

BAB II

TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Penelitian yang bertujuan dalam melakukan prediksi terhadap suatu kejadian sudah banyak dilakukan dan sudah banyak diterapkan dalam berbagai bidang. Dalam penelitian yang sudah dilakukan sebelumnya menunjukkan bahwa memprediksi suatu kejadian atau permasalahan sangat diperlukan untuk dijadikan perencanaan maupun persiapan yang lebih baik dari sebelumnya, serta memudahkan dan membantu dalam pengambilan keputusan untuk menciptakan masa depan yang lebih baik.

Penelitian dengan judul “*Prediction of Seismic Slope Stability Through Combination of Particle Swarm Optimization*” dilakukan oleh Behrouz Gordan, Danial Jahed Armaghani, Mohsen Hajihassani, dan Masoud Monjezi pada tahun 2015. Penelitian tersebut digunakan untuk memprediksi stabilitas lereng seismik ketika terjadi gempa. Dua metode yaitu Jaringan Saraf Tiruan (JST) *Backpropagation* dan kombinasi JST dengan *Particle Swarm Optimization* (BP-PSO) dirancang untuk memprediksi faktor keamanan pada lereng homogeneous. Dari penelitian tersebut ditemukan bahwa metode kombinasi BP-PSO memiliki performa yang lebih baik dibandingkan dengan metode JST *Backpropagation*. Nilai R² (koefisien determinasi) dari pengujian dataset sebesar 0,915 didapat dari metode JST *Backpropagation* dan 0,986 dari metode kombinasi BP-PSO [18].

Penelitian yang berjudul “Particle Swarm Optimization Sebagai Penentu Nilai Bobot pada Artificial Neural Network Berbasis Backpropagation untuk Prediksi Tingkat Penjualan Minyak Pelumas Pertamina” yang dilakukan oleh Muhammad Ali Ridla membuktikan bahwa algoritma JST memiliki kemampuan yang baik dalam melakukan prediksi. Akan tetapi terdapat kelemahan yaitu tingkat konvergensi yang lambat, yang disebabkan bobot awal yang dipilih secara random. Maka dari itu dilakukan optimasi menggunakan PSO. Dengan menggunakan PSO

didapatkan akurasi yang lebih tinggi dengan hasil RMSE sebesar 0,059 sedangkan untuk JST *backpropagation* tanpa optimasi dengan RMSE 0,111 [13].

Penelitian berikutnya yang berjudul “Aplikasi Jaringan Saraf Tiruan dan Particle Swarm Optimization untuk Peramalan Indeks Harga Saham Bursa Efek Indonesia” yang dilakukan oleh Desy Wartati dan Nur Aini Masruroh didapatkan hasil bahwa model yang dibuat dengan kombinasi algoritma *backpropagation* dan PSO menghasilkan waktu komputasi, *error* model, dan *error* peramalan yang lebih kecil daripada model *backpropagation*. Dilihat dari nilai MAPE, peramalan yang dilakukan dengan kedua model tersebut termasuk *highly accurate* karena MAPE yang dihasilkan lebih kecil dari 10%, yaitu 2,55% untuk *backpropagation* dan 1,88% untuk *backpropagation* dengan PSO [19].

Penelitian lain yang berjudul “Penerapan Algoritma Backpropagation Untuk Prediksi Kebutuhan Air Bersih pada PDAM Intan Banjar” yang dilakukan oleh Bahar dan Surya Ade Yahya pada tahun 2019 ini bertujuan untuk memprediksi kebutuhan air bersih pada PDAM Intan Banjar. Pada penelitian tersebut menggunakan algoritma *backpropagation* yang bisa digunakan untuk memprediksi data *time series*. Penelitian tersebut menghasilkan prediksi dengan RMSE beragam tergantung jumlah data dan struktur JST *backpropagation* yang digunakan, yaitu 0,1600 pada bulan Januari, 0,1048 pada bulan Februari dan 0,0877 pada bulan Maret 2015. Meskipun selisih yang didapat masih cukup besar dibanding perkiraan manual. Yang terkecil mencapai 56.282 dan yang terbesar mencapai 197.326 [2].

Penelitian selanjutnya yang berjudul “Peramalan Jumlah Penjualan Menggunakan Jaringan Saraf Tiruan Backpropagation pada Perusahaan Air Minum Dalam Kemasan” dilakukan oleh Nur Fitriyaningsih Hasan, Kusri, dan Hanif Al Fatta yang bertujuan untuk memprediksi jumlah penjualan Air Minum dalam Kemasan (AMDK). Berdasarkan pelatihan JST *backpropagation* model dengan *multi-layer* 12 *input layer*, 10 *hidden layer* dan 1 *output layer* diperoleh hasil peramalan penjualan AMDK 600ml untuk periode 2019 memiliki selisih 2% dengan data aktual. Dengan nilai MSE sebesar 0,00043743 dan nilai MAPE 6,88% [7].

Penelitian selanjutnya adalah penelitian yang dilakukan oleh Nabilla Putri Sakinah, Imam Cholissodin, dan Agus Wahyu Widodo dengan judul “Prediksi Jumlah Permintaan Koran Menggunakan Metode Jaringan Saraf Tiruan Backpropagation” pada tahun 2018. Data yang digunakan merupakan data jumlah penjualan koran Radar Madura tahun 2015. Berdasarkan hasil pengujian, didapatkan jumlah iterasi terbaik sebesar 200 dan *learning rate* terbaik sebesar 0,6. Data latih dan data uji menghasilkan nilai *error rate* terkecil yaitu 0,0162 [20].

Ringkasan penelitian yang relevan ditunjukkan pada Tabel 2.1 di bawah.

Tabel 2.1 Penelitian terdahulu

No.	Judul	Comparing	Contrasting	Criticize	Synthesize	Summarize
1	<i>Prediction of Seismic Slope Stability Through Combination of Particle Swarm Optimization</i>	Penelitian dilakukan untuk memprediksi stabilitas lereng seismik dengan membandingkan dua model, yaitu JST <i>Backpropagation</i> dan model kombinasi BP-PSO. Prediksi dilakukan dengan menggunakan variabel independen	Penelitian dilakukan untuk mencari arsitektur terbaik pada kedua model, kemudian dilakukan optimalisasi pada JST <i>backpropagation</i> dengan menggunakan PSO untuk menentukan nilai bobot yang digunakan untuk	Tidak melakukan percobaan pada <i>learning rate</i> dari arsitektur model yang akan diterapkan untuk melakukan prediksi stabilitas lereng seismik.	Penelitian dilakukan dengan melakukan pemodelan prosedur untuk mendapat dataset yang cocok. Kemudian dilakukan perancangan arsitektur dengan performa terbaik dipilih untuk dilakukan pelatihan dan	Penelitian dengan menggunakan kombinasi BP-PSO memiliki performa yang lebih baik. Nilai R2 (koefisien determinasi) yang didapat dari model kombinasi BP-PSO sebesar 0,982 untuk pelatihan dan 0,986 untuk pengujian.

		berupa ketinggian lereng, gradien, koefisien, sudut gesekan dan percepatan puncak tanah. Kemudian dilakukan analisis sensitivitas dalam melakukan pemodelan arsitektur.	memprediksi stabilitas lereng seismik.		pengujian pada semua dataset.	
2	<i>Particle Swarm Optimization</i> Sebagai Penentu Nilai Bobot pada <i>Artificial Neural Network</i> Berbasis <i>Backpropagation</i> untuk Prediksi	Penelitian dilakukan terhadap data penjualan minyak pelumas menggunakan JST <i>backpropagation</i> dan PSO. PSO	Membahas mengenai cara mengoptimalkan JST <i>backpropagation</i> dengan PSO untuk menentukan nilai bobot yang	Dataset yang digunakan tidak disebutkan dalam penelitian. Implementasi yang dilakukan hanya menggunakan	Penelitian dilakukan dengan mengumpulkan data, kemudian dilakukan <i>preprocessing</i> . Setelah itu dilakukan	Penelitian dengan JST <i>backpropagation</i> dan PSO menghasilkan akurasi yang lebih tinggi dari pada JST

	<p>Tingkat Penjualan Minyak Pelumas Pertamina</p>	<p>digunakan untuk menutupi kelemahan JST yaitu pada penentuan nilai bobot terbaik. Implementasi dilakukan pada aplikasi <i>rapidminer</i>.</p>	<p>digunakan untuk memprediksi penjualan minyak pelumas pertamina.</p>	<p><i>rapidminer</i> yang dalam kerjanya terlalu instan.</p>	<p>pelatihan terhadap data menggunakan PSO dan JST <i>backpropagation</i>. Kemudian dilakukan pengujian terhadap data menggunakan <i>model attribute weight</i> PSO dan JST <i>backpropagation</i>. Kemudian dilakukan evaluasi untuk melihat keberhasilan algoritma.</p>	<p><i>backpropagation</i> tanpa menggunakan PSO.</p>
--	---	---	--	--	---	--

3	<p>Aplikasi Jaringan Saraf Tiruan dan <i>Particle Swarm Optimization</i> untuk Peramalan Indeks Harga Saham Bursa Efek Indonesia</p>	<p>Penelitian ini dilakukan terhadap data Indeks Harga Saham Gabungan Maret 2016 sampai Februari 2017 dengan menggunakan JST dan PSO. PSO digunakan untuk mendapatkan bobot dengan waktu komputasi yang cepat dan hasil yang optimal.</p>	<p>Membahas mengenai pengaplikasian JST <i>backpropagation</i> yang dioptimasi menggunakan PSO.</p>	<p>Tidak menyebutkan jumlah data yang digunakan.</p>	<p>Penelitian dilakukan dengan pengumpulan data, kemudian memilih data yang akan digunakan untuk melakukan prediksi. Membagi data menjadi data latih dan data uji. Tahap selanjutnya adalah membangun model JST dan PSO. Model JST <i>backpropagation</i> digunakan untuk pelatihan dan pengujian,</p>	<p>Penelitian dengan JST <i>backpropagation</i> dan PSO, diketahui bahwa model yang dibuat dengan JST <i>backpropagation</i> dan PSO menghasilkan waktu komputasi, <i>error</i> model, dan <i>error</i> peramalan yang lebih kecil daripada model JST <i>backpropagation</i> tanpa PSO.</p>
---	--	---	---	--	--	---

					sedangkan PSO hanya untuk pelatihan. Kemudian dilakukan validasi pada tahap pengujian. Hasil dari peramalan dua model dibandingkan berdasarkan waktu dan nilai <i>error</i> .	
4	Penerapan Algoritma <i>Backpropagation</i> Untuk Prediksi Kebutuhan Air Bersih pada	Penelitian ini dilakukan terhadap data time series yang berisi data pemakaian air, produksi air dan jumlah	Penelitian ini dilakukan untuk memprediksi jumlah produksi air bersih yang tepat, agar mencegah	Tidak ada optimasi pada JST <i>backpropagation</i> dalam penentuan bobot awal, serta tahapan penelitian	Penelitian dilakukan dengan menentukan arsitektur JST <i>backpropagation</i> , kemudian menentukan	Prediksi menghasilkan tingkat RMSE yang beragam tergantung pada jumlah data dan struktur JST

	PDAM Intan Banjar	pelanggan yang didapat Laporan PDAM Intan Banjar.	yang dari Data Intan air bersih yang akan berdampak pada penambahan biaya bahan baku penjernih air.	terjadinya kekurangan ataupun kelebihan air yang akan berdampak pada penambahan biaya bahan baku penjernih air.	yang kurang terperinci.	kebutuhan sistem. Normalisasi data kemudian dilakukan perhitungan. Setelah itu dilakukan pengujian sistem untuk menguji akurasi, yang setelah itu dibandingkan hasil prediksi manual dan hasil prediksi dari model.	backpropagation. Pada bulan Januari 2015 menghasilkan RMSE 0,1600, pada bulan Februari 0,1048 dan yang paling kecil pada bulan Maret menghasilkan RMSE 0,0877.
5	Peramalan Jumlah Penjualan Menggunakan	Penelitian dilakukan terhadap	Penelitian dilakukan untuk mengetahui	Tidak ada optimasi pada JST backpropagation	Penelitian dilakukan dari mengidentifikasi	Berdasarkan pelatihan JST backpropagation	

	Jaringan Syaraf Tiruan <i>Backpropagation</i> pada Perusahaan Air Minum Dalam Kemasan	penjualan produk AMDK dengan menggunakan JST backpropagation. Data yang digunakan merupakan data penjualan produk AMDK 600ml tahun 2017-2018.	tingkat akurasi dari penggunaan JST backpropagation dalam meramal penjualan AMDK pada Unit Bisnis AMDK Perusahaan ABC Kota Jayapura.	dalam penentuan bobot awal.	masalah, kemudian studi literatur. setelah itu pengumpulan data lalu melakukan analisa terhadap training dan testing untuk kebutuhan peramalan penjualan.	model arsitektur JST multi-layer 12 input layer, 10 hidden layer dan 1 output layer dihasilkan nilai MSE sebesar 0,00043743 dan nilai MAPE 6,88%.
6	Prediksi Jumlah Permintaan Koran Menggunakan Metode Jaringan Syaraf Tiruan <i>Backpropagation</i>	Penelitian ini dilakukan terhadap data jumlah penjualan koran Radar Madura tahun 2015	Penelitian dilakukan untuk memprediksi jumlah koran yang harus diproduksi untuk mengurangi terjadinya resiko kerugian	Dataset tidak disebutkan dalam penelitian.	Penelitian dilakukan dengan mengumpulkan data permintaan koran, kemudian dilakukan normalisasi. Setelah itu	Penelitian menghasilkan nilai error rate terkecil yaitu 0,0162 dengan jumlah iterasi terbaik sebesar 200, dan learning

		menggunakan JST backpropagation.	dikarenakan kelebihan jumlah produksi.		pelatihan JST, setelah selesai dilakukan peramalan JST sehingga didapatkan hasil peramalan. Keluaran yang dihasilkan merupakan peramalan yang akan dihasilkan dari data uji.	rate terbaik 0,6 dengan 100 data latih dan 10 data uji.
--	--	----------------------------------	--	--	--	---

2.2 Dasar Teori

2.2.1 Air

Air merupakan senyawa kimia yang jumlahnya sangat banyak di alam. Air memiliki bentuk cair yang tidak memiliki warna, bau dan juga tidak memiliki rasa. Air mempunyai titik beku 0°C pada tekanan 1 atm, titik didih 100°C dan kerapatan $1,0 \text{ g/cm}^3$ pada suhu 4°C . Ukuran molekul air sangat kecil, biasanya berdiameter sekitar 3 \AA (diameter $0,3 \text{ nm}$ atau $3 \times 10^{-8} \text{ cm}$). Wujud air dapat berupa cairan, gas (uap air) dan padatan (es) [21].

Jumlah air yang terdapat di permukaan bumi ini relatif tetap, meskipun air mengalami pergerakan arus, bersirkulasi karena pengaruh cuaca dan juga mengalami perubahan bentuk. Sirkulasi dan perubahan bentuk antara lain melalui air permukaan berubah menjadi uap (penguapan), air yang mengikuti sirkulasi di dalam tubuh tumbuhan (transpirasi) dan air yang mengikuti sirkulasi di tubuh manusia dan hewan (pernafasan). Sebagian air hujan jatuh langsung ke permukaan, dan sebagian air hujan merembes ke celah-celah bebatuan di dalam tanah, sehingga berubah menjadi air tanah. Air tanah dangkal akan diserap tanaman, sedangkan air tanah dalam akan menjadi mata air. Sirkulasi dan perubahan fisis akan berlangsung hingga akhir zaman [21].

2.2.2 PDAM

PDAM (Perusahaan Daerah Air Minum) adalah badan usaha milik daerah dengan fungsi pelayanan memproduksi air minum dan air bersih bagi masyarakat. Oleh karena itu, PDAM diharapkan dapat memberikan pelayanan air bersih, mendistribusikannya secara merata kepada seluruh lapisan masyarakat, mendorong perkembangan dunia usaha, dan menetapkan struktur tarif yang sesuai dengan tingkat kemampuan masyarakat. Artinya, PDAM memiliki dua fungsi yaitu melayani masyarakat dan meningkatkan pendapatan daerah [1].

2.2.3 Prediksi

Forecasting (prediksi) merupakan proses memperkirakan keadaan pada masa yang akan datang dengan melakukan sebuah analisis terhadap data yang ada pada masa lampau. Data-data sebelumnya digabungkan dengan metode yang telah ditentukan secara terstruktur untuk mendapatkan perkiraan di masa mendatang

[13]. Prediksi digunakan untuk menentukan urutan proses yang akan dilakukan sistem pada data masukan guna mendapatkan keluaran yang diinginkan [22][23].

Berdasarkan teknik yang dipakai untuk melakukan suatu prediksi, maka prediksi dapat dibagi menjadi dua, yaitu prediksi kualitatif dan prediksi kuantitatif [24].

1. Prediksi kualitatif

Prediksi kualitatif didasarkan pada data kualitatif di masa lampau. Metode ini digunakan jika data masa lampau dari variabel yang akan diprediksi tidak ada, tidak cukup ataupun memiliki kredibilitas yang kurang dipercaya. Hasil prediksi yang dibuat sangat tergantung pada orang yang menyusunnya. Ini penting karena hasilnya prediksi ditentukan berdasarkan penilaian atau pendapat, pengetahuan dan pengalaman dari penyusun. Oleh karena itu, metode kualitatif ini disebut juga *judgement, subjective, intuitive* [24].

2. Prediksi kuantitatif

Prediksi kuantitatif didasarkan pada data kuantitatif di masa lampau. Hasil prediksi sangat tergantung pada metode yang digunakan dalam prediksi tersebut. Dengan metode yang berbeda akan diperoleh hasil yang berbeda juga. Hal yang perlu diperhatikan pada penggunaan metode tersebut adalah baik atau tidaknya metode yang digunakan, dan juga sangat ditentukan dari penyimpangan antara hasil prediksi dengan kenyataan yang terjadi. Metode yang baik yaitu metode yang bisa memberikan nilai-nilai perbedaan atau penyimpangan yang mungkin. Prediksi kuantitatif hanya bisa digunakan jika tiga kondisi berikut ini terpenuhi: Ada informasi tentang kondisi lain, informasi bisa dikuantifikasi dalam bentuk data, dapat diasumsikan bahwa pola masa lalu akan terus berlanjut di masa depan [24].

2.2.4 *Machine Learning*

Machine Learning (ML) adalah cabang dari kecerdasan buatan yang secara sistematis menerapkan algoritma untuk memadukan hubungan yang mendasar antara data dan informasi. Tujuan dari ML adalah untuk memprediksi peristiwa atau skenario masa depan yang tidak diketahui oleh komputer. ML merupakan bidang studi yang memberikan komputer kemampuan untuk belajar tanpa diprogram

secara eksplisit [25]. Secara singkat, ML dapat diartikan sebagai proses mengubah data menjadi informasi [26].

Secara umum, algoritma ML bisa dikategorikan menjadi tiga kategori yaitu:

1. *Supervised Learning*

Supervised learning merupakan mekanisme pembelajaran yang menyimpulkan hubungan yang mendasari antara data yang diamati (data masukan) dan variabel target (variabel atau label dependen) [25].

2. *Unsupervised Learning*

Unsupervised learning dirancang untuk menemukan struktur tersembunyi dalam kumpulan data yang tidak berlabel, di mana keluaran yang diinginkan tidak diketahui [25]. Mekanisme ini bermanfaat pada kasus-kasus untuk menemukan hubungan implisit dari dataset yang tidak memiliki label yang ada [27].

3. *Reinforcement Learning*

Metode *reinforcement learning* melibatkan eksplorasi urutan tindakan atau perilaku adaptif oleh *intelligent agent* dalam memberikan lingkungan dengan memberikan motivasi untuk memaksimalkan umpan balik kumulatif [25]. *Reinforcement learning* terinspirasi dari kebiasaan makhluk hidup dalam belajar dan bertindak. Dalam reinforcement learning tidak ada data latih. Data-data diperoleh berdasarkan pengalaman [27].

2.2.5 Jaringan Saraf Tiruan (JST)

Jaringan Saraf Tiruan (JST) merupakan model komputasi yang didasarkan pada *Biological Neural Network*. JST juga dikenal sebagai *Neural Network* (NN). Istilah tiruan di sini digunakan karena jaringan saraf ini mensimulasikan cara kerja jaringan saraf biologis yang ada pada manusia dan diterapkan dengan menggunakan program komputer dalam proses pengambilan keputusan [28][29]. JST dapat belajar dari pengalaman, melakukan generalisasi atas contoh yang diperolehnya dan mengabstraksi karakteristik esensial masukan bahkan untuk data yang tidak relevan [30]. Dengan kemampuan JST yang sangat baik, beberapa aplikasi JST memang sangat cocok untuk diterapkan pada kasus klasifikasi, asosiasi, prediksi, optimasi dan *self-organizing* [7][31].

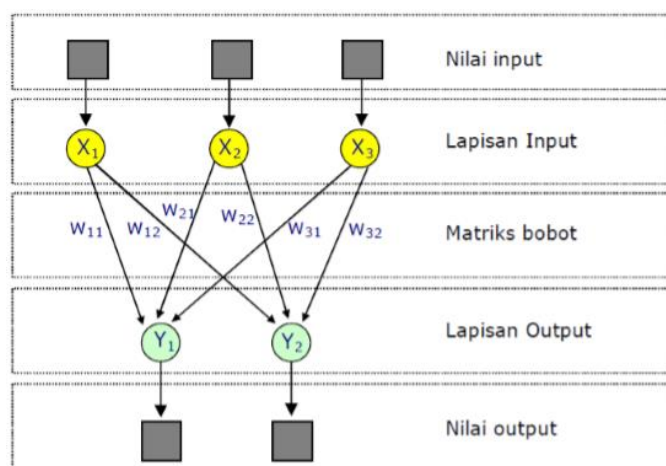
Secara umum JST terdiri dari 3 (tiga) lapisan, yaitu [32]:

1. *Input layer*, terdiri dari beberapa neuron yang menerima sebuah masukan dari lingkaran luar. Neuron tersebut tidak melakukan suatu perubahan dalam masukan, tetapi hanya mengirimkannya ke *processing elements* yang berdekatan dengan lapisan tersembunyi (*hidden layer*) [32].
2. *Hidden layer*, terdiri dari beberapa neuron yang mempunyai jenis menerima masukan yang dipindahkan dari *input layer*, membuat perubahan, dan mengirimkan keluaran ke lapisan berikutnya [32].
3. *Output layer*, terdiri dari beberapa neuron yang menerima *output* dari *hidden layer* dan mengirimkannya kepada *user* [32].

Pengaturan dari saraf-saraf dalam lapisan (*layers*) dan pola hubungannya di antara lapisan-lapisan disebut arsitektur jaringan [32]. Ada beberapa arsitektur JST diantaranya yaitu Jaringan *Single Layer* dan Jaringan *Multi-Layer* [7].

1. Jaringan *Single Layer*

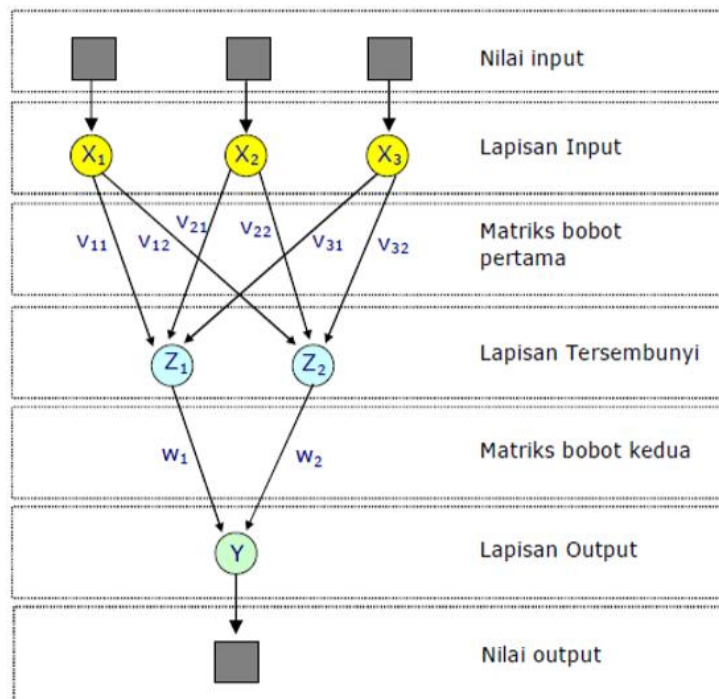
Jaringan yang memiliki arsitektur jenis ini hanya memiliki satu buah lapisan bobot koneksi, jaringan ini terdiri unit-unit *input* yang menerima sinyal dari luar dan unit-unit *output* di mana kita bisa membaca respon dari JST tersebut [7]. Gambar untuk arsitektur jaringan *Single Layer* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Arsitektur jaringan *Single Layer* [7]

2. Jaringan *Multi-Layer*

Merupakan jaringan dengan satu atau lebih lapisan tersembunyi. *Multi-Layer* ini memiliki kemampuan lebih dalam memecahkan masalah bila dibandingkan dengan *Single Layer*. Namun jaringan *Multi-Layer* dalam pelatihannya jauh lebih rumit [7]. Gambar untuk arsitektur jaringan *Multi-Layer* dapat dilihat pada Gambar 2.2.



Gambar 2.2 Arsitektur jaringan *Multi-Layer* [7]

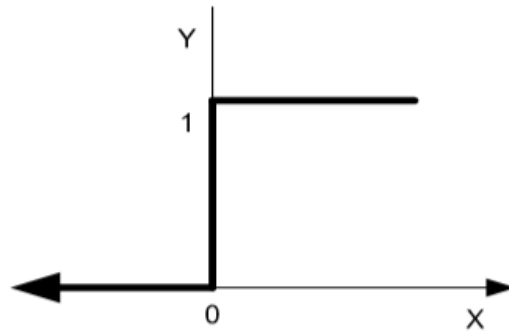
2.2.6 Fungsi Aktivasi

Fungsi aktivasi merupakan bagian penting dari perhitungan keluaran algoritma [7]. Beberapa fungsi aktivasi yang digunakan dalam JST adalah:

1. Fungsi *Threshold* (batas ambang)

Fungsi *threshold* merupakan fungsi *threshold* biner. Untuk kasus bilangan bipolar, maka angka 0 diganti dengan angka -1. Adakalanya dalam JST ditambahkan suatu unit yang nilainya selalu 1. Unit tersebut dikenal dengan bias. Bias dapat dipandang sebagai sebuah *input* yang nilainya selalu 1. Bias berfungsi untuk mengubah *threshold* menjadi sama dengan 0 [33].

$$f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2.1)$$

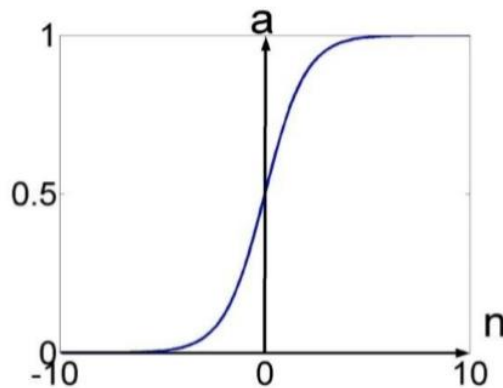


Gambar 2.3 Fungsi aktivasi *threshold* [33]

2. Fungsi *Sigmoid*

Fungsi ini sering digunakan karena nilai fungsinya yang sangat mudah untuk didiferensialkan [33].

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.2)$$



Gambar 2.4 Fungsi aktivasi *sigmoid* [33]

3. Fungsi Identitas

Digunakan jika keluaran yang dihasilkan oleh JST merupakan sembarang bilangan riil (bukan hanya pada range [0,1] atau [1,-1] [33].

$$f(x) = x \quad (2.3)$$

2.2.7 Backpropagation

Backpropagation merupakan salah satu algoritma pembelajaran dalam jaringan saraf tiruan [34]. *Backpropagation* merupakan algoritma pembelajaran yang terawasi (*supervised learning*) dan pada umumnya digunakan oleh *perceptron* dengan lapisan yang banyak untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada *hidden layer* [7][35]. *Backpropagation* diperkenalkan oleh Rumelhart, Hinton dan William pada tahun 1986, kemudian Rumelhart dan Mc Clelland mengembangkannya pada tahun 1988. Algoritma ini termasuk dalam *supervised learning* di mana tanda dari metode ini yaitu meminimalkan *error* pada output yang dihasilkan oleh jaringan. Algoritma *backpropagation* untuk *neural network* biasanya diaplikasikan pada jaringan berlapis banyak (*multilayer*). Algoritma ini setidaknya mempunyai bagian *input*, bagian *output* dan beberapa lapis yang berada di antara input dan output. Lapisan di tengah ini biasa disebut dengan lapisan tersembunyi (*hidden layer*). *Hidden layer* bisa berjumlah satu, dua, tiga dan seterusnya. Output lapis terakhir dari *hidden layer* langsung dipakai sebagai *output* dari *neural network* [20].

Algoritma *backpropagation* menggunakan *error output* untuk mengubah bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan *error*, harus menyelesaikan fase propagasi maju (*forward propagation*) terlebih dahulu. Dalam proses propagasi maju, fungsi aktivasi sigmoid digunakan untuk mengaktifkan neuron. Pada awal proses pembelajaran bobot hubungan diperoleh secara acak. Algoritma pembelajaran, memodifikasi bobot tersebut dalam setiap *epoch* sampai pembelajaran selesai. Jika proses telah mencapai *epoch* maksimum yang ditargetkan atau telah mencapai nilai *error* yang ditargetkan maka proses berhenti. Koleksi bobot hubungan yang tersimpan digunakan sebagai bobot hubungan pada proses pengujian [32].

Proses pembelajaran *backpropagation* dimulai dengan menentukan target *error*, *maksimum epoch*, *learning rate* (α) dan memasukkan bobot awal secara acak ke dalam model [32]. Algoritma pembelajaran metode *backpropagation* dengan satu *hidden layer* dengan fungsi aktivasi sigmoid biner range [0,1] adalah sebagai berikut:

1. Langkah 0:
Inisialisasi bobot-bobot garis dengan nilai acak kecil.
2. Langkah 1:
Jika kondisi berhenti belum terpenuhi, lakukan langkah 2 sampai 8
3. Langkah 2:
Untuk masing-masing pasangan *training* lakukan proses *feedforward* dengan 3 langkah.

Fase *feedforward*

4. Langkah 3:
Masing-masing unit *input* (X_i , $i=1, \dots, n$) menerima sinyal *input* X_i dan menyebarkan ke sinyal unit lapisan atas (*hidden unit*).
5. Langkah 4:
Masing-masing unit tersembunyi (Z_j , $j=1, \dots, p$) nilai *input* dihitung menggunakan persamaan (2.4)

$$z_{net_j} = v_{j0} + \sum_{t=1} x_t v_{jt} \quad (2.4)$$

Kemudian menjumlahkan bobot sinyal dan mengaplikasikan fungsi aktivasi untuk menghitung sinyal output menggunakan persamaan (2.5) dan mengirimkan sinyal tersebut ke semua unit pada lapisan atasnya (unit *output*).

$$z_j = f(Z_{net_j}) \quad (2.5)$$

6. Langkah 5:
Masing-masing unit *output* (Y_k , $k=1, \dots, m$) nilai *input* dihitung menggunakan persamaan (2.6)

$$y_{net_k} = w_{k0} + \sum_{j=1} z_j w_{kj} \quad (2.6)$$

Menjumlahkan jumlah sinyal *input* dan mengaplikasikan fungsi aktivasi persamaan (2.7) untuk menghitung sinyal output.

$$Y_k = f(y_{net_k}) \quad (2.7)$$

Fase *backpropagation*

7. Langkah 6:

Masing-masing unit *output* (Y_k , $k=1, \dots, m$) menerima sebuah pola target yang bersesuaian dengan pola *training input*, menghitung informasi kesalahan menggunakan persamaan (2.8)

$$\delta_k = (t_k - y_k) f'(y_{net_k}) = (t_k - y_k) y_k (1 - y_k) \quad (2.8)$$

Kemudian menghitung koreksi bobot (ini digunakan untuk memperbaiki w_{jk}) dengan persamaan (2.9)

$$\Delta W_{kj} = a \delta_k \cdot z_j \quad (2.9)$$

Setelah itu menghitung korelasi bias (ini digunakan untuk memperbaiki w_{0k}) dengan persamaan (2.10), setelah itu mengirimkan δ_k ke unit dalam lapisan paling atas.

$$\Delta W_{k0} = a \delta_k \quad (2.10)$$

8. Langkah 7:

Masing-masing unit tersembunyi (Z_j , $j=1, \dots, p$) menjumlahkan input delta (dari unit lapisan atas) dengan menggunakan persamaan (2.11)

$$\delta_{net_j} = (x + a)^n = \sum_{k=1}^m \delta_k w_{kj} \quad (2.11)$$

Kalikan nilai dengan turunan fungsi aktivasi untuk menghitung informasi kesalahan menggunakan persamaan (2.12)

$$\delta_j = \delta_{net_j} \cdot f'(z_{net_j}) \quad (2.12)$$

Kemudian hitunglah koreksi bobot (digunakan untuk memperbaiki v_{ij}) dengan menggunakan persamaan (2.13).

$$\Delta V_{ji} = a\delta_j x_i \quad (2.13)$$

Setelah itu hitung koreksi bias (digunakan untuk memperbaiki v_{0j}) menggunakan persamaan (2.14)

$$\Delta V_{j0} = a\delta_j \quad (2.14)$$

Fase *update* bobot dan bias

9. Langkah 8

Masing-masing unit *output* ($Y_k, k=1, \dots, m$) memperbaiki bobot dan bias ($Z_j, j=0, \dots, p$) dengan menggunakan persamaan (2.15)

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (2.15)$$

Kemudian masing-masing unit tersembunyi ($Z_j, j=1, \dots, p$) memperbaiki bobot dari bias ($X_i, i=1, \dots, n$) menggunakan persamaan (2.16)

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \quad (2.16)$$

10. Langkah 9:

Tes kondisi berhenti.

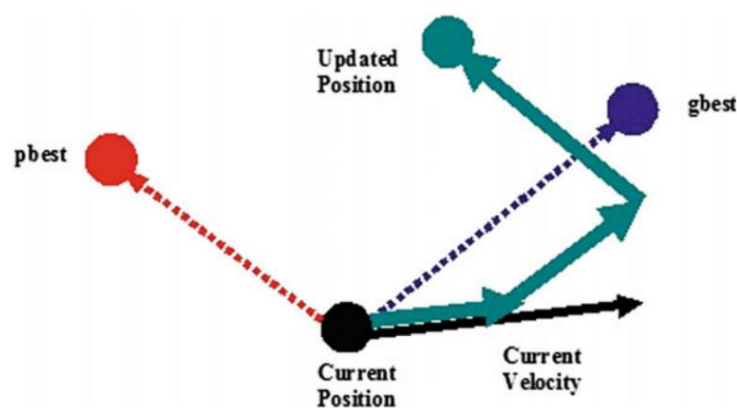
2.2.8 Swarm Intelligent (SI)

Swarm Intelligent (SI) adalah disiplin ilmu yang melibatkan sistem alam dan buatan. Terdiri dari banyak individu yang menggunakan kontrol desentralisasi untuk berkoordinasi dan mengatur diri sendiri (*self-organization*) [36][37]. Secara khusus, disiplin ini berfokus pada perilaku kolektif yang dihasilkan oleh interaksi lokal antara satu individu dan individu lain, dan antara individu dan lingkungan.

Kemudian, logika SI menginspirasi sebagian besar peneliti untuk mengembangkan algoritma berbasis kecerdasan hewan, khususnya untuk menyelesaikan masalah optimasi. Berbagai algoritma pada perilaku kolektif diusulkan untuk memecahkan masalah optimasi. Masalah optimasi menyebabkan perlunya dilakukan penelitian untuk menentukan nilai minimum atau maksimum dari suatu fungsi. Nilai tersebut disebut sebagai nilai optimal [36].

2.2.9 Particle Swarm Optimization (PSO)

Salah satu algoritma SI adalah *Particle Swarm Optimization* (PSO). PSO adalah teknik optimasi populasi yang dikembangkan oleh James Kennedy dan Russ Eberhart pada tahun 1995 [37]. Teknik ini terinspirasi berdasarkan perilaku sosial kawanan burung atau kawanan ikan [38]. Setiap burung dianggap sebagai partikel yang merepresentasikan solusi dari suatu permasalahan yang memiliki posisi (x) dan kecepatan (v). PSO memiliki dua fungsi utama, yaitu *update* kecepatan dan *update* posisi yang digunakan untuk memperbarui kecepatan dan posisi partikel sehingga akan terus dipercepat sampai mendekati posisi partikel terbaik dari partikel tersebut (*local best*) dan partikel terbaik dari seluruh kawanan (*global best*) sampai kondisi minimum *error* tercapai. PSO dimulai dengan sekumpulan partikel (solusi) yang dibangkitkan secara acak [36].



Gambar 2.5 Ilustrasi pergerakan partikel pada proses PSO [39]

Dibandingkan dengan algoritma *Ant Colony* dan EA, PSO hanya membutuhkan operator matematika lama, lebih sedikit komputasi dan umumnya

memiliki baris kode yang lebih sedikit. Dengan demikian secara komputasi lebih sederhana baik dalam hal memori dan *requirements*. PSO populer karena kesederhanaan implementasinya dan kemampuannya yang menjadi konvergen dengan cepat dan memberikan solusi yang dapat diterima [17].

Beberapa istilah umum yang biasa digunakan dalam PSO dapat didefinisikan sebagai berikut [29]:

1. *Swarm*, populasi dari suatu algoritma.
2. *Particle*, anggota (individu) pada suatu *swarm*. Setiap *particle* merepresentasikan suatu solusi yang potensial pada permasalahan yang diselesaikan. Posisi dari suatu *particle* adalah ditentukan oleh representasi solusi saat itu.
3. *Pbest (Personal best)*, posisi *Pbest* suatu *particle* yang menunjukkan posisi *particle* yang dipersiapkan untuk mendapatkan suatu solusi terbaik.
4. *Gbest (Global best)*, posisi terbaik *particle* pada *swarm* atau posisi terbaik di antara *Pbest* yang ada.
5. *Velocity (v)*, vector yang menggerakkan proses optimasi yang menentukan arah di mana suatu *particle* diperlukan untuk berpindah (*move*) untuk memperbaiki posisinya semula atau kecepatan yang menggerakkan proses optimasi yang menentukan arah di mana *particle* diperlukan untuk berpindah dan memperbaiki posisinya semula.
6. *Inertia weight, inertia weight* disimbolkan dengan w , parameter ini digunakan untuk mengontrol dampak dari adanya *velocity* yang diberikan oleh suatu *particle*.
7. *Learning rates (c_1 dan c_2)*, suatu konstanta untuk menilai kemampuan *particle* (c_1) dan kemampuan sosial *swarm* (c_2) yang menunjukkan bobot dan *particle* terhadap memorinya.

Algoritma PSO meliputi langkah berikut:

1. Langkah 1:
Bangkitkan posisi awal sejumlah partikel sekaligus kecepatan awalnya secara random.
2. Langkah 2:

Evaluasi *fitness* dari masing-masing partikel berdasarkan posisinya.

3. Langkah 3:

Tentukan partikel dengan *fitness* terbaik dan tetapkan sebagai *Gbest*. Untuk setiap partikel, *Pbest* awal akan sama dengan posisi awal.

4. Langkah 4:

Ulangi langkah 5 sampai 8 sampai *stopping criteria* terpenuhi.

5. Langkah 5:

Menggunakan *Pbest* dan *Gbest* yang ada, perbarui kecepatan setiap partikel menggunakan persamaan (2.17)

$$v_j^{t+1} = w \cdot v_j^t + c_1 \cdot r_1 (pBest_j^t - x_j^t) + c_2 \cdot r_2 (gBest_j^t - x_j^t) \quad (2.17)$$

Lalu dengan kecepatan baru yang didapat, perbarui posisi setiap partikel menggunakan persamaan (2.18)

$$x_j^{t+1} = x_j^t + v_j^{t+1} \quad (2.18)$$

di mana:

v_j^t : kecepatan partikel j pada iterasi ke- t

x_j^t : posisi partikel j pada iterasi ke- t

w : bobot inersia

c_1 & c_2 : *learning rates*

r_1 & r_2 : bilangan acak yang bernilai 0 sampai 1

$pBest_j^t$: posisi terbaik dari partikel j pada iterasi ke- t

$gBest_j^t$: global optimal dari partikel j pada iterasi ke- t

6. Langkah 6:

Evaluasi *fitness* dari setiap partikel.

7. Langkah 7:

Tentukan partikel dengan *fitness* terbaik, dan tetapkan sebagai *Gbest*. Untuk setiap partikel, tentukan *Pbest* dengan membandingkan posisi sekarang dengan *Pbest* dari iterasi sebelumnya.

8. Langkah 8:
Cek *stopping criteria*.