

# PREDIKSI DATA DERET WAKTU MULTIVARIAT MENGGUNAKAN *ECHO STATE NETWORK* DAN *HARMONY SEARCH*

Muhammad Muharrom Al Haromainy<sup>1</sup>), Chastine Fatichah<sup>2</sup>), dan Ahmad Saikhu<sup>3</sup>)

<sup>1, 2, 3</sup>)Departemen Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember

Sukolilo, Surabaya

e-mail: muharrom.18051@mhs.its.ac.id<sup>1</sup>), chf@cs.its.ac.id<sup>2</sup>), saikhu@if.its.ac.id<sup>3</sup>)

## ABSTRAK

*Prediksi data deret waktu multivariat banyak diterapkan di berbagai bidang, seperti bidang industri, kesehatan, dan ekonomi. Beberapa metode dapat digunakan untuk membentuk model prediksi, seperti Artificial Neural Network (ANN) dan Recurrent Neural Network (RNN). Namun, metode tersebut mempunyai nilai error lebih besar daripada metode pengembangan dari RNN, yaitu Echo State Network (ESN). Metode ESN mempunyai beberapa parameter global, seperti jumlah reservoir dan leaking rate. Penentuan nilai parameter sangat berpengaruh terhadap performa model prediksi yang dihasilkan. Metode optimasi Harmony Search (HS) diusulkan untuk memberikan solusi penentuan parameter metode ESN. Metode HS dipilih karena lebih mudah diterapkan dan berdasarkan penelitian lain metode HS mendapat nilai optimum lebih baik daripada metode meta-heuristik yang lain. Metode yang dibandingkan dalam penelitian ini adalah RNN, ESN, dan ESN-HS. Root Mean Square Error (RMSE) dan Mean Absolute Percent Error (MAPE) digunakan untuk mengukur tingkat kesalahan hasil peramalan. ESN mendapatkan nilai error lebih kecil daripada RNN, dan ESN-HS menghasilkan nilai error paling kecil di antara uji coba yang lain, yaitu  $0,782e-5$  untuk RMSE dan  $0,28\%$  untuk MAPE. Metode optimasi HS berhasil mendapatkan kombinasi parameter global yang sesuai untuk model prediksi ESN.*

**Kata Kunci:** Data deret waktu multivariat, echo state network, harmony search, optimasi, model prediksi.

# PREDICTION OF MULTIVARIATE TIME SERIES DATA USING *ECHO STATE NETWORK* AND *HARMONY SEARCH*

Muhammad Muharrom Al Haromainy<sup>1</sup>), Chastine Fatichah<sup>2</sup>), dan Ahmad Saikhu<sup>3</sup>)

<sup>1, 2, 3</sup>) Department of Informatics, Faculty of Intelligent Electrical and Information Technology, Sepuluh Nopember Institute of Technology

Sukolilo, Surabaya

e-mail: muharrom.18051@mhs.its.ac.id<sup>1</sup>), chf@cs.its.ac.id<sup>2</sup>), saikhu@if.its.ac.id<sup>3</sup>)

## ABSTRACT

*Multivariate time series data prediction is widely applied in various fields such as industry, health, and economics. Several methods can form prediction models, such as Artificial Neural Network (ANN) and Recurrent Neural Network (RNN). However, this method has an error value more significant than the development method of RNN, namely the Echo State Network (ESN). The ESN method has several global parameters, such as the number of reservoirs and the leaking rate. The determination of parameter values dramatically affects the performance of the resulting prediction model. The Harmony Search (HS) optimization method is proposed to provide a solution for determining the parameters of the ESN method. The HS method was chosen because it is easier to implement, and based on other research, the HS method gets the optimum value better than other meta-heuristic methods. The methods compared in this study are RNN, ESN, and ESN-HS. Root Mean Square Error (RMSE) and Mean Absolute Percent Error (MAPE) are used to measure the error rate of forecasting results. ESN got a smaller error value than RNN, and ESN-HS produced a minor error value among the other trials, namely  $0.782e-5$  for RMSE and  $0.28\%$  for MAPE. The HS optimization method has successfully obtained the appropriate global parameters for the ESN prediction model.*

**Keywords:** Multivariate time series data, echo state network, harmony search, optimization, prediction models.

## I. PENDAHULUAN

**P**REDIKSI data deret waktu multivariat merupakan salah satu bidang penelitian yang penting. Prediksi data deret waktu banyak diterapkan di berbagai macam bidang, seperti bidang industri, kesehatan, dan ekonomi. Selama bertahun-tahun, para peneliti mengusulkan banyak teknik untuk memodelkan dan memprediksi data deret waktu [1], [2]. Penelitian [3] tentang prediksi penyimpangan trek yang sangat penting untuk pemeliharaan

dan manajemen kereta api. Kemudian [4] melakukan prediksi kecepatan angin karena berpengaruh tinggi terhadap tenaga yang dihasilkan dan didasari peningkatan permintaan akan energi terbarukan dari hari ke hari.

Beberapa metode yang digunakan untuk melakukan prediksi data deret waktu, seperti metode Artificial Neural Network (ANN) untuk mengetahui keterlambatan troposfer basah guna untuk memprediksi cuaca secara realtime [5]. Penelitian lain, metode ANN digunakan untuk memprediksi kecepatan angin [6], memprediksi data eksresi gen [7], [8], dan masih banyak yang lainnya.

Metode *Recurrent Neural Network* (RNN) menghasilkan nilai akurasi lebih baik dibandingkan ANN [9] karena memiliki sistematisa perhitungan bobot secara berulang. RNN dapat menyimpan memori (*feedback loop*) dengan melakukan perulangan dalam arsitekturnya kemudian menggunakannya untuk prediksi sehingga tidak membuang informasi dari masa lalu. RNN dapat secara otomatis mempelajari semantik tingkat tinggi [10]. Penelitian [11] menggunakan RNN untuk prediksi lintasan kendaraan menggunakan data lalu lintas perkotaan. Rusia juga menggunakan penelitian hasil ilmiah untuk membantu mengurangi biaya dan meningkatkan efisiensi energi menggunakan jaringan saraf berulang [12] karena membantu kesalahan prediksi yang menyebabkan biaya lebih mahal.

Model prediksi yang paling banyak digunakan adalah jaringan saraf berulang [13]. Namun, salah satu kelemahan RNN adalah kesulitan dalam mengadopsi bobot [14]. Berbagai algoritma telah digunakan untuk melatih RNN, namun algoritma ini tetap mengalami kompleksitas dan komputasi tinggi, proses pelatihan lambat, dan potensi ketidakstabilan [15]. Metode pengembangan dari RNN adalah *Echo State Network* (ESN) model prediksi deret waktu nonlinier multivariat yang kuat dan adaptif [16]. Penelitian [17] menerapkan ESN untuk memprediksi konsumen energi gedung kantor dan mendapatkan hasil yang baik. Konsep utama algoritma ini adalah melatih bobot keluaran sendiri dengan metode regresi linier sederhana. Bobot yang dilatih hanya pada lapisan ketiga [18]. Berbeda dengan RNN, di mana semua bobot (tidak hanya bobot pada lapisan output) harus dilatih [19].

Terlepas dari keunggulan yang disebutkan sebelumnya, pengaturan parameter ESN, seperti inisialisasi parameter jumlah reservoir masih dilakukan dengan percobaan beberapa kali hingga mendapatkan nilai parameter yang sesuai [20]. Di mana, reservoir adalah lapisan kedua dari ESN yang digunakan untuk ekspansi data input. Kemudian, salah satu parameter yang utama menurut [21] adalah *leaking rate*. Sehingga dua parameter tersebut perlu dioptimasi untuk mendapatkan nilai yang optimal. Penentuan nilai parameter pada metode ESN sangat berpengaruh terhadap model yang dihasilkan, sehingga memengaruhi hasil prediksi. Maka dari itu, diperlukan algoritma optimasi. Pada penelitian ini menggunakan algoritma *Harmony Search* (HS) untuk optimasi, karena dibandingkan algoritma optimasi lain, HS lebih mudah untuk diaplikasikan dan hanya membutuhkan beberapa parameter untuk ditentukan sebelum menjalankan algoritma HS. Algoritma lain seperti *Genetic Algorithm* (GA) juga dapat digunakan, namun pada penelitian [22] HS mendapatkan hasil optimasi yang lebih baik. Pada penelitian [23] juga menggunakan ESN dan metode optimasi, namun berbeda studi kasus dan tidak dijelaskan apa saja yang dioptimasi. Penelitian ini dilakukan optimasi terhadap dua parameter, yaitu jumlah reservoir dan nilai *leaking rate*. Jika semua parameter dioptimasi, waktu komputasi yang dibutuhkan semakin lama.

Penelitian ini mengusulkan metode untuk prediksi data multivariat time series menggunakan Echo State Network dan optimasi parameter menggunakan algoritma *Harmony Search* sehingga diharapkan mendapatkan parameter yang optimal dan menghasilkan prediksi lebih baik.

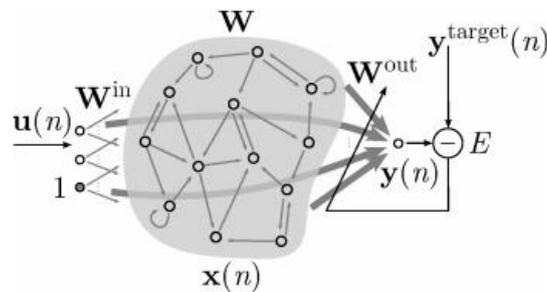
## II. KAJIAN PUSTAKA

Data deret waktu ada yang berjenis *univariate* dan *multivariate*. Data deret waktu *univariate* biasanya hanya memiliki satu atribut yang digunakan untuk proses prediksi, misalnya data suhu udara setiap jam. Sedangkan *multivariate* memiliki lebih dari satu atribut. Setiap atribut tidak hanya bergantung pada data masa lalu, tetapi juga memiliki keterkaitan pada atribut lain. Pengujian korelasi antar atribut dapat dilakukan dengan beberapa metode, salah satunya *Pearson correlation test*. Berdasarkan metode tersebut, jika diperoleh *p-value* kurang dari *alpha* sebesar 0,05, berarti mempunyai korelasi dan dapat digunakan atribut tersebut.

Penelitian ini menggunakan metode ESN, metode ESN berdasar pada metode *reservoir computing*. Komputasi reservoir mempunyai tiga lapisan. Pertama adalah lapisan *input*, berupa vektor memiliki panjang sesuai dengan atribut data yang digunakan. Lapisan ini terdiri dari beberapa neuron yang saling terhubung secara acak. Lapisan pertama dan lapisan kedua dihubungkan oleh vektor bobot yang terbentuk secara acak sebagai faktor pengali nilai *input* menuju lapisan *reservoir*. Neuron pada *reservoir* juga terhubung dengan neuron lain yang ada di dalam *reservoir* dan terhubung secara acak. Hubungan antarneuron berlangsung secara berulang menggunakan konsep RNN membuat reservoir memiliki memori untuk menyimpan *input* secara sekuen [24].

Lapisan ketiga adalah *readout*, terbentuk vektor dengan ukuran yang sesuai dengan jumlah *output* yang ditentukan. Lapisan kedua dan ketiga terhubung oleh nilai bobot, namun yang membedakan dengan bobot input, lapisan ini mengalami proses pembelajaran yang dilakukan dengan perhitungan mundur seperti *backpropagation*. Perbaikan bobot hanya terjadi pada lapisan *readout*. Beberapa metode menggunakan pengembangan dari

*reservoir computing* adalah *Echo State Network* (ESN) [24].



Gambar 1. Echo state network model [24].

*Echo State Network* (ESN) menyediakan arsitektur dan prinsip pembelajaran yang didasari oleh Recurrent Neural Network (RNN). Gagasan utama adalah untuk menggerakkan jaringan saraf berulang secara acak, besar, memperbaiki RNN dengan sinyal input sehingga menginduksi di setiap neuron dalam jaringan “reservoir”. RNN adalah *Machine Learning* (ML) yang paling mirip dengan otak biologis, substrat kecerdasan alami [21].

Alternatif penggunaan RNN telah diusulkan dengan *Echo State Network* dalam ML dan *Liquid State Machine* (LSM) dalam ilmu *neuroscience*. Tren yang dimulai oleh ESN dan LSM dikenal dengan *Reservoir Computing* (RC) [25]. RC saat ini merupakan area penelitian yang produktif. Pada *benchmark* yang dirancang untuk menantang akuisisi memori jangka pendek, ESN masih mengungguli RNN [26]. Kesederhanaan yang tampak dari ESN memerlukan beberapa pengalaman. Ada sejumlah hal yang bisa jadi salah. Secara khusus, generasi awal jaringan *reservoir* dipengaruhi oleh beberapa parameter *global*, dan ini harus ditetapkan secara tepat. Model dasar ESN ditunjukkan pada Gambar 1.

Metode asli *Reservoir Computing* yang diperkenalkan dengan ESN sebagai berikut:

1. Menghasilkan *reservoir* RNN yang besar dan acak ( $W^{in}, W, \alpha$ ).
2. Menjalankan dengan *input* pelatihan  $u(n)$  dan mengumpulkan status aktivasi *reservoir* yang sesuai  $x(n)$
3. Menghitung bobot *readout* linier dari  $W^{out}$  dari *reservoir* menggunakan regresi linier, meminimalkan MSE antara  $y(n)$  dan  $y^{target}(n)$ .
4. Menggunakan jaringan terlatih pada data *input* baru  $u(n)$  menghitung  $y(n)$  dengan menggunakan hasil *output* bobot  $W^{out}$  yang terlatih.

Dalam praktiknya, penting untuk diingat bahwa *reservoir* bertindak sebagai ekspansi non linier dan sebagai memori *input*  $u(n)$  secara bersamaan. Ada hubungan antara metode RC dan kernel dalam ML. *Reservoir* dapat dilihat sebagai ekspansi non linier berdimensi tinggi  $x(n)$  dari input sinyal  $u(n)$ . Pada saat yang sama, *reservoir* berfungsi sebagai memori, menyediakan konteks temporal. Lapisan kedua dari ESN disebut *reservoir*. ESN menggunakan arsitektur atau *framework* dari *reservoir computing*.

*Harmony Search* (HS) adalah algoritma pencarian *meta-heuristik* yang mencoba meniru proses improvisasi musisi dalam menemukan harmoni yang sesuai [27]. HS mudah diterapkan, menyatu dengan cepat untuk menemukan solusi optimal dan cukup baik dalam waktu komputasi yang tidak terlalu lama. HS menunjukkan kinerja yang menjanjikan dalam masalah optimisasi. Algoritma *meta-heuristik* bisa menjadi alternatif yang efisien untuk menyelesaikan masalah optimisasi yang kompleks. Sebuah harmoni didefinisikan oleh hubungan khusus antara beberapa gelombang suara yang mempunyai frekuensi berbeda. Kualitas improvisasi harmoni ditentukan oleh estimasi estetika. Untuk meningkatkan estimasi estetika dan menemukan harmoni terbaik, para musisi melakukan latihan demi latihan.

Ada kesamaan antara improvisasi musisi dan optimisasi. Optimisasi tujuannya adalah untuk menemukan global optimum dari fungsi tujuan yang sedang dipertimbangkan dengan menyetel sejumlah variabel keputusan yang telah ditentukan. Dalam masalah optimisasi variabel keputusan membuat vektor solusi. Kemudian, nilai-nilai variabel keputusan dimasukkan ke dalam fungsi objektif dan kualitas vektor solusi dihitung. Vektor solusi diperbarui selama iterasi terjadi hingga global optimum diperoleh.

Perbandingan proses improvisasi musik dan optimisasi terdapat beberapa kesamaan. Pertama, dalam proses musik, kualitas harmoni ditentukan oleh estimasi estetika. Dalam proses optimisasi kualitas vektor solusi ditentukan nilai fungsi objektif. Kedua, dalam proses musik, tujuan utamanya adalah untuk mendapatkan harmoni terbaik. Dalam proses optimisasi tujuan utamanya adalah untuk mendapatkan *global optimum*. Ketiga, dalam proses musik, musisi mengubah nada instrumen. Algoritma optimisasi mengubah nilai-nilai variabel keputusan. Keempat, dalam proses musik, setiap upaya untuk memainkan harmoni disebut latihan. Dalam optimisasi, setiap upaya untuk memperbarui vektor solusi disebut iterasi.

Optimasi merupakan usaha untuk mendapatkan solusi terbaik di antara banyak solusi. Algoritma HS, solusi yang layak disebut harmoni dan setiap variabel keputusan solusinya adalah sesuai dengan yang ditetapkan. HS termasuk *memory harmony* (HM) di mana sejumlah harmoni ( $N$ ) yang telah ditentukan untuk disimpan. Misalkan tujuannya adalah untuk meminimalkan/memaksimalkan fungsi *fitness* ( $f$ ). Masalah optimasi didefinisikan sebagai berikut:

$$\Delta f(x^*) = 0 \tag{1}$$

Di mana  $f$  adalah fungsi *fitness*,  $x^*$  adalah calon penyelesaian atau titik optimal. Bila  $\Delta f(x^*) = 0$  dan  $H(x^*)$  definit positif, maka  $x^*$  titik maksimum. Bila  $\Delta f(x^*) = 0$  dan  $H(x^*)$  definit negatif, maka  $x^*$  titik minimum.

Evaluasi yang digunakan dalam penelitian ini menggunakan kesalahan nilai prediksi  $y^{target}$  terhadap  $y$ . Memungkinkan ada residual yang didapatkan dari perbedaan  $y - y^{target}$  dan kesalahan estimasi atau prediksi. Proses prediksi menggunakan ESN diterapkan benar sehingga kesalahan prediksi dapat diminimalkan. Untuk mengetahui kinerja ESN dalam memprediksi, digunakan rumus *Root Mean Square Error* (RMSE) dan *Mean Absolute Percentage Error* (MAPE).

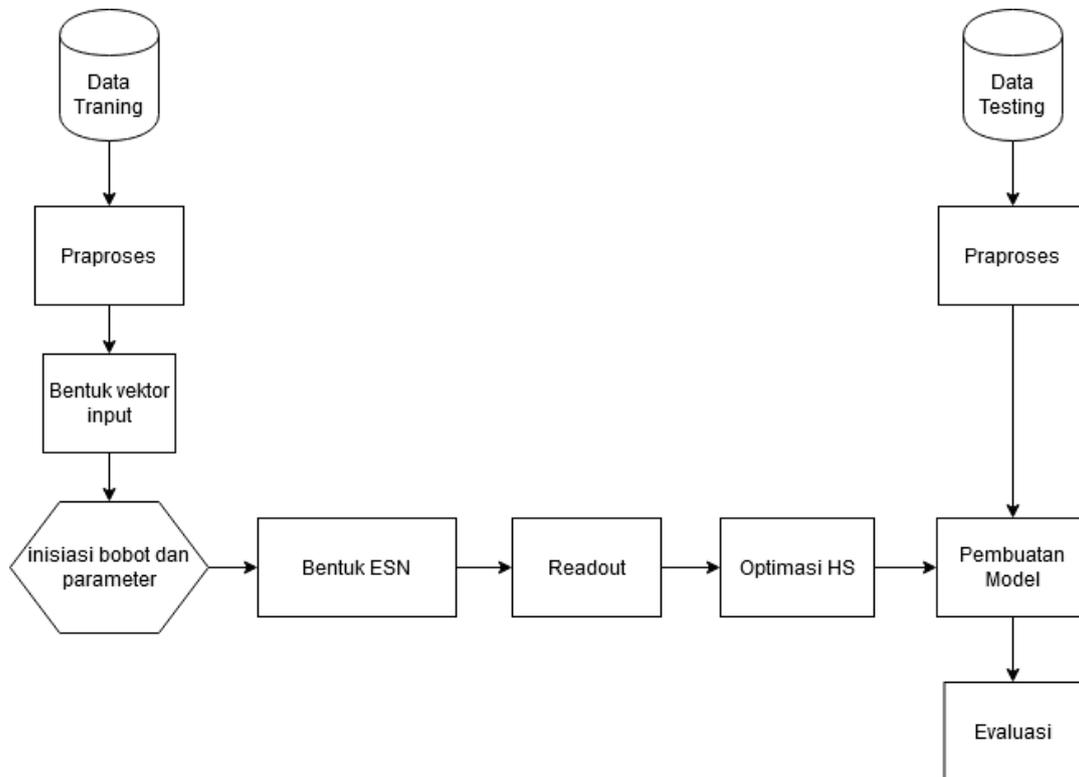
$$RMSE = E(y, y^{target}) = \frac{1}{N_y} \sum_{i=1}^{N_y} \sqrt{\frac{1}{T} \sum_{n=1}^T (y_i(n) - y_i^{target}(n))^2} \tag{2}$$

$$MAPE = E(y, y^{target}) = \frac{\sum_{i=1}^n \left| \left( \frac{y_i - y_i^{target}}{y_i} \right) 100 \right|}{n} \tag{3}$$

### III. METODE YANG DIAJUKAN

Tahap pertama yang dilakukan pada penelitian ini dengan mengumpulkan dan memahami berbagai publikasi hasil penelitian terbaru yang berkaitan dengan model-model prediksi, terutama metode ESN dan algoritma optimasi. Solusi permasalahan dan landasan masalah didapatkan dari berbagai sumber referensi yang digunakan.

Gambar 2 menjelaskan bahwa tahap pertama melakukan persiapan data yang digunakan sebagai training dan testing. Data awal dilakukan praproses terlebih dahulu. Proses berikutnya adalah membentuk vektor input data. Setelah didapatkan nilai input, dilanjutkan membuat nilai acak terhadap bobot yang menghubungkan antara lapisan input dan reservoir. Hubungan membentuk ukuran yang sesuai antara jumlah input dan jumlah neuron dalam reservoir.



Gambar 2. Diagram alir implementasi model prediksi ESN.

TABEL 1  
Data *Google Stock Price*.

Date	Open	High	Low	Close	Adj Close	Volume
2005-06-15	136,986	138,132	133,215	136,886	136,886	419226
2005-06-16	136,617	138,630	136,025	138,201	138,201	250181
2005-06-17	138,979	139,626	137,434	139,626	139,626	209469
2005-06-18	137,529	143,297	135,357	142,814	142,814	422069
2005-06-19	143,497	144,607	141,952	143,382	143,382	303779

A. *Pengambilan Data*

Data yang digunakan pada penelitian ini berupa data deret waktu multivariat diambil dari situs Yahoo Finance misalnya data *Google Stock Price*. Data tersebut mempunyai beberapa atribut dan mempunyai periode harian seperti ditunjukkan pada Tabel 1. Sebelum data digunakan untuk prediksi, ditentukan terlebih dahulu atribut prediktor (X) dan atribut yang diprediksi (Y). Pada penelitian ini yang diprediksi adalah atribut *Open*.

Setelah data dibagi menjadi 2 bagian (prediktor dan yang diprediksi), prediktor dicek terlebih dahulu apakah ada korelasi atau hubungan dengan atribut *Open*. Pada penelitian ini menggunakan *pearson's correlation test*. Hipotesis untuk uji korelasi pearson ditunjukkan pada persamaan 4. Hipotesis diterima jika  $p\text{-value} = 0,0 < \alpha = 0,05$ . Dengan tingkat kepercayaan di atas 95%, atribut yang menjadi prediktor mempunyai korelasi.

$$H_0 : \rho = 0; H_1 : \rho \neq 0; \tag{4}$$

B. *Praproses*

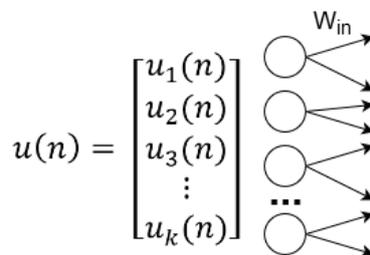
Dataset yang digunakan terdiri dari beberapa atribut dan mempunyai satuan atau selisih angka yang jauh. Mulai satuan hingga ratusan, sehingga dapat menurunkan atau memperburuk performa model prediksi. Maka dari itu, perlu dilakukan normalisasi data seperti ditunjukkan pada Tabel 2. Normalisasi yang digunakan adalah *MinMax*.

C. *Membuat Vektor Input*

Setelah data melewati tahap normalisasi, setiap baris data akan dijadikan input berupa vektor  $u(n)$ . Vektor input  $u(n)$  merupakan elemen dari  $R^{Nu}$ . Vektor input  $u(n)$  selanjutnya dihubungkan dengan reservoir menggunakan bobot  $W_{in}$ . Bobot  $W_{in}$  menghubungkan setiap elemen  $u(n)$  tepat satu neuron dari reservoir seperti ditunjukkan pada Gambar 3.

TABEL 2  
Data Setelah Normalisasi.

Open	High	Low	Close	Adj Close	Volume
0,450	0,553	0,123	0,723	0,723	1,000
0,420	0,653	0,113	0,613	0,613	0,991
0,423	0,153	0,213	0,443	0,443	0,899
0,412	0,353	0,103	0,411	0,411	0,900
0,881	0,189	0,001	0,900	0,900	0,923



Gambar 3. Vektor input  $u(n)$ .

$$W_{in} = \begin{bmatrix} W_{1,1} & W_{1,2} & W_{1,3} & \dots & W_{1,500} \\ W_{2,1} & W_{2,2} & W_{2,3} & \dots & W_{2,500} \\ W_{3,1} & W_{3,2} & W_{3,3} & \dots & W_{3,500} \\ W_{4,1} & W_{4,2} & W_{4,3} & \dots & W_{4,500} \end{bmatrix}$$

Gambar 4. Bobot input  $W_{in}$ .

#### D. Inisiasi Bobot dan Parameter Global

Setelah membentuk vektor input  $u(n)$ , vektor tersebut dihubungkan dengan bobot  $W_{in}$  yang dibentuk dengan acak. Ukuran matriks  $W_{in}$  sama seperti dua matriks yang dihubungkan. Vektor  $u(n)$  memiliki ukuran  $N_u$ , sedangkan vektor reservoir berukuran  $N_x$ . Maka dari itu diperoleh ukuran matriks  $W_{in}$  adalah  $N_x \times N_u$  atau ditulis  $W_{in} \in \mathbb{R}^{N_x \times N_u}$ . Nilai awal parameter global mulai ukuran reservoir  $N$ , spectral radius dimulai dengan nilai awal 1, leaking rate, bobot input  $W_{in}$ , dan input scaling yang mengatur jarak atau selisih antar input bobot. Permisalan menggunakan empat atribut dan ukuran reservoir adalah 500, maka matriks  $W_{in}$  seperti Gambar 4.

#### E. Reservoir

Hasil proses pada bagian C dan D yaitu pemetaan dari vektor input  $u(n)$  dengan  $W_{in}$  menjadi vektor *reservoir*  $x(n)$ . Nilai vektor  $x(n)$  terpengaruh oleh setiap neuron yang ada, di mana setiap neuron dalam reservoir dihubungkan dengan bobot  $W_x$ . Karena setiap neuron terhubung satu sama lain sebanyak jumlah neuron atau  $N_x$ , maka  $W_x$  membentuk matrik bobot berukuran  $N_x \times N_x$  atau dapat dituliskan dengan notasi  $W_x \in \mathbb{R}^{N_x \times N_x}$ . Nilai setiap neuron didapatkan menggunakan persamaan 5 dan 6.

$$\tilde{x}(n) = \tanh(W^{in}[1; u(n)] + Wx(n - 1)) \quad (5)$$

$$x(n) = (1 - \alpha) \times (n - 1) + \alpha \tilde{x}(n) \quad (6)$$

#### F. Readout

Setelah menyelesaikan tahapan pada lapisan reservoir, dilanjutkan menghitung output pada lapisan readout. Nilai output  $y(t)$  adalah sebuah vektor yang berukuran  $N_y$ . Terdapat bobot  $W_{out}$  yang menghubungkan antara lapisan reservoir dan readout. Karena bobot  $W_{out}$  berada di antara lapisan reservoir dan readout, menghasilkan matriks bobot berukuran  $N_y \times N_x$  atau dapat dituliskan dengan  $W_{out} \in \mathbb{R}^{N_y \times N_x}$ , nilai lapisan readout didapatkan menggunakan persamaan 7. Kemudian,  $W_{fb}$  atau bobot yang menghubungkan antara readout untuk kembali ke reservoir memiliki ukuran yang sama seperti  $W_{out}$ , hanya saja dilakukan transpose.

$$y(n) = W^{out}[1; u(n); x(n)] \quad (7)$$

#### G. Optimasi Parameter ESN menggunakan Harmony Search

Optimasi parameter bertujuan untuk mendapatkan parameter terbaik untuk implementasi metode ESN. Terdapat dua parameter yang dioptimasi, yaitu jumlah reservoir dan nilai *leaking rate*. Solusi terbaik direpresentasikan jika mendapatkan nilai minimal fungsi *fitness*. Fungsi *fitness* yang digunakan adalah RMSE dan MAPE. Perancangan metode HS dilakukan dengan beberapa tahapan:

##### 1) Inisialisasi harmony memory

$N$  harmoni diinisiasi untuk menentukan berapa jumlah solusi yang disimpan di *harmony memory*. HM direpresentasikan seperti persamaan 8. Di mana  $x_{i,1}, x_{i,2}, \dots, x_{i,d}$  adalah kombinasi nilai random dari parameter yang akan dioptimasi, dan  $f$  adalah nilai *fitness* yang diperoleh.

$$HM = \begin{bmatrix} \text{Harmony 1} \\ \text{Harmony 2} \\ \vdots \\ \text{Harmony N} \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,d} \\ x_{2,1} & x_{2,2} & \dots & x_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N,1} & x_{N,2} & \dots & x_{N,d} \end{bmatrix} = \begin{bmatrix} f1 \\ f2 \\ \vdots \\ fN \end{bmatrix} \quad (8)$$

##### 2) Improvisasi harmoni baru dan Replacement

Improvisasi harmoni baru berarti membuat angka random untuk parameter yang dioptimasi, jika nilai *fitness* yang didapatkan lebih kecil, akan disimpan dan menggantikan nilai *fitness* terburuk di harmoni memori. Apabila nilai *fitness* lebih tinggi, maka tidak disimpan di harmoni memori. Setelah itu berulang untuk mengimprovisasi harmoni baru hingga kondisi terpenuhi.

##### 3) Stopping criterion

Kebanyakan cara berhenti algoritma optimasi adalah untuk mencapai jumlah iterasi yang telah ditentukan ( $t_{max}$ ). Jika kriteria ini belum terpenuhi, diulangi langkah kedua sampai kriteria untuk berhenti terpenuhi.

##### 4) Hasil akhir

Harmoni terbaik yang disimpan di harmoni memori dikembalikan sebagai solusi optimal.

#### H. Proses Learning

Proses terakhir setelah mendapatkan nilai dari lapisan readout, selanjutnya adalah proses pembelajaran pada matriks  $W_{out}$ . Hasil ini akan dimasukkan ke dalam proses algoritma harmony search untuk dihitung nilai *error*,

kondisi akan berhenti ketika mendapatkan hasil *error* yang ditargetkan atau paling kecil, dapat dilihat pada Gambar 5. Inisialisasi parameter global HS seperti Harmony Memory Size ditentukan sesuai jumlah solusi terbaik yang ingin disimpan, HMCR berkisar 0,7 sampai 0,99, PAR umumnya berkisar 0,1 sampai 0,5, dan *stop creation* sejumlah iterasi untuk melakukan improvisasi.

#### IV. HASIL DAN PEMBAHASAN

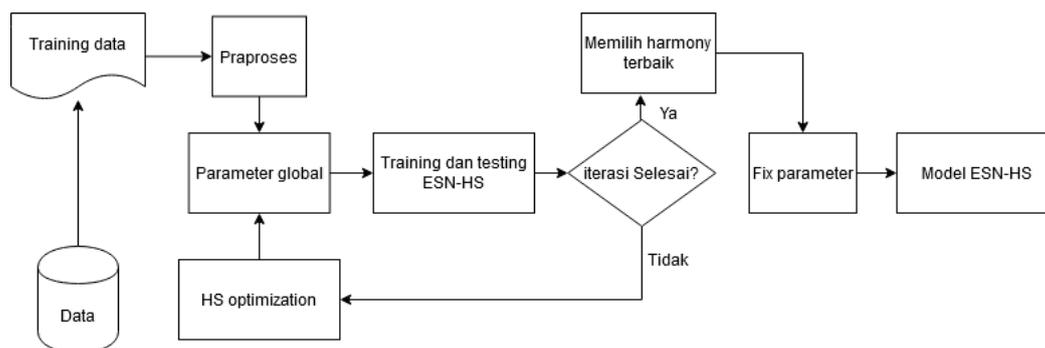
Uji coba yang dilakukan menggunakan data deret waktu multivariat, di mana sudah ditentukan bahwa atribut yang diprediksi adalah *Open*. Sedangkan atribut lain sebagai prediktor. Atribut *volume* tidak digunakan, karena setelah dilakukan *Perason's correlation test*, atribut tersebut tidak mempunyai korelasi dengan variabel *Open*. Hasil tes korelasi ditunjukkan pada Tabel 3. Setelah atribut dipilih, dilanjutkan proses normalisasi data yang bertujuan untuk homogenitas karena dapat memengaruhi hasil prediksi.

Pada penelitian ini melakukan skenario uji coba perbandingan metode prediksi, yaitu RNN, ESN, dan ESN yang dioptimasi menggunakan HS. Semua uji coba menggunakan data google stock price mulai tahun 2005 hingga tahun 2020. Hasil pengujian dapat dilihat pada Gambar 6, Gambar 7, dan Gambar 8.

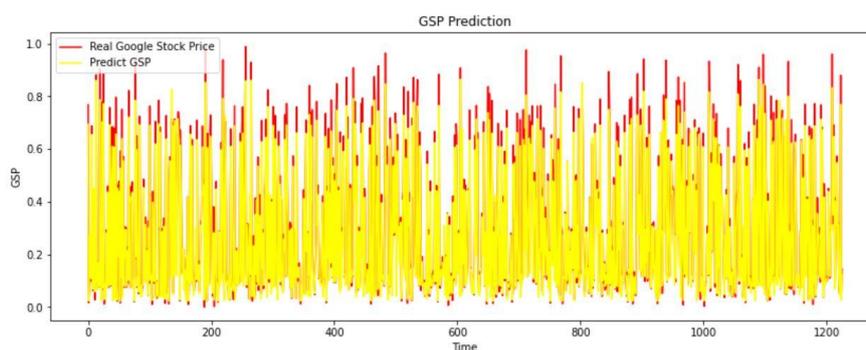
Uji coba dievaluasi menggunakan *Root Mean Square Error (RMSE)* dan *Mean Absolute Percent Error (MAPE)*. Uji coba pertama seperti ditunjukkan pada Gambar 6 mendapatkan hasil yang cukup baik. Namun masih terlihat perbandingan nilai kesalahan yang cukup lebar antara data asli dan data hasil prediksi. Sedangkan pada Gambar 7 mendapatkan hasil lebih baik dari pada uji coba pertama, terlihat pada Gambar 7 warna kuning lebih dominan daripada merah. Berarti model prediksi menggunakan ESN lebih baik daripada model RNN. Karena model ESN sangat dipengaruhi oleh nilai variabel, seperti jumlah reservoir dan nilai *leaking rate*. Pada uji coba ketiga, metode ESN dioptimasi menggunakan metode *harmony search* agar mendapatkan nilai yang sesuai untuk variabel di ESN. Sehingga seperti ditunjukkan pada Gambar 8, rentang kesalahan lebih kecil dibandingkan uji coba kedua. Hasil evaluasi ditunjukkan pada Tabel 4. Lama komputasi ketika implementasi metode HS tergantung pada jumlah iterasi dan harmoni memori yang ditetapkan. Algoritma HS diperlukan karena penentuan nilai parameter pada model prediksi ESN sangat berpengaruh.

TABEL 3  
Pearson's Correlation Test.

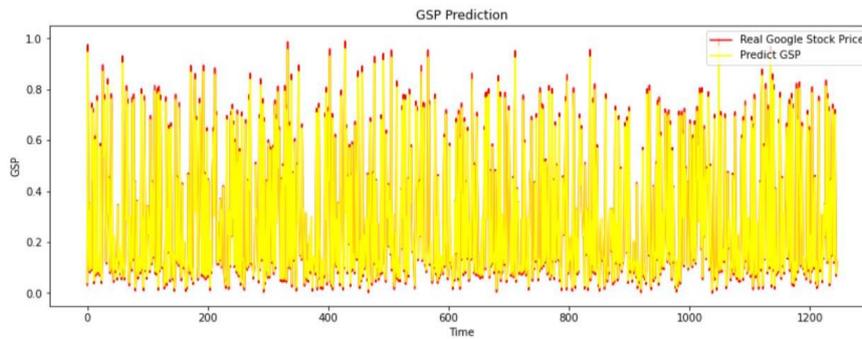
	High	Low	Close	Adj Close	Volume
Nilai Kepercayaan	99,98%	99,98%	99,97%	99,97%	-54,76%
p-value	0,0	0,0	0,0	0,0	> 0,05



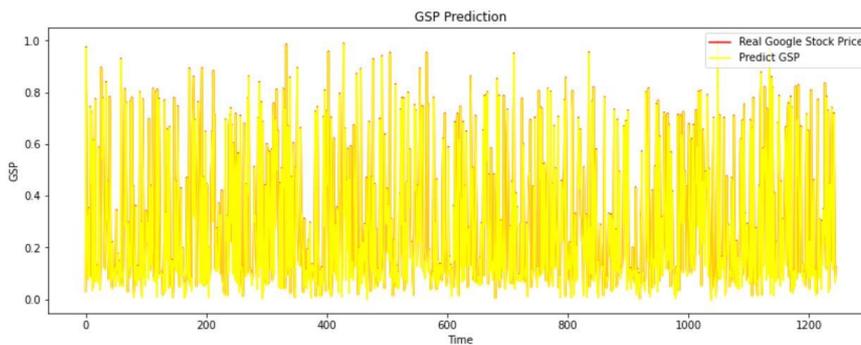
Gambar 5. Flow chart ESN-HS.



Gambar 6. Hasil uji coba metode RNN.



Gambar 7. Hasil uji coba metode ESN.



Gambar 8. Hasil uji coba ESN-HS.

TABEL 4  
Hasil evaluasi RMSE dan MAPE.

Uji Coba	RMSE	MAPE
RNN	0,1184	26,23%
ESN	0,0001	0,95%
ESN-HS	0,782e-5	0,28%

### V. KESIMPULAN

Ketika membuat model prediksi dan menggunakan data deret waktu berbentuk multivariat, dilakukan praproses seperti uji korelasi antar fitur. Karena, tidak semua fitur terdapat hubungan sehingga memengaruhi hasil prediksi. Beberapa uji coba model prediksi telah dilakukan dan mendapatkan hasil yang memuaskan. Metode ESN mendapatkan nilai *error* lebih kecil daripada RNN, dilanjutkan uji coba metode ESN dioptimasi menggunakan harmony search menghasilkan parameter input yang sesuai sehingga mendapatkan nilai eror lebih kecil. Metode ESN-HS mendapatkan nilai *error* 0,782e-5 (RMSE) dan 0,28% (MAPE). Pada penelitian selanjutnya, metode ESN dapat dimodifikasi pada bagian reservoir dan dioptimasi menggunakan algoritma lain. Kemudian, penentuan parameter yang harus dioptimasi dapat dikaji lebih lanjut, karena tidak harus semua parameter dari ESN dioptimasi.

### DAFTAR PUSTAKA

- [1] F. M. Bianchi, E. D. E. Santis, A. Rizzi, A. Sadeghian, and S. Member, "Short-Term Electric Load Forecasting Using Echo State Networks and PCA Decomposition," *IEEE Access*, vol. 3, pp. 1931–1943, 2015.
- [2] T. Mikolov, W.-T. Yih, and G. Zweig, "Linguistic Regularities in Continuous Space Word Representations," Association for Computational Linguistics, 2013.
- [3] Z. Ma, Y. Dong, C. Wang, and X. U. N. Shao, "Forecast of Non-Equal Interval Track Irregularity Based on Improved Grey Model and PSO-SVM," *IEEE Access*, vol. 6, pp. 34812–34818, 2018.
- [4] K. M. Begam and S. N. Deepa, "Optimized nonlinear neural network architectural models for multistep wind speed forecasting ☆," *Comput. Electr. Eng.*, vol. 78, pp. 32–49, 2019.
- [5] M. O. Selbesoglu, "Prediction of tropospheric wet delay by an artificial neural network model based on meteorological and GNSS data," *Eng. Sci. Technol. an Int. J.*, 2019.
- [6] R. K. B. Navas, S. Prakash, and T. Sasipraba, "Artificial Neural Network based computing model for wind speed prediction: A case study of Coimbatore, Tamil Nadu, India," *Physica A*, 2019.
- [7] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, p. e00938, Nov. 2018.
- [8] M. Daoud, M. Mayo, P. O. Box, and N. Zealand, "A survey of neural network-based cancer prediction models from microarray data," *Artif. Intell. Med.*, vol. 97, no. October 2017, pp. 204–214, 2019.
- [9] K. Ivandera and M. Mustikasari, "Implementasi Metode Recurrent Neural Network Pada Text the Implementation of Text Summarization with Abstractive," vol. 6, no. 4, 2019.

- [10] M. Usama, B. Ahmad, W. Xiao, M. S. Hossain, and G. Muhammad, "Self-attention based recurrent convolutional neural network for disease prediction using healthcare data," *Comput. Methods Programs Biomed.*, p. 105191, 2019.
- [11] S. Choi, J. Kim, and H. Yeo, "Attention-based Recurrent Recurrent Neural Network Network for Urban Vehicle Trajectory Prediction," *Procedia Comput. Sci.*, vol. 151, no. 2018, pp. 327–334, 2019.
- [12] S. A. Zagrebina, V. G. Mokhov, and V. I. Tsimbol, "Electrical Energy Consumption Prediction is based on the Recurrent Neural Network," *Procedia Comput. Sci.*, vol. 150, pp. 340–346, 2019.
- [13] X. Cai, N. Zhang, G. K. Venayagamoorthy, and D. C. Wunsch, "Time series prediction with recurrent neural networks trained by a hybrid PSO – EA algorithm," vol. 70, pp. 2342–2353, 2007.
- [14] M. Xu, M. Han, and H. Lin, "Wavelet-denoising multiple echo state networks for multivariate time series prediction," *Inf. Sci. (Ny)*, 2018.
- [15] J. Schmidhuber, "Deep Learning in Neural Networks : An Overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.
- [16] L. Shen, J. Chen, Z. Zeng, J. Yang, and J. Jin, "A novel echo state network for multivariate and nonlinear time series prediction," *Appl. Soft Comput. J.*, vol. 62, pp. 524–535, 2018.
- [17] G. Shi, D. Liu, and Q. Wei, "Energy consumption prediction of office buildings based on echo state networks," *Neurocomputing*, pp. 1–11, 2016.
- [18] S. Lokse, F. Maria, and R. Jenssen, "Training Echo State Networks with Regularization Through Dimensionality Reduction," *Cognit. Comput.*, no. 1, 2017.
- [19] U. D. Schiller and J. J. S. Ã, "Analyzing the weight dynamics of recurrent learning algorithms," vol. 63, pp. 5–23, 2005.
- [20] N. Chouikhi, B. Ammar, N. Rokbani, and A. M. Alimi, "PSO-based analysis of Echo State Network parameters for time series forecasting," *Appl. Soft Comput. J.*, vol. 55, pp. 211–225, 2017.
- [21] M. Lukoševičius, "A Practical Guide to Applying Echo State Networks," pp. 659–686, 2012.
- [22] Y. Kim, Y. Yoon, and Z. Woo, "A comparison study of harmony search and genetic algorithm for the max-cut problem," *Swarm Evol. Comput.*, vol. 44, no. August 2017, pp. 130–135, 2019.
- [23] J. Saadat, P. Moallem, and H. Koofgar, "Training Echo Estate Neural Network Using Harmony Search Algorithm," vol. 15, no. 1, pp. 163–179, 2017.
- [24] A. Klibisz, "A Primer on Reservoir Computing," pp. 1–14, 2016.
- [25] D. Verstraeten, B. Schrauwen, M. D. Haene, and D. Stroobandt, "An experimental unification of reservoir computing methods," vol. 20, pp. 391–403, 2007.
- [26] H. Jaeger, "Long Short-Term Memory in Echo State Networks : Details of a Simulation Study Long Short-Term Memory in Echo State Networks : Details of a Simulation Study," no. 27, 2012.
- [27] A. Askarzadeh and E. Rashedi, *Harmony Search Algorithm*, no. April. 2017.