

# **APLIKASI JARINGAN NEURAL UNTUK PEMODELAN DAN PREDIKSI CURAH HUJAN**

**Dadang Subarna**

Peneliti Pusat Pemanfaatan Sains Atmosfer dan Iklim, LAPAN  
dangsub@yahoo.com

## **RINGKASAN**

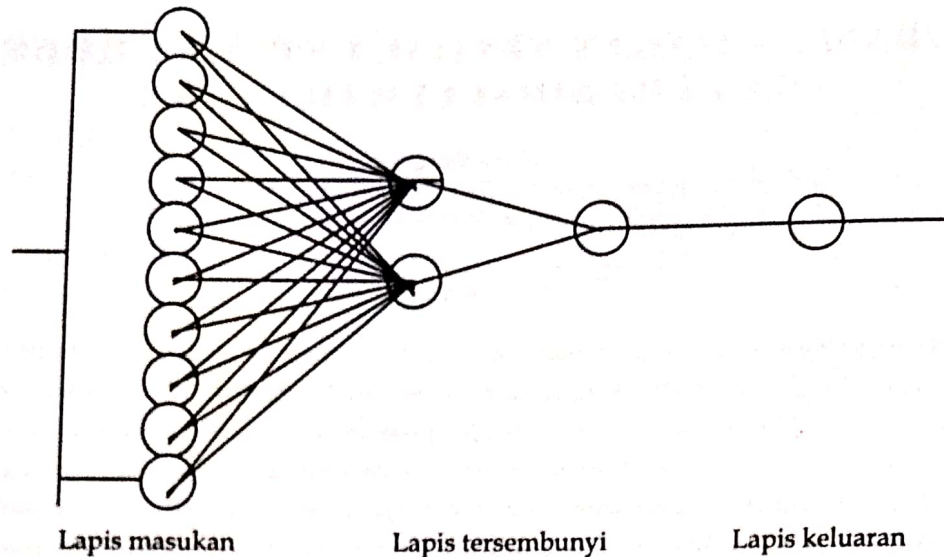
Aplikasi jaringan neural umpan maju untuk prediksi dan pemodelan nonlinear telah diteliti untuk data pentad curah hujan dari kota Jakarta. Desain model jaringan neural perlu dioptimalisasi baik jumlah masukan, jumlah neuron (unit), jumlah lapisan tersembunyi maupun aturan pembelajaran karena akan menentukan kinerja dari model jaringan neural dan nilai korelasi dari hasil prediksinya. Pada penelitian ini digunakan model jaringan neural dengan spesifikasi masukan 10 neuron, dua lapis tersembunyi masing-masing lapis kedua 2 neuron, lapis ketiga 1 neuron dan keluaran 1 neuron. Dengan menggunakan model jaringan neural tersebut maka didapat hasil prediksi dengan keakuratan yang ditunjukkan dengan nilai koefisien korelasi yaitu nilai korelasi ( $r \sim 0,40$ ) untuk daerah Jakarta.

## **1 PENDAHULUAN**

Dalam beberapa dekade terakhir, para peneliti dari berbagai bidang kajian seperti rekayasa, fisika, sains kognitif, kedokteran, statistik, dan ekonomi telah melakukan kontribusi yang penting dalam memahami pengembangan dan penerapan sistem artifisial yang memodelkan aspek tertentu dalam model dan fungsi dari kecerdasan manusia. Kemampuan memahami pengalaman-pengalaman masa lalu untuk meramalkan masa datang dari kecerdasan manusia ini dicoba diaplikasikan dalam masalah prediksi.

Jaringan neural artifisial adalah suatu model yang berusaha menyaingi atau menyerupai suatu jaringan-jaringan neural dalam biologi khususnya kemampuan otak manusia. Meskipun neuron artifisial merupakan analogi dari neuron dalam biologi namun jaringan neural artifisial masih jauh dari deskripsi yang realistis tentang bagaimana cara kerja otak sesungguhnya. Namun demikian jaringan neural artifisial melengkapi suatu pengkayaan, kemampuan dan maket kerja pemodelan yang menarik dengan potensi aplikasi dalam berbagai bidang sains. Sebagai contoh aplikasinya adalah Elman (1990) untuk pembelajaran dan representasi struktur temporal dalam bahasa, Jordan (1990)

untuk kontrol dan pembelajaran pergerakan robot secara halus. Gencay dan Dechert (1992), Gencay (1996) dan Gencay dan Decert (1996) dalam pengkodean galau deterministik dan berderau serta estimasi eksponen Lyapunov. Keberhasilan ini dan pada bidang-bidang lainnya merupakan manfaat tambahan sebagai alat yang tersedia dalam prediksi dan pemodelan deret waktu nonlinear. Khusus untuk bidang meteorologi perkembangannya memang belum sepesat bidang lain, namun dalam jaringan neural umpan maju sinyal dari satuan masukan secara langsung dihubungkan dengan satuan keluaran melalui fungsi keluaran. Bentuk awal dari fungsi keluaran adalah suatu fungsi ambang yang mengambil suatu nilai dari 0 atau 1 yang ditentukan oleh suatu parameter ambang. Satuan keluaran diaktivasi bila nilai fungsi adalah 1 dan nonaktivasi bila nilainya yang lain. Sesuai perjanjian fungsi keluaran ini disebut fungsi aktivasi. Suatu jaringan biasanya terdiri dari bagian lapisan masukan, lapisan pertengahan, dan lapisan keluaran. Lapisan pertengahan biasanya disebut dengan lapisan tersembunyi. Suatu model jaringan neural umpan maju dengan lapisan tersembunyi diilustrasikan dalam Gambar 1-1.



Gambar 1-1: Suatu model jaringan neural umpan maju dengan dua lapisan tersembunyi

## 2 PERMASALAHAN DALAM PREDIKSI

Prediksi sangat syarat dengan berbagai asumsi dan batasan-batasan dan erat kaitannya dengan ilmu keputusan (*decision science*), sains manajemen dan perencanaan skenario (*scenarios planning*). Prediksi adalah suatu usaha memahami masa yang akan datang yang serba tidak pasti (*uncertainty*) dengan memakai memori masa lalu. Memori masa lalu itu bisa berupa jejak yang ditinggalkan sebagai hasil dari suatu proses, data, informasi dan sebagainya. Tidak ada kata "pasti" dalam usaha prediksi. Namun dalam ketidakpastian itu muncul harapan, tantangan dan peluang. Usaha untuk memahami masa depan telah lama dilakukan orang, terbukti munculnya istilah-istilah yang berkaitan dengan nujum, ramal-meramal dan lain sebagainya. Secara saintifik usaha untuk memahami masa depan muncul dalam ilmu statistik berdasarkan data yang diamati dari suatu proses. Dari sana muncul istilah kepeluangan (*probability*), kemungkinan (*possibility*), *plausibility*, tingkat kepercayaan, derajat kesamaran dan sebagainya. Usaha prediksi dalam data deret waktu meliputi interpolasi dan ekstrapolasi data. Pendekatan linear (*arima*, *sarima*, *marima* dan sejenisnya) telah lama dilakukan para predikter.

Dalam dekade terakhir ini muncul pendekatan nonlinear (polinom, fungsi basis radial, jaringan neural, polinom ortogonal dan sejenisnya). Pada penelitian ini akan digunakan model jaringan neural umpan maju untuk prediksi curah hujan. Untuk jaringan neural beberapa pendekatan menurut tipe jaringannya meliputi linear, MLP (*Multi Layer Perceptron*), RBF (*Radial Basis Function*), PNN (*Probabilistic Neural Network*), GRNN (*Generalized Neural Network*), SOFM (*Self Organizing Feature Map*), PCN (*Principal Component Network*), CN (*Clustering Network*) dan sejenisnya. Pada makalah ini diterapkan model MLP untuk memodelkan data curah hujan dan bagaimana agar mendapatkan MLP yang optimal untuk maksud prediksi.

## 3 METODOLOGI DAN PERUMUSAN

Kontras dengan teknik linear tradisional dalam statistik, tidak ada suatu metode yang dikenal saat ini yang secara otomatis mendapatkan jaringan neural optimal dalam mencocokkan dengan kumpulan data tertentu. Seseorang biasanya menjalankan algoritma pembelajaran beberapa lama terhadap desain jaringan neural tertentu, dengan memilih jaringan yang paling baik (mungkin beberapa

yang paling, baik). Lalu harus memilih tipe jaringan neural, jumlah variabel input dan lapisan tersembunyi serta setting berbagai parameter kontrol dalam algoritma pembelajaran yang mungkin mempengaruhi unjuk kerja akhir dari jaringan. Oleh karena itu sejumlah eksperimen dengan berbagai desain disajikan lalu jaringan yang terbaik dipilih. Selama proses eksperimen, *desainer* (perancang) harus memandu agar tidak *over-learning* dengan menggunakan teknik "terhenti dini". Teknik khusus seperti regulasi dan analisis sensitivitas dapat disebar untuk membantu proses desain.

Pencarian cerdas mengikuti suatu proses yang sama meskipun dalam kasus ini *heuristic expertise* dari suatu *desainer* jaringan neural diganti dengan algoritma pencarian yang menggunakan teknik *state-of-the-art* untuk menentukan pemilihan masukan, jumlah unit tersembunyi dan faktor kunci lainnya dalam desain jaringan. Sehingga pencarian cerdas mencari jaringan optimal dari beberapa tipe (MLP dan RBF) secara simultan. Pencarian cerdas dapat mencari untuk jangka waktu tak terbatas (kecuali dalam kasus tertentu yang sederhana, seperti jaringan linear, pencarian berakhir dengan sendirinya) meskipun setelah beberapa periode waktu tak diketahui tak mungkin membuat kemajuan lebih lanjut. Pencarian cerdas memerlukan waktu lebih bila menjalankan tugas tertentu, khususnya fitur seleksi (penentuan input otomatis) dan untuk suatu bentangan yang kurang maka penentuan kompleksitas (penentuan jumlah unit tersembunyi otomatis). Jika terdapat persoalan besar dengan puluhan ribu variabel input dan ribuan atau puluhan ribu kasus, bermanfaat sekali menemukan kemudahan dengan menggunakan metode ini.

Dari Gambar 1-1 terlihat bahwa jaringan neural ini menggunakan tiga lapis neuron, dimana terdapat dua lapis tersembunyi dan satu lapis keluaran. Pada neuron lapis pertama menerima masukan, di sini jumlah

bobot ( $w$ ) sama dengan jumlah masukan kali jumlah neuron pertama. Kemudian neuron lapis kedua menerima masukan dari keluaran pada neuron lapis pertama sehingga jumlah bobot sama dengan neuron lapis pertama kali neuron lapis kedua. Terakhir adalah neuron ketiga atau keluaran dimana akan menerima masukan dari neuron lapis kedua karena keluaran hanya satu neuron maka jumlah bobot sama dengan jumlah neuron pada lapis kedua.

Untuk menentukan jumlah neuron pada masing-masing lapisan dilakukan dengan coba-coba atau pencarian cerdas dan yang kesalahannya dianggap paling kecil dan memberikan koefisien korelasi paling besar adalah yang digunakan sebagai model dalam penelitian ini. Pada penelitian ini diperoleh lapis pertama 10 neuron, lapis kedua 2 neuron, lapis ketiga 1 neuron dan keluaran 1 neuron. Algoritma pengolahan dengan metode jaringan neural umpan maju (R. Gencay dan T. Liu, 1996) adalah sebagai berikut.

Bila masukan  $x_j = (x_1, x_{t-1}, \dots, x_{t-k})$  dimana di sini  $k=10$  maka keluaran dari jaringan neural dengan  $q$  neuron tersembunyi pada lapis pertama adalah

$$h_{i,j} = \Psi \left( \gamma_{io} + \sum_{j=1}^n \gamma_{ij} x_j \right) \quad i=1, \dots, q \quad (3-1)$$

Keluaran dari lapis kedua

$$o_{i,j} = \Phi \left( \beta_{io} + \sum_{j=1}^n \beta_{ij} h_{i,j} \right) \quad (3-2)$$

Keluaran dari lapis ketiga

$$o_j = \Omega \left( \alpha_o + \sum_{j=1}^n \alpha_j o_{i,j} \right) \quad (3-3)$$

dimana  $\alpha, \beta, \gamma$  adalah parameter-parameter kontrol yang diestimasi pada saat pembelajaran

dan  $\Psi, \Phi, \Omega$  adalah fungsi aktivasi. Pada penelitian ini fungsi aktivasi menggunakan fungsi sigmoid yaitu

$$F = \frac{1}{1 + e^{-a}} \quad (3-4)$$

Seperti terlihat pada Gambar 1-1, neuron (unit) tersembunyi dari jaringan neural umpan maju tidak dinamis yaitu jaringan neuron pada lapis tersembunyi itu tidak bergantung nilai lalu yang dihasilkan dari jaringan. Sinyal hanya menjalar dalam satu arah tanpa umpan balik (*feedback*). Dengan alasan ini maka jaringan tersebut disebut jaringan umpan maju (*feedforward*).

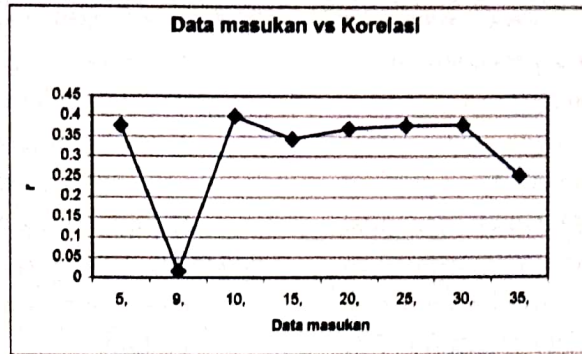
#### 4 HASIL DAN PEMBAHASAN

Dengan menggunakan algoritma yang diuraikan dalam bagian 4 maka pertama-tama ditentukan dulu jumlah masukan data sebelumnya yang optimal untuk data curah hujan lalu dicari desain model jaringan yang paling baik dengan cara coba-coba atau pencarian cerdas untuk melakukan pembelajaran dan prediksi. Didapat hasil sebagai mana pada Gambar 4-2.

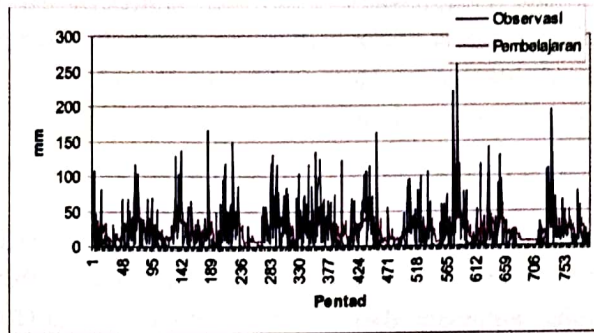
Dari Gambar 4-1 terlihat bahwa jumlah masukan 10 memberikan nilai koefisien korelasi yang paling besar, maka nilai inilah yang diambil sebagai jumlah masukan untuk model jaringan neural.

Untuk menentukan jumlah neuron pada masing-masing lapisan dilakukan dengan coba-coba atau pencarian cerdas dan yang kesalahannya dianggap paling kecil serta memberikan

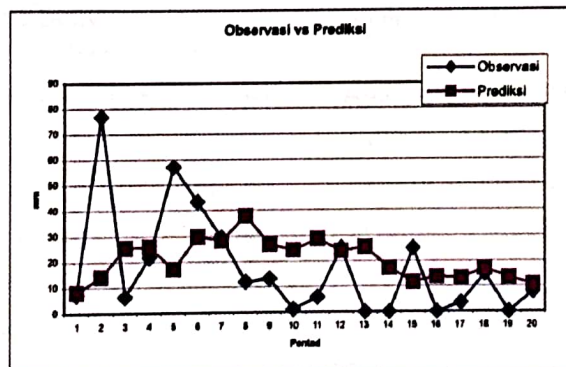
koefisien korelasi paling besar adalah yang digunakan sebagai model dalam penelitian ini. Pada penelitian ini diperoleh lapis pertama 10 neuron, lapis kedua 2 neuron, lapis ketiga 1 neuron dan keluaran 1 neuron, seperti tampak pada Gambar 1-1. Dengan menggunakan jumlah masukan dan model jaringan neural yang paling baik yang telah didapat tersebut maka hasil pembelajaran dan prediksi data curah hujan pentad dari beberapa kota di Jawa. Untuk melakukan prediksi kita tinjau data curah hujan dalam selang waktu antara  $t_0$  dan  $t_1$ . Hasil prediksi dibuat merentang pada selang waktu yang lebih lebar dari selang waktu semula, misalnya dari  $t_0$  ke  $t_1 + \Delta t$ . Deret waktu data curah hujan dalam selang antara  $t_1$  dan  $t_1 + \Delta t$  adalah hasil prediksinya. Untuk data curah hujan kota Jakarta di dapat seperti pada Gambar 4-2.



Gambar 4-1: Jumlah data masukan dan koefisien korelasi untuk menentukan jumlah masukan optimal



Gambar 4-2: Data observasi dan pembelajaran model MLP 10-2-1-1 untuk data pentad (1988-1998) daerah Jakarta



Gambar 4-3: Data observasi dan prediksi 20 titik ke depan dengan model MLP 10-2-1-1 untuk data pentad (1988-1998) daerah Jakarta

Dengan perhitungan statistik didapat Tabel 4-1, di bawah ini

Tabel 4-1: DATA JAKARTA DAN VALIDASI HASIL PREDIKSINYA

Rata-rata data	23.77011
Standar deviasi data	33.48089
Rata-rata kesalahan	-0.45778
Standar deviasi kesalahan	30.67222
Rata-rata kesalahan mutlak	21.31523
Rasio standar deviasi	0.916111
Koefisien korelasi	0.400962

## 5 KESIMPULAN

Dari hasil pembahasan dan penelitian yang dilakukan selama ini dapat disimpulkan bahwa jaringan neural dapat digunakan untuk membantu prediksi curah hujan di kota-kota yang rawan terhadap banjir. Penentuan jumlah masukan sangat signifikan terhadap nilai koefisien korelasi hasil prediksi. Penentuan desain model jaringan neural juga sangat penting terhadap unjuk kerja model jaringan neural dan terhadap nilai koefisien korelasinya. Untuk penentuan desain ini dapat dilakukan

dua pendekatan yaitu secara coba-coba (*Heuristic Expert*) atau secara pencarian cerdas (*Intelligent searching*). Banyaknya lapisan tersembunyi (*Hidden Layer*) pada suatu desain model jaringan neural tidak meningkatkan unjuk kerja model jaringan neural dan nilai korelasinya. Banyaknya neuron yang menyusun jaringan neural belum tentu akan berkontribusi terhadap unjuk kerja dan nilai korelasinya. Dari hasil-hasil yang telah dikaji dan pembahasan yang dilakukan disarankan agar penggunaan jaringan neural ini terus dikembangkan untuk model prediksi curah hujan dengan mencoba menggunakan model-model modifikasi arsitektur jaringan neural dan mencoba menggunakan jenis-jenis aturan pembelajaran (*Learning Rule*) yang lain selain umpan maju serta membandingkan dengan model-model komputasi lain seperti selular automata dan lain-lain.

#### DAFTAR RUJUKAN

- Bayong, T. H. K., 1999. *Klimatologi Umum*, Penerbit ITB Bandung.
- Demuth. H., Beale. M, 1995. *Neural Network Toolbox*, The Math Works Inc.

- Elman, J.L., 1990. Finding Structure in Time, *Cognitive Science*, 14, 179-211.
- Freeman. A.J., Skapura. M.D, 1992. *Neural Networks, Algorithms, Applications And Programming Techniques*, Addison-Wesley Publishing Company, Inc.
- Fu.Limin, 1992. *Neural Network In Computer Intelligence*, McGraw-Hill International Edition.
- Gencay, R., 1996. A Statistical Framework for Testing Chaotic Dynamics Via Lyapunov Exponent", *Physica D*, 89, 261-266.
- Gencay, R. and Liu, T., 1996. Nonlinear modelling and Prediction with Feedforward and Recurrent Network, *Physica Letters, A* 187, 397-403.
- Gencay, R. and W.D. Dechert, 1992. An Algorithm for the  $n$  Lyapunov Exponents of  $n$ -Dimensional Unknown Dynamical System, *Physica D*, 59,142-17.
- Yang. Q. C, Bhargaya. K.V, 1991. Optimum Selection Of Error Control Coding Using Neural Network, *IEEE*, pp.1074-1082.