

Hybrid Jaringan Saraf Tiruan Backpropagation dengan Firefly Algorithm dan Simulated Annealing untuk Peramalan Curah Hujan di Surabaya

Dicky Zulfikar Zurkarnain¹, Auli Damayanti^{2,*} & Edi Winarko³

^{1,2,3}Departemen Matematika Fakultas Sains dan Teknologi Universitas Airlangga, Surabaya, Indonesia

*Corresponding author Email: auli-d@fst.unair.ac.id

Abstrak. Indonesia mempunyai berbagai jenis iklim. Salah satu parameter iklim adalah curah hujan. Curah hujan yang dapat menjadi sumber bencana adalah curah hujan ekstrem, yaitu kondisi curah hujan yang cukup tinggi/rendah dari rata-rata kondisi normalnya. Informasi tentang peramalan curah hujan sangat berguna khususnya bagi pemerintah kota Surabaya dalam mengantisipasi kemungkinan kejadian-kejadian atau bencana yang diakibatkan oleh curah hujan ekstrem seperti, kekeringan, banjir, pohon tumbang, rusaknya fasilitas umum, dll. Tujuan dari penulisan skripsi ini adalah untuk mendapatkan nilai peramalan curah hujan di Surabaya pada bulan yang akan datang menggunakan Hybrid Jaringan Saraf Tiruan Backpropagation dengan Firefly Algorithm dan Simulated Annealing. Proses diawali dengan input dan normalisasi data, kemudian dilanjutkan dengan proses pelatihan untuk mencari bobot dan bias yang optimal. Setelah diperoleh bobot dan bias yang optimal, kemudian melakukan uji validasi, dan dilanjutkan dengan proses peramalan. Pada proses peramalan curah hujan, data yang digunakan sebanyak 120 data curah hujan bulanan dari bulan Januari 2008 hingga bulan Desember 2017 dengan ketentuan 80% data untuk pelatihan dan 20% data untuk uji validasi. Data yang digunakan, selanjutnya dilatih kemudian dicarilah Mean Square Error (MSE) dan bobot yang optimal. Bobot optimal yang diperoleh, selanjutnya diuji dengan uji validasi untuk mengetahui seberapa baik pola yang dikenali. Berdasarkan implementasi pada data curah hujan tersebut, diperoleh nilai MSE hasil pelatihan sebesar 0.0395384228 dan nilai selisih rata-rata sebesar 3,75382. Sedangkan hasil peramalan untuk 3 bulan berikutnya yaitu bulan Januari hingga Maret 2018 berturut-turut adalah 6.1451, 8.5459, dan 7.7391.

Keywords: Jaringan Saraf Tiruan Backpropagation, Firefly Algorithm, Simulated Annealing, Peramalan, Curah Hujan.

1 Pendahuluan

Indonesia mempunyai berbagai jenis iklim. Secara geografis Indonesia terletak di antara Samudra Pasifik dan Samudra Hindia, antara benua Asia dan benua Australia, dan pada pertemuan dua rangkaian pegunungan, yaitu Sirkum Pasifik dan Sirkum Mediterranean. Sebagian besar wilayah Indonesia merupakan lautan, sedangkan wilayah daratan pada umumnya memiliki permukaan yang bergunung-gunung. Kondisi wilayah tersebutlah yang menyebabkan adanya keragaman iklim di wilayah Indonesia.

Salah satu parameter iklim adalah curah hujan. Curah hujan menjadi sumber daya alam yang amat dibutuhkan, juga dapat menjadi sumber bencana. Curah hujan yang dapat menjadi sumber bencana adalah curah hujan ekstrem, yaitu kondisi curah hujan yang cukup tinggi/rendah dari rata-rata kondisi normalnya. Curah hujan ekstrem biasanya disebabkan oleh fenomena siklus El-Nino dan La-Nina. El-Nino menyebabkan kekeringan karena curah hujan sangat rendah, dan La-Nina menyebabkan banjir karena curah hujan sangat tinggi [1].

Surabaya adalah ibukota provinsi Jawa Timur dengan jumlah populasi penduduk yang padat. Informasi tentang peramalan curah hujan sangat berguna khususnya bagi pemerintah kota Surabaya dalam mengantisipasi kemungkinan kejadian-kejadian atau bencana yang diakibatkan oleh curah hujan ekstrem seperti, kekeringan, banjir, pohon tumbang, rusaknya fasilitas umum, dll. Di kota Surabaya sendiri, banjir masih sering terjadi jika ada hujan dengan intensitas tinggi. Menurut [2], Surabaya adalah salah satu kota dengan persoalan banjir yang besar karena selalu mengalami banjir setiap tahunnya dengan kerugian yang cukup besar. Kerugian dan kerusakan akibat banjir adalah sebesar dua pertiga dari semua bencana alam yang terjadi.

Jaringan saraf tiruan (*Artificial Neural Network*) adalah salah satu cabang dari ilmu kecerdasan buatan (*Artificial Intelligent*) yang sering digunakan untuk masalah pengelompokan dan pengenalan suatu pola. Karena jaringan saraf tiruan dirancang seperti cara kerja otak manusia. Sehingga jaringan saraf tiruan lebih unggul dibandingkan dengan algoritma konvensional lainnya khususnya untuk pengelompokan dan pengenalan pola. Jaringan saraf tiruan juga menggunakan proses pembelajaran atau pelatihan dalam prosesnya. Proses pelatihan data ini bertujuan untuk memperoleh bobot dan bias yang optimal yang sesuai dengan pola data yang dilatih [3]. Jaringan saraf tiruan mempunyai kemampuan belajar yang baik dan dapat dengan mudah beradaptasi dengan lingkungannya. Jaringan saraf tiruan dapat mengenali pola dari berbagai macam distribusi data termasuk data yang tidak teratur dan tidak berkaitan [4].

Jaringan saraf tiruan *Backpropagation* adalah salah satu diantara banyak metodologi yang sangat luas digunakan untuk menyelesaikan masalah pengenalan pola karakter. Struktur Jaringan saraf tiruan *Backpropagation* cukup sederhana, algoritmanya matang dan dapat diaplikasikan di berbagai bidang. *Backpropagation* adalah metode *gradient descent* yang mana persentase kesalahan dihitung dengan memperhatikan bobot yang diberikan pada *input* awal dengan perambatan balik dari *output layer* menuju *hidden layer* dan selanjutnya ke *input layer* [5].

Jaringan Saraf Tiruan *Bacpropagation* bekerja dengan baik pada permasalahan pelatihan sederhana. Jika kompleksitas permasalahan meningkat, maka daya kerja dari *backpropagation* ini turun drastis kedalam minimum lokal, dan proses pelatihannya sangat lambat karena proses pencarian bobot. Untuk mengatasi keterbatasan ini,

beberapa pengombinasian Jaringan saraf tiruan *Backpropagation* telah dikembangkan, yaitu mengombinasikan dengan algoritma metaheuristik seperti *Particle Swarm Optimization* (PSO), *Firefly Algorithm* (FA), dan *Genetic Algorithm* (GA) yang dapat dimasukkan ketika proses pelatihan jaringan [5].

Algoritma Kunang-kunang (*Firefly Algorithm*) adalah algoritma metaheuristik yang terinspirasi dari kecerdasan koloni hewan (*Swarm Intelligence*) yaitu perilaku berkedipnya cahaya kunang-kunang dan cara komunikasi kunang-kunang melalui cahaya yang dimilikinya [6]. Algoritma kunang-kunang mempunyai kelebihan yaitu tingginya tingkat konvergensi dan masing-masing individu *firefly* memperoleh optimum global dengan jumlah iterasi yang sedikit [5].

Simulated Annealing (SA) adalah salah satu algoritma untuk optimasi yang terinspirasi dari bidang metalurgi *annealing*, yaitu suatu teknik yang digunakan dalam mempelajari proses pembentukan kristal dalam suatu materi. Agar dapat terbentuk susunan kristal yang sempurna, diperlukan pemanasan sampai tingkat tertentu, kemudian dilanjutkan dengan pendinginan yang perlahan-lahan dan terkendali dari materi tersebut. Algoritma ini dapat digunakan untuk mencari pendekatan terhadap solusi optimum lokal dari suatu permasalahan [7]. Berbeda dari banyak metode yang ada, algoritma *Simulated Annealing* tidak mudah terjebak di minimum lokal. Kemampuan algoritma *Simulated Annealing* dalam menyelesaikan permasalahan optimasi yang kompleks sangat bagus [8].

Berdasarkan penelitian [5], untuk meningkatkan hasil solusi dari jaringan saraf tiruan dengan proses pelatihan *Backpropagation*, digunakan algoritma optimasi untuk menentukan bobot optimal dalam tahap pembelajaran. Dalam [9] proses optimasi bobot jaringan saraf tiruan *Backpropagation* menggunakan *Firefly Algorithm* untuk memperoleh bobot optimal dibutuhkan iterasi lebih sedikit dibandingkan dengan menggunakan *Genetic Algorithm* (GA). Dalam [8] proses pelatihan jaringan saraf tiruan dengan algoritma optimasi telah menunjukkan bahwa metode tersebut menjadi pendekatan yang efektif untuk mencari solusi optimal. Pada kombinasi jaringan saraf tiruan dengan *Simulated Annealing* (SA), *Simulated Annealing* digunakan pada proses pelatihan pada jaringan saraf tiruan. Oleh karena itu, pada penelitian kali ini, Jaringan saraf tiruan *Backpropagation* digunakan untuk peramalan curah hujan di Surabaya dan *Firefly Algorithm* serta *Simulated Annealing* digunakan untuk optimasi bobotnya. Dengan menggunakan Jaringan saraf tiruan *Backpropagation* untuk peramalan diharapkan dapat membantu pemerintah kota Surabaya dalam pengambilan keputusan serta mengambil langkah antisipatif sebelum datangnya curah hujan ekstrem, sehingga bisa mengurangi dampak dan kerugian yang diakibatkan oleh curah hujan ekstrem.

Dengan adanya kekurangan dan kelebihan yang ada pada jaringan saraf tiruan dan *firefly algorithm* ini, maka menarik untuk dilakukan pengombinasian (*hybrid*) jaringan saraf tiruan metode *extreme learning machine* dengan *firefly algorithm* pada proses pelatihan

jaringan, agar pencarian bobot dan bias dalam proses pelatihannya menjadi lebih cepat. Dengan demikian diharapkan setelah proses pelatihan berlangsung, diperoleh solusi yang lebih baik sehingga akurat untuk memprediksi nilai tukar rupiah terhadap dolar Amerika Serikat.

2 Landasan Teori

Beberapa penjelasan mengenai teori yang berkaitan dengan penelitian dijelaskan pada bagian ini.

A. Curah Hujan

Curah hujan dapat diartikan sebagai ketinggian air yang terkumpul dalam tempat yang datar, tidak menguap, tidak meresap, dan tidak mengalir. Untuk mengukur curah hujan, digunakan alat yang disebut Observarium dan umumnya curah hujan dinyatakan dalam milimeter. Curah hujan satu milimeter artinya pada luasan satu meter persegi dalam tempat yang datar tertampung air setinggi satu milimeter atau tertampung air sebanyak satu liter (1 inchi=25,4 mm). jika curah hujan dihitung pada rentang kumulatif tertentu maka disebut curah hujan kumulatif dan perbandingan antara jumlah curah hujan selama waktu tertentu yang diterapkan (satu periode musim kemarau atau musim hujan) dengan jumlah curah hujan normalnya (rata-rata selama 30 tahun) disebut sebagai sifat hujan [10]. Sifat hujan berdasarkan BMKG dibagi menjadi tiga sifat, yaitu atas normal, normal dan bawah normal. Hujan dikatakan normal apabila tinggi hujan yang terjadi pada suatu musim berada pada selang antara 85% samapai 115% dari nilai rata-rata hujan jangka panjang. Dikatakan bawah normal apabila tinggi hujan kurang dari 85 % dari nilai rata-rata dan diatas normal apabila tinggi hujan lebih besar dari 115% dari nilai rata-rata. Pada saat fenomena La-Nina berlangsung, biasanya sifat hujan di Indonesia khususnya yang memiliki pola hujan monsoon akan dia atas normal, sebaliknya kalau fenomena El-Nino yang berlangsung sifat hujan akan dibawah normal.

B. Jaringan Saraf Tiruan Backpropagation

Jaringan saraf *backpropagation* merupakan jaringan saraf yang menggunakan arsitektur *multilayer*. Dalam jaringan, selain terdapat unit-unit *input*, unit-unit tersembunyi (*hidden*) dan *output* jugaterdapat bias yang diberikan pada unit-unit tersembunyi dan *output*. Unit *input* akan dilambangkan dengan X , unit *hidden* dilambangkan dengan Z , dan unit *output* dilambangkan dengan Y . Sedangkan, untuk bobot antara X dan Z dilambangkan dengan v dan bobot antara Z dan Y dilambangkan dengan w . Untuk bias, biasanya dipakai indeks indeks v_{01} dan w_{01} . Pada intinya, pelatihan dengan metode *backpropagation* terdiri atas tiga langkah, yaitu :

1. Umpan maju, data dimasukkan ke input jaringan (*feedforward*). Saat umpan maju, setiap unit input (X_i) akan menerima sinyal *input* dan akan menyebarkan sinyal

tersebut pada tiap unit *hidden* (Z_j). Setiap unit *hidden* kemudian akan menghitung aktivasinya dan mengirim sinyal (z_j) ke tiap unit *output*. Kemudian, setiap unit *output* (Y_k) juga akan menghitung aktivasinya (y_k) untuk menghasilkan respon terhadap *input* yang diberikan jaringan.

2. Propogasi error, perhitungan dan propagasi balik dari *error* yang bersangkutan. Saat proses pelatihan, setiap unit *output* membandingkan aktivasinya (y_k) dengan nilai target untuk menentukan besarnya *error*. Berdasarkan *error* tersebut, dihitung factor δ_k . Faktor δY_k digunakan untuk mendistribusikan *error* dan *output* kembali ke *layer* sebelumnya. Dengan cara yang sama, faktor δZ_k juga dihitung pada unit *hidden* Z_j . Faktor δ_k digunakan untuk memperbaharui bobot antara *hidden layer* dan *input layer*.
3. Pembaharuan bobot dan bias. Setelah semua faktor δ ditentukan, bobot untuk semua *layer* diperbaharui secara bersamaan. Pembaharuan bobot w_{jk} (dari unit *hidden* Z_j ke unit *output* Y_k) dilakukan berdasarkan faktor δY_k dan aktivasi z_j dari unit *hidden* Z_j . Sedangkan, pembaharuan bobot v_{ij} (dari unit input X_i ke unit *hidden* Z_j) dilakukan berdasarkan faktor δZ_j , dan aktivasi x_i , dari *input*.

Pada tiap langkah yang dijalani, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi *sigmoid biner*. Fungsi aktivasi tersebut digunakan pada proses *input* ke *hidden* maupun dari *hidden* ke *output* [3].

C. Firefly Algorithm

Menurut [6], terdapat dua hal yang berkaitan dan sangat penting dalam *Firefly Algorithm* yaitu intensitas cahaya dan fungsi keatraktifan. Diasumsikan bahwa keatraktifan dipengaruhi tingkat intensitas cahaya. Dalam kasus optimasi maksimum, tingkat intensitas cahaya pada sebuah *firefly* x dapat dilihat sebagai,

$$I(x) \approx f(x) \quad (1)$$

dengan nilai I adalah tingkat intensitas cahaya pada *firefly* x yang sebanding dengan solusi fungsi tujuan permasalahan yang akan dicari $f(x)$. Nilai keatraktifan *firefly* (β) adalah relatif, karena dipengaruhi besar kecilnya intensitas cahaya. Sehingga hasil penilaian akan berbeda juga tergantung dari jarak antar *firefly* (r_{ij}). Selain itu, intensitas cahaya akan menurun dilihat dari sumbernya dikarenakan terserap oleh media, contohnya udara. Fungsi keatraktifan ($\beta(r)$) diberikan sebagai berikut:

$$\beta(r) = \beta_0 e^{-\gamma r^2} \quad (2)$$

dengan β_0 adalah nilai keatraktifan *firefly* awal, γ adalah koefisien penyerapan cahaya, r adalah jarak antar *firefly*. Seberapa besar suatu *firefly* dapat melihat cahaya *firefly* yang lain dipengaruhi oleh faktor jarak. Semakin jauh jarak antar *firefly* semakin redup cahaya

dari *firefly* yang berkedip. Jarak antara *firefly* i dan j pada posisi x masing-masing adalah x_i dan x_j yang dapat dihitung menggunakan rumus sebagai berikut:

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_i^k - x_j^k)^2} \quad (3)$$

dengan d adalah jumlah komponen tiap individu *firefly*, x_i^k adalah komponen ke- k dari *firefly* ke- i (x_i) dan x_j^k adalah komponen ke- k dari *firefly* ke- j (x_j). Pergerakan dari *firefly* i yang bergerak menuju *firefly* j yang mempunyai intensitas cahaya yang lebih cerah dapat dilihat dari persamaan berikut:

$$x_i^{baru} = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha (rand - 0.5) \quad (4)$$

dengan $\beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i)$ adalah perpindahan *firefly* yang terjadi karena adanya daya tarik (keatraktifan *firefly*) dan $\alpha (rand - 0.5)$ adalah pergerakan random *firefly* dengan α adalah koefisien parameter random dan $rand$ adalah bilangan random berdistribusi uniform pada interval $[0,1]$. Pada sebagian besar implementasi, digunakan $\beta_0 = 1$, $\alpha \in [0,1]$ dan $\gamma \in [0, \infty)$.

Langkah-langkah dasar dari *Firefly Algorithm* untuk menyelesaikan suatu permasalahan sebagai berikut:

1. Inisialisasi parameter *Firefly Algorithm*, yaitu koefisien penyerapan cahaya (γ), banyaknya *firefly* (m), keatraktifan awal (β_0), koefisien parameter random (α).
2. Mendefinisikan fungsi objektif (x) dari permasalahan yang akan diselesaikan dengan *Firefly Algorithm*.
3. Membangkitkan secara random populasi awal dengan *generate firefly* sebanyak m .
4. Menentukan intensitas cahaya tiap *firefly* (x) berdasarkan nilai fungsi tujuan (x).
5. Membandingkan intensitas cahaya masing-masing *firefly* dengan *firefly* lainnya. Jika intensitas cahaya *firefly* i lebih rendah dari *firefly* j maka terjadi pergerakan dari *firefly* i ke *firefly* j .
6. Menentukan *G-best*. Untuk iterasi pertama, *firefly* terbaik (*firefly* dengan intensitas cahaya terbesar) adalah *G-best*.
7. Melakukan proses di atas sampai batas iterasi dipenuhi.

D. Simulated Annealing

Nama dan inspirasi dari *Simulated Annealing* berasal dari bidang metalurgi yaitu *Annealing*. *Annealing* adalah suatu teknik yang digunakan dalam mempelajari proses pembentukan kristal dalam suatu materi. Agar dapat terbentuk susunan kristal yang sempurna, diperlukan pemanasan sampai tingkat tertentu, kemudian dilanjutkan dengan pendinginan yang perlahan-lahan dan terkendali dari materi tersebut. Pemanasan materi

diawal proses annealing memberikan kesempatan pada atom-atom dalam materi tersebut untuk bergerak secara bebas. Proses pendinginan yang perlahan-lahan memungkinkan atom-atom dari materi tersebut untuk menemukan tempat yang optimum, yaitu kondisi dimana energi internal yang dibutuhkan atom itu untuk mempertahankan posisinya adalah minimum.

Prosedur *Simulated Annealing* secara garis besar dapat dijabarkan sebagai berikut [7]:

1. Inisialisai temperatur awal $T_{sekarang}$.
2. Inisialisasi populasi awal v_i .
3. Evaluasi populasi awal sebagai solusi sementara $f(v_i)$.
4. Modifikasi populasi awal v_m dan evaluasi kembali sebagai solusi baru $f(v_m)$.
5. Jika $f(v_m) \leq f(v_i)$ maka solusi sementara sama dengan solusi baru. Jika tidak maka dibangkitkan dengan acak bilangan real r pada interval $[0,1]$. Hitung probabilitas $p = e^{\frac{\Delta f}{T}}$ dengan Δf merupakan selisih dari $f(v_i)$ dan $f(v_m)$, Sedangkan T adalah suhu saat ini. Jika $p > r$ maka solusi baru tetap diterima sebagai solusi sementara.
6. Jika sudah mencapai $T_i < T_{akhir}$, maka algoritma berhenti. Jika tidak, hitung perubahan temperatur $T_i = tT_{sekarang}$ dengan $0 < t < 1$ dan kembali ke Langkah 4.

3 Metode Penelitian

Langkah-langkah yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Melakukan studi pustaka mengenai curah hujan, peramalan, *firefly algorithm* (FA), algoritma *simulated annealing* (SA) dan jaringan saraf tiruan *backpropagation* (JSTB).
2. Melakukan normalisasi data. Proses normalisasi, dilakukan pada data yang akan digunakan dan dirubah pada interval $[0,1]$. Rumus normalisasi yang digunakan untuk mengubah data ke interval $[0, 1]$ adalah :

$$x = \frac{0,8 \times (x_d - \min(x_d))}{(\max(x_d) - \min(x_d))} + 0.1 \quad (5)$$

dengan :

x_d = data asli yang belum dilakukan proses normalisasi

$\min(x_d)$ = nilai minimum pada seluruh data

$\max(x_d)$ = nilai maksimum pada seluruh data

3. Melakukan pembagian data menjadi data pelatihan dan data validasi dengan komposisi 80% dari total data sebagai data pelatihan 20% dari total data sebagai

data validasi.

4. Merancang data yang akan menjadi *input*. Jumlah neuron *input* yang digunakan adalah sebanyak 3 unit yang artinya mewakili data sebanyak 3 bulan. Sedangkan target output yang digunakan adalah data bulan ke-4 dan seterusnya. Berikut rancangan data yang nantinya akan digunakan disajikan pada Tabel 1.

Tabel 1 Rancangan data

Pola ke-	Data Input			Target
	x1	x2	x3	
1	Bulan ke-1	Bulan ke-2	Bulan ke-3	Bulan ke-4
2	Bulan ke-2	Bulan ke-3	Bulan ke-4	Bulan ke-5
3	Bulan ke-3	Bulan ke-4	Bulan ke-5	Bulan ke-6
.
.
.
<i>n-3</i>	Bulan ke-(<i>n-3</i>)	Bulan ke-(<i>n-2</i>)	Bulan ke-(<i>n-1</i>)	Bulan ke- <i>n</i>

5. Merencanakan arsitektur jaringan saraf tiruan dengan metode *Backpropagation*. Pada penelitian ini, jaringan hanya terdiri dari 3 *layer*, yaitu sebuah *input layer*, sebuah *hidden layer*, dan sebuah *output layer*. Rancangan jumlah unit pada tiap *layer*:
 - a. Pada *input layer* terdapat jumlah neuron sebanyak 3 unit, yaitu data setiap 3 bulan.
 - b. Pada *hidden layer* terdapat jumlah neuron sebanyak 3 unit.
 - c. Pada *output layer* terdapat jumlah neuron sebanyak 1 unit
 - d. Fungsi aktivasi yang digunakan adalah fungsi aktivasi *sigmoid biner*.
6. Menerapkan *Hybrid Jaringan Saraf Tiruan Backpropagation* dengan *Firefly Algorithm* dan *Simulated Annealing* pada proses pelatihan data dengan langkah-langkah sebagai berikut :
 - a. Mempersiapkan data yang akan digunakan untuk pelatihan.
 - b. Menentukan banyaknya populasi *firefly* (n), keatraktifan awal (β_0), koefisien penyerapan udara (γ), koefisien nilai *random* (α), target error (*maks_error*), maksimum iterasi (*maks_iterasi*), suhu awal (T_0), suhu akhir (T_a), dan koefisien penurunan suhu (t).
 - c. Membangkitkan populasi awal pada FA dengan cara membangkitkan bilangan real secara random pada interval [0,1] sebanyak n populasi. Jumlah komponen tiap-tiap individu *firefly* sebanyak 16 unit. Populasi yang dibangkitkan merupakan bobot dan bias yang akan digunakan dalam pelatihan JSTB.

- d. Konversi individu *firefly* dalam FA menjadi bobot dan bias dalam JST.
- e. Melakukan proses pelatihan Jaringan Saraf Tiruan *Backpropagation* pada bobot dan bias masing-masing individu *firefly*. Selanjutnya didapatkan nilai MSE untuk masing-masing individu *firefly*.
- f. Menghitung nilai intensitas cahaya dari nilai MSE yang diperoleh tiap individu *firefly* menggunakan rumus berikut:

$$I(x) = \frac{1}{1+MSE} \quad (5)$$

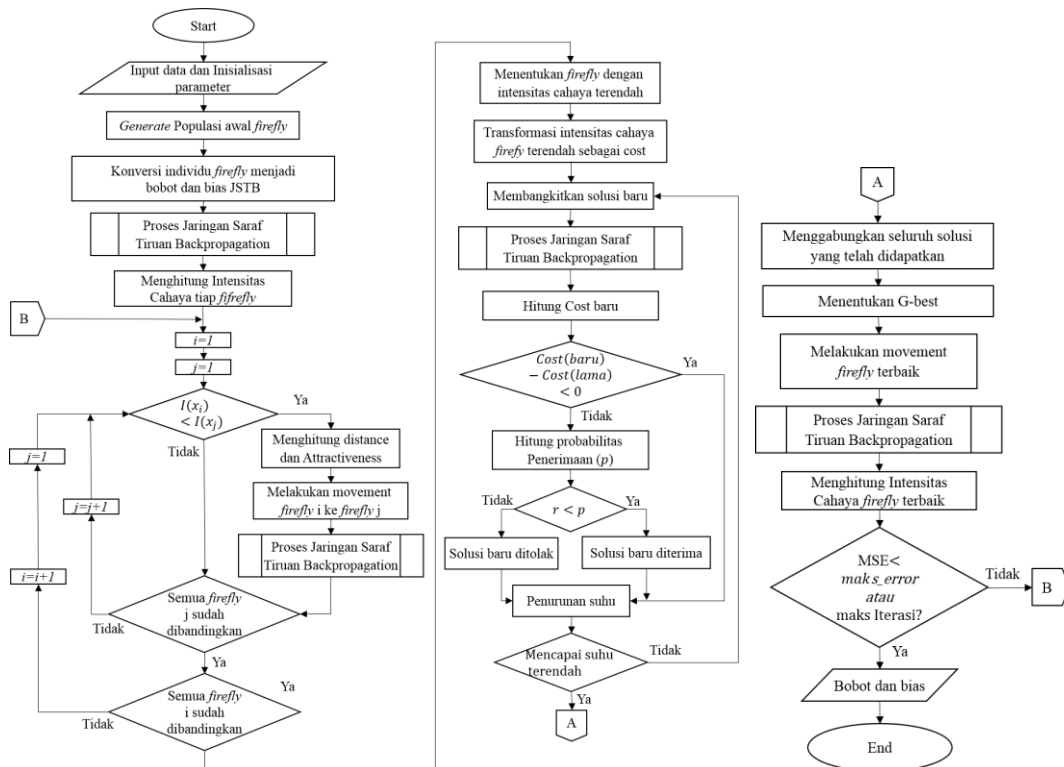
- g. Membandingkan intensitas cahaya *firefly* ke- *i* dengan *firefly* ke- *j*. Jika intensitas cahaya *firefly* ke- *i* lebih besar dari *firefly* ke- *j*, ulangi langkah ini dengan $j = j + 1$. Jika intensitas cahaya *firefly* ke- *i* lebih kecil dari *firefly* ke- *j*, hitung jarak antara *firefly* ke- *i* dan *firefly* ke- *j*, hitung nilai keaktraktifan *firefly* dan lakukan pergerakan dari *firefly* ke- *i* menuju *firefly* ke- *j*.
- h. Mengevaluasi fungsi tujuan (MSE) dan intensitas cahaya setelah pergerakan seperti Langkah 6.e-f.
- i. Ulangi langkah 6.g dengan $i = i + 1$ sampai seluruh *firefly* dibandingkan.
- j. Tentukan *firefly* terburuk sebagai solusi awal dalam proses *Simulated Annealing* dan hitung *cost* dari solusi awal tersebut menggunakan rumus :

$$C(x) = \frac{1}{I(x)} \quad (6)$$

- k. Melakukan proses *Simulated Annealing* dengan langkah-langkah sebagai berikut :
 - i. Membangkitkan solusi baru, yaitu dengan memodifikasi *firefly* terburuk menggunakan mutasi.
 - ii. Mengevaluasi solusi baru, proses evaluasi dilakukan seperti Langkah 6.e-f, dengan intensitas cahaya yang didapat kemudian ditransformasikan menjadi *cost* untuk solusi baru.
 - iii. Hitung selisih *cost* lama dan *cost baru* (Δf).
 - iv. Jika Δf negatif maka solusi baru diterima, jika tidak maka *random* bilangan real $r \in [0,1]$, hitung probabilitas penerimaan, apabila $r < p$ maka solusi baru diterima, jika tidak maka solusi baru ditolak.
 - v. Lakukan penurunan suhu. Cek apakah $T < T_{akhir}$. Jika iya, proses *Simulated Annealing* dihentikan. Jika tidak, maka ulangi dari Langkah 6.k.i.
- l. Menggabungkan seluruh solusi yang telah didapatkan.
- m. Menentukan *G-best* yaitu calon solusi terbaik. Pada iterasi pertama, *firefly* terbaik menjadi *G-best*. Untuk iterasi lainnya, bandingkan *firefly* terbaik dengan *G-best*. Jika intensitas cahaya *firefly* terbaik lebih besar daripada intensitas cahaya *G-best* maka *firefly* tersebut akan menjadi *G-best*.
- n. Update *G-best* sebagai calon solusi terbaik.

- o. Melakukan pergerakan pada *firefly* terbaik. Setelah itu, hitung MSE dan intensitas cahaya *firefly* setelah melakukan pergerakan.
 - p. Cek apakah nilai MSE *G-best* yang diperoleh kurang dari *maks_error* atau maksimum iterasi (*maks_iterasi*) telah terpenuhi. Jika iya, proses dilanjutkan ke Langkah 6.q. Jika tidak, proses dilanjutkan ke Langkah 6.g.
 - q. Menyimpan hasil bobot.
7. Melakukan uji validasi data pelatihan menggunakan nilai bobot dan bias yang diperoleh dari proses pelatihan. Langkah-langkah uji validasi sebagai berikut:
- a. *Input* data validasi yang akan diuji.
 - b. Menginisialisasi bobot dan bias optimal dalam JSTB. Inisialisasi ini didapatkan dari langkah 6.
 - c. Masukkan parameter yang digunakan pada proses pelatihan.
 - d. Lakukan proses *feedforward* dalam JSTB.
 - e. Menampilkan *output* dari proses *feedforward*.

Flowchart proses Jaringan Saraf Tiruan *Backpropagation* dengan *Firefly Algorithm* dan *Simulated Annealing* dapat dilihat pada Gambar 1.



Gambar 1 Flowchart proses Jaringan Saraf Tiruan *Backpropagation* dengan *Firefly Algorithm* dan *Simulated Annealing*

8. Melakukan denormalisasi data.
9. Mengimplementasikan program yang telah dibuat pada masalah peramalan curah hujan.

4 Hasil dan Pembahasan

Untuk menyelesaikan masalah peramalan curah hujan di kota Surabaya menggunakan *Hybrid* jaringan saraf tiruan *Backpropagation* dengan *Firefly Algorithm* dan *Simulated Annealing* telah dibuat program untuk membantu proses perhitungan menggunakan bahasa pemrograman Java dengan bantuan *software* Netbeans IDE 8.0.2.

Proses awal pada program ini adalah pelatihan data. Pada proses pelatihan diawali dengan memasukkan nilai parameter-parameter yang dibutuhkan. Parameter tersebut antara lain adalah banyaknya *firefly*, batas error, maksimal iterasi, koefisien penyerapan udara, keatraktifan awal, koefisien random, suhu awal, suhu akhir, dan koefisien penurunan suhu. Nilai parameter yang digunakan pada pelatihan data ini yaitu maksimum iterasi = 10000, batas error = 0.001, nilai keatraktifan awal = 1, suhu awal = 10000, suhu akhir = 1, dan koefisien penurunan suhu = 0.1. Sedangkan banyaknya *firefly*, koefisien penyerapan udara, dan koefisien random merupakan parameter yang diubah-ubah. Setelah input parameter selesai, maka program akan menghasilkan bobot dan bias optimal dengan nilai MSE untuk masing-masing percobaan. Kemudian dipilih nilai MSE yang terkecil. Berikut hasil percobaan pelatihan dapat dilihat pada Tabel 2.

Berdasarkan Tabel 2, pada percobaan ke-18 diperoleh nilai MSE terkecil dari proses pelatihan adalah sebesar 0.0395384228 dengan dengan jumlah individu *firefly* yang digunakan adalah sebanyak 20, koefisien penyerapan udara (γ) = 1,5, nilai keatraktifan awal (β_0) = 1, dan nilai koefisien random α = 0,8. Nilai bobot dan bias terbaik dapat dilihat pada Tabel 3 dan Tabel 4. Nilai bobot dan bias terbaik yang dihasilkan dari proses pelatihan, selanjutnya digunakan pada uji validasi.

Setelah proses pelatihan dilakukan, langkah selanjutnya adalah proses validasi terhadap data yang belum pernah digunakan dalam proses pelatihan. Uji validasi ini dilakukan untuk melihat seberapa besar eror hasil peramalan curah hujan di kota Surabaya. Pada proses uji validasi dilakukan proses *feedforward* jaringan saraf tiruan menggunakan bobot dan bias terbaik yang diperoleh dari proses pelatihan. Setelah memperoleh nilai *output* atau hasil peramalan pada proses *feedforward*, selanjutnya dihitung hasil selisih eror yang dihasilkan. Hasil uji validasi dapat dilihat pada Tabel 5.

Tabel 2 Hasil percobaan pelatihan data

No.	<i>firefly</i>	Koefisien udara (γ)	Koefisien random (α)	MSE
1	10	0.5	0.1	0.0400038357
2			0.5	0.0419995020
3			0.8	0.0419963027
4		1	0.1	0.0408890845
5			0.5	0.0421927186
6			0.8	0.0426184953
7		1.5	0.1	0.0416505263
8			0.5	0.0427456853
9			0.8	0.0412489815
10	20	0.5	0.1	0.0407439307
11			0.5	0.0415554002
12			0.8	0.0436114638
13		1	0.1	0.0412850129
14			0.5	0.0423823425
15			0.8	0.0406383275
16		1.5	0.1	0.0417282268
17			0.5	0.0425468934
18			0.8	0.0395384228
19	50	0.5	0.1	0.0406854723
20			0.5	0.0422393247
21			0.8	0.0429385087
22		1	0.1	0.0413094229
23			0.5	0.0427697167
24			0.8	0.0419941870
25		1.5	0.1	0.0420737806
26			0.5	0.0416733926
27			0.8	0.0406338550

Tabel 3 Bobot dan bias terbaik dari *input* ke *hidden*

	h1	h2	h3
x1	-0.5929	-0.8284	-0.3976
x2	-1.1577	-0.5770	-0.9172
x3	-0.7693	-1.1497	-0.7742
b1	-1.1722	-0.7547	-0.4571

Tabel 4 Bobot dan bias terbaik dari *hidden* ke *output*

	o1
h1	-0.9679
h2	-0.9298
h3	-0.5955
b2	-0.3268

Tabel 5 Hasil uji validasi

No	Data Aktual	Hasil Peramalan	Selisih Error
1	6.92222	6.27962	0.6426
2	14.52273	6.00431	8.51842
3	4.97073	6.08599	1.11526
4	2.81739	5.93822	3.12083
5	2.36154	5.58203	3.22049
6	4.19773	4.95321	0.75548
7	7.90238	4.94887	2.95351
8	7.33333	5.32369	2.00964
9	14.56667	5.60126	8.96541
10	17.83462	6.12963	11.70499
11	12.05122	6.5675	5.48372
12	9.90851	6.6472	3.26131
13	5.53617	6.43234	0.89617

Berdasarkan hasil uji validasi tersebut, diperoleh selisih error rata-rata sebesar 3.75382. Setelah proses uji validasi selesai, kemudian dilanjutkan ke proses peramalan. Proses peramalan ini dilakukan untuk meramalkan curah hujan di kota Surabaya pada bulan berikutnya. Pada proses peramalan dilakukan proses *feedforward* jaringan saraf tiruan menggunakan bobot dan bias terbaik yang diperoleh dari proses pelatihan. Data yang

digunakan adalah data 3 bulan terakhir yaitu bulan Oktober hingga Desember 2017. Data yang digunakan untuk peramalan dapat dilihat pada Tabel 6.

Hasil peramalan untuk 3 bulan berikutnya yaitu bulan Januari hingga Maret 2018 dapat dilihat pada Tabel 7.

Tabel 6 Data untuk peramalan

	Rata-rata curah hujan (mm)
Okt-17	1.4929
Nov-17	11.6491
Des-17	9.7897

Tabel 7 Hasil peramalan

	x1	x2	x3	Hasil Peramalan
Jan-18	1.4929	11.6491	9.7897	6.1451
Feb-18	11.6491	9.7897	6.1451	8.5459
Mar-18	9.7897	6.1451	8.5459	7.7391

5 Kesimpulan

Berdasarkan implementasi yang dilakukan pada data curah hujan di kota Surabaya diperoleh hasil MSE terbaik pelatihan sebesar 0.0395384228. Nilai MSE cukup kecil sehingga bobot optimal yang dihasilkan mampu mengenali pola data dengan baik. Pada uji validasi diperoleh rata-rata selisih eror sebesar 3.75382. Sedangkan hasil peramalan untuk 3 bulan berikutnya yaitu bulan Januari hingga Maret 2018 berturut-turut adalah 6.1451, 8.5459, 7.7391.

6 Daftar Pustaka

- [1] Katarina, Tumiar M., 2014, *Klimatologi Dasar : Unsur Iklim dan Proses Pembentukan Iklim*, Penerbit Graha Ilmu, Yogyakarta.
- [2] Kodoatie, Robert J., 2013, *Rekayasa dan Manajemen Banjir Kota*. Penerbit Andi, Yogyakarta.
- [3] Fausett, L., 2003, *Fundamental of Neural Network Architecture, Algorithm, and Application*, Printice-Hall, Inc, London.
- [4] Roosmalita, Nadia S., Wayan Firdaus M., Aji Prasetya W., 2016, *Backpropagation on Neural Network Method for Inflation Rate Forecasting in Indonesia*, International Journal Advance Soft Computer Applications, Vol. 8 No. 3.

- [5] Frolic A., I.M., & J, Bhuvana. 2014, May. *Backpropagation Based Firefly Algorithm for Character Recognition*. Chennai, India.
- [6] Yang, X. S., 2010, *Engineering Optimization: An Introduction with Metaheuristic Applications*, Wiley & Sons, Inc, New Jersey, USA.
- [7] Krikpatrick, S., Gelatt, C. D. and Vecchi, M. P., 1983, *Optimization by Simulated Annealing*, Science, 220,671-680.
- [8] Mohammad S.M., Elham S.M., Pengcheng Jiao, 2017, *Next generation prediction model for daily solar radiation on horizontal surface using a hybrid neural network and simulated annealing method*, Energy Conversion and Management 153 (2017) 671–682.
- [9] Nandy, Suharshan, Partha P.S. dan Achintya D., 2012, *Analysis of a Nature Inspired Firefly Algorithm based Back-propagation Neural Network Training*, International Journal of Computer Applications (0975- 8887), Vol 43.
- [10] Rachmawati, Kusratmoko E, Damayanti A., 2004, *Peristiwa Banjir Tahun 1996 dan 2002 di Daerah Aliran Ciliwung*. Makalah Semiloka Pengelolaan Tata Air & Sampah Jakarta. 14 Agustus 2004.