelembaban, Suhu udara, dan tekanan)	Artificial Neural Network	-	Indian Meteorological Center	MSE = 0.9325 - 3.5321
A Deep Learning Approach to Predict Weather Data Using Cascaded LSTM Network [7]	4 (Kecepatan Angin, Kelembaban, Titik Embun, dan Suhu Udara)	LSTM (Long Short Term Memory)	Moving Average	https://saskatoon.weatherstats.ca	MSE = 0.0003 - 0.003

Weather Forecasting using Machine Learning Techniques [8]	7(Tekanan, Suhu Udara, Kelembaban, Arah Angin, Tutupan Awan, Jarak Pandang, dan Titik Embun)	Support Vector Machine , Artificial Neural Netowk, dan SVM, ANN, dan Reccurent Neural Network	Normalisasi	Indian Meteorological Center	RMSE = 1.4 - 6.7
Weather Forecasting System with the use of Neural Network and Backpropagation Algorithm [9]	5 (Suhu Udara, Titik Embun, Kelembaban, Tekanan, dan Curah Hujan)	Artifical Neural Network dan Backpropagation	Normalisasi	Meteorological website: https://www.wunderground.com	Error = 0.07 dan Effisiensi = 95.93
Jaringan Saraf Tiruan Backpropagation Untuk Prediksi Curah Hujan di Wilayah Kabupaten Wonosobo [13]	1 (Curah Hujan)	Backpropagataion	Normlisasi	BMKG Wonosobo	MSE: 0.00099899

Rainfall prediction with backpropagation method [2]	5 (Suhu Udara, Kelembaban, Durasi Penyinaran Matahari, Kecepatan Angin, dan Curah Hujan)	Backpropagation	Normalisasi	Japan Meteorological Agency Website	MSE = 0.3416
Application Of Backpropagation Neural Networks In Predicting Rainfall Data In Ambon City [14]	3 (Suhu udara, kecepatan angin, dan curah hujan)	Backpropagation	-	BMKG Provinsi Maluku	Akurasi: 80% dan MSE: 0.022

Pemanfaatan Artificial Neural Network dan Teknik Backpropagation Pada Prakiraan Cuaca Jangka Pendek	6 Unsur (Suhu, Kelembaban, Tekanan, Jarak Pandang, Tinggi Awan, dan Cuaca)	Backpropagation	Standarisasi	BMKG Lampung	Prakiraan hujan tanpa klasifikasi: Akurasi:0.44-0.86 Prakiraan hujan terklasifikasi: Akurasi:0.57-0.89

Hampir seluruh jurnal, pada studi literatur, menggunakan normalisasi pada preprocessing. Ini dilakukkan, karena normalisasi tidak merubah nilai dari sebuah data, tetapi membuat model menjadi lebih efektif pada proses train (latih). Hanya satu jurnal yang menggunakan *moving average* [7], yang digunakan untuk proses *cleaning data*. Data yang digunakan pada [6], [8], [10]–[14] menggunakan data pengamatan langsung,[2], [7], [9] sedangkan data yang digunakan pada adalah data dari model cuaca. Berdasarkan studi literatur pada tabel 2.1, juga didapatkan, algoritma JST dengan teknik BPNN dapat memprakirakan cuaca dengan hasil yang baik.

2.2. Prakiraan Cuaca

Cuaca adalah kondisi terkini dari atmosfer pada suatu lokasi. Cuaca bersifat dinamis, yang menyebabkan cuaca dapat berubah-ubah secara cepat, bergantung pada waktu dan lokasinya. Atmosfer bersifat dinamis, sehingga selalu bergerak sepanjang waktu yang dipengaruhi dengan berbagai fenomena yang terjadi secara lokal hingga global [15].

Hujan adalah salah satu bentuk presipitasi (endapan) berupa air yang berasal dari uap air di atmosfer dan turun ke permukaan bumi [16]. Dalam proses pembentukannya, hujan dipengaruhi oleh dipengaruhi beberapa faktor yang terdapat di atsmofer dan saling berkaitan satu sama lain seperti suhu udara, tekanan udara, kelembaban udara, visibility, tutupan awan, dan kecepatan angin [17].

Suhu udara adalah derajat panas dari aktivitas molekul di atmofer, sedangkan kelembaban udara menyatakan banyaknya kandungan uap air di udara. Tekanan udara adalah tekanan pada suatu lokasi yang disebabkan oleh berat dari udara. Perbedaan tekanan udara menyebabkan timbulnya kecepatan angin yang merupakan pergerakan horizontal udara terhadap permukaan bumi pada suatu waktu yang diperoleh. Kelembaban udara mempengaruhi visibilitas atau jarak pandang horizontal dan tutupan awan atau banyaknya jumlah awan yang terbentuk di atmosfer.

Prakiraan dapat didefinisikan sebagai penggunaan data sekarang dan data yang telah lalu untuk memperkirakan kejadian yang akan datang [18]. Sedangkan prakiraan cuaca adalah penggunaan aplikasi sains dan teknologi untuk memprakirakan kondisi atmosfer pada lokasi tertentu *(weather forecasting)*. Prakiraan cuaca ini termasuk curah hujan hujan, suhu dan kelembaban udara, tekanan tutupan awan, dan kecepatan angin.

Prakiraan cuaca akan menurun bersamaan dengan lamanya waktu prakiraan. Berdasarkan lamanya waktu prakiraan, prakiraan cuaca dapat diklasifikasikan sebagai *nowcasting* (0-3 jam), *shortcasting* (3-6 jam), *forecasting* (6-48 jam), dan *longcasting* (dapat mencapai 6 hari ke depan).

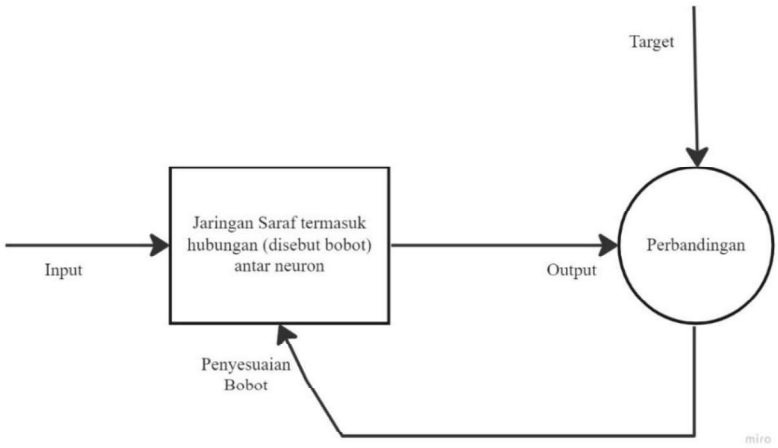
2.3. Jaringan Saraf Tiruan

Secara sederhana, data statistik yang dimodelkan secara non-linear [18]. JST dapat menemukan pola pada data dengan memodelkan hubungan input dan output yang kompleks. Sederhananya, JST adalah tiruan otak manusia yang dapat belajar hal-hal baru dan beradaptasi pada lingkungan yang berubah. Otak dapat membuat keputusan yang didasrkan pada kondisi yang tidak sempurna dan informasi yang tidak jelas [20].

JST merepresentasikan otak manusia dalam cara beikut[21]:

a. JST mendapat pengetahuan melalui pembelajaran
b. Pengetahuan JST disimpan dalam koneksi interneuron yang kuat yang dikenal sebagai sinaptik

JST disusun dari elemen sederhana yang beroperasi secara paralel. JST dilatih untuk membangun suatu fungsi tertentu. Pelatihan dilakukan dengan cara perbaikan nilai-nilai pembobot (weight) ANN hingga output ANN sama dengan target output.



Gambar 2.1 Struktur jaringan saraf tiruan

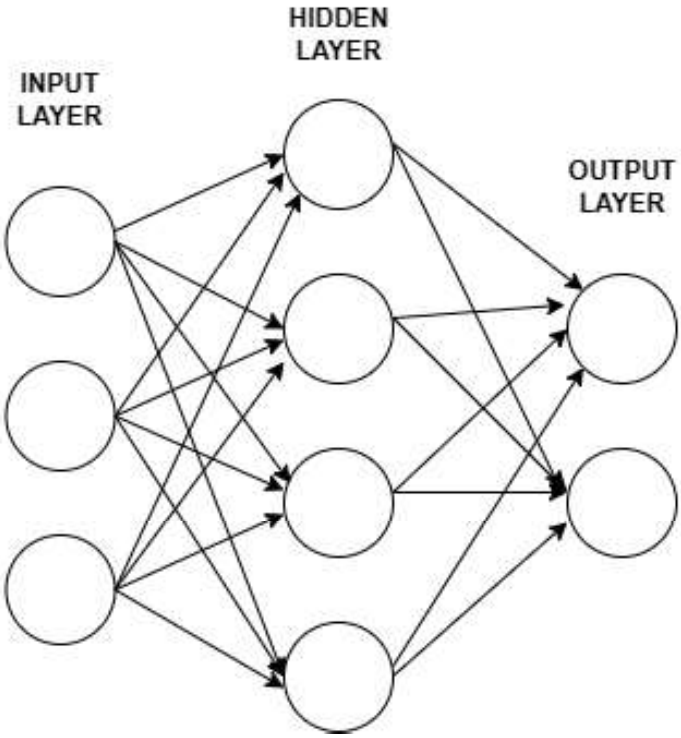
2.3.1. Komponen JST

Seperti halnya otak manusia, jaringan syaraf juga terdiri atas beberapa neuron dan ada hubungan antar neuron tersebut. Neuron-neuron tersebut akan mentransformasikan informasi yang diterima melalui sambungan keluarnya menuju ke neuron-neuron yang lain. Pada jaringan syaraf, hubungan ini dikenal dengan nama bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tersebut.

Neuron ini sebenarnya mirip dengan sel neuron biologis. Neuron-neuron buatan tersebut bekerja dengan cara yang sama pula dengan neuron biologis. Informasi (disebut dengan: input) akan dikirim ke neuron dengan bobot kedatangan tertentu. Input ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan ini kemudian akan dibandingkan dengan suatu nilai ambang(threshold) tertentu melalui fungsi aktivasi setiap neuron.

Apabila input tersebut melewati suatu nilai ambang tertentu, maka neuron tersebut akan diaktifkan, tapi kalau tidak, maka neuron tersebut tidak akan diaktifkan.

Apabila neuron tersebut diaktifkan, maka neuron tersebut akan mengirimkan output melalui bobot-bobot outputnya kesemua neuron yang berhubungan dengannnya.



Gambar 2.2 Komponen Jaringan Saraf Tiruan

Pada Jaringan syaraf tiruan, neuron-neuron akan dikumpulkan dalam lapisan (layer) yang disebut dengan lapisan neuron (neuron layer). Neuron-neuron pada satu lapisan akan dihubungkan dengan lapisan-lapisan sebelum dan sesudahnya (kecuali lapisan input dan lapisan output). Informasi yang diberikan pada jaringan syaraf akan dirambatkan lapisan ke lapisan. Mulai dari lapisan input sampai ke lapisan output melalui lapisan lainnya, yang sering disebut sebagai lapisan tersembunyi (hidden layer) [22].

2.3.2. Standarisasi Data

Pada penelitian ini pre-processing dikaukan dengan menstandarisasi data, yang memastikan data tetap konsisten. Standarisasi dapat membuat algorimta lebih

efektif dan effisien pada data beukuran besar dengan menghilangkan redundasi dan menstandarisasi informasi. Persamaan dari standarisasi:

$$s = \frac{x - \mu}{\sigma} \tag{1}$$

dimana x adalah nilai input, z adalah hasil standarisasi, μ adalah rata-ratanya, dan σ adalah standar deviasi.

2.3.3. Fungsi Aktivasi

Ada beberapa fungsi aktivasi yang sering digunakan dalam jaringan syaraf tiruan, antara lain [23]:

1. Fungsi ReLU (Rectified Linear Unit)

ReLU adalah fungsi aktivasi non-linear yang digunakan pada jaringan neuron *multi-layer* atau jaringan neural dalam (*deep neural network*). Fungsi ini dituliskan sebagai berikut:

$$f(x) = \max(0, x) \tag{2}$$

dimana x adalah nilai input. Berdasarkan persamaan (2), output dari ReLU adalah nilai maksimum antara 0 dan nilai input. Sehingga,

$$f(x) = \begin{cases} 0, jika\ x < 0 \\ x, jika\ x \geq 0 \end{cases} \tag{3}$$

2. Fungsi Log Sigmoid

Fungsi sigmoid adalah fungsi matematika yang memiliki karakteristik kurva berbentuk “S”, yang mengubah nilai antara rentang 0 dan 1. Fungsi sigmoid disebut juga dengan kurva sigmoidal atau fungsi logistik. Ini adalah salah satu fungsi aktivasi non-linier yang paling banyak digunakan.

$$z = f(a) = \frac{1}{1 + e^{-a}} \quad (4)$$

dimana:

$f(a)$ adalah fungsi dari a atau fungsi dari nilai net input

e^{-a} adalah eksponen dari $-a$

2.3.4. Algoritma Backpropagation

JST mendukung berbagai tipe pembelajaran atau algorimta latih, yang salah satunya adalah teknik *Back-Propagation Neuran Network* (BPN) [24]. Keuntungan utama dari metode BPN dapat mendekati fungsi yang memiliki banyak kelas. Metode ini lebih efisien dibandingkan dengan differensiasi numerik. Beberapa analisis menunjukkan bahwa penggunaan prakiraan cuaca dengan JST memiliki efisiensi yang lebih baik jika dibantu dengan teknik BPN.

BPN terdiri atas setidaknya tiga layer, yaitu input layer, paling sedikit satu hidden layer, dan terakhir output layer [25]. Proses data prakiraan cuaca secara real-time menunjukkan bahwa prakiraan cuaca yang berdasarkan pada teknik BPN meningkatkan bukan hanya akurasi dari prakiraan dengan model, tetapi juga prakiraan cuaca lokal dalam lokasi yang lebih sempit [21].

Algoritma pembelajaran BPN dapat dibagi ke dalam dua fase, yaitu propagasi dan pembaruan nilai bobot [26]. Pada setiap fase propagasi mengikuti langkah-langkah berikut:

1. Propagasi maju dari input pola pelatihan diberikan melalui jaringan saraf untuk menghasilkan output
2. *Back Propagation* dari output aktivasi propagasi melalui JST menggunakan target pola pelatihan untuk menghasilkan delta atau selisih untuk semua output dan neuron lainnya

Pada setiap update nilai bobot dilakukan langkah:

1. Mengkalikan input aktivasinya dengan delta output untuk mendapatkan gradien nilai bobot
2. Tambahkan nilai gradien ke dalam nilai bobot, untuk memberbarui nilai bobotnya.

Rasio gradien bobot berpengaruh pada kecepatan dan kualitas latih, yang disebut kecepatan latih. Tanda gradien bobot menunjukkan kesalahan yang meningkat, dimana semakin besar nilai gradien, maka semakin besar nilai kesalahan. Fase 1 dan 2 terus diulang sampai mendapatkan kinerja JST dan akurasi yang memuaskan.

Algoritma *backpropagation* dapat ditampilkan pada langkah berikut:

1. Inisiasi semua bobot dengan nilai bilangan acak kecil
2. Jika kondisi penghentian belum terpenuhi, maka lakukan langkah 2-9
3. Untuk setiap data latih, lakukan langkah 3-8
4. Tiap unit input menerima sinyal dan meneruskan ke hidden layer di atasnya
5. Hitung semua unit output di hidden layer $z_j (j = 1,2, \dots, p)$

$$a_j = b + \sum_{i=1}^{n} x_i v_{ji} \qquad (5.a)$$

$$z_j = f(a_j) = \max(0, a_j) \qquad (5.b)$$

Dimana a adalah net input, b adalah bias dari input, x_i adalah input, v_{ji} adalah bobot dari input, dan $z_j$adalah hasil dari hidden layer.

6. Hitung semua output hasil dari setiap unit $y_k (k = 1,2, \dots, m)$

$$y_{net_k} = c + \sum_{j=1}^{p} z_j w_{kj} \qquad (6.a)$$

$$yk = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}} \qquad (6.b)$$

Dimana y_{net_k} net output dari hidden layer, c adalah bias dari dari hidden layer, w_{kj} adalah bobot dari hidden layer, dan y_k adalah hasil output akhir.

7. Hitung faktor δ unit keluaran berdasarkan error dari setiap unit $y_k(k = 1,2, \dots m)$

$$\delta_k = (t_k - y_k)f'(y_{net_k}) = (t_k - y_k)y_k(1 - y_k) \quad (7)$$

δ_k merupakan error yang akan dipakai dalam perubahan bobot layer di bawahnya dan t_k adalah hasil sebenarnya, pada data latih. Hitung suku perubahan bobot ΔW_{kj} (yang akan dipakai nanti untuk merubah suatu bobot W_{kj}) dengan laju perubahan α

$$\Delta W_{kj} = \alpha\delta_k z_j, \qquad k = 1,2, \dots . . m, \quad j = 0,1, \dots, p \quad (8)$$

8. Hitung faktor δ pada hidden layer berdasarkan kesalahan di setiap hidden layer $z_j(j = 1,2, \dots . p)$

$$\delta_{net_j} = \sum_{k=1}^{m} \delta_k W_{kj} \quad (9)$$

Faktor δ_k dari hidden layer:

$$\delta_j = \delta_{net_j} f'(a_j) = \delta_{net_j} z_j(1 - z_j) \quad (10)$$

Hitung suku perubahan bobot V_{ji}(yang nanti akan dipakai untuk merubah bobot v_{ji})

$$\Delta V_{ji} = \alpha\delta_j x_i, \qquad j = 1,2, \dots , \qquad i = 0,1, . . , n \quad (11)$$

9. Hitung semua perubahan bobot. Perubahan bobot dari hidden layer ke output:

$$W_{kj}(baru) = W_{kj}(lama) + \Delta W_{kj},$$
$$(k = 1,2, . . m \; dan \; j = 0,1, . . p) \quad (12)$$
$$V_{ji}(baru) = V_{ji}(lama) + \Delta V_{ji},$$
$$(j = 1,2, . . p \; dan \; i = 0,1, \dots . n) \quad (13)$$

2.4. Evaluasi

2.4.1. Confussion Matrix

Untuk mengevaluasi kinerja dari suatu model dalam memprediksi sebuah data yang berbentuk katagori, seperti menentukan hujan atau tidaknya. Maka diperlukan

matrix konfusi (*confussion matrix*). Matrix konfusi ini digunakan untuk melihat seberapa besar prediksi benar yang sebenarnya atau salah yang sesalahnya [27].

Penggambaran dari matrik konfusi dapat dilihat pada tabel 2.2, dimana nilai *True Positive* (TP) akan menggabarkan nilai yang diprediksi hujan dan keadaan sebenarnya juga hujan. Nilai *False Positive* (FP) akan menunjukkan nilai yang diprediksi hujan sedangkan keadaan sebenarnya tidak hujan. Nilai *False Negative* (FN) akan menunjukkan nilai yang diprediksi tidak hujan sedangkan keadaan sebenarnya hujan. Sedangkan nilai *True Negative* (TN) akan menunjukkan salah yang sesalah-salahnya.

Tabel 2.2 Perbandingan Matrix Konfusi

		Keadaan Sebenarnya	
		Benar	Salah
Model Prediksi	Benar	*True Positive (TP)*	*False Positive* (FP)
	Salah	*False Negative* (FN)	*True Negative* (TN)

Untuk menghitung seberapa besar akurasi dari model dalam memprediksi kondisi sebenarnya maka, digunakan akurasi (ACC),

$$ACC = \frac{TP + TN}{P + N} \tag{14}$$

Selain menggunakan akurasi, untuk mengevaluasi sebuah model juga digunakan recall,

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

Serta presisi,

$$Presisi = \frac{TP}{TP + FP} \tag{16}$$