

Optimasi Jaringan Syaraf Tiruan dengan Algoritma Genetika untuk Peramalan Curah Hujan

Aziz Kustiyo¹, Agus Buono¹ dan Novi Apriyanti²

¹Staf Departemen Ilmu Komputer, FMIPA IPB

²Alumni Departemen Ilmu Komputer, FMIPA IPB

Abstrak

Informasi tentang banyaknya curah hujan sangat berguna bagi para petani dalam mengantisipasi kemungkinan terjadinya peristiwa-peristiwa ekstrim (kekeringan dan banjir) yang akan berakibat kegagalan dalam proses produksinya. Dengan demikian, ketersediaan informasi ini memerlukan suatu metode peramalan curah hujan yang akurat. Beberapa penelitian yang sudah dilakukan belum memberikan hasil yang memuaskan.

Algoritma genetika standar adalah metode yang digunakan untuk optimasi konfigurasi neuron pada lapisan tersembunyi dalam Jaringan Syaraf Tiruan (JST) ini. Teknik pembelajaran JST yang digunakan dalam metode peramalan curah hujan ini adalah JST propagasi balik standar dengan arsitektur banyak lapis yaitu satu lapisan input, satu lapisan tersembunyi, dan satu lapisan output. Keakuratan hasil prediksi JST diukur berdasarkan R^2 dan RMSE-nya.

Ujicoba terhadap sistem telah dilakukan JST dalam peramalan curah hujan dan hasilnya dibandingkan dengan metode *principal component regression* (PCR) dan JST propagasi balik standar dengan hasil terbaiknya dengan jumlah neuron 9 dan laju pembelajaran 0.02. Hasil peramalan curah hujan menggunakan metode PCR menghasilkan R^2 sebesar 63%, sedangkan peramalan dengan menggunakan metode jaringan syaraf tiruan propagasi balik menghasilkan R^2 sebesar 74%, dan JST yang dioptimasi menjadi 87%. Hal ini menunjukkan adanya peningkatan R^2 . Berdasarkan uji coba tersebut dapat diperoleh kesimpulan bahwa optimasi JST dengan algoritma genetika dapat digunakan untuk memperbaiki tingkat pendugaan curah hujan dengan menggunakan data GCM daripada metode PCR dan JST propagasi balik standar.

Kata Kunci : *artificial intelligence, neural network, genetics algorithm, principle component analysis, determinant coefficient,*

PENDAHULUAN

Latar Belakang

Informasi tentang banyaknya curah hujan sangat berguna bagi para petani dalam mengantisipasi kemungkinan terjadinya peristiwa-peristiwa ekstrim yang tidak diinginkan seperti kekeringan dan banjir yang akan berakibat kegagalan dalam proses produksinya. Informasi curah hujan yang diperlukan petani harus tersedia sebelum kegiatan proses produksi dimulai sehingga perlu diadakannya kajian tentang peramalan curah hujan.

Dengan menggunakan teknologi di bidang *Artificial Intelligence*, yaitu teknologi Jaringan Syaraf Tiruan (JST), maka identifikasi pola data dari sistem peramalan curah hujan dapat dilakukan dengan metode pendekatan pembelajaran. Berdasarkan kemampuan belajar yang dimilikinya, maka JST dapat dilatih untuk mempelajari dan menganalisa pola data masa lalu dan berusaha mencari suatu formula atau fungsi yang akan menghubungkan pola data pada masa lalu. JST juga dapat mencari formula atau fungsi yang akan menghubungkan pola data pada masa lalu dengan keluaran yang diinginkan pada saat ini.

Saat ini model-model peramalan yang ada belum menggunakan data sirkulasi atmosfer (data spasial-temporal) yang diperoleh dari *output* model sirkulasi umum (*General Circulation Model* – GCM), untuk mengantisipasi keragaman perubahan curah hujan dalam ruang dan waktu. Sebagai contoh, Lestari (2000) melakukan penelitian untuk menentukan model time series ARMA dari banyaknya hari

hujan di Jember berdasarkan data hari hujan yang dihitung per bulan. Penelitian yang pernah dilakukan menggunakan data GCM yaitu menggunakan *Principal Component Regression* (PCR) atau regresi komponen utama dalam *statistical downscaling* (Fitriadi 2004) dan JST propagasi balik standar (Normakristagaluh 2004).

Untuk itulah pada penelitian ini dilakukan pendugaan dengan menggunakan JST yang dioptimasi menggunakan *Genetic Algorithm* (GA) untuk mendukung peramalan curah hujan berdasarkan faktor-faktor dominan yang mempengaruhi curah hujan di Indonesia.

Tujuan

Tujuan penelitian ini adalah mengembangkan model JST dengan optimasi struktur neuron lapis tersembunyi menggunakan GA untuk peramalan curah hujan.

Ruang Lingkup Penelitian

Ruang lingkup penelitian ini meliputi :

1. Perlakuan *preprocessing* terhadap data *input* yang digunakan, yaitu dengan mengelompokkan data. Setiap kelompok data dinormalisasi, kemudian dilakukan *Principal Components Analysis* (PCA) untuk mereduksi dimensi data spasial suhu.
2. Teknik pembelajaran JST yang digunakan adalah propagasi balik standar dengan satu lapisan *input*, satu lapisan tersembunyi, dan satu lapisan *output*.
3. Jumlah neuron pada lapis tersembunyi berjumlah 16, baik pada percobaan JST standar maupun pada

percobaan JST optimasi menggunakan algoritma genetika.

4. Optimasi struktur neuron lapis tersembunyi menggunakan GA berada pada selang solusi 0-16 dan tujuan fungsi *fitness*-nya adalah untuk mengoptimalkan nilai R^2 .
5. Pada tahap *postprocessing*, hasil prediksi (nilai *output*) JST diukur berdasarkan R^2 dan RMSE-nya.

TINJAUAN PUSTAKA

Model Sirkulasi Umum

Model sirkulasi umum merupakan model matematika yang menggambarkan hubungan atau interaksi berbagai proses fisik yang berlangsung di atmosfer, laut dan daratan (Ratag dalam Normakristagaluh 2004). Model ini menduga perubahan unsur-unsur cuaca secara regional pada *grids* berukuran 3° atau 4° sampai 10° menurut lintang dan bujur, dan dapat digunakan untuk peramalan atau menilai dampak yang mungkin timbul apabila terjadi perubahan di udara, laut, dan daratan.

Curah Hujan

Curah hujan dibatasi sebagai tinggi air hujan (dalam mm) yang diterima di permukaan sebelum mengalami aliran permukaan, evaporasi, dan peresapan/perembesan ke dalam tanah (Handoko dalam Normakristagaluh 2004). Data hujan mempunyai variasi yang sangat besar dibandingkan unsur-unsur iklim yang lain, baik variasi menurut tempat maupun waktu. Untuk mendapatkan gambaran wilayah diperlukan pengamatan yang cukup panjang dan kerapatan jaringan stasiun pengamatan yang memadai. Curah hujan yang diamati pada stasiun klimatologi meliputi tinggi hujan (curah hujan), jumlah hari hujan, dan intensitas hujan.

Normalisasi

Normalisasi data GCM dilakukan dengan mengurangi nilai setiap *grid* dengan nilai rata-rata seluruh *grid* dibagi standar deviasinya.

$$grid_normal_i = (grid_i - \text{rata-rata})$$

$$\text{rata-rata} = \frac{1}{n} \sum_{i=1}^n grid_i$$

$$stddev = \sqrt{\frac{1}{n} \sum_{i=1}^n (grid_i - \text{rata-rata})^2}$$

Tujuan normalisasi adalah untuk mendapatkan nilai rata-rata nol dan standar deviasi satu.

Principal Components Analysis (PCA)

PCA atau analisis komponen utama yang disebut juga transformasi *Karhunen-Loeve* merupakan suatu teknik untuk mereduksi p peubah (variabel) pengamatan menjadi q peubah baru yang saling *ortogonal*, di mana masing-masing q peubah baru tersebut merupakan kombinasi linear dari p peubah lama. Pemilihan q peubah baru tersebut sedemikian rupa sehingga keragaman yang dimiliki oleh p peubah lama, sebagian besar dapat diterangkan atau dimiliki oleh q peubah baru. PCA akan cukup efektif jika antar p peubah asal memiliki korelasi yang cukup tinggi.

Misalnya diberikan objek pengamatan vektor *Grid* dengan dimensi p (peubah), $Grid = [grid_1, grid_2, grid_3, \dots, grid_p]$ yang akan direduksi menjadi vektor $Y = [y_1, y_2, y_3, \dots, y_q]$, di mana $q \ll p$ tanpa kehilangan informasi secara berarti, yang dapat dilihat sebagai berikut:

$$y_1 = a_{11}grid_1 + a_{12}grid_2 + \dots + a_{1p}grid_p = (a_{11} \ a_{12} \ \dots \ a_{1p}) * \begin{pmatrix} grid_1 \\ grid_2 \\ \vdots \\ grid_p \end{pmatrix} = \underline{a}^T \underline{grid}$$

$$y_2 = a_{21}grid_1 + a_{22}grid_2 + \dots + a_{2p}grid_p = (a_{21} \ a_{22} \ \dots \ a_{2p}) * \begin{pmatrix} grid_1 \\ grid_2 \\ \vdots \\ grid_p \end{pmatrix} = \underline{a}^T \underline{grid}$$

:

$$y_q = a_{q1}grid_1 + a_{q2}grid_2 + \dots + a_{qp}grid_p = (a_{q1} \ a_{q2} \ \dots \ a_{qp}) * \begin{pmatrix} grid_1 \\ grid_2 \\ \vdots \\ grid_p \end{pmatrix} = \underline{a}^T \underline{grid}$$

untuk memaksimalkan ragam pada y maka harus dicari nilai \underline{a} sehingga $y = \underline{a}^T \underline{\Sigma} \underline{a}$ dengan kendala $\underline{a}^T \underline{a} = 1$. Dengan menggunakan pengganda *Lagrange* dibuat suatu fungsi $F = \underline{a}^T \underline{\Sigma} \underline{a} - \lambda(\underline{a}^T \underline{a} - 1)$ yang selanjutnya akan dimaksimalkan dengan cara menurunkan F terhadap parameter-parameternya dan turunannya sama dengan 0, sehingga diperoleh:

$$\frac{\delta F}{\delta \underline{a}} = 2\underline{\Sigma} \underline{a} - 2\lambda \underline{a} = 0$$

$$\underline{\Sigma} \underline{a} = \lambda \underline{a}$$

di mana $\underline{\Sigma}$ adalah matriks peragam

$$(\text{covariance}) : \underline{\Sigma} = \begin{pmatrix} \tau_{11} & \tau_{12} & \dots & \tau_{1p} \\ \tau_{21} & \tau_{22} & \dots & \tau_{2p} \\ \vdots & \vdots & \dots & \vdots \\ \tau_{p1} & \tau_{p2} & \dots & \tau_{pp} \end{pmatrix} \quad \text{dengan}$$

$$\tau_{jk} = \frac{1}{n-1} \sum_{i=1}^n (grid_{ij} - \overline{grid_j})(grid_{ik} - \overline{grid_k})$$

\underline{a} adalah vektor ciri (*eigenvectors*),

λ adalah akar ciri (*eigenvalue*).

Setelah didapatkan matriks *covariance* $\underline{\Sigma}$ selanjutnya akan dicari vektor dan akar ciri yang bersesuaian dengan matriks *covariance* tersebut, melalui persamaan berikut:

$$\underline{\Sigma} \underline{a} = \lambda \underline{a}$$

$$\underline{\Sigma} \underline{a} - \lambda \underline{a} = 0$$

$$(\underline{\Sigma} - \lambda \underline{I}) \underline{a} = 0$$

Dari persamaan tersebut dapat disimpulkan $|\underline{\Sigma} - \lambda \underline{I}| = 0$ dan akan diperoleh suatu persamaan polinom derajat p di dalam λ . Persamaan polinom tersebut akan memiliki p akar, yaitu $\lambda_1, \lambda_2, \dots, \lambda_p$ yang merupakan akar ciri dari matriks *covariance* $\underline{\Sigma}$. Akar-akar ciri tersebut kemudian digunakan untuk menentukan vektor-vektor cirinya. Vektor-vektor ciri tersebut sebagai koefisien komponen utama.

Akar-akar ciri dengan vektor ciri yang bersesuaian tersebut kemudian disusun terurut menurun sehingga memenuhi $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_p \geq 0$

Penentuan proporsi dari nilai vektor-vektor ciri yang digunakan dapat dihitung dengan persamaan:

$$\text{Proporsi} = \frac{\sum_{i=1}^q \lambda_i}{\sum_{i=1}^n \lambda_i}$$

Pada akhirnya PCA ini hanyalah akan mentransfer variabel-variabel yang berkorelasi menjadi variabel-variabel yang tidak berkorelasi.

Tujuan metode PCA di atas adalah untuk menentukan faktor-faktor yang menunjukkan seluruh kemungkinan variasi pada keseluruhan data melalui sebagian kecil faktor-faktor dari keseluruhan data (Dillon dan Goldstein 1984).

Jaringan syaraf tiruan (JST)

Menurut Fauset (1994), JST memiliki karakteristik-karakteristik sebagai berikut:

1. Pola hubungan antar-neuron yang disebut arsitektur.
2. Metode penentuan bobot pada hubungan yang disebut pelatihan (*training*) atau pembelajaran (*learning*).
3. Fungsi aktivasi yang dijalankan masing-masing neuron pada *input* jaringan untuk menentukan *output*.

JST seperti beberapa metode statistik mampu mengolah data yang besar dan membuat suatu prediksi yang kadang-kadang keakuratannya lebih baik, tetapi biasanya lebih lambat.

JST Propagasi Balik Standar

Menurut Fauset (1994), ada tiga tahap pelatihan pada JST propagasi balik, yaitu pelatihan *input* yang bersifat umpan maju (*feedforward*), penghitungan propagasi balik galat yang diperoleh, dan penyesuaian bobot. Jaringan ini menggunakan metode pembelajaran terbimbing (*supervised learning*). Adapun cara kerja JST diawali dengan inialisasi bobot dan bias. Hal ini berpengaruh dalam kecepatan JST memperoleh kekonvergenan.

Teknik inialisasi yang digunakan dalam penelitian ini adalah inialisasi *Nguyen-Widrow*. Inialisasi ini umumnya mempercepat proses pembelajaran dibandingkan dengan inialisasi acak (Fauset 1994). Inialisasi *Nguyen-Widrow* didefinisikan sebagai persamaan berikut:

a. Hitung harga faktor pengali β

$$\beta = 0.7 p^{-1/n}$$

di mana:

β = faktor pengali

n = jumlah neuron lapisan *input*

p = jumlah neuron lapisan tersembunyi

b. Untuk setiap unit tersembunyi ($j=1, 2, \dots, p$):

- Hitung v_{ij} (lama) yaitu bilangan acak di antara -0.5 dan 0.5 (atau di antara $-\gamma$ dan $+\gamma$). Pembaharuan bobot v_{ij} (lama) menjadi v_{ij} baru yaitu:

$$v_{ij}(\text{baru}) = \frac{\beta v_{ij}(\text{lama})}{\|v_{ij}(\text{lama})\|}$$

- Tetapkan Bias.

v_{0j} = Bobot pada bias bernilai antara $-\beta$ dan β .

Selanjutnya pola *input* dan target dimasukkan ke dalam jaringan. Pola *input* ini akan berubah bersamaan dengan propagasi pola tersebut ke lapisan-lapisan berikutnya hingga menghasilkan *output*. *Output* ini akan dibandingkan dengan target, jika hasil perbandingan menghasilkan nilai yang sama maka proses pembelajaran akan berhenti.

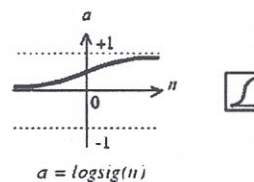
Namun jika berbeda maka JST mengubah bobot yang ada pada hubungan antar-neuron dengan suatu aturan tertentu agar nilai keluaran lebih mendekati nilai target. Proses pengubahan bobot adalah dengan cara mempropagasikan kembali nilai koreksi galat keluaran JST ke lapisan-lapisan sebelumnya (propagasi balik).

Kemudian dari lapisan *input*, pola akan diproses lagi untuk mengubah nilai bobot sampai akhirnya memperoleh *output* JST baru. Demikian seterusnya proses ini dilakukan berulang-ulang sampai nilai yang sama atau minimal sesuai dengan galat yang telah didefinisikan. Proses penyesuaian bobot ini disebut pembelajaran.

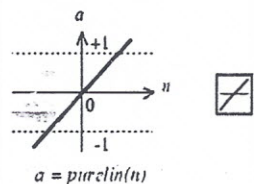
Dalam metode pembelajaran propagasi balik, fungsi pada *input* dan *output* haruslah berbentuk fungsi yang dapat dideferensialkan karena pada proses propagasi (umpan maju atau propagasi balik) penghitungan nilai didasarkan pada fungsi yang dipakai. Pada umpan maju, fungsi yang dipakai adalah fungsi yang telah ditentukan untuk JST, sedangkan pada propagasi balik, fungsi yang digunakan adalah fungsi diferensialnya.

Fungsi yang digunakan dalam penelitian ini adalah sigmoid biner (Gambar 1) pada lapisan tersembunyi dan fungsi linear (Gambar 2) pada lapisan *output*. Fungsi sigmoid ini memiliki daerah hasil pada interval 0 sampai dengan 1. Turunan fungsi sigmoid biner didefinisikan sebagai persamaan berikut:

$$f'(x) = f(x) [1 - f(x)]$$



Gambar 1. Grafik fungsi sigmoid biner dengan range (0,1).



Gambar 2. Grafik fungsi linear.

Algoritma Genetika

Algoritma ini terinspirasi oleh teori evolusi Darwin. GA adalah algoritma pencarian yang berdasarkan pada mekanisme sistem natural, yakni genetika dan seleksi alam. Dalam aplikasi GA, variabel solusi dikodekan ke dalam struktur *string* yang merepresentasikan barisan gen, yang merupakan karakteristik dari solusi problem.

Berbeda dengan teknik pencarian konvensional, GA berangkat dari himpunan solusi yang dihasilkan secara acak. Himpunan ini disebut populasi. Sedangkan setiap individu dalam populasi disebut kromosom yang merupakan representasi dari solusi. Kromosom-kromosom berevolusi dalam suatu proses iterasi yang berkelanjutan yang disebut generasi. Pada setiap generasi, kromosom dievaluasi berdasarkan suatu fungsi evaluasi. Setelah beberapa generasi maka GA akan konvergen pada kromosom terbaik, yang diharapkan merupakan solusi optimal.

Di dalam GA cara merepresentasikan permasalahan dalam kromosom merupakan suatu hal yang penting. Dasar-dasar di dalam GA yaitu fungsi evaluasi, seleksi, dan operator genetika. Struktur umum dari GA sebagai berikut (Gen & Cheng, 1997):

Prosedur Algoritma Genetika

```

Begin
  t ← 0;
  inisialisasi P(t);
  evaluasi P(t);
  while (kondisi berhenti belum dicapai) do
    rekombinasi P(t) menghasilkan C(t);
    evaluasi C(t);
    seleksi P(t + 1) dari P(t) dan C(t);
    t ← t + 1;
  end
End
    
```

Langkah evaluasi yaitu *string* dikonversi ke parameter fungsi, fungsi obyektifnya dievaluasi, kemudian fungsi objektif tersebut diubah ke dalam *fitness* yang digunakan untuk maksimasi *problem*, *fitness* sama dengan fungsi obyektifnya. *Output* dari fungsi *fitness* dipergunakan sebagai dasar untuk menyeleksi individu pada generasi berikutnya.

Pada proses seleksi digunakan *roulette wheel*. *Roulette wheel* menyeleksi populasi baru dengan distribusi peluang yang berdasarkan nilai *fitness*.

Operator genetika dipergunakan untuk mengkombinasi (*modifikasi*) individu dalam aliran populasi guna mencetak individu pada generasi berikutnya. Ada dua operator genetika yaitu *crossover* dan mutasi.

Crossover membangkitkan generasi baru dengan mengganti sebagian informasi dari *parents* (orang tua/ induk). Sedangkan mutasi menciptakan individu baru dengan melakukan modifikasi satu atau lebih gen dalam individu yang sama.

Ketepatan Pendugaan Model Regresi Linear

Ketepatan atau keakuratan sebuah model regresi dapat dilihat dari koefisien determinasi (R^2) dan *Root Mean Square*

Error (RMSE). R^2 menunjukkan proporsi jumlah kuadrat total yang dapat dijelaskan oleh sumber keragaman peubah bebas, sedangkan RMSE menunjukkan seberapa besar simpangan nilai dugaan terhadap nilai aktualnya. Kita juga dapat menunjukkan bahwa R^2 adalah kuadrat dari korelasi antara nilai vektor observasi (*the vector of observations*) y dengan nilai vektor penduganya (*the vector of fitted values*) \hat{y} (Douglas dan Elizabeth dalam Normakristagaluh 2004). Menurut Walpole (1982) R^2 dan RMSE dirumuskan sebagai berikut:

$$R^2 = \frac{\left[\sum_{i=1}^n (\hat{y}_i - \bar{y})(y_i - \bar{y}) \right]^2}{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2 \sum_{i=1}^n (y_i - \bar{y})^2}$$

di mana:
 y_i = nilai-nilai aktual
 \hat{y}_i = nilai-nilai prediksi

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (X_i - F_i)^2}{n}}$$

di mana:
 X_i = nilai aktual pada waktu ke- t
 F_i = nilai dugaan pada waktu ke- t

Nilai R^2 berada pada selang 0 sampai 1. Kecocokan model dikatakan semakin baik jika R^2 mendekati 1 dan RMSE mendekati 0.

METODOLOGI

Data

Data yang diperlukan dalam penelitian ini adalah:

1. Data GCM ($X_{(t,x,g)}$)
 Data ini diperoleh dari NCEP/NCAR (<http://www.irdl.columbia.edu> [2 Maret 2003]) sebagai salah satu sumber *output* model GCM dengan skala (*grid*) $2.8^\circ \times 2.8^\circ$ ($1^\circ = 110$ km). Domain cakupan data GCM ini adalah $23^\circ\text{LU}-23^\circ\text{LS}$ dan $80^\circ\text{BT}-60^\circ\text{BT}$, yang mencakup wilayah Indonesia dan lautan Pasifik. Pada penelitian ini, domain yang digunakan adalah $21.9^\circ\text{LU}-21.9^\circ\text{LS}$ dan $82.5^\circ\text{BT}-157.5^\circ\text{BT}$ yang mencakup wilayah DAS Saguling Jawa Barat. Peubah-peubahnya akan ditentukan berdasarkan mekanisme fisik yang berkaitan erat dengan curah hujan, seperti *sea surface temperature* (suhu) pada 41×24 *grid* dari tahun 1986–2002. Data GCM (*domain*) yang digunakan dipilih berdasarkan kriteria oleh Von Storch (Bergant dalam Normakristagaluh 2004) yang menyatakan bahwa jumlah *domain* minimal 8×8 *grid* di sekitar wilayah respon. Domain yang dipilih adalah yang berkorelasi tinggi dengan wilayah respon.
2. Data Curah Hujan ($X_{(t)}$)
 Respon yang digunakan dalam penelitian ini adalah curah hujan rata-rata dari tahun 1986–2002 di DAS

Saguling yang diperoleh dari Badan Pengkajian dan Penerapan Teknologi (BPPT).

Preprocessing Data dengan PCA

Pertama, dilakukan normalisasi terhadap sejumlah data *training* yang akan digunakan dan dihasilkan rataan dan simpangan bakunya. Kedua, dilakukan proses PCA pada data *training* hasil normalisasi tersebut sehingga didapatkan kombinasi linear atau komponen-komponen utamanya yang selanjutnya disebut sebagai matriks transformasi. Dengan dilakukan perkalian antara data *training* dengan matriks transformasi tersebut didapatkan data *training* yang baru. Ketiga, dilakukan normalisasi data *testing* dengan menggunakan rataan dan standar deviasi yang diperoleh dari data *training* kemudian dilakukan perkalian dengan matriks transformasi sehingga diperoleh data *testing* yang baru. Selanjutnya data *training* dan data *testing* yang baru tersebut digunakan oleh proses pada JST standar dan JST dengan GA.

Faktor dan Taraf

Pada penelitian ini terdapat tiga faktor yaitu persentase data training JST, persentase keragaman data pada hasil PCA, dan laju pembelajaran JST. Tiap-tiap faktor tersebut terdiri dari beberapa taraf (Tabel 1).

Tabel 1. Faktor dan taraf

No.	Faktor	Taraf
1.	Persentase data <i>training</i> yang digunakan JST	<ul style="list-style-type: none"> • 95% • 90 % • 70 % • 50 %
2.	Persentase keragaman data pada <i>preprocessing</i> data dengan PCA (<i>Principal Components Analysis</i>)	<ul style="list-style-type: none"> • 80 % • 90 % • 95 % • 99% • 99.8%
3	Laju pembelajaran (<i>learning rate</i>)	<ul style="list-style-type: none"> • 0.5 • 0.2 • 0.1 • 0.05 • 0.01

Jaringan Syaraf Tiruan

Arsitektur JST yang digunakan serta parameter-parameternya disajikan pada Tabel 2. Arsitektur JST terdiri dari dua *layer*, dengan satu *layer hidden*. Metode inialisasi bobot yang digunakan adalah Nguyen-Widrow.

Optimasi JST dengan GA

Penggunaan GA dalam optimasi JST dilakukan untuk mendapatkan struktur neuron lapis tersembunyi yang optimal. Tingkat pengenalan JST dalam peramalan yang tinggi akan didapat apabila seluruh neuron pada lapis tersembunyi memberikan kontribusi nilai objektif yang tinggi, dalam hal ini penulis menekankan nilai R^2 . Apabila neuron yang memberikan kontribusi R^2 yang kecil dapat dihilangkan, sedangkan yang memberikan kontribusi R^2 besar dapat dipertahankan, maka JST ini dapat diharapkan memberikan nilai R^2 yang lebih tinggi.

Tabel 2. Struktur JST yang digunakan dalam penelitian

KARAKTERISTIK	SPESIFIKASI
Arsitektur	2 layer
Neuron Input	Dimensi PCA (untuk grid-grid lokal GCM)
Neuron Layer Hidden	Tanpa GA, dengan GA
Neuron Layer Output	1 (target data Curah Hujan)
Fungsi Aktivasi Layer Hidden	Sigmoid Biner
Fungsi Aktivasi Layer Output	Linear
Inialisasi Bobot	Nguyen-Widrow
Toleransi galat	0.001
Target Epoch	1500
Laju Pembelajaran	0.01 0.05 0.1 0.2 0.5

Penghilangan neuron yang kurang bermanfaat ini dapat dilakukan dengan dua cara, yaitu dengan membuang sejumlah bobot dari setiap neuron yang memberi kontribusi R^2 kecil atau dengan membuang sejumlah neuron yang berarti membuang seluruh bobot keterhubungan dari neuron yang kurang bermanfaat (Kusumoputro 2004). Pada penelitian ini digunakan pendekatan kedua yaitu membuang bobot keterhubungan dari neuron yang memberikan kontribusi R^2 kecil.

Dalam penelitian ini jumlah perlakuan dengan menggunakan optimasi GA dan tanpa GA masing-masing sebanyak 4 (faktor 1) X 5 (faktor 2) X 5 (faktor 3) yaitu 100 perlakuan dengan perulangan sebanyak 10 kali untuk tiap perlakuan dengan tujuan untuk memperoleh rata-rata R^2 dan $RMSE$ yang memiliki simpangan baku terkecil.

Spesifikasi GA yang digunakan

Pada penelitian ini digunakan GA dengan parameter sebagai berikut :

1. Representasi Solusi : *Binary String*.
2. Populasi : 60 individu.
3. Jumlah Generasi : 100 generasi.
4. Pindah Silang : *Simple Crossover (0.6)*
5. Mutasi : *Binary Mutation (0.0333)*
6. Mekanisme Seleksi : *Roulette Whell*.

Model GA ini berdasarkan model yang pernah dicoba oleh Kusumoputro (2004). Pengkodean kromosom yang dipakai adalah *string* biner, dengan tiap bit dalam *string* kromosom merepresentasikan sebuah neuron. Bit yang bernilai 1 merepresentasikan neuron yang dipertahankan dan bit yang bernilai 0 merepresentasikan neuron yang dibuang.

Representasi Solusi pada Algoritma Genetika

Pengkodean *string biner* sepanjang 16 bit diambil dari jumlah neuron maksimum yang ditentukan pada lapis tersembunyi. Hal ini berdasarkan penelitian oleh Kusumoputro (2004). Selanjutnya bit-bit yang panjangnya 16 tersebut (mulai dari semua bit yang bernilai 0 semua sampai dengan bit yang bernilai 1 semua) masing-masing dikonversi menjadi bilangan berbasis sepuluh sehingga diperoleh selang pencarian bilangan bulat antara 0-65535. Proses konversi dari deret biner sepanjang 16 bit menjadi bilangan bulat x dengan selang [0,65535] dilakukan dengan cara :

$$\langle b_{16} b_{15} b_{14} \dots b_0 \rangle_2 = \left(\sum_{i=0}^{16} b_i \cdot 2^i \right)_{10} = x'$$

Proses konversi bilangan berbasis sepuluh, dalam hal ini merupakan ruang pencarian [0,65535] dihitung dengan cara :

$$x = 0 + x' \cdot \frac{65535}{2^{16} - 1}$$

Dengan selang pencarian 0-65535 maka diambil bilangan acak untuk dikonversi menjadi bilangan berbasis dua (*biner*) dengan panjang 16 bit yang digunakan dalam inisialisasi populasi.

Respon (Fungsi Fitness)

Respon yang diambil dalam penelitian ini adalah nilai R^2 yang dicapai berdasarkan persentase data *training*, persentase keragaman PCA yang digunakan, dan berbagai nilai laju pembelajaran yang digunakan.

Sedang fungsi *fitness* yang dipakai adalah memaksimalkan nilai r (koefisien korelasi) yang secara otomatis memaksimalkan nilai R^2 . Perhitungannya dilakukan sebagai berikut.

$$f(r, RMSE) = r + \frac{1}{RMSE + 0,001}$$

di mana :

r = koefisien korelasi

$RMSE$ = besar simpangan nilai dugaan terhadap nilai aktualnya.

Nilai 0.001 pada persamaan tersebut merupakan suatu konstanta (bilangan) yang dianggap sangat kecil sehingga dalam proses GA fungsi *fitness* tersebut tidak menghasilkan nilai tak hingga (ketika $RMSE$ bernilai 0). Nilai 0.001 merupakan nilai toleransi galat yang digunakan dalam JST standar.

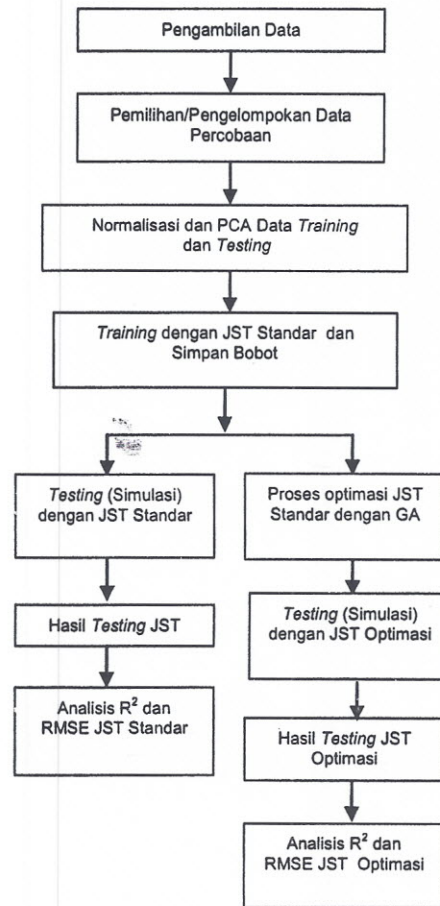
Penambahan nilai $1/(RMSE+0.001)$ dalam fungsi *fitness* diharapkan dapat dihasilkan sebuah nilai R^2 yang optimum berpasangan dengan dengan nilai $RMSE$ yang lebih baik. Akan tetapi dalam penelitian ini fungsi *fitness* yang dibuat bukan untuk meminimumkan nilai $RMSE$ melainkan memaksimalkan nilai R^2 sehingga untuk hasil $RMSE$ dalam percobaan sangat bervariasi (bisa lebih tinggi dari pada JST standar atau lebih rendah).

Gambar 3 menunjukkan diagram proses pendugaan yang dilakukan dalam penelitian ini.

Perangkat Keras dan Lunak yang Digunakan

Penelitian ini akan menggunakan perangkat keras dan lunak, di antaranya :

- a. Microsoft Windows XP Professional
- b. Matlab 6.5
- c. Microsoft Excel XP
- d. AMD Athlon XP 2400 GHz
- e. Memori DDR 256 MB
- f. Hardisk 80 GB



Gambar 3. Diagram proses pendugaan.

Perancangan Penelitian

Tahapan-tahapan yang dilakukan dalam perancangan yaitu dengan merepresentasikan data mentah yang diperoleh menjadi masukan yang dapat diolah menjadi masukan metode pembelajaran dalam JST propagasi balik standar. Setelah dilakukan pembelajaran, akan diperoleh bobot pelatihan yang akan digunakan dalam *testing* pada JST standar dan proses optimasi JST dengan GA.

HASIL DAN PEMBAHASAN

Perbandingan R^2 dan $RMSE$ JST Standar

Dalam percobaan JST standar, kelompok data pertama diperoleh R^2 maksimum sebesar 0.11099 (11,099%) pada keragaman data 99,8% dan laju pembelajaran 0,1 dengan $RMSE$ sebesar 83.512. Untuk kelompok data kedua diperoleh R^2 maksimum sebesar 0.18374 (18.374%) pada keragaman data 95% dan laju pembelajaran 0.05 dengan $RMSE$ 70.258. Untuk kelompok data ketiga diperoleh R^2 maksimum sebesar 0.42052 (42.052%) pada keragaman data 95% dan laju pembelajaran 0.5 dengan $RMSE$ 62.125. Untuk kelompok data keempat diperoleh R^2 maksimum sebesar 0.48179 (48.179%) pada keragaman data 95% dan laju pembelajaran 0.1 dengan $RMSE$ 76.603 (Tabel 3).

Tabel 3. Perbandingan R² dan RMSE JST standar

Kelompok Data	R ²	RMSE
1 (50% data training)	0.11099	83.512
2 (70% data training)	0.18374	70.258
3 (90% data training)	0.42052	62.125
4 (95% data training)	0.48179	76.603

Perbandingan R² dan RMSE JST Optimasi dengan GA

Dalam percobaan menggunakan JST yang sudah dioptimasi dengan GA, kelompok data pertama diperoleh R² maksimum sebesar 0.38732 (38,732%) pada keragaman 95% dan laju pembelajaran 0,01 dengan RMSE sebesar 189.083. Kelompok data kedua diperoleh R² maksimum sebesar 0.37746 (37.746%) pada keragaman 99.8% dan laju pembelajaran 0,2 dengan RMSE 61.893. Kelompok data ketiga diperoleh R² maksimum sebesar 0.51598 (51.598%) pada keragaman 99% dan laju pembelajaran 0,05 dengan RMSE 132.864. Kelompok data keempat diperoleh R² maksimum sebesar 0.87717 (87.717%) pada keragaman 99.8% dan laju pembelajaran 0,1 dengan RMSE 78.472 (Tabel 4).

Tabel 4. Perbandingan R² dan RMSE JST optimasi

Kelompok Data	R ²	RMSE
1 (50% data training)	0.38732	189.083
2 (70% data training)	0.37746	61.893
3 (90% data training)	0.51598	132.864
4 (95% data training)	0.87717	78.472

Perbandingan R² dan RMSE JST Standar dan JST Optimasi

Tabel 5. Perbandingan R² JST standar dan JST optimasi

Kelompok Data	R ²	
	JST Standar	JST Optimasi
1 (50% data training)	0.11099	0.38732
2 (70% data training)	0.18374	0.37746
3 (90% data training)	0.42052	0.51598
4 (95% data training)	0.48179	0.87717

Tabel 6. Perbandingan RMSE JST standar dan JST optimasi

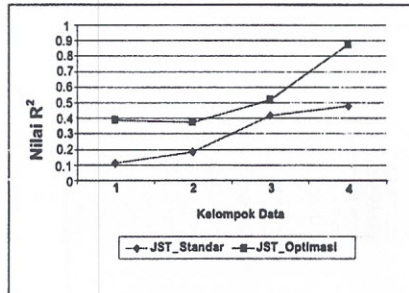
Kelompok Data	RMSE	
	JST Standar	JST Optimasi
1 (50% data training)	83.512	189.083
2 (70% data training)	70.258	61.893
3 (90% data training)	62.125	132.864
4 (95% data training)	76.603	78.472

Sesuai hasil yang terdapat di dalam Tabel 5 terlihat bahwa untuk kedua metode yang digunakan, nilai R² terbaik terletak pada kelompok data keempat. Hal ini berarti bahwa pembagian data menjadi data training dan

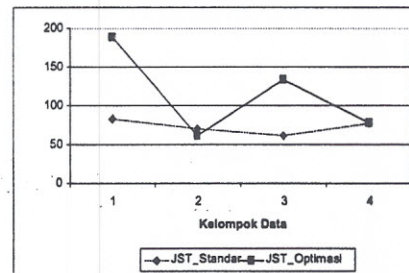
data testing cukup berpengaruh terhadap peningkatan nilai R². Untuk perbandingan nilai R² bagi kedua metode bisa dilihat bahwa nilai R² JST yang sudah dioptimasi lebih tinggi dari pada JST standar.

Berdasarkan Tabel 6 terlihat bahwa nilai RMSE untuk kedua metode cenderung bervariasi tidak tergantung pada pembagian kelompok data. Hal ini berarti bahwa pembagian data menjadi data training dan data testing tidak berpengaruh terhadap penurunan nilai RMSE.

Sebagian besar JST yang sudah dioptimasi nilai RMSE-nya selalu lebih besar dibanding JST standar kecuali pada kelompok data kedua (Gambar 4). Hal ini disebabkan karena dalam optimasi menggunakan GA fungsi fitness yang digunakan adalah memaksimalkan nilai R² bukanlah meminimumkan nilai RMSE sehingga konfigurasi neuron yang didapat merupakan neuron-neuron yang memberikan kontribusi nilai R² yang tinggi (Gambar 5).



Gambar 4. Perbandingan RMSE JST standar dan JST optimasi.

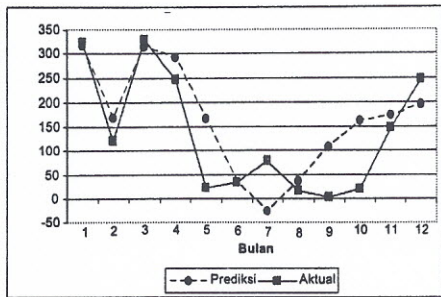


Gambar 5. Perbandingan nilai R² metode JST standar dan JST optimasi.

Arsitektur Terbaik

Pola input ternyata sangat berpengaruh terhadap nilai R² dalam JST. Seperti telah dijelaskan bahwa dalam penelitian ini data dikelompokkan dalam berbagai kelompok percobaan dengan membagi data menjadi data training dan data testing.

Pada kasus ini, semakin banyak bulatan pada nilai prediksi yang menempel pada kotak nilai aktual, maka error pun semakin kecil. Nilai R² sebesar 0.48179 dapat diartikan bahwa 48.179% di antara keragaman dalam nilai-nilai prediksi dapat dijelaskan oleh hubungan linearnya dengan nilai aktual. Pada Gambar 6 curah hujan prediksi pada bulan Juli bernilai negatif. Hal ini dapat diartikan bahwa pada bulan tersebut curah hujan dianggap 0 atau tidak ada hujan.



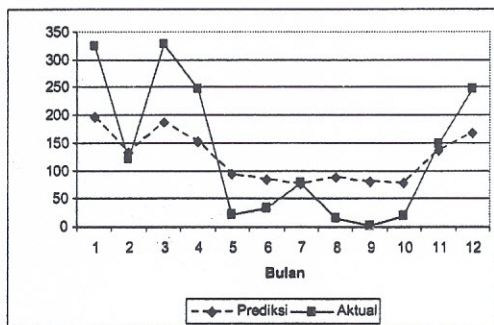
Gambar 6. Nilai prediksi dan aktual arsitektur terbaik JST standar.

Pada percobaan menggunakan JST yang sudah dioptimasi menggunakan GA diperoleh arsitektur terbaik pada kelompok data keempat dengan nilai R^2 sebesar 0.87717 (87.717%) dan RMSE sebesar 78.472 dengan konfigurasi struktur neuron pada lapisan tersembunyi

0 1 0 1 1 1 0 0 1 0 1 1 0 1 0 0

yang berarti sebanyak delapan neuron pada lapisan tersembunyi yang dibuang.

Pada Gambar 7 terdapat beberapa bulatan nilai prediksi yang hampir berhimpit dengan nilai aktual sehingga nilai error pun bisa semakin kecil. Nilai R^2 sebesar 0.87717 dapat diartikan bahwa 87.717% di antara keragaman dalam nilai-nilai prediksi dapat dijelaskan oleh hubungan linearnya dengan nilai aktual.



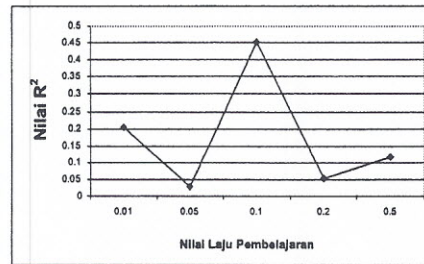
Gambar 7. Nilai prediksi dan aktual arsitektur terbaik JST optimasi.

Pengaruh Laju Pembelajaran terhadap R^2

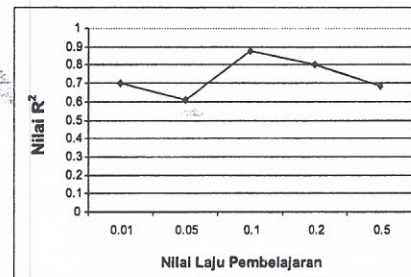
Berdasarkan Gambar 8 bisa dilihat bahwa nilai laju pembelajaran terhadap nilai R^2 JST standar pada kelompok data dan PCA arsitektur terbaik, yaitu 95% data *training* dan PCA 95% sangat bervariasi. Pada grafik bisa dilihat bahwa laju pembelajaran 0.1 memiliki nilai R^2 maksimum.

Hal yang sama juga terjadi pada JST optimasi. Pada kelompok data dan PCA arsitektur terbaik, yaitu 95% data *training* dan PCA 99.8%, nilai R^2 yang diperoleh sangat bervariasi. Pada grafik bisa dilihat bahwa laju pembelajaran 0.1 memiliki nilai R^2 maksimum.

Dari Gambar 9 dan Gambar 10, untuk setiap laju pembelajaran hasil R^2 rata-rata pada JST optimasi selalu lebih baik dari hasil R^2 rata-rata pada JST standar.



Gambar 8. Pengaruh laju pembelajaran terhadap R^2 JST standar.



Gambar 9. Pengaruh laju pembelajaran terhadap R^2 JST optimasi.

Perbandingan dengan Hasil Penelitian Sebelumnya

Hasil R^2 dalam penelitian menggunakan JST yang dioptimasi lebih baik dari pada penelitian sebelumnya yang menggunakan JST propagasi balik standar dan metode PCR.

Tabel 8. Perbandingan nilai R^2 dan RMSE dengan penelitian sebelumnya

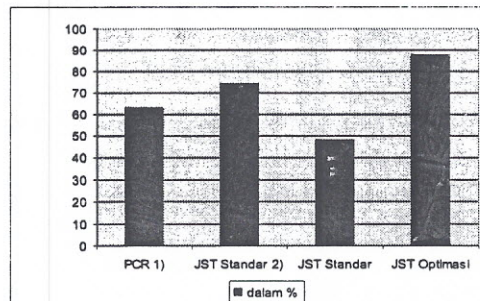
Metode	R^2	RMSE
PCR ¹⁾	63.16%	74.24
JST Standar ²⁾	74.02%	65.16
JST Standar	48.179%	76.603
JST Optimasi	87.717%	78.472

Keterangan :

¹⁾ Fitriadi 2004.

²⁾ Normakristagaluh 2004.

Nilai R^2 74.02% pada percobaan dengan JST standar oleh Normakristagaluh (2004) terletak pada jumlah neuron lapisan tersembunyi sebanyak 9 dan laju pembelajaran 0.02 yang merupakan arsitektur terbaik.



Gambar 10. Perbandingan R^2 beberapa penelitian.

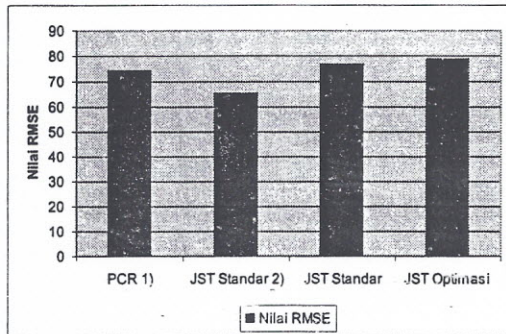
Keterangan :

¹⁾ Fitriadi 2004

²⁾ Normakristagaluh 2004

Dari Tabel 8 terlihat bahwa nilai R^2 penelitian ini yang menggunakan optimasi lebih besar disebabkan karena di dalam GA terdapat pembuangan neuron yang memberikan kontribusi nilai R^2 yang kecil dibuang sehingga yang tersisa adalah neuron-neuron yang memberikan kontribusi nilai R^2 yang lebih besar.

Untuk nilai RMSE belum terjadi penurunan dari penelitian sebelumnya. Hal ini disebabkan karena fungsi *fitness* yang digunakan bukan untuk meminimumkan nilai RMSE.



Gambar 11. Perbandingan RMSE beberapa penelitian.

Keterangan :

¹⁾ Fitriadi 2004.

²⁾ Normakristagaluh 2004.

KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan penelitian yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut :

1. Algoritma genetika dapat digunakan dalam optimasi salah satu parameter dalam JST, dalam hal ini struktur *neuron* pada satu lapis tersembunyi untuk meningkatkan nilai pendugaan berupa nilai R^2 , yaitu meningkat dari 48.179% menjadi 87.717% dengan RMSE 78.472.
2. Arsitektur terbaik dengan nilai R^2 terbaik pada percobaan ini terjadi pada kelompok data keempat yaitu 95% data *training* dan 5% data *testing* untuk kedua percobaan dan pada *learning rate* 0,1, yaitu 48.179% untuk JST standar dan 87.717% untuk JST optimasi.
3. Model JST optimasi selalu menghasilkan nilai R^2 yang lebih tinggi dari pada JST standar (bertambah 9.5% sampai 38.5%).

Saran

Penelitian ini masih memiliki banyak kekurangan sehingga masih perlu pengembangan penelitian selanjutnya.

Oleh sebab itu disarankan hal-hal sebagai berikut :

1. Penambahan parameter GCM selain suhu dan curah hujan, seperti kelembaban, arah angin, dan parameter lain yang berkorelasi dengan curah hujan.
2. Menambahkan mekanisme elitisme dan operator *crossover* yang berbeda.
3. Penelitian terhadap optimasi nilai RMSE.

DAFTAR PUSTAKA

- Fauset, L. 1994. *Fundamentals of Neural Networks*. Prentice Hall, New Jersey.
- Fitriadi. 2004. *Kombinasi Model Regresi Komponen Utama dan Arima dalam Statistical Downscaling*. Skripsi. Departemen Statistika FMIPA IPB, Bogor.
- Fu, LM. 1994. *Neural Networks in Computer Intelligence*. Singapore : Mc Graw-Hill.
- Gen, M. & Runwei C. 1997. *Genetics Algorithms and Engineering Design*. John Wiley & Sons, Inc. Canada.
- Kusumoputro, B. 2004. *Pengembangan Sistem Pengenal Wajah secara 3 Dimensi menggunakan Hemisphere Structure of Neural Networks dan Optimasi Struktur menggunakan Algoritma Genetika*. Makalah SNIKTI V. Fakultas Ilmu Komputer, UI, Depok.
- Lestari B. 2000. *Pemodelan dan Peramalan banyaknya Hari Hujan di Jember dengan Proses ARMA*. Majalah Matematika dan Statistika. Volume 1, No. 1, pp: 61-72
- Michalewicz, Z. 1995. *Genetic Algorithms + Data Structures = Evolution Programs*. W. H. Freeman and Company. New York.
- Normakristagaluh, P. 2004. *Penerapan Jaringan Syaraf Tiruan untuk Peramalan Curah Hujan dalam Statistical Downscaling*. Skripsi. Departemen Ilmu Komputer FMIPA IPB. Bogor.
- Suyanto. 2005. *Algoritma Genetika dalam Matlab*. Yogyakarta : Andi
- Walpole, E.R. 1995. *Pengantar Statistika*. PT Gramedia Pustaka Utama, Jakarta.