

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Sebelumnya

Penelitian yang dilakukan sebagai bentuk antisipasi terhadap kejadian yang akan datang diperlukan untuk melakukan perencanaan maupun persiapan yang lebih baik agar tidak memunculkan permasalahan yang sama seperti kejadian sebelumnya. Beberapa penulis, tidak sedikit yang melakukan penelitian dengan mengangkat tema prediksi mengenai suatu kejadian atau permasalahan dalam berbagai bidang. Dari penelitian-penelitian yang sudah dilakukan oleh beberapa penulis tersebut menunjukkan bahwa memprediksikan suatu kejadian atau permasalahan sebagai data yang akan dijadikan bentuk antisipasi dalam perencanaan maupun persiapan yang lebih baik dari sebelumnya sangat diperlukan untuk menciptakan masa depan yang lebih baik.

Penelitian sebelumnya yang berjudul “Optimasi Algoritma Neural Network dengan Algoritma Genetika dan Particle Swarm Optimization untuk Memprediksi Hasil Pemilukada” yang dilakukan oleh Muhammad Badrul membuktikan bahwa algoritma *neural network* memiliki kemampuan yang baik dalam memprediksi, namun terdapat kelemahan pada penentuan arsitektur, bobot awal, dan bobot akhir. Oleh karena itu, dilakukan kombinasi dengan algoritma genetika dan *particle swarm optimization*. Dari kedua algoritma tersebut, telah dibuktikan bahwa kombinasi *neural network* dan *particle swarm optimization* memiliki akurasi yang lebih tinggi dari kombinasi *neural network* dan algoritma genetika, dimana algoritma *particle swarm optimization* berperan dalam pengoptimalan bobot pada algoritma *neural network*[10].

Penelitian lain yang berjudul “Analisis Jaringan Syaraf Tiruan untuk Prediksi Volume Ekspor dan Impor Migas di Indonesia” yang dilakukan oleh Yuli Andriani, Hotmalina Silitonga, dan Anjar Wanto membuktikan bahwa penggunaan algoritma jaringan saraf tiruan *backpropagation* dalam volume ekspor dan impor migas di Indonesia memiliki kemampuan yang baik dengan nilai MSE mendekati 0 yaitu 0,0281641257[5].

Penelitian selanjutnya yang berjudul “*Application of Hybrid Optimized BP Network Model to Estimate Water Quality Parameters of Beihai Lake in Beijing*” yang dilakukan oleh Jianzhuo Yan, Zongbao Xu , Yongchuan Yu, Hongxia Xu, dan Kaili Gao membuktikan bahwa kelemahan *backpropagation neural networks* dapat diatasi dengan melakukan kombinasi menggunakan algoritma optimasi, seperti algoritma genetika dan *particle swarm optimization*. Pada penelitian ini, estimasi terhadap parameter kualitas air di Danau Beihai dilakukan menggunakan empat metode, yaitu BPNN, BPNN-PSO, BPNN-GA, dan PSO-GA-BPNN[15].

Penelitian lain yang berjudul “Penerapan Jaringan Syaraf Tiruan Untuk Peramalan Penjualan Dalam Mendukung Pengembangan Agroindustri Coklat di Kabupaten Blitar” yang dilakukan oleh Hendri Cahya Aprilianto, Sri Kumalaningsih, dan Imam Santoso telah membuktikan bahwa algoritma jaringan saraf tiruan *backpropagation* mampu menghasilkan nilai MSE yang baik yaitu sebesar 0,0001148. Hal tersebut menunjukkan kemampuan algoritma jaringan saraf tiruan *backpropagation* dalam memprediksi jumlah penjualan coklat di Kabupaten Blitar, dimana data yang digunakan berupa data *time series* yaitu data jumlah penjualan coklat dari tahun 2013-2017[13].

Penelitian selanjutnya dengan judul “Optimasi Jaringan Saraf Tiruan Backpropagation Dengan Particle Swarm Optimization Untuk Prediksi Pasang Surut Air Laut” yang dilakukan oleh Nerfita Nikentari, Hendra Kurniawan, Nola Ritha dan Denny Kurniawan telah membuktikan bahwa kombinasi kinerja algoritma jaringan saraf tiruan *backpropagation* dengan algoritma *particle swarm optimization* mampu memprediksi pasang surut air laut dengan akurasi yang dihasilkan sebesar 91.56%[7]. Pernyataan ini membuktikan bahwa perpaduan antara algoritma jaringan saraf tiruan *backpropagation* dan algoritma *particle swarm optimization* dapat digunakan dalam memecahkan masalah prediksi. Ringkasan penelitian yang relevan ditunjukkan pada Tabel 2.1.

Tabel 2.1 Penelitian Terdahulu

No.	Judul	Comparing	Contrasting	Criticize	Synthesize	Summarize
1	Optimasi Algoritma Neural Network dengan Algoritma Genetika dan Particle Swarm Optimization untuk Memprediksi Hasil Pemilukada	Penelitian ini dilakukan terhadap hasil Pemilihan umum Kepala Daerah dan Wakil Kepala Daerah menggunakan algoritma <i>neural network</i> dengan algoritma genetika dan <i>particle swarm optimization</i> .	Penelitian dilakukan untuk menciptakan hasil prediksi yang akurat, dimana hasil prediksi tersebut mempengaruhi aspek social, ekonomi, keamanan, dan lain-lain.	Dataset yang digunakan tidak disebutkan dalam penelitian.	Penelitian dilakukan dengan pengumpulan sejumlah data, kemudian data tersebut diproses menggunakan algoritma jaringan saraf tiruan (<i>Neural Network</i>) dengan algoritma genetika dan <i>particle swarm optimization</i> , kemudian dilakukan perbandingan tingkat akurasi yang lebih tinggi antara <i>neural network</i> dan algoritma genetika dengan <i>neural network</i> dan <i>particle swarm</i>	Penelitian dengan kombinasi <i>neural network</i> dan <i>particle swarm optimization</i> menghasilkan akurasi yang lebih tinggi dari kombinasi <i>neural network</i> dan algoritma genetika.

No.	Judul	Comparing	Contrasting	Criticize	Synthesize	Summarize
					<i>optimization.</i>	
2	Analisis Jaringan Syaraf Tiruan untuk prediksi volume ekspor dan impor migas di Indonesia	Penelitian ini dilakukan terhadap volume ekspor dan impor migas di Indonesia dengan menggunakan jaringan saraf tiruan <i>backpropagation</i> . Data yang digunakan bersumber dari dokumen kepabean Ditjen Bea dan Cukai yaitu Pemberitahuan Ekspor Barang (PEB) dan Pemberitahuan Impor Barang (PIB) tahun 1996 hingga 2016.	Penelitian dilakukan untuk mengetahui volume perkembangan ekspor dan impor di Indonesia di bidang migas di masa yang akan datang dengan menggunakan jaringan syaraf tiruan <i>backpropagation</i> .	Belum adanya optimasi terhadap penentuan bobot pada jaringan saraf tiruan <i>backpropagation</i> .	Penelitian dimulai dengan pengumpulan data, lalu melakukan studi Pustaka, kemudian mengidentifikasi masalah, preproses, menguji hasil pengolahan data, memprediksi, dan mengevaluasi akhir.	Penelitian menghasilkan akurasi antara 80% hingga 90% dengan arsitektur <i>backpropagation</i> 12-5-1 dengan <i>Means Square Error (MSE)</i> sebesar 0,0281641257.
3	<i>Application of Hybrid Optimized BP Network Model to Estimate Water Quality Parameters of Beihai Lake in Beijing</i>	Penelitian ini dilakukan terhadap parameter kualitas air Sungai Beihai, Beijing. Data yang digunakan yaitu data <i>time series</i> berupa <i>monitoring</i> kualitas air dalam 120 h pada bulan Agustus 2013.	Penelitian dilakukan untuk memberikan informasi prediksi kualitas air kepada bagian perlindungan para lingkungan air sebagai antisipasi dalam mengatasi krisis air bersih.	Tidak ditunjukkan contoh data asli yang digunakan dalam penelitian..	Penelitian dimulai dengan pengumpulan data, lalu melakukan studi Pustaka, kemudian menganalisa data, lalu melakukan pengolahan data, kemudian melakukan prediksi menggunakan BPNN, BPNN-	Penelitian menghasilkan nilai RMSE pada BPNN sebesar 1,2733, BPNN-GA sebesar 0,4019, BPNN-PSO sebesar 0,7873, dan PSO-GA-BPNN sebesar 0,3596.

No.	Judul	Comparing	Contrasting	Criticize	Synthesize	Summarize
					PSO, BPNN-GA, dan PSO-GA-BPNN, setelah itu mengevaluasi hasil akhir prediksi.	
4	Penerapan Jaringan Syaraf Tiruan Untuk Peramalan Penjualan Dalam Mendukung Pengembangan Agroindustri Coklat di Kabupaten Blitar	Penelitian dilakukan terhadap jumlah penjualan coklat di Kabupaten Blitar. Data yang digunakan yaitu data jumlah penjualan coklat di Kabupate Blitar dari tahu 2013-2017.	Penelitian dilakukan untuk mengetahui nilai prediksi jumlah penjualan coklat di Kabupaten Blitar sebagai antisipasi dalam menangani peningkatan biaya produksi dan biaya inventori jika terdapat produk yang tidak terjual.	Tidak adanya optimasi pada jaringan saraf tiruan <i>backpropagation</i> dalam penentuan arsitektur, bobot awal dan bias awal serta tidak adanya tingkat akurasi pada hasil penelitian.	Penelitian dimulai dengan pengumpulan data, lalu penentuan responden yang berjumlah 3 orang, kemudian dilakukan pengolahan data, lalu memprediksi jumlah penjualan coklat, dan evaluasi hasil prediksi menggunakan perhitungan MSE.	Penelitian menghasilkan akurasi berupa nilai <i>Mean Squared Error</i> sebesar 0,0001148.
5	Optimasi Jaringan Saraf Tiruan Backpropagation Dengan Particle Swarm Optimization Untuk Prediksi Pasang Surut Air Laut	Penelitian dilakukan terhadap data pasang surut air laut dengan perjam variabel jam dan tinggi (m) yang berjumlah 1000 data.	Penelitian dilakukan untuk mengetahui waktu terjadinya pasang surut yang mempengaruhi kesejahteraan hidup sumber daya perairan.	Penggunaan parameter yang digunakan memerlukan variasi yang lebih banyak pada pengujian parameter	Penelitian dimulai dengan pengumpulan data pasang surut air laut perjam dengan variabel jam dan	Penelitian menghasilkan MSE terbaik sebesar 0,004283.

No.	Judul	Comparing	Contrasting	Criticize	Synthesize	Summarize
				metode jaringan saraf tiruan dan <i>particle swarm optimization</i> .	tinggi air laut, kemudian dilakukan prediksi optimasi bobot menggunakan algoritma PSO, lalu bobot yang telah dioptimasi menjadi input pada prediksi menggunakan algoritma jaringan saraf tiruan. Kemudian, hasil akhir dievaluasi dengan menggunakan perhitungan MSE.	

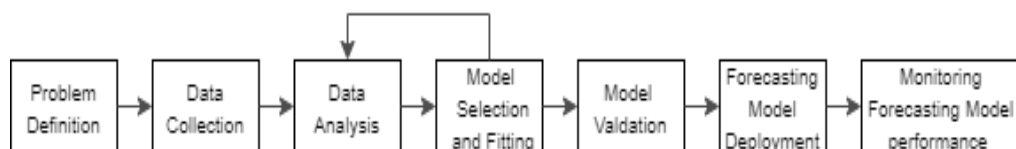
2.2 Dasar Teori

Pada bagian dasar teori, penulis menguraikan teori-teori berdasarkan hasil studi pustaka melalui jurnal dan buku sebagai landasan pengetahuan dalam melakukan penelitian. Teori-teori tersebut meliputi prediksi, penyakit ginjal kronis, jaringan saraf tiruan, arsitektur jaringan saraf tiruan, fungsi aktivasi, propagasi balik (*backpropagation*), tahapan *backpropagation*, *swarm intelligence*, *particle swarm optimization* (PSO), tahapan *particle swarm optimization*, normalisasi dan denormalisasi, dan *mean squared error* (MSE).

2.2.1 Prediksi (Peramalan)

Prediksi atau peramalan merupakan aktifitas memperkirakan sesuatu yang akan terjadi di masa yang akan datang dengan memanfaatkan serangkaian data mengenai hal-hal yang sudah terjadi di waktu sebelumnya yang telah mengalami penganalisaan secara ilmiah[16]. Prediksi juga dapat disebut sebagai alat yang termasuk efisien dan efektif dalam perencanaan, terutama dalam bidang ekonomi. Pengetahuan mengenai keadaan di masa mendatang memiliki peran yang sangat penting dalam memahami kebaikan dan keburukan keadaan yang sudah atau sedang terjadi dan juga bertujuan untuk mempersiapkan segala hal dengan baik untuk masa depan yang lebih baik[17]. Prediksi dilakukan dengan tujuan mendapatkan informasi yang memiliki probabilitas tertinggi di masa yang akan datang[5].

Pendekatan pada prediksi merupakan perkembangan kedua pendekatan yaitu pendekatan stokastik dan deterministik. Secara garis besar, prediksi banyak menggunakan pendekatan deterministik. Pendekatan deterministik merupakan kondisi di saat t , $t-1$ dan $t+1$ dalam periode harian[18]. Metode prediksi dapat dilakukan secara kualitatif melalui pendapat para pakar atau secara kuantitatif dengan perhitungan secara matematis. Salah satu metode prediksi kuantitatif adalah menggunakan analisis deret waktu (*time series*)[5].



Gambar 2.1 Proses Prediksi[17]

Pada Gambar 2.1 ditunjukkan diagram proses prediksi, dimana proses pertama yaitu *problem definition* yang merupakan proses memahami lebih dalam mengenai permasalahan yang akan diprediksi agar sesuai atau tidak jauh dengan ekspektasi seseorang yang melakukan

prediksi. *Data collection* merupakan proses mencari keterkaitan antara variabel-variabel yang ada di data sebelumnya dengan permasalahan yang akan diprediksi, karena tidak semua variabel yang ada pada data sebelumnya berpengaruh terhadap permasalahan yang akan diprediksi. *Data analysis* merupakan proses penting dalam model prediksi yang akan digunakan, karena dalam proses ini terjadi pengenalan pola dari data yang akan digunakan yang akan mempengaruhi penggunaan data dalam proses prediksi[19].

Model selection and fitting merupakan proses memilih satu atau lebih model prediksi serta disesuaikan dengan data yang sudah dianalisis. *Model validation* merupakan proses evaluasi terhadap model prediksi yang digunakan, dimana pada proses ini ditentukan seberapa sesuai performansi model prediksi untuk diimplementasikan pada aplikasi yang diharapkan. *Forecasting model deployment* merupakan proses yang melibatkan penggunaan model prediksi dan hasil prediksi yang didapatkan sebagai bahan yang dipelajari dan dipahami. *Monitoring forecasting model performance* merupakan proses yang dilakukan secara berkelanjutan untuk memastikan kinerja model prediksi yang digunakan tetap sesuai dengan target yang ingin dicapai[19].

2.2.2 Penyakit Ginjal Kronis

Ginjal merupakan bagian dari organ tubuh yang memiliki fungsi dalam pengaturan keseimbangan cairan pada tubuh, keseimbangan basa-asam pada darah serta pengaturan konsentrasi garam pada darah[20]. Selain itu, ginjal juga berfungsi untuk menjaga kestabilan komposisi elektrolit seperti fosfat, potasium, dan sodium serta membuat eritrosit dan menjaga kesehatan pada tulang[1]. Penyakit ginjal kronis merupakan kondisi ginjal yang mengalami kerusakan dimana *Glomerulus Filtrate Rate* (GFR) kurang dari 60mL/menit/1,73m² dalam jangka waktu 3 bulan maupun lebih, sementara pada tahap akhir penyakit ginjal kronis memiliki

ginjal dengan *Glomerulus Filtrate Rate* kurang dari 15mL/menit/1,73m² dengan pelayanan dialisis maupun tidak[21].

Penanganan penyakit ginjal kronis terbagi dalam tiga jenis yaitu, hemodialisis, transplantasi ginjal dan dialisis peritoneal. Pada Peraturan Menteri Kesehatan Nomor 812 Tahun 2010 tentang Penyelenggaraan Pelayanan Dialisis pada Fasilitas Pelayanan Kesehatan, dialisis merupakan bagian dari pelayanan medis dalam menangani pasien penyakit ginjal kronis, dimana pelayanan ini dilakukan dengan terapi untuk mengganti fungsi ginjal sementara demi mempertahankan kualitas hidup pasien. Hemodialisis merupakan salah satu bagian dari dialisis berupa terapi pengganti fungsi ginjal dengan alat khusus yang berfungsi dalam pengeluaran toksis uremic serta pengatur cairan elektrolit pada tubuh. Dialisis peritoneal juga merupakan bagian dari dialisis yang memanfaatkan peritoneum pasien sebagai membrane semipermeable, diantaranya yaitu terapi pengganti ginjal dengan mempergunakan peritoneum pasien sebagai membran semipermeabel, antara lain *Ambulatory Peritoneal Dialysis* (APD) dan *Continous Ambulatory Peritoneal Dialysis* (CAPD)[1].

Pelayanan medis terhadap pasien penyakit ginjal kronis dengan hemodialisis dilakukan dengan pengeluaran darah dari tubuh yang dipompa untuk memasuki mesin dialisis, dimana terjadi pembuangan toksin dengan cara filtrasi atau penyaringan, lalu kembali dialirkan ke tubuh pasien[22]. Akses sirkulasi yang terdapat pada proses hemodialisis terdiri dari beberapa jenis diantaranya yaitu Femoral, AV (*Arterio Venous*) Shunt dan Double Lumen. Akses sirkulasi femoral merupakan tindakan penyuntikan melalui daerah di atas paha dan daerah tangan. AV (*Arterio Venous*) Shunt merupakan tindakan penyuntikan yang dilakukan pada daerah kedua tangan dalam memperbesar aliran darah vena untuk memudahkan proses hemodialisis. *Double Lumen* merupakan tindakan penyuntikan yang dilakukan melalui clavicula[23].

2.2.3 Jaringan Saraf Tiruan

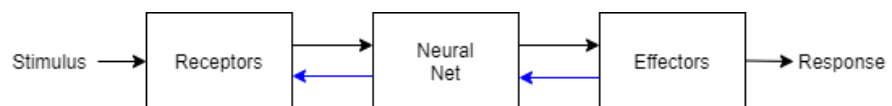
Jaringan saraf tiruan (*Artificial Neural Network*) merupakan algoritma *Neural Network* yang disebut sebagai sistem yang melakukan pemrosesan terhadap informasi dengan metode yang terinspirasi dari cara sistem saraf pada otak manusia dalam melakukan tugasnya[24]. Jaringan saraf tiruan termasuk dalam simpulan umum model matematis yang didasarkan oleh pemahaman dari manusia, algoritma jaringan saraf tiruan melakukan proses langsung dengan angka. Oleh karena itu, data yang tidak berbentuk numerik memerlukan proses pengolahan data agar data menjadi data numerik[25].

Tabel 2.2 menunjukkan keterkaitan jaringan saraf tiruan dengan konsep biologi menurut Medsker dan Liebowitz dalam Turban, dimana pada konsep biologi node (simpul) dalam jaringan saraf tiruan merupakan soma, input pada jaringan saraf tiruan merupakan dendrit, output pada jaringan saraf tiruan merupakan akson, bobot pada jaringan saraf tiruan merupakan sinapsis, jumlah neuron pada jaringan saraf tiruan terdiri dari beberapa neuron, sementara dalam konsep biologi neuron yang tersusun banyak (10^9), dan kecepatan pada jaringan saraf tiruan tinggi sementara pada konsep biologi memiliki kecepatan yang rendah[26].

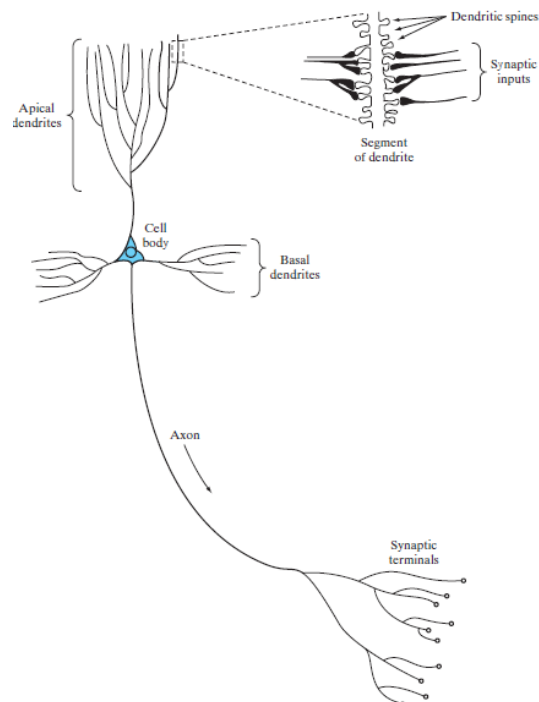
Tabel 2.2 Keterkaitan Antara JST dan Biologi[24]

Biologi	Jaringan Saraf tiruan
Soma	Node (simpul)
Dendrit	Input
Akson	Output
Sinapsis	Weight (bobot)
Neuron yang tersusun banyak (10^9)	Beberapa neuron
Kecepatan rendah	Kecepatan tinggi

Otak manusia merupakan pusat kerja sistem yang direpresentasikan dengan jaringan saraf yang menerima informasi secara terus-menerus, kemudian merasakan dan membuat keputusan terhadap informasi tersebut. Pada Gambar 2.2, ditunjukkan dua pasang tanda panah, dimana tanda panah yang mengarah dari kiri ke kanan menunjukkan transmisi dari informasi-informasi yang membawa sinyal melalui sistem. Sementara itu, anak panah yang mengarah dari kanan ke kiri menunjukkan umpan balik dari sistem. Stimulus yang berasal dari tubuh manusia atau dari lingkungan dikonversi oleh reseptor menjadi impuls listrik yang menyampaikan informasi untuk saraf. Lalu, efektor melakukan konversi terhadap impuls listrik yang dihasilkan oleh jaringan saraf menjadi tanggapan yang dapat dilihat sebagai *output* dari sistem saraf[27].

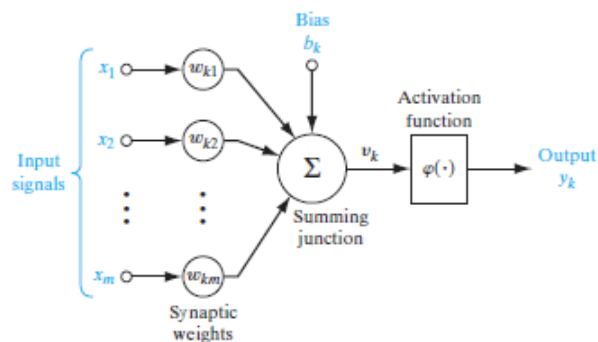


Gambar 2.2 Diagram Blok Sistem Saraf Manusia[25]



Gambar 2.3 Struktur Neuron Pada Otak Manusia[25]

Gambar 2.3 merupakan struktur neuron pada otak manusia yang terdiri dari beberapa bagian yaitu dendrit, akson dan sinapsis. Dendrit memiliki peran sebagai pengirim impuls yang telah diterima menuju bagian badan dari sel saraf. Kemudian, akson memiliki peran sebagai pengirim impuls dari bagian badan sel saraf menuju jaringan yang lain. Lalu, terdapat sinapsis yang memiliki fungsi sebagai sebuah unit fungsional dengan posisi di antara dua buah sel saraf. Sinapsis memiliki kemampuan yang tidak stabil atau dapat menurun maupun meningkat bergantung pada besarnya tingkat propagasi sinyal yang dapat diterimanya. Impuls-impuls dari sinyal atau disebut sebagai informasi akan mengalami penerimaan oleh neuron yang lain jika telah memenuhi batasan yang disebut sebagai nilai ambang atau threshold[28].



Gambar 2.4 Struktur Unit Jaringan Saraf Tiruan[25]

Gambar 2.4 menunjukkan struktur unit jaringan saraf tiruan, ditunjukkan struktur unit jaringan saraf tiruan yang terdiri dari tiga elemen dasar, yaitu[27]:

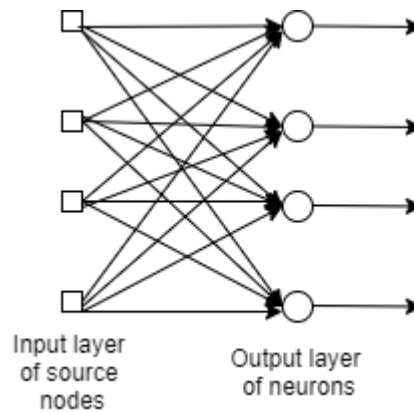
1. Satu set dari sinapsis, atau penghubung yang masing-masing digolongkan oleh bobot atau kekuatannya.
2. Sebuah penambah untuk menjumlahkan sinyal-sinyal input yang diukur dari kekuatan sinaptik masing-masing neuron.
3. Sebuah fungsi aktivasi untuk membatasi amplitudo output dari neuron. Fungsi ini bertujuan membatasi jarak amplitude yang

diperbolehkan oleh sinyal output menjadi sebuah angka yang terbatas.

Jaringan saraf tiruan memiliki tiga lapisan dalam struktur arsitekturnya, yaitu *input layer*, *hidden layer* dan *output layer*. *Input layer* yang berperan sebagai dendrit merupakan sebuah lapisan dengan peran sebagai unit yang menyatakan pola apa saja yang akan digunakan sebagai masukan pada jaringan. *Hidden layer* merupakan sebuah lapisan yang menghubungkan *input layer* dan *output layer*. *Hidden layer* dapat berjumlah lebih dari satu pada sebuah jaringan dalam beberapa kasus tertentu. *Output layer* yang berperan sebagai akson merupakan lapisan paling akhir pada jaringan saraf tiruan yang memiliki peran sebagai tempat keluaran yang dapat dijadikan sebagai representasi sebuah pola pada beberapa penerapan[25].

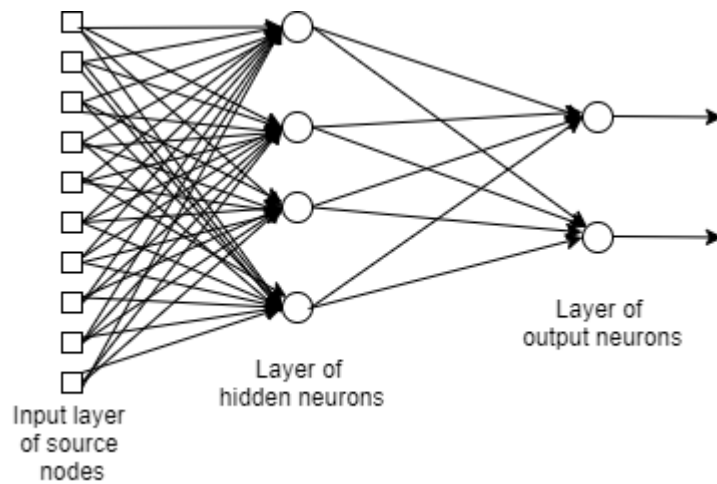
2.2.4 Arsitektur Jaringan Saraf Tiruan

Secara umum, arsitektur jaringan saraf tiruan terbagi dalam tiga jenis yaitu, layer tunggal umpan maju (*Single layer feedforward network*), multi layer umpan maju (*Multilayer feedforward network*) dan jaringan saraf tiruan berulang (*Recurrent Network*). Pada Gambar 2.5 ditunjukkan *single layer feedforward network* atau jaringan saraf tiruan dengan layer tunggal umpan maju yang memiliki *input layer* yang terhubung langsung pada *output layer*. *Input layer* disusun oleh neuron yang terhubung pada bobot *output layer* pada satu alur maju, dimana peran *output layer* yaitu menghasilkan *output* tanpa melibatkan *layer* lain pada proses komputasi[27].



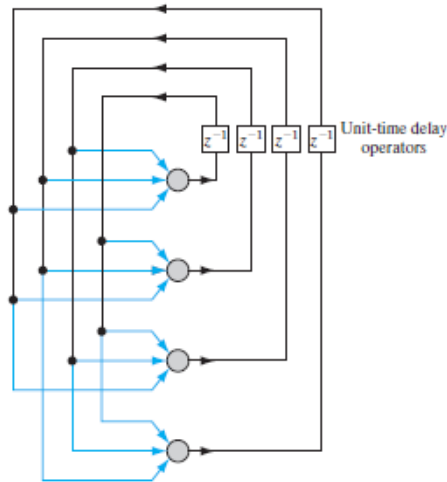
Gambar 2.5 Single Layer Feedforward Network[25]

Multilayer feedforward network (Multi layer umpan maju) yang ditunjukkan pada Gambar 2.6 merupakan arsitektur jaringan saraf tiruan yang memiliki layer tersembunyi (*hidden layer*). *Hidden layer* tersebut terletak di antara *input layer* dan *output layer* yang terdiri dari neuron-neuron tersembunyi. Neuron tersembunyi tersebut berfungsi dalam melakukan perhitungan yang dimulai pada *input layer*, lalu diteruskan kepada *output layer*. Jumlah *hidden layer* yang digunakan dapat lebih dari satu *hidden layer* yang disesuaikan dengan permasalahan ataupun kejadian yang hendak diselesaikan. Koneksi antara *input layer* dan *hidden layer* yaitu himpunan bobot, lalu koneksi antara *hidden layer* dan *input layer* terhubung dengan himpunan bobot yang memiliki proses *feedforward*, namun tidak dilakukan proses sebaliknya[27].



Gambar 2.6 Multilayer Feedforward Network[25]

Recurrent network atau yang disebut sebagai jaringan saraf tiruan berulang yang ditunjukkan pada Gambar 2.7 memiliki ciri khas yaitu adanya *looping* atau perulangan yang berjumlah paling sedikit satu *looping* (perulangan) umpan balik. Perulangan umpan balik tersebut memiliki fungsi sebagai yang menambahkan kemampuan mempelajari karakter dari data set yang digunakan dengan sifat sementara. *Recurrent network* dengan model versi Elman menggunakan proses *learning* dengan membuat salinan yang berfungsi sebagai perpanjangan *input layer*. Salinan atau *context input* merupakan salinan neuron-neuron dari *hidden layer* di *input layer*. *Context input* atau salinan ini memiliki fungsi lain yaitu sebagai penyimpan keadaan atau status sebelumnya yang dimiliki oleh *hidden layer*. Status tersebut akan diteruskan melalui penyampaian kembali kepada *hidden layer* [27].



Gambar 2.7 Recurrent Network

2.2.5 Fungsi Aktivasi

Fungsi aktivasi disebut sebagai proses yang perlu dilakukan dalam jaringan saraf tiruan pada saat data melalui neuron[29]. Fungsi aktivasi dalam jaringan saraf tiruan berperan sebagai penentu sinyal aktivasi, dimana sinyal aktivasi diperlukan pada pengaktifan dan pemadaman penjalaran sinyal dari sebuah neuron. Fungsi aktivasi juga digunakan untuk menentukan keluaran (output) dari suatu neuron[25].

Fungsi aktivasi terdiri dari beberapa jenis, diantaranya yaitu: fungsi identitas, fungsi undak biner (*hard limit*), fungsi undak biner (*threshold*), fungsi sigmoid biner dan fungsi sigmoid bipolar.

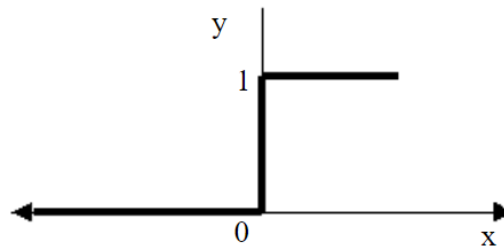
Fungsi identitas lebih sering digunakan pada jaringan lapis tunggal atau perceptron. Fungsi ini memiliki *output* dengan nilai yang sama dengan nilai *input* yang digunakan. Bentuk grafik yang akan terbentuk dari fungsi ini yaitu linier atau garis lurus. Fungsi identitas dituliskan dalam persamaan[26]:

$$f(x) = x \rightarrow \forall x \quad (2.1)$$

Fungsi aktivasi lainnya yaitu fungsi undak biner (*hard limit*) seperti pada Gambar 2.8 yang digunakan untuk melakukan perubahan terhadap nilai kontinu data *input* menjadi data *output* dengan nilai biner (0 atau 1).

Nilai biner tersebut memiliki batasan pemisah yang bernilai 0 (nol). Fungsi undak biner (*hard limit*) memiliki persamaan[29]:

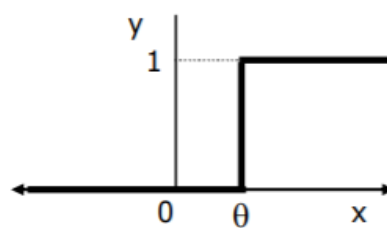
$$y = f(x) = \begin{cases} 0, & \text{jika } x < 0 \\ 1, & \text{jika } x \geq 0 \end{cases} \quad (2.2)$$



Gambar 2.8 Grafik Fungsi Aktivasi Undak Biner (*Hard Limit*)[27]

Fungsi undak biner (*threshold*) seperti pada Gambar 2.9, memiliki kemiripan dengan fungsi undak biner (*hard limit*), dimana kegunaannya yaitu untuk melakukan perubahan terhadap nilai kontinu pada data *input* agar menjadi data *output* yang bernilai biner (0 atau 1). Namun, batasan pemisah nilai biner pada fungsi undak biner (*threshold*) bernilai tidak 0 (nol). Fungsi undak biner (*threshold*) memiliki persamaan[29]:

$$y = f(x) = \begin{cases} 0, & \text{jika } x < \theta \\ 1, & \text{jika } x \geq \theta \end{cases} \quad (2.3)$$

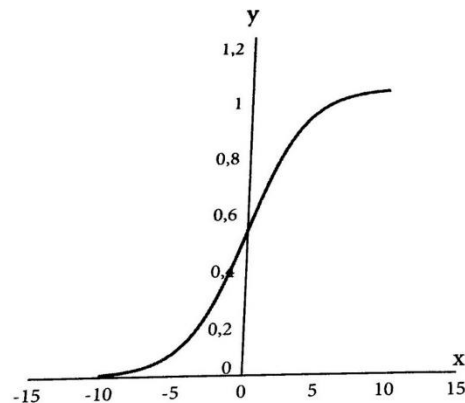


Gambar 2.9 Grafik Fungsi Aktivasi Undak Biner (*Threshold*)[27]

Fungsi sigmoid biner seperti yang ditunjukkan pada Gambar 2.10 merupakan fungsi aktivasi yang memiliki ketergantungan terhadap steepness parameter (σ) yang bernilai 1. Fungsi sigmoid biner biasanya digunakan pada jaringan saraf tiruan dengan metode propagasi balik (*backpropagation*). Pada metode propagasi balik fungsi aktivasi harus memenuhi persyaratan apabila fungsi aktivasi akan dipakai diantaranya

kontinu, diferensiabel, dan fungsinya tidak menurun secara monoton. Sigmoid biner merupakan fungsi aktivasi yang memenuhi ketiga syarat tersebut[25]. Fungsi sigmoid biner memiliki rumus[27]:

$$f(x) = \frac{1}{1+e^{-\sigma x}} \quad (2.4)$$



Gambar 2.10 Grafik Fungsi Aktivasi Sigmoid Biner[25]

Fungsi sigmoid bipolar memiliki kemiripan dengan fungsi sigmoid biner, dimana fungsi ini juga memiliki ketergantungan terhadap steepness parameter (σ). Namun, fungsi sigmoid bipolar disebut sebagai perluasan dari fungsi sigmoid biner, dimana sumbu x diperluas hingga mencapai nilai negatif. Oleh karena itu, steepness parameter yang bernilai 1 membuat fungsi sigmoid bipolar menghasilkan nilai *output* antara -1 sampai 1. Fungsi sigmoid bipolar memiliki rumus[26]:

$$g(x) = 2f - 1 = \frac{2}{1+e^{-\sigma x}} - 1 = \frac{1-e^{-\sigma x}}{1+e^{-\sigma x}} \quad (2.5)$$

2.2.6 Propagasi Balik (*Backpropagation*)

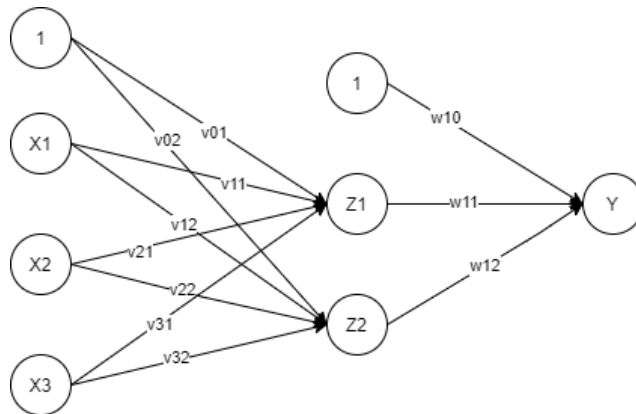
Jaringan saraf tiruan *backpropagation* atau jaringan saraf tiruan propagasi balik termasuk dalam metode yang tidak jarang digunakan untuk melakukan pemecahan masalah yang berstatus cukup kompleks. Permasalahan yang dapat diatasi dengan metode *backpropagation*, salah satunya yaitu prediksi atau peramalan. Propagasi balik atau *backpropagation* juga memiliki sebutan lain yaitu aturan delta yang

mengalami generalisasi (*generalized delta rule*). Jaringan ini bekerja dengan memasukan *input* pada jaringan untuk kemudian menghasilkan *output*. Nilai *output* aktual dan *output* target yang memiliki nilai perbedaan akan mengalami propagasi balik atau dikembalikan kepada *hidden layer* yang digunakan sebagai input dalam modifikasi terhadap bobot, sehingga *output* dari jaringan semakin dekat dengan nilai *output* target [25].

Backpropagation termasuk dalam jenis arsitektur jaringan saraf tiruan yang memiliki banyak lapisan yaitu *multilayer feedforward network*. Pada *multilayer feedforward network*, setiap *input layer* memiliki koneksi dengan *hidden layer* (lapisan tersembunyi) dan setiap *hidden layer* memiliki koneksi dengan *output layer* [25].

Jumlah neuron pada *hidden layer* termasuk dalam parameter penentuan arsitektur jaringan saraf tiruan *backpropagation*. Penggunaan jumlah neuron pada *hidden layer* yang terlalu banyak dapat menyebabkan *overfitting*, dimana adanya ketidakseimbangan antara kemampuan pemrosesan informasi yang dimiliki jaringan saraf tiruan dengan jumlah terlalu banyak dan keterbatasan jumlah informasi pada *training set*, akibatnya neuron pada *hidden layer* tidak dapat dilatih seluruhnya. Sementara itu, penggunaan jumlah neuron pada *hidden layer* yang terlalu sedikit akan menyebabkan *underfitting*, dimana neuron terlalu sedikit untuk mendapatkan informasi sinyal pada data yang digunakan. Pada penentuan jumlah neuron yang akan digunakan pada *hidden layer*, disebutkan beberapa aturan, yaitu[30]:.

- Jumlah neuron pada *hidden layer* berada di antara ukuran *input layer* dan ukuran *output layer*.
- Jumlah neuron pada *hidden layer* yaitu $2/3$ dari ukuran *input layer* ditambahkan dengan ukuran *output layer*.
- Jumlah neuron pada *hidden layer* yaitu lebih kecil dari dua kali ukuran *input layer*.



Gambar 2.11 Arsitektur Backpropagation

Pada Gambar 2.11 unit masukan (input) x_i memiliki bobot garis yaitu v_{ji} yang menjadi penghubung unit masukan x_i dengan unit z_j , dimana unit bias selalu bernilai 1 dan memiliki bobot garis yaitu v_{j0} sebagai penghubung antara unit bias dengan unit z_j . Unit *hidden layer* z_j memiliki bobot garis yaitu w_{kj} sebagai penghubung antara unit *hidden layer* z_j dengan unit keluaran (output) Y_k dan unit bias memiliki bobot garis w_{k0} sebagai penghubung antara unit bias dengan unit keluaran y_k [25]. Inisialisasi bobot yang digunakan pada jaringan saraf tiruan *backpropagation* mempengaruhi kemampuan pelatihan jaringan saraf tiruan, secara umum bobot yang digunakan dalam jaringan saraf tiruan yaitu nilai acak antara -1 dan +1, kemudian nilai acak antara -5 dan +5[31].

2.2.7 Tahapan *Backpropagation*

Jaringan saraf tiruan *backpropagation* memiliki tiga fase yang terbilang umum di dalamnya, yaitu fase umpan maju (*feedforward*), fase propagasi mundur (*backpropagation*) dan fase perubahan bobot. Fase maju atau *feedforward* merupakan fase yang menunjukkan perhitungan semua pola masukan dengan cara maju yang dimulai dengan perhitungan dari *input layer* sampai pada *output layer* yang memanfaatkan fungsi aktivasi yang akan digunakan. Fase propagasi mundur atau *backpropagation* yaitu sebuah fase yang menunjukkan perhitungan selisih di antara keluaran jaringan dan keluaran target, dimana dalam proses

tersebut dilakukan pemeriksaan kesamaan dari nilai keluaran jaringan dengan nilai keluaran target. Kemudian, selisih atau yang disebut sebagai *error* mengalami proses propagasi mundur dengan menghitung besarnya *error* (kesalahan) yang berguna dalam fase perubahan bobot, dimana bobot garis mengalami perbaikan dengan memperbaharui (*update*) nilai semua bobot untuk mengurangi nilai *error* [25].

Penjelasan lebih lanjut mengenai tahapan-tahapan pada jaringan saraf tiruan *backpropagation* dengan fungsi aktivasi sigmoid biner pada range [0,1] yaitu sebagai berikut[29]:

Langkah 1 : Inisialisasi seluruh bobot garis secara *random* atau acak, maksimal iterasi, dan toleransi *error*.

Langkah 2 : Lakukan selama kondisi berhenti masih belum terpenuhi, mulai dari tahapan alur maju (*feedforward*)

Fase Satu : Fase Maju atau *feedforward*

Langkah 3 : Setiap unit input menerima sinyal dan diteruskan ke unit tersembunyi (hidden layer).

Langkah 4 : Pada unit tersembunyi z_j ($j = 1, \dots, p$) menghitung jumlah isyarat masukan sehingga menghasilkan keluaran :

$$z_{netj} = v_j0 + \sum x_i v_{kj} \quad (2.6)$$

Lalu, menghitung fungsi aktivasi sigmoid biner pada unit tersembunyi,

$$z_j = f(z_{netj}) = \frac{1}{(1+e^{-z_{netj}})} \quad (2.7)$$

Langkah 5 : Pada unit output y_k ($k=1, \dots, m$) dihitung nilai keluarannya.

$$y_{netk} = w_k0 + \sum z_j w_{kj} \quad (2.8)$$

Lalu, menghitung fungsi aktivasi sigmoid biner pada unit keluaran.

$$y_k = f(y_{netk}) = \frac{1}{(1+e^{-y_{netk}})} \quad (2.9)$$

Fase Dua : Propagasi Mundur/Backpropagation

Langkah 6 : Nilai keluaran dari dari tiap-tiap unit output Y_k ($k = 1, \dots, m$) dihitung nilai error:

$$\delta k = (tk - yk)f'(y_{netk}) = (tk - yk)yk(1 - yk) \quad (2.10)$$

Faktor δk merupakan nilai error yang dipakai pada perubahan bobot layar dibawahnya. Hitung nilai perubahan bobot W_{kj} dimana nilai tersebut digunakan untuk merubah bobot W_{kj} dengan laju pemahaman α .

$$\Delta w_{kj} = \alpha \delta k w_{kj} \quad (2.11)$$

Dengan : $k = 1, \dots, m$ dan $j = 0, \dots, p$

Langkah 7 : Faktor δ yang terdapat pada unit layar tersembunyi (hidden layer) dihitung berdasarkan error disetiap unit tersembunyi z_j ($j = 1, \dots, p$).

$$\delta_{netj} = \sum \delta_j w_{kj} \quad (2.12)$$

Faktor error δ di unit tersembunyi :

$$\delta_j = \delta_{netj} f'(z_{netk}) = \delta_{netj} z_j (1 - z_j) \quad (2.13)$$

Hitung selisih perubahan bobot V_{ji} :

$$\Delta v_{ji} = \alpha \delta_j x_i \quad (2.14)$$

Dengan : $j = 1, \dots, p$ dan $i = 0, 1, \dots, n$

Fase Tiga : Perubahan Bobot

Langkah 8 : Hitung seluruh pergantian bobot

Pergantian bobot garis yang mengarah ke unit output :

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

Dengan : $k = 1, \dots, m$ dan $j = 0, 1, \dots, p$

Pergantian bobot garis yang mengarah ke unit tersembunyi :

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

Dengan : $j = 1, \dots, p$ dan $i = 0, 1, \dots, n$.

2.2.8 *Swarm Intelligence*

Swarm Intelligence atau kecerdasan berkelompok adalah disiplin keilmuan dari sistem cerdas yang berhubungan dengan sistem alami dan buatan, yang terdiri dari banyak individu (populasi) yang berkoordinasi menggunakan konsep kontrol desentralisasi (kecerdasan sosial dalam berkelompok) dan *self-organized* (kecerdasan personal/ terorganisir secara mandiri). *Swarm Intelligence* secara singkat bisa juga disebut sebagai

kecerdasan berkelompok[32]. Secara umum konsep *swarm intelligence* dapat di-petakan seperti Gambar 1.2[32].



Gambar 2.12 Konsep Dasar *Swarm Intelligence*

Gambar 2.12 menunjukkan konsep dasar *Swarm Intelligence* yaitu desentralisasi terhadap kumpulan permasalahan dengan karakter yang sulit dilakukan pemodelan dalam bentuk matematika kompleks dengan waktu yang sangat lama. *Swarm Intelligence* merupakan *nature-inspired meta-heuristics*, dimana algoritma yang termasuk dalam *swarm intelligence* terinspirasi dari alam. Algoritma *swarm intelligence* memiliki kemampuan mengoptimasi pencarian solusi yang berbeda dengan berbagai teknik di dalamnya yang didasarkan pada intuisi maupun aturan-aturan yang ada melalui pendekatan empiris[32].

2.2.9 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) merupakan algoritma yang termasuk memiliki kemampuan untuk mengoptimasi algoritma lainnya, dimana algoritma ini terinspirasi dari kebiasaan sosial atau disebut dengan perilaku dari kelompok ikan atau burung. *Particle swarm optimization* diasumsikan sebagai kelompok burung yang melakukan pencarian makanan di sebuah daerah, dimana tidak semua burung di daerah tersebut mengetahui titik lokasi makanan yang dituju, sehingga burung yang tidak mengetahui letak makanan tersebut menyerap informasi dari burung yang tahu ataupun lebih dekat dengan lokasi makanan yang dituju.[10].

Perumpamaan kawanan burung dengan kecerdasan yang berbeda pada setiap individu, namun burung-burung tersebut akan mengikuti kebiasaan (*rule*) seperti berikut ini[33]:

1. Jarak antara seekor burung dengan burung yang lainnya tidak terlalu dekat.
2. Burung tersebut mengambil arah terbang ke tempat populasi burung yang lainnya.
3. Burung memposisikan diri sesuai dengan posisi burung yang lainnya dan menjaga jarak agar tidak terlalu jauh.

Kawanan burung memiliki perilaku sebagai hasil dari kombinasi tiga faktor sederhana, yaitu kohesi (melakukan pergerakan bersama), separasi (menjaga jarak untuk tidak terlalu dekat), dan penyesuaian (mengikuti arah tujuan yang sama dengan kawanan). *Particle swarm optimization* dikembangkan berdasarkan model pengembangan yang akan mengalami proses simulasi dalam dimensi dengan sejumlah iterasi. Proses ini menyebabkan arah posisi partikel semakin mendekat (meminimalkan atau memaksimalkan) dengan target yang dituju. Model pengembangan *Particle swarm optimization* yaitu sebagai berikut[33]:

1. Saat seekor burung mendekat ke arah target atau makanan (fungsi tujuan maksimum atau minimum, maka ketika itu juga burung mengirimkan sinyal informasi kepada burung yang lain dalam suatu populasi tertentu.
2. Burung yang lain tersebut secara tidak langsung akan mengikuti arah menuju target atau makanan.
3. Beberapa hal bergantung pada ingatan atau memori burung mengenai jalur yang pernah dilewati sebelumnya.

Particle swarm optimization termasuk dalam bagian metode metaheuristik dengan dasar populasi kelompok burung maupun ikan pada proses pencarian sebuah solusi. Setiap populasi kelompok ikan atau burung memiliki individu yang berpotensi memberikan pengaruh kepada individu lainnya Individu yang dimaksud berperan sebagai partikel atau titik pada suatu ruang dan waktu.[34].

Particle swarm optimization melakukan pencarian menggunakan populasi (*swarm*) dari individu (partikel) yang akan diperbaharui dari

iterasi. *Particle swarm optimization* memiliki beberapa parameter seperti posisi, kecepatan, kecepatan maksimum, konstanta percepatan, dan berat inersia. *Particle swarm optimization* memiliki perbandingan lebih atau bahkan pencarian kinerja lebih unggul untuk banyak masalah optimasi dengan lebih cepat dan tingkat *konvergensi* yang lebih stabil[10].

Pada algoritma PSO pencarian solusi dilakukan secara acak dari suatu populasi, dimana setiap partikel berkaitan dengan posisi dan kecepatan dalam melakukan pencarian baru secara dinamis berdasarkan perilaku mereka. Setiap partikel memiliki nilai *fitness* yang harus dievaluasi untuk setiap generasi berdasarkan *local best (pbest)* dan *global best (gbest)* yang merupakan pengalaman dari setiap partikel dalam menghasilkan solusi terbaik. Pengalaman tersebut dapat digunakan sebagai parameter *weight inertia* dalam menentukan pengaruh kecepatan sebelumnya dengan kecepatan baru.[34].

Adapun parameter yang digunakan dalam proses algoritma PSO adalah sebagai berikut[34]:

- a. Jumlah partikel (*Number of particles*) merupakan faktor yang dianggap sangat penting dalam melakukan penyelesaian masalah.
- b. Bobot Inersia (*Inertia Weight*), memainkan peran yang sangat penting dalam kecepatan partikel dari algoritma PSO.
- c. Faktor pembelajaran (*Learning factors*). Parameter *c1* merupakan pengakuan koefisien, sedangkan *c2* adalah komponen social.
- d. Rentang dan dimensi partikel (*Range and dimension of particles*). Dimensi partikel dan rentang ditentukan berdasarkan masalah yang dioptimalkan.
- e. Kecepatan (*Velocity*) merupakan perubahan maksimum pada setiap partikel, dapat diambil selama iterasi yang didefinisikan sebagai kecepatan maksimum.
- f. Menghentikan kondisi (*Stopping condition*) merupakan salah satu cara apabila kriteria yang dicari sudah tercapai.

Istilah yang digunakan dalam penerapan algoritma PSO sebagai berikut[34]:

- a. *Swarm* : populasi dari sekawanan partikel
- b. *Particle* : individu pada suatu *swarm*. Setiap partikel mempresentasikan suatu solusi dari permasalahan yang diselesaikan.
- c. *Pbest* : suatu partikel yang menunjukkan posisi terbaik.
- d. *Gbest* : posisi terbaik dari seluruh partikel yang ada dalam suatu *swarm*.
- e. *Velocity* : kecepatan yang dimiliki oleh setiap partikel dalam menentukan arah perpindahan suatu partikel untuk memperbaiki posisi semula.
- f. c_1 dan c_2 : c_1 merupakan konstanta pembelajaran kognitif, dan c_2 konstanta pembelajaran sosial

Metode optimasi yang didasarkan pada *swarm intelligence* ini disebut algoritma *behaviorally inspired* sebagai alternatif dari algoritma genetika, yang sering disebut *evolution-based procedures*. Algoritma PSO ini awalnya diusulkan oleh J. Kennedy dan R. C. Eberhart. Pada konteks optimasi multi-variabel, kawanan diasumsikan mempunyai ukuran tertentu atau tetap dengan setiap partikel posisi awalnya terletak di suatu lokasi yang acak dalam ruang multidimensi. Setiap partikel diasumsikan memiliki dua karakteristik posisi dan kecepatan[35].

PSO memiliki tiga komponen utama diantaranya: partikel, komponen kognitif dan komponen sosial, serta kecepatan partikel. Setiap partikel merepresentasikan solusi penyelesaian. Pembelajaran partikel terdiri dari dua faktor yaitu pengalaman partikel (disebut *cognitive learning*) dan kombinasi pembelajaran dari keseluruhan *swarm* (disebut *social learning*). *Cognitive learning* sebagai *pBest* yaitu posisi terbaik yang pernah dicapai sebuah partikel, sedangkan *social learning* sebagai *gBest* yaitu posisi terbaik dari keseluruhan partikel dalam *swarm*. *pBest* dan *gBest* untuk menghitung kecepatan partikel, kecepatan untuk menghitung posisi selanjutnya[35].

2.2.10 Tahapan *Particle Swarm Optimization*

Tahapan *Particle swarm optimization* terdiri dari beberapa langkah, sebelum melakukan penjabaran tahapan tersebut perlu dibentuk sebuah fungsi. Sebagai contoh, fungsi diasumsikan seperti di bawah ini[33]:

$$\min F(X) \text{ dengan } X(B) \leq X \leq X(B) \quad (2.15)$$

dimana,

X : posisi awal

$F(X)$: fungsi tujuan

$X(B)$: batas bawah dari X

$X(A)$: batas atas dari X

Penjabaran tahapan *Particle swarm optimization* yaitu sebagai berikut:

Langkah 0: Proses Inisialisasi

Proses inisialisasi dilakukan untuk membangkitkan himpunan solusi baru secara acak/random yang terdiri atas sejumlah string dimensi partikel dan ditempatkan pada penampungan yang disebut populasi. Dalam tahap ini harus ditentukan ukuran populasi (*popSize*). Inisialisasi ukuran swarm (jumlah partikel) bertujuan untuk mengurangi jumlah evaluasi fungsi yang dibutuhkan dan juga untuk menemukan solusi. Ukuran swarm disarankan tidak terlalu besar nilainya, agar perhitungan tidak semakin lebih panjang. Ukuran swarm juga sebaiknya tidak terlalu kecil, sehingga banyak kemungkinan posisi menuju solusi terbaik atau optimal. Secara umum ukuran swarm yang digunakan antara 20 sampai 30[33].

Sebagai contoh, partikel ke- i dinyatakan sebagai: $x_i = (x_{i1}, x_{i2}, \dots, x_{ij})$ dalam ruang d -dimensi. Posisi terbaik sebelumnya dari partikel ke- i disimpan dan dinyatakan sebagai $pbest_i = (pbest_{i,1}, pbest_{i,2}, \dots, pbest_{i,j})$. Indeks partikel terbaik diantara semua partikel dalam kawanan group dinyatakan sebagai $gbest_j$. Kecepatan partikel dinyatakan sebagai: $v_i = (v_{i,1}, v_{i,2}, \dots, v_{i,j})$ [36].

Langkah 1: Proses Update Kecepatan

Kecepatan partikel dapat bergerak menuju nilai yang besar dengan cepat, terutama pada saat posisi partikel berada jauh dengan posisi tetangga dan individu terbaik. Hal ini dapat menyebabkan partikel tersebut memiliki kecenderungan berada di luar batas solusi. Berdasarkan hal tersebut, diperlukan inisialisasi terhadap batas nilai kecepatan atas (*velocity maximum*) dan batas nilai kecepatan bawah (*velocity maximum*) untuk mencegah partikel berada terlalu jauh dari batas solusi. Batasan kecepatan yang digunakan yaitu[32]:

$$vmax_j = \frac{k(xmax_j - xmin_j)}{2} \quad k \in (0,1) \quad (2.17)$$

$$vmin_j = -vmax_j \quad (2.18)$$

dimana,

jika $v_{i,j}^{t+1} > vmax_j$ maka $v_{i,j}^{t+1} = vmax_j$

jika $v_{i,j}^{t+1} < -vmax_j$ maka $v_{i,j}^{t+1} = -vmax_j$

Update Kecepatan dilakukan untuk menentukan arah perpindahan posisi partikel yang ada di populasi. Perhitungan update kecepatan yaitu[32]:

$$v_{i,j}^{t+1} = w * v_{i,j}^t + c1 * R * (Pbest_{i,j}^t - x_{i,j}^t) + c2 * R * (Gbest_{i,j}^t - x_{i,j}^t) \quad (2.19)$$

Dimana:

$v_{i,j}$ = Kecepatan partikel ke-i pada iterasi ke-i

w = Faktor bobot inersia

c1, c2 = Konstanta akselerasi (learning rate)

R = Bilangan random (0-1)

$x_{i,j}$ = Posisi saat ini dari partikel ke-i pada iterasi ke-j

$pbest_i$ = Posisi terbaik sebelumnya dari partikel ke-i

$gbest_j$ = Partikel terbaik diantara semua partikel dalam satu kelompok atau populasi

t = iterasi

Langkah 2: Proses Update Posisi dan Hitung Fitness

Setelah menghitung kecepatan dilakukan perhitungan posisi dengan rumus sebagai berikut[32]:

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} \quad (2.20)$$

Dimana:

$x_{i,j}^t$ = Posisi pada iterasi ke t

$v_{i,j}^{t+1}$ = Kecepatan pada iterasi ke t+1

Kemudian, setelah menghitung posisi, dilanjutkan dengan menghitung *fitness*. Nilai *fitness* merupakan nilai partikel yang berupa fungsi x dimana x merupakan posisi dari partikel.

Langkah 3: Proses Update *Pbest* dan *Gbest*

Update *Pbest* dilakukan dengan cara membandingkan antara *Pbest* dan *fitness* pada iterasi sebelumnya dengan hasil dari update posisi dengan persamaan sebagai berikut[32]:

$$Pbest_{i,j}^{t+1} = \begin{cases} Pbest_{i,j}^t & \text{jika } fitness(x_{i,j}^{t+1}) \leq fitness Pbest_{i,j}^t \\ x_{i,j}^{t+1} & \text{jika } fitness(x_{i,j}^{t+1}) > fitness Pbest_{i,j}^t \end{cases} \quad (2.21)$$

Langkah 1,2 dan 3 dilakukan secara berulang kali sampai mencapai iterasi maksimum atau telah mencapai kondisi konvergen. Kondisi konvergen yaitu ketika posisi seluruh partikel mengarah ke nilai yang sama. Secara umum, kondisi berhenti ditentukan, seperti jumlah selisih solusi pada saat ini atau sekarang dengan solusi sebelumnya yang bernilai kecil[33].

2.2.11 Normalisasi dan Denormalisasi

Normalisasi data pada prediksi menggunakan jaringan saraf tiruan *Backpropagation* merupakan salah satu proses pengubahan nilai-nilai pada data yang digunakan kedalam sebuah range tertentu. Data yang dilakukan perubahan nilai adalah data latih dan data uji, dimana jaringan saraf tiruan hanya mampu melakukan pengolahan terhadap nilai antara -1 sampai dengan 1. Metode yang digunakan untuk melakukan normalisasi data yaitu metode *min-max normalization* untuk membuat data asli menjadi nilai dalam range 0,1 sampai 0,9 dengan rumus di bawah ini[25]:

$$X' = \frac{0,8(X-b)}{a-b} + 0,1 \quad (2.22)$$

Dimana,

X' = data ternormalisasi

X = data aktual

a = nilai maksimum data asli

b = nilai minimum data asli

Data hasil normalisasi yang sudah melewati proses pelatihan dan pengujian akan dilakukan proses denormalisasi untuk mengembalikan nilai menjadi bentuk nilai data awal sebelum dinormalisasi.[37] Rumus denormalisasi data yaitu[38]:

$$y' = \left(\frac{(y-0,1)(max-min)}{0,8} \right) + min \quad (2.23)$$

Dimana,

y' : output yang telah didenormalisasi

y : output

max : data dengan nilai tertinggi

min : data dengan nilai terendah

2.2.12 Mean Squared Error (MSE)

Mean Squared Error (MSE) disebut sebagai salah satu metode untuk mengevaluasi prediksi atau peramalan. Nilai MSE yang semakin kecil menunjukkan keakuratan prediksi yang semakin besar. MSE dapat mengatur tingkat kesalahan peramalan dengan nilai tinggi karena dalam MSE dilakukan pengkuadratan pada kesalahan[39]. Berikut ini adalah rumus perhitungan nilai MSE[40]:

$$MSE = \frac{\sum_{t=1}^n |a-b|^2}{n} \quad (2.24)$$

Dimana,

a = Nilai aktual

b = Nilai hasil peramalan

n = Jumlah data