

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Hujan

Hujan adalah proses pengembalian air yang telah diuapkan ke atmosfer menuju ke permukaan bumi. Pengembalian ini akibat dari udara yang naik hingga melewati ketinggian kondensasi dan berubah menjadi awan. Di dalam awan terjadi proses tumbukan dan penggabungan antar butir-butir air yang akan meningkatkan massa dan volume butir air, jika butiran air akan turun dalam bentuk hujan. Agar terjadi hujan terdapat tiga faktor utama yang penting, yaitu: massa udara yang lembab, inti kondensasi (seperti partikel debu, kristal garam), dan suatu sarana sebagai tempat berlangsungnya proses pendinginan akibat udara. Pengangkatan massa ke udara ke atmosfer dapat berlangsung dengan cara-cara pendinginan siklonik, orografis, dan konvektif (Iskandar, 2012).

Menurut Iskandar (2012), tiga-tipe hujan yang umum dijumpai di daerah tropis dapat disebutkan sebagai berikut:

a. Hujan konveksional (*Convictional storms*)

Tipe hujan ini disebabkan oleh adanya beda panas yang diterima permukaan tanah dengan panas yang diterima oleh lapisan udara di atas permukaan tanah tersebut. Sumber panas di daerah tropis adalah berasal dari matahari. Beda panas ini biasanya terjadi pada akhir musim kering yang akan menyebabkan hujan dengan intensitas tinggi sebagai hasil proses kondensasi massa air basah pada ketinggian di atas 15 km. Mekanisme terjadinya hujan tipe konvektif secara singkat adalah sebagai berikut: ketika lapisan udara

di atas permukaan tanah menjadi lebih panas daripada lapisan udara di atasnya, maka berlangsunglah gerakan massa udara panas tersebut ke tempat yang lebih tinggi. Massa udara panas yang bergerak ke tempat yang lebih tinggi tersebut pada saatnya akan terkondensasi. Pada proses ini terjadi pelepasan tenaga panas yang akan menyebabkan udara menjadi tambah panas, dan dengan demikian, mendorong udara panas tersebut bergerak lebih tinggi lagi sampai ketinggian tertentu di mana uap air panas tersebut membeku dan jatuh sebagai hujan oleh adanya gravitasi. Tipe hujan konvektif biasanya dicirikan dengan intensitas yang tinggi berlangsung relative cepat, dan mencakup wilayah yang tidak terlalu luas. Tipe hujan konvektif inilah yang seringkali digunakan untuk membedakan dari tipe hujan yang sering dijumpai di daerah beriklim sedang (tipe hujan frontal) dengan intensitas hujan lebih sedang.

b. Hujan Frontal (*Frontal/cyclonic storms*)

Tipe hujan yang umumnya disebabkan oleh bergulungnya dua massa udara yang berbeda suhu dan kelembapan. Pada tipe hujan ini, massa udara lembap yang hangat dipaksa bergerak ke tempat yang lebih tinggi (suhu lebih rendah dengan kerapatan udara dingin lebih besar). Tergantung pada tipe hujan yang dihasilkannya, hujan frontal dapat dibedakan menjadi hujan frontal dingin dan hangat. Hujan frontal dingin biasanya mempunyai kemiringan permukaan frontal yang besar dan menyebabkan gerakan massa udara ke tempat yang lebih tinggi cepat sehingga bentuk hujan yang dihasilkan adalah hujan lebat dalam waktu singkat. Sebaliknya, pada hujan

frontal hangat, kemiringan permukaan frontal tidak terlalu besar sehingga gerakan massa udara ke tempat yang lebih tinggi dapat dilakukan dengan perlahan-lahan (proses pendinginan berlangsung bertahap). Tipe hujan yang dihasilkannya adalah hujan yang tidak terlalu lebat dan berlangsung dalam waktu lebih lama (hujan dengan intensitas rendah). Hujan badai dan hujan monsoon adalah tipe hujan frontal yang lazim dijumpai.

c. Hujan Orografik (*Orographic storm*)

Jenis hujan yang umum terjadi di daerah pegunungan, yaitu ketika massa udara bergerak ke tempat yang lebih tinggi mengikuti bentang lahan pegunungan sampai saatnya terjadi proses kondensasi. Ketika massa udara melewati daerah bergunung, pada lereng di mana angin berhembus (*windward side*) terjadi hujan orografik. Sementara pada lereng dimana gerakan massa udara tidak atau kurang berarti (*leeward side*), udara yang turun akan mengalami pemanasan dengan sifat kering, dan daerah ini disebut daerah “bayangan” dan hujan yang terjadi disebut hujan di daerah “bayangan” (jumlah hujan lebih kecil daripada hujan yang terjadi di daerah *windward side*). Besarnya intensitas hujan orografik cenderung menjadi lebih besar dengan meningkatnya ketebalan lapisan udara lembap di atmosfer yang bergerak ke tempat yang lebih tinggi. Tipe hujan orografik dianggap sebagai pemasok air tanah, danau, bendungan, dan sungai karena berlangsung di daerah hulu DAS.

2.2 Karakteristik Hujan

Karakteristik hujan berupa jumlah, intensitas hujan, lama hujan, dan frekuensi hujan. Jumlah hujan yang jatuh di permukaan bumi dinyatakan dalam kedalaman air (biasanya mm), jumlah hujan dianggap terdistribusi secara merata pada seluruh daerah tangkapan air. Intensitas hujan adalah jumlah curah hujan dalam satu satuan waktu, dinyatakan dalam mm/jam, mm/hari, mm/bulan, mm/tahun dan berturut-turut sering disebut hujan jam-jaman, harian, mingguan, bulanan, tahunan dan sebagainya (Triatmodjo, 2013). Lama hujan adalah periode hujan jatuh, dinyatakan dalam menit, jam, dan lain-lain. Luas daerah yang terkena hujan biasanya dapat dianggap sama (Seyhan, 1990).

Tabel 2.1 merupakan keadaan hujan dan intensitas hujan. Tabel tersebut menunjukkan bahwa curah hujan tidak bertambah sebanding dengan waktu. Jika durasi waktu lebih lama, penambahan curah hujan adalah lebih kecil dibanding dengan penambahan waktu, karena hujan tersebut bisa berkurang atau berhenti (Triatmodjo, 2013).

Durasi hujan adalah waktu yang dihitung dari saat hujan mulai turun sampai berhenti, yang biasanya dinyatakan dalam jam. Intensitas hujan rerata adalah perbandingan antara kedalaman hujan dengan intensitas hujan, misalnya hujan dalam 5 jam menghasilkan kedalaman 5 mm, yang berarti intensitas hujan rerata adalah 10 mm/jam. Demikian juga hujan dalam 5 menit sebesar 6 mm, yang berarti intensitas reratanya adalah 72 mm/jam (Triatmodjo, 2013).

Tabel 2.1 Keadaan hujan dan intensitas hujan

Keadaan Hujan	Intensitas Hujan	
	1 Jam	24 Jam
Hujan Sangat Tinggi	<1	<5
Hujan Ringan	1-5	5-20
Hujan Normal	5-10	20-50
Hujan Lebat	10-20	50-100
Hujan Sangat Lebat	>20	>100

Menurut Linsley (1996), bentuk-bentuk hujan adalah sebagai berikut:

1. Gerimis (*drizzle*), yang kadang-kadang disebut *mist*, terdiri dari tetes-tetes air yang tipis, biasanya dengan diameter antara 0,1 dan 0,5 mm, dengan kecepatan jatuh yang demikian lambatnya sehingga kelihatan seolah-olah melayang dengan intensitas jarang melebihi 1 mm/jam.
2. Hujan (*Rain*), terdiri dari tetes-tetes air yang mempunyai diameter lebih besar dari 0,05 mm.
3. *Glase* dalam selimut es, biasanya bersih dan halus, yang terbentuk pada permukaan yang terbuka oleh pembekuan atau air yang sangat dingin yang diendapkan oleh hujan atau gerimis. Berat jenisnya dapat mencapai 0,8 sampai 0,9.
4. Rime adalah endapan butiran es yang tak tembus cahaya dan berwarna putih, yang kurang lebih dipisahkan oleh udara yang tertangkap dan terbentuk oleh pembekuan air dingin dengan sangat cepat menimpa benda-benda yang terbuka. Berat jenisnya dapat serendah 0,2 sampai 0,3.

5. Salju adalah campuran kristal-kristal es yang sebagian besar berbentuk heksagonal yang kompleks dan bercabang, dan umumnya menggumpal menjadi kumpulan salju (*snowflake*), diameternya dapat mencapai beberapa inci. Berat jenis rata-rata nya sering dianggap sebesar 0,1.
6. Hujan es (*hail*) adalah hujan dalam bentuk bola-bola es, yang dihasilkan dalam awan-awan konvektif, kebanyakan *cumulonimbus*. Batu-batu es (*hailstones*) dapat berbentuk sferadional, kerucut, atau bentuk yang tidak beraturan, dan diameternya berkisar dari sekitar 5 sampai 125 mm. berat jenisnya sekitar 0,8 mm.
7. *Sleet* (hujan yang bercampur es dan salju) terdiri dari butir-butir es yang bulat, pejal, dan tembus cahaya, yang terbentuk oleh pembekuan tetes air hujan yang turun atau pembekuan kembali sebagai besar kristal es yang mencair yang jatuh melalui suatu lapisan udara dengan temperature di bawah titik beku di dekat permukaan bumi.

2.3 Data Hujan

Besaran hujan merupakan masukan terpenting dalam analisis curah hujan, sehingga dapat dipahami apabila kesalahan yang terbawa dalam data hujan terlalu besar maka hasil analisisnya pantas diragukan (Agustin, 2010).

Seringkali data hujan dari suatu stasiun hujan tidak terekam, dan jika itu terjadi akan sangat merugikan karena berpengaruh pada kevalidan data. Menurut Agustin (2010), satu seri data hujan untuk satu stasiun tertentu, dimungkinkan sifatnya tidak valid. Data semacam ini tidak dapat langsung digunakan dalam analisis. Ketidakvalidan dapat saja terjadi karena berbagai sebab, yaitu:

1. Alat ukur yang diganti spesifikasi yang berbeda atau alat yang sama, tetapi dipasang dengan patokan aturan yang berbeda.
2. Alat ukur dipindahkan dari tempat semula, tetapi secara administratif nama stasiun tersebut tidak berubah, misalnya karena masih dalam satu desa yang sama.
3. Alat ukur sama, tempat tidak dipindahkan, tetapi lingkungan berubah, misalnya semula dipasang di tempat ideal menjadi berubah karena bangunan atau pohon besar.

Satuan di dalam mengukur curah hujan adalah millimeter. Jumlah curah hujan 1 mm, menunjukkan tinggi air hujan yang menutupi permukaan sebesar 1 mm zat cair dan tidak meresap ke dalam tanah atau menguap ke atmosfer (Tjasyono, 2004).

Tujuan utama dari setiap metode pengukuran adalah untuk mendapatkan contoh yang benar-benar mewakili seluruh kawasan, karena itu di dalam memasang suatu penakar haruslah dijamin dengan persyaratan di bawah ini (Seyhan, 1990):

- a. Percikan tetesan hujan ke dalam atau ke luar penampung harus dicegah.
- b. Kehilangan air dari reservoir oleh penguapan haruslah se-minimal mungkin.
- c. Jika ada, hujan haruslah melebur.

Pemilihan suatu tipe penakar hujan tertentu dan lokasinya di suatu tempat bergantung pada beberapa faktor, diantaranya disebutkan di bawah ini (Seyhan, 1990):

- a. Dapat dipercaya (ketelitian pengukuran)
- b. Tipe data yang diperlukan (menit, harian, dan lain-lain)
- c. Tipe yang akan diukur (adanya salju, tebal salju)
- d. Dapat diperbandingkan dengan penakar hujan lain yang ada
- e. Biaya instalasi dan perawatan
- f. Mudahnya perawatan (deteksi kebocoran)
- g. Mudahnya pengamatan
- h. Gangguan oleh hewan dan manusia

Sesudah tipe penakar hujan dipilih, maka langkah selanjutnya adalah memutuskan jumlah minimum penakar yang dibutuhkan untuk suatu kawasan. Pengajuan ini tergantung pada maksud tujuan penelitian, posisi geografis kawasan (aspek iklim mikro seperti pengaruh orografi), dan urbanisasi kawasan tersebut (Seyhan, 1990).

Alat penakar curah hujan dinamakan pluviometer atau penakar hujan (*rain gauge*). Alat penakar hujan menurut Seyhan (1990), diklasifikasi berdasarkan suatu kombinasi pendekatan yaitu:

1. Penakar hujan bukan pencatat, diantaranya yaitu:
 - a. Penakar hujan baku (standar).
 - b. Penakar hujan penyimpanan (atau penjumlah).
 - c. Penakar hujan searah tanah.
 - d. Penakar hujan acuan internasional (*International Reference Precipitation Gauge*).
 - e. RADAR (*Radio Detecting and Ranging*)

2. Penakar hujan otomatis (*Automatic Rainfall Recorder*)

Semua penakar hujan otomatis akan mencatat data (dalam jumlah hujan) secara kontinu (interval 1 menit, 5 menit, 10 menit, dan lain-lain) maupun secara berkala pada beberapa macam grafik, pita berlubang, pita magnet, film, sinyal-sinyal listrik, dan lain-lain. Berikut ini adalah macam alat penakar hujan otomatis yang dipasang di tanah:

- a. Penakar hujan otomatis tipe penimbangan.
- b. Penakar hujan otomatis tipe pelampung.
- c. Penakar hujan otomatis tipe ember-tumpah (*tipping-bucket*).
- d. Pengindera jauh.

2.4 *Artificial Neural Network*

Usaha manusia dalam mengembangkan suatu sistem yang meniru kemampuan dan perilaku makhluk hidup telah berlangsung selama beberapa dekade belakang ini. *Artificial neural network* merupakan hasil perkembangan ilmu dan teknologi yang kini sedang berkembang pesat. *Artificial neural network* yang berupa susunan sel-sel *neural network* (*neuron*) dibangun berdasarkan prinsip-prinsip organisasi otak manusia. Perhatian yang besar pada *artificial neural network* disebabkan adanya keunggulan yang dimilikinya seperti kemampuan untuk belajar komputasi paralel, kemampuan untuk memodelkan fungsi non linier dan sifat *fault tolerance* (Kusumadewi, 2003).

Artificial neural network adalah suatu metode pengelompokan dan pemisahan data yang prinsip kerjanya sama seperti *neural network* pada manusia. Elemen mendasar dari paradigma tersebut adalah struktur yang baru dari sistem

pemrosesan informasi. *Artificial neural network* dibentuk untuk memecahkan suatu masalah tertentu seperti pengenalan pola atau klasifikasi karena proses pembelajaran (Puspitaningrum, 2006).

Perancangan suatu *artificial neural network* selain memperhatikan struktur hubungan antara simpul masukan dengan simpul keluaran, perlu ditentukan juga cara atau metode pembelajarannya. Belajar bagi *artificial neural network* adalah cara memperbaharui bobot *sinapsis* disesuaikan dengan isyarat masukan dan keluaran yang diharapkan. Secara umum suatu *artificial neural network* dibentuk atas sejumlah *neuron* sebagai unit pengolah informasi sebagai dasar operasi untuk menjalankan fungsi atau tugasnya (Siang, 2005).

Sesuai dengan sistem kerjanya, struktur *artificial neural network* terdiri dari tiga lapisan yaitu lapisan *input*, lapisan antara *input* dan *output* atau lapisan tersembunyi dan lapisan *output*. Masing-masing lapisan diberikan pembobot yang akan mentransformasi nilai *input* menjadi nilai *output*. Setiap lapisan terdiri dari beberapa *neuron* dan antar *neuron* ini akan terhubung dengan *neuron-neuron* lain pada lapisan terdekat (Ripley, 1996).

Artificial neural network mempunyai beberapa kelebihan di antaranya:

- a) *Artificial neural network* bersifat non linier

Jaringan cocok digunakan untuk sistem-sistem dengan kompleksitas permasalahan yang tinggi.

- b) *Artificial neural network* bersifat adaptif

Jaringan belajar dari data yang diberikan padanya dan menghasilkan pemecahan (hubungan pemetaan) antara masukan dan keluarannya.

- c) *Artificial neural network* bersifat tahan terhadap kesalahan

Jaringan secara keseluruhan masih mampu bekerja walaupun terjadi kesalahan/kerusakan sampai tingkat tertentu.

- d) *Artificial neural network* dapat melakukan generalisasi

Jaringan dapat mengolah data yang belum pernah dilatihkan padanya berdasarkan apa yang telah diperoleh selama belajar.

Fausett (1994) menyatakan bahwa *artificial neural network* pada umumnya digunakan untuk tugas atau pekerjaan yang kurang praktis jika dikerjakan secara manual. Kegunaan *artificial neural network* dalam kehidupan nyata antara lain:

- a. Perkiraan fungsi atau analisis regresi, peramalan dan termasuk prediksi *time series* serta *modeling*.
- b. Klasifikasi, termasuk pengenalan pola dan pengenalan urutan serta pengambilan keputusan dalam pengurutan.
- c. Pengolahan data, termasuk di dalamnya penyaringan, pengelompokan dan kompresi.
- d. Robotik.

Menurut Siang (2005) aplikasi *artificial neural network* adalah sebagai berikut:

- a. Pengenalan pola (*Pattern Recognition*)

Artificial neural network dipakai untuk mengenali pola (misalnya: huruf, angka, suara dan tanda tangan) yang sudah sedikit berubah. Hal ini mirip dengan otak manusia yang masih mampu mengenali orang yang sudah beberapa waktu tidak dijumpainya (mungkin wajah/bentuk tubuhnya sudah sedikit berubah).

b. *Signal Processing*

Artificial neural network dapat dipakai untuk menekan *noise* dalam saluran telepon.

c. Peramalan

Artificial neural network juga dapat dipakai untuk meramalkan apa yang akan terjadi di masa yang akan datang berdasarkan pola kejadian yang

2.5 Arsitektur *Artificial Neural Network*

Pemodelan struktur pemrosesan informasi terdistribusi dilakukan dengan menentukan pola hubungan antar sel-sel saraf buatan. Pola hubungan yang umum adalah hubungan antar *layer* (lapisan). Setiap lapisan terdiri dari sekumpulan sel saraf buatan (*unit*) yang memiliki fungsi tertentu, misalnya fungsi masukan (*masukan*) atau fungsi keluaran (*keluaran*).

Sistem saraf buatan terdiri dari tiga lapisan unit, yaitu:

1. Unit masukan

Pada Gambar 2.6.1 unit masukan dinotasikan dengan i . Unit masukan ini menerima data dari jaringan saraf luar. Aktivasi unit-unit lapisan masukan menunjukkan informasi dasar yang kemudian digunakan dalam jaringan saraf buatan.

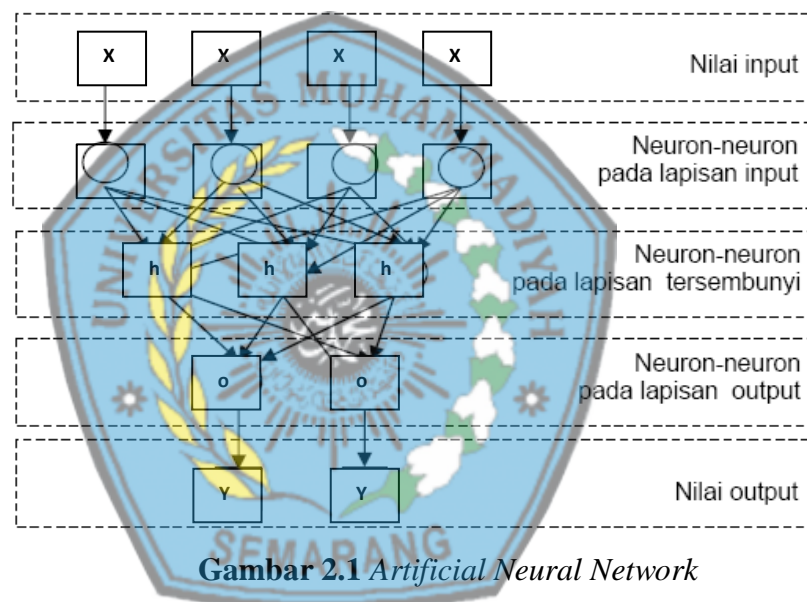
2. Unit tersembunyi

Unit tersembunyi dinotasikan dengan h pada Gambar 2.1. Unit tersembunyi menerima dan mengirim sinyal ke jaringan saraf. Aktivasi setiap unit-unit lapisan tersembunyi ditentukan oleh aktivasi dari unit-unit

masukan dan bobot dari koneksi antara unit-unit masukan dan unit-unit lapisan tersembunyi.

3. Unit keluaran

Unit keluaran dinotasikan dengan o . Unit keluaran mengirim data ke jaringan saraf. Karakteristik dari unit-unit keluaran tergantung dari aktivasi unit-unit lapisan tersembunyi dan bobot antara unit-unit lapisan tersembunyi dan unit-unit keluaran. Keluaran ini biasanya berupa nilai diskrit yang merupakan hasil klasifikasi dari data masukan.



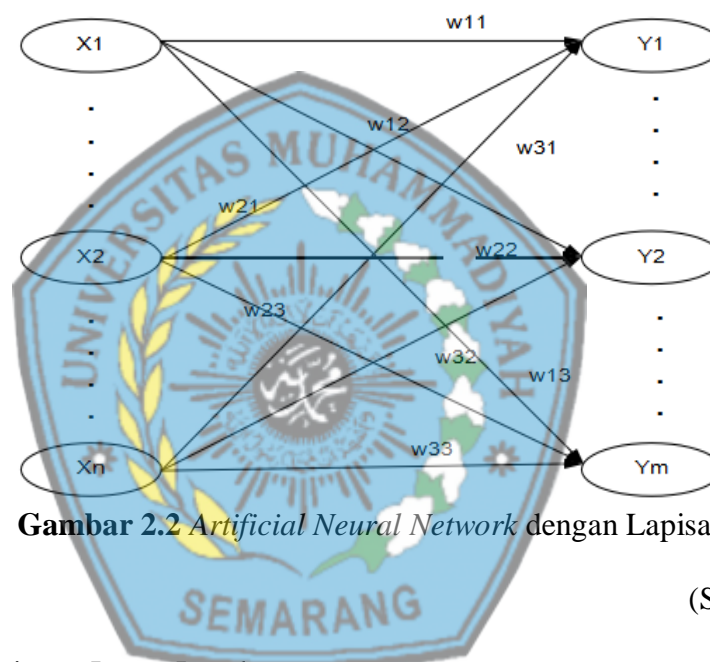
Sedangkan tipe arsitektur jaringan saraf buatan ada tiga yaitu

1. Jaringan dengan lapisan tunggal (*Single Layer Net*)

Jaringan dengan lapisan tunggal terdiri dari 1 *layer input*, 1 *layer output* dan 1 lapisan bobot koneksi (Fausett, 1994). Pada jaringan ini, sekumpulan masukan *neuron* dihubungkan langsung dengan sekumpulan keluaran nya. Sinyal mengalir searah dari lapisan masukan sampai lapisan keluaran.

Setiap simpul dihubungkan dengan simpul lainnya yang berada di atasnya dan di bawahnya, tetapi tidak dengan simpul yang berada pada lapisan yang sama. Model yang masuk kategori ini antara lain: *Adaline*, *Hopfield*, *perceptron*, *LVQ*, dan lain-lain (Siang, 2005).

Pada gambar 2.2 berikut diperlihatkan jaringan dengan n unit masukan (x_1, x_2, \dots, x_n) dan m keluaran (y_1, y_2, \dots, y_m).



Gambar 2.2 Artificial Neural Network dengan Lapisan Tunggal

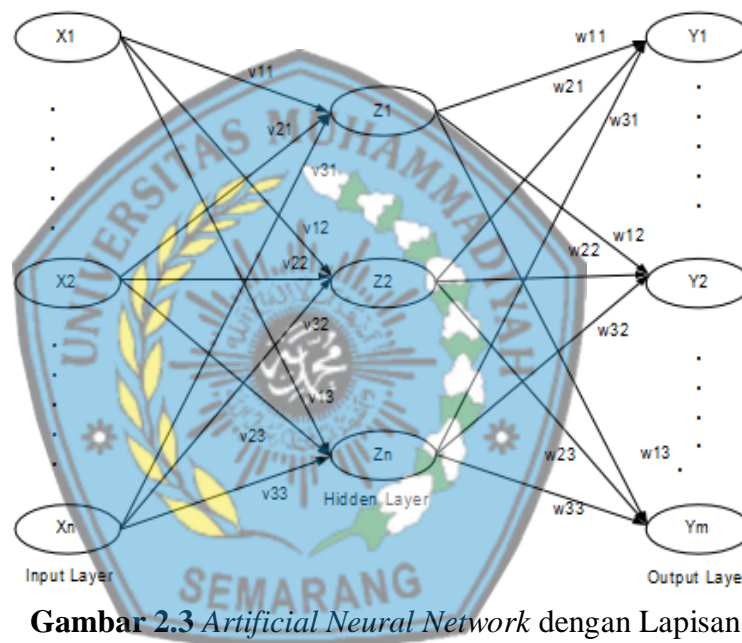
(Siang, 2005)

2. Jaringan Layar Jamak

Jaringan dengan lapisan jamak memiliki ciri khas tertentu yaitu memiliki tiga jenis lapisan yakni *layer input*, *layer output*, dan *layer* tersembunyi (Fausett, 1994). Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan jaringan dengan lapisan tunggal. Namun, proses pelatihan sering membutuhkan waktu yang cenderung lama tentu dengan pembelajaran yang lebih rumit. Akan tetapi, pada beberapa kasus pembelajaran pada jaringan yang

mempunyai banyak lapisan ini lebih sukses dalam menyelesaikan masalah. Model yang termasuk kategori ini antara lain: *Madaline* dan *Backpropagation* (Siang, 2005).

Pada gambar 2.3 berikut diperlihatkan jaringan dengan n unit masukan (x_1, x_2, \dots, x_n) dan layar tersembunyi yang terdiri dari m (y_1, y_2, \dots, y_m) dan 1 unit keluaran.



Gambar 2.3 Artificial Neural Network dengan Lapisan Jamak

(Siang, 2005)

2.6 Metode *Backpropagation*

Seperti yang telah dijelaskan di atas *artificial neural network* menyerupai otak manusia dalam dua hal, salah satunya adalah pengetahuan diperoleh jaringan melalui proses belajar. Pada otak manusia, informasi yang dilewatkan dari satu *neuron* ke *neuron* yang lainnya berbentuk rangsangan listrik melalui *dendrite*. Jika rangsangan tersebut diterima oleh suatu *neuron*, maka *neuron* tersebut akan membangkitkan *output* ke semua *neuron* yang berhubungan dengannya sampai

informasi tersebut sampai ke tujuannya yaitu terjadi suatu reaksi. Jika rangsangan yang diterima terlalu halus, maka *output* yang dibangkitkan oleh *neuron* tersebut tidak akan direspon. Tentu saja sangatlah sulit untuk memahami bagaimana otak manusia bisa belajar. Selama proses pembelajaran, terjadi perubahan yang cukup berarti pada bobot-bobot yang menghubungkan antar *neuron* (Kusumadewi, 2003).

Backpropagation atau propagasi balik merupakan suatu teknik pembelajaran atau pelatihan *supervised learning* yang paling banyak digunakan. Metode ini merupakan salah satu metode yang sangat baik dalam menangani masalah pengenalan pola-pola kompleks. Di dalam jaringan propagasi balik ini, setiap unit yang berada di lapisan *input* terhubung dengan setiap unit yang ada di lapisan *output*. Jaringan ini terdiri dari banyak lapisan. Ketika jaringan diberikan pola masukan sebagai pola pelatihan, maka pola tersebut menuju unit-unit lapisan tersembunyi untuk selanjutnya diteruskan pada unit-unit di lapisan keluaran. Kemudian unit-unit lapisan keluaran akan memberikan respons sebagai keluaran *artificial neural network*. Saat hasil keluaran tidak sesuai dengan yang diharapkan, maka keluaran akan disebarkan mundur pada lapisan tersembunyi kemudian dari lapisan tersembunyi menuju lapisan masukan (Puspitaningrum, 2006).

Kusumadewi (2003) menyatakan bahwa algoritma belajar *backpropagation* adalah sebagai berikut:

1. Perhitungan nilai keluaran *neuron* pada lapisan tersembunyi dan *output*.

2. Perhitungan kesalahan dalam proses belajar disebut dengan fungsi aktivasi.
3. Perhitungan hubungan *neuron-neuron* dalam lapisan tersembunyi maupun lapisan *output*, di mana persamaan yang digunakan untuk perhitungan hubungan baik untuk lapisan tersembunyi maupun lapisan *output*, tergantung dari fungsi aktivasi yang digunakan.
4. Perhitungan nilai perubahan bobot dan *bias*.

$$\text{Perubahan bobot: } \Delta W_{ij}(x, y) = \Delta \delta_{ij}^x \quad (2.1)$$

$$\text{Perubahan bias: } \Delta q_i(x, y) = \Delta \delta_i \quad (2.2)$$

5. Perhitungan nilai dan bobot *bias* baru.

$$\text{Bobot baru: } w_{ij}^{i+1} = w_{ij}^t + \Delta w_{ij}^t = \text{momentum} w_{ij}^{i+1} \quad (2.3)$$

$$\text{Bias baru: } q_j^{i+1} = q_j^t + \Delta q_j^t \quad (2.4)$$

6. Langkah-langkah tersebut diulang sampai didapatkan nilai keluaran kecil sehingga mencapai pemberhentian *error* yang diharapkan.

Momentum digunakan untuk mempercepat tercapainya kekonvergenan nilai kesalahan. Penyesuaian bobot yang dilakukan saat $t+1$, dipengaruhi oleh bobot yang ada pada saat t dan $t+1$, hal ini dimaksudkan untuk menjaga perubahan bobot agar terjadi dalam arah yang sama, di mana:

w_{ij}^{i+1} : bobot koneksi antara *neuron* pada lapisan tersembunyi I dan *neuron* pada lapisan keluaran j pada saat $t+1$.

w_j^t : bobot koneksi antara *neuron* pada lapisan tersembunyi I dan *neuron* pada lapisan keluaran j pada saat t .

w_{ij}^{t+1} : bobot koneksi antara *neuron* pada lapisan tersembunyi I dan *neuron* pada lapisan keluaran j pada saat $t+1$.

2.6.1 Tahapan Metode *Backpropagation*

- Inisialisasi bobot (ambil bobot awal dengan nilai random yang cukup kecil).
- Tetapkan : Maksimum *Epoch*, target *Error*, dan *Learning Rate* (α).
- Inisialisasi : $Epoch = 0$, $mean\ square\ error = 1$

Siang (2005) menyatakan bahwa algoritma pelatihan untuk jaringan *backpropagation* dengan satu layer tersembunyi (dengan fungsi aktivasi *sigmoid biner*), adalah sebagai berikut:

Langkah 1: Menetapkan nilai *input* (Kusumadewi, 2003).

Langkah 2: Membuat inisialisasi semua bobot dengan bilangan acak kecil (Siang, 2005).

Fase I : Propagasi Maju (*Forward Propagation*)

Langkah 3 : Tiap unit masukan X_i ($i = 1, 2, 3, \dots, n$) menerima sinyal X_i dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya.

Langkah 4: Tiap-tiap unit lapisan tersembunyi Z_j ($j = 1, 2, 3, \dots, p$) menjumlahkan sinyal-sinyal *input* terbobot:

$$Z_{in_j} = V_{oj} + \sum_{i=1}^n X_i V_i \quad (2.5)$$

Kemudian dihitung nilai keluaran dengan menggunakan fungsi aktivasi:

$$Z_j = f(Z_{in_j}) = \frac{1}{1 + e^{-Z_{in_j}}} \quad (2.6)$$

Dan dikirimkan sinyal tersebut ke semua unit di lapisan atasnya.

Langkah 5: Hasil fungsi tersebut dikirim ke semua unit pada lapisan di atasnya. Tiap-tiap unit keluaran Y_k ($k = 1, 2, \dots, m$) menjumlahkan sinyal-sinyal *input* terbobot:

$$Y_{in_k} = W_k + \sum_{j=1}^p Z_j W_{jk} \quad (2.7)$$

Kemudian dihitung nilai keluaran dengan menggunakan fungsi aktivasi:

$$Y_k = f(Y_{in_k}) = \frac{1}{1 + e^{-Y_{in_k}}} \quad (2.8)$$

Dan dikirimkan sinyal tersebut ke semua unit di lapisan atasnya.

Fase II : Propagasi Mundur (*Backpropagation*)

Langkah 6: Tiap-tiap unit keluaran Y_k ($k = 1, 2, \dots, m$) menerima target pola yang berhubungan dengan pola *input* pembelajaran, hitung informasi *error*.

$$\delta_k = (t_k - Y_k) f'(Y_{in_k}) \quad (2.9)$$

$$= (t_k - Y_k) y_k (1 - Y_k) \quad (2.10)$$

Kemudian dihitung koreksi nilai bobot yang kemudian akan digunakan untuk memperbaharui nilai W_{jk} :

$$\Delta W_{jk} = \alpha \delta_k Z_j \quad (2.11)$$

Hitung koreksi nilai *bias* yang kemudian akan digunakan untuk memperbaharui W_{ok} .

$$\Delta W_{ok} = \alpha \delta_k \quad (2.12)$$

Kirimkan δ_k ke unit-unit yang ada di lapisan bawahnya.

Langkah 7: Tiap-tiap unit lapisan *hidden* Z_j ($j = 1, 2, 3, \dots, p$) menjumlahkan *delta input* (dari unit-unit yang berada pada lapisan di atasnya):

$$\delta_{in_j} = \sum_{k=1}^m \delta_k W_{jk} \quad (2.13)$$

Kemudian nilai tersebut dikalikan dengan nilai turunan dari fungsi aktivasi untuk menghitung informasi kesalahan:

$$\delta_j = \delta_{in_j} f'(Z_{in_j}) \quad (2.14)$$

Hitung koreksi nilai bobot yang kemudian digunakan untuk memperbarui V_{ij} :

$$\Delta V_{ij} = \alpha \delta_j X_i \quad (2.15)$$

Hitung juga koreksi *bias* (yang nantinya akan digunakan untuk memperbaiki nilai V_{oj}):

$$\Delta V_{jk} = \alpha \delta_j \quad (2.16)$$

Fase III : Perubahan Bobot

Langkah 8: Tiap-tiap unit *output* Y_k ($k=1,2,3,\dots,m$) memperbaiki *bias* dan bobotnya ($j=1,2,3,\dots,p$):

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (2.17)$$

Tiap-tiap unit *output* Z_j ($j=1,2,3,\dots,p$) memperbaiki *bias* dan bobotnya ($i=1,2,3,\dots,n$):

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (2.18)$$

Langkah 9: Menghitung *mean square error* (MSE), jika MSE belum lebih kecil daripada *target error*, maka langkah 2-8 terus dilakukan (Kusumadewi, 2003).

Setelah pelatihan selesai, jaringan dapat dipakai untuk pengenalan pola. Dalam hal ini, langkah 4 dan 5 saja yang dipakai untuk menentukan keluaran. Apabila fungsi yang dipakai bukan fungsi *sigmoid biner*, maka langkah 4 dan 5 harus disesuaikan, begitu juga turunannya pada langkah 6 dan 7.

Pada praktiknya, perancangan arsitektur *artificial neural network backpropagation* sangat tergantung pada masalah yang akan diselesaikan. Himpunan masukan berdimensi besar atau jumlah kelas keluaran yang diinginkan besar, maka diperlukan jumlah *node* pada lapisan tersembunyi yang lebih banyak, atau diperlukan lebih dari satu lapisan tersembunyi, tetapi tentu saja ada batas optimal nya untuk kedua parameter tersebut (Siang, 2005).

Keterangan simbol-simbol pada algoritma *artificial neural network backpropagation* dijelaskan pada tabel 2.6.1 berikut sebagai berikut:

Tabel 2.2 Simbol-simbol Algoritma *Backpropagation*

No	Simbol	Keterangan
1	X	<i>Input</i> vektor pelatihan, X_i ($i = 1, 2, 3, \dots, n$)
2	Y	<i>Output</i> vektor target, Y_k ($k = 1, 2, 3, \dots, m$)
3	δ_k	Informasi tentang kesalahan pada unit Y_k yang disebarkan kembali ke unit tersembunyi
4	δ_j	Informasi tentang kesalahan dari lapisan <i>output</i> ke unit tersembunyi Z_j
5	α	Laju pembelajaran (<i>learning rate</i>)
6	X_i	Unit <i>input</i> i
7	V_{oj}	<i>Bias</i> pada lapisan tersembunyi j
8	Z_j	Unit tersembunyi j
9	Z_in_j	<i>Input</i> jaringan ke Z_j
10	V_{ok}	<i>Bias</i> pada unit tersembunyi k
11	Y_k	Unit <i>output</i> i
12	Y_in_k	<i>Input</i> jaringan ke Y_k

2.6.2 Pemilihan Bobot Awal

Bobot awal akan mempengaruhi apakah jaringan mencapai titik minimum lokal atau global, dan seberapa cepat konvergensinya (Siang, 2005). Bobot yang menghasilkan nilai turunan aktivasi yang kecil se-dekat mungkin dihindari karena akan menyebabkan perubahan bobotnya menjadi sangat kecil. Demikian pula nilai bobot awal tidak boleh terlalu

besar karena nilai turunan fungsi aktivasinya menjadi target yang sangat kecil juga. Oleh karena itu dalam *backpropagation*, bobot dan *bias* diisi dengan bilangan acak kecil.

Biasanya bobot awal diinisialisasi secara random dengan nilai -0.5 sampai 0.5 (atau -1 sampai 1) (Kusumadewi, 2003). Tetapi untuk inisialisasi bobot dan *bias* dari lapisan masukan ke lapisan tersembunyi, Nguyen dan Widrow (1990) mengusulkan cara sehingga menghasilkan iterasi lebih cepat (Siang, 2005).

Algoritma inisialisasi Nguyen dan Widrow adalah sebagai berikut:

- a) Inisialisasi semua bobot v_{ij} (*lama*) dengan bilangan acak dalam interval $[-0,5;0,5]$.
- b) Hitung $\|v_j\| = \sqrt{v_{j1}^2 + v_{j2}^2 + \dots + v_{jn}^2}$
- c) Bobot yang dipakai sebagai inisialisasi adalah $v_j = \frac{\beta v_{j1}(\text{lama})}{\|v_j\|}$
- d) *Bias* yang dipakai sebagai inisialisasi adalah v_{oj} , yaitu bilangan acak antara $-\beta$ dan β .

2.6.3 Lama Iterasi

Tujuan utama penggunaan *backpropagation* adalah untuk mendapatkan keseimbangan antara pengenalan pola pelatihan secara benar dan respon yang baik untuk pola lain yang sejenis (data pengujian) (Siang, 2005). Jaringan dapat dilatih terus menerus hingga semua pola pelatihan dikenali dengan benar. Akan tetapi hal itu tidak menjamin jaringan akan mampu mengenali pola pengujian dengan tepat. Jadi

