

# IDENTIFIKASI SISTEM NONLINIER DENGAN MENGGUNAKAN RECURRENT NEURAL NETWORK DAN ALGORITMA DEAD-ZONE KALMAN FILTER

Rully Soelaiman<sup>1</sup>, Rangga Rifa'i<sup>1</sup>

<sup>1</sup>Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Kampus Keputih, Sukolilo, Surabaya 60111, Indonesia  
Email : rully@is.its.ac.id, rank2@cs.its.ac.id

## Abstrak

Identifikasi sistem nonlinier telah terbukti dapat diselesaikan menggunakan jaringan saraf tiruan dengan berbagai algoritma pembelajaran bobot. Algoritma pembelajaran normal seperti backpropagation (BP), memiliki beberapa kelemahan antara lain proses konvergensi yang masih lambat, beban komputasional yang meningkat seiring bertambahnya iterasi dan proses pembelajaran sensitif terhadap noise. Oleh karena itu algoritma pembelajaran extended Kalman filter (EKF) yang memiliki konvergensi cepat diusulkan walaupun perhitungannya lebih rumit dan masih sensitif terhadap noise.

Pada makalah ini, algoritma (EKF) diimplementasikan pada state-space recurrent neural network (RNN) untuk mengidentifikasi sistem nonlinier. Algoritma ini digunakan untuk mengupdate bobot pada lapisan tersembunyi dan lapisan keluaran dari jaringan saraf tiruan berdasarkan identifikasi error antara target dengan hasil estimasi state. Algoritma EKF dimodifikasi dengan algoritma dead-zone untuk memperbaiki kekuatan Kalman filter. Teori kestabilan Lyapunov digunakan untuk membuktikan bahwa Algoritma Kalman filter dengan dead-zone stabil.

Untuk membuktikan hasil identifikasi yang baik dengan dead-zone Kalman filter (DKF) maka digunakan algoritma BP sebagai pembanding. Dari hasil uji coba dan evaluasi, dapat disimpulkan bahwa hasil identifikasi dengan DKF lebih akurat dan efisien serta lebih cepat konvergensinya dibandingkan dengan algoritma pembanding.

Kata kunci : sistem nonlinier, identifikasi sistem, recurrent neural network, dead-zone Kalman filter.

## 1. Pendahuluan

Seiring perkembangan zaman peran komputer semakin mendominasi kehidupan. Saat ini, suatu

sistem sudah dapat melakukan pembelajaran untuk mengenali perilaku lingkungannya. Salah satunya adalah identifikasi sistem nonlinier seperti permasalahan mekanika dan dinamika dengan algoritma pembelajaran tertentu. Identifikasi sistem adalah mengatur suatu sistem agar keluaran dari sistem mampu mengikuti targetnya. Identifikasi sistem nonlinier menjadi sangat rumit karena komputer hanya dapat menyelesaikan perhitungan yang linier. Metode yang tepat akan dapat memodelkan sistem nonlinier secara akurat mengingat solusi eksak dari permasalahan nonlinier sangat sulit.

Identifikasi sistem nonlinier dapat diselesaikan dengan menggunakan suatu metode yang dinamakan Jaringan Syaraf Tiruan (JST). Jaringan syaraf tiruan adalah suatu teknik pemrosesan informasi yang terinspirasi oleh sistem saraf biologi dengan melakukan proses pembelajaran seperti pengenalan pola. Teknik jaringan saraf tiruan sangat efektif untuk mengidentifikasi sistem nonlinier yang rumit. Menurut Simon Haykin (1999), jaringan syaraf tiruan dapat diklasifikasikan menjadi dua yaitu *feedforward* dan *recurrent*.

Dalam terminologi matematis, sistem nonlinier adalah sistem yang tidak memenuhi prinsip superposisi. Sebuah sistem nonlinier merupakan sistem yang tidak bisa diprediksi berdasarkan perilaku terdahulunya dan memiliki cukup besaran yang tidak diketahui. Sistem nonlinier ini merupakan konsep dari sistem dinamik. Beberapa permasalahan timbul saat dilakukan prediksi pada bentuk sistem yang belum pernah dilatih sebelumnya. Permasalahan-permasalahan tersebut antara lain adalah prediksi sistem untuk waktu yang lama kurang akurat. Hal ini disebabkan jaringan belajar mengenali bentuk sistem lewat pasangan *input-output* dari kecepatan perubahan sistem. Sehingga jaringan syaraf tiruan yang biasa tidak dapat menangkap perilaku sistem untuk waktu yang lama dengan baik.

Penelitian-penelitian lebih lanjut telah berhasil dilakukan untuk memperkenalkan algoritma-algoritma baru dalam pengidentifikasian sistem nonlinear. Namun, karena adanya permasalahan yang masih muncul yaitu proses konvergensi yang lambat dan sensitif terhadap *noise* maka hasil dari pengidentifikasian sistem nonlinear tidak selalu memuaskan. Dalam jurnal ilmiah yang ditulis oleh Jose' de Jesu's Rubio dan Wen Yu pada tahun 2006 diusulkan suatu metode yang dapat memecahkan masalah identifikasi nonlinier yaitu RNN dengan algoritma DKF. RNN yang menggabungkan *feedback* memiliki kemampuan penggambaran yang sangat kuat. EKF juga memiliki konvergensi yang lebih cepat karena interaksi yang terjadi sedikit dan *learning rate* bukan konstanta positif. Algoritma EKF ini akan dimodifikasi dengan algoritma *dead-zone* agar lebih stabil. Algoritma DKF ini akan digunakan untuk mengubah bobot pada lapisan tersembunyi dan lapisan keluaran dari *state-space recurrent neural network* berdasarkan identifikasi *error* antara keluaran jaringan dengan target. Hal ini akan dilakukan secara terus-menerus hingga keluaran dari jaringan sesuai dengan target atau plant sistem.

## 2. Dasar Teori

### 2.1 Sistem Nonlinier

Sistem nonlinier merupakan bagian dari sistem dinamik. Sistem nonlinier adalah suatu sistem yang dinyatakan dalam persamaan nonlinier dan tidak memenuhi prinsip superposisi atau keluaran dari sistem tidak sebanding dengan masukannya. Kasus-kasus sistem nonlinier merupakan kasus dimana variabel-variabel yang akan diselesaikan tidak dapat ditulis dalam kombinasi linier dari komponen-komponen yang independen.

Sistem nonlinier yang diidentifikasi dalam makalah ini adalah sistem nonlinear dinamik dengan kontrol *input* atau vektor masukan yang dinotasikan pada persamaan (1).

$$x(k+1) = f[x(k), u(k)] \quad (1)$$

dimana

$u(k) \in \mathfrak{R}^m$  adalah kontrol *input*, nilai dari  $|u(k)|^2 \leq \bar{u}$ ,  $x(k) \in \mathfrak{R}^n$  adalah vektor *state*, dengan  $f$  tidak diketahui.

Contoh sistem nonlinier dengan kontrol *input* yang digunakan adalah sebagai berikut :

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+2) \end{bmatrix} = \begin{bmatrix} -a_1 x_1 \\ -a_2 x_2 \end{bmatrix} + \begin{bmatrix} \beta_1 \text{sign}(x_2) \\ \beta_2 \text{sign}(x_1) \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$u_1 = u_2 = 0.3\cos(3\pi kTs) + 0.1\cos(4\pi kTs) + 0.6\cos(\pi kTs)$$

Walaupun beberapa hubungan fisis seringkali dinyatakan dengan persamaan linear, tetapi dalam kebanyakan kasus hubungan yang sebenarnya adalah tidak benar-benar linear.

### 2.2 Identifikasi Sistem Nonlinier

Identifikasi sistem adalah pendekatan yang bersifat percobaan untuk memodelkan suatu proses atau plant dari parameter yang tidak diketahui. Pada konteks ini memodelkan berarti membangun representasi matematis dari perilaku dinamis suatu sistem atau proses baik dalam waktu atau frekuensi domain. Identifikasi sistem juga dikenal dengan proses evaluasi dari pembangunan model disetiap waktu atau frekuensi domain. Evaluasi dapat dilakukan dengan pengujian ke dalam pendekatan model yang dipakai dengan memasukkan data masukan yang ada. Dari hasil analisa data masukan ke dalam pemodelan maka akan didapatkan hasil keluaran model. Hasil keluaran model ini kemudian dibandingkan dengan data keluaran sistem sehingga dapat dievaluasi dengan memperbaiki parameter tertentu dalam model yang digunakan.

Identifikasi sistem meliputi beberapa proses tahapan antara lain perencanaan yang bersifat percobaan, pemilihan struktur model yang akan dipakai, estimasi parameter, dan validasi model. Langkah-langkah identifikasi sistem akan terus berulang sampai didapatkan hasil pembangunan model yang memuaskan. Untuk membangun identifikasi model yang parameternya cocok pada suatu plant dari sistem nonlinier yang dinamik terdapat 2 jenis prosedur yaitu

#### 1. Model *State-Space*.

Identifikasi sistem nonlinier dapat dilakukan dengan menggunakan model *state-space* yang dinyatakan dalam persamaan.

$$x(k+1) = f[x(k), u(k)]$$

$$y(k) = h(x(k))$$

dimana fungsi  $f(\cdot)$  dan  $h(\cdot)$  merupakan fungsi nonlinier bernilai vektor yang diasumsikan tidak diketahui.  $u(k)$  adalah vektor masukan yang berukuran  $m \times 1$  dan  $x(k)$  adalah vektor keluaran dari lapisan *hidden* pada waktu ke  $k$  yang berukuran  $q \times 1$ .

#### 2. Model *Input-Output*.

Model *input-output* digunakan untuk identifikasi sistem nonlinier yang hanya dapat diketahui nilai keluarannya. Model ini tidak menggunakan *state*. Sistem menggunakan satu masukan dan satu keluaran.  $y(k)$  adalah

keluaran sistem dan  $u(k)$  adalah nilai masukan untuk waktu ke  $k$ . Model ini menggunakan *single input-single output* yang salah satu modelnya adalah NARX (*nonlinear autoregressive with exogenous inputs*) model.

$$\hat{y}(k+1) = \varphi(y(k)...y(k-q+1), u(k), \dots, u(k-q+1))$$

### 2.3 Identifikasi Sistem dengan State-Space Recurrent Neural Network

Teknik JST sangat efektif dalam mengidentifikasi sistem nonlinier yang kompleks ketika model informasi yang lengkap sudah tidak bisa didapatkan. Jenis model arsitektur jaringan yang digunakan dalam makalah ini adalah RNN yang memiliki *feedback loop* pada arsitektur jaringannya. RNN memiliki kemampuan penggambaran yang sangat bagus dan dapat mengatasi kelemahan *feedforward*. RNN dapat memperkirakan kecepatan perubahan bentuk sistem dengan baik pada sub jaringannya berdasarkan interpolasi daerah domain dalam satu selang waktu tertentu atau jumlah iterasi sehingga dapat memprediksi bentuk lintasan sistem untuk waktu yang lama. Selain itu diharapkan RNN ini dapat memperkirakan jalannya sistem untuk selang waktu yang berbeda-beda.

Model *state-space recurrent neural network* yang digunakan untuk mengidentifikasi sistem nonlinier pada persamaan (1) dengan  $n$  buah input-output dan buah lapisan  $m$  hidden adalah sebagai berikut:

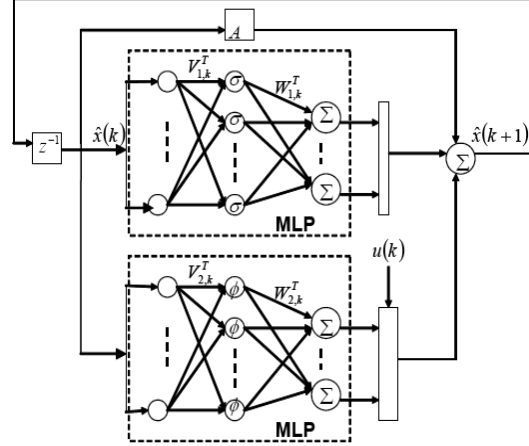
$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + V_{1,k} \sigma[W_{1,k}x(k)] \\ &+ V_{2,k} \phi[W_{2,k}x(k)]u(k) \end{aligned} \quad (2)$$

dimana  $\hat{x}(k) \in \mathcal{R}^n$  adalah internal state vektor,  $A \in \mathcal{R}^{n \times n}$  adalah matrik tetap,  $x(k) \in \mathcal{R}^n$  adalah state vektor,  $u(k) \in \mathcal{R}^m$  adalah vektor masukan. Nilai  $u(k)$  dan  $x(k)$  diketahui.

$W_{1,k}, W_{2,k} \in \mathcal{R}^{m \times n}$  adalah bobot pada lapisan hidden sedangkan  $V_{1,k}, V_{2,k} \in \mathcal{R}^{n \times m}$  adalah bobot pada lapisan output.  $\hat{x}(k+1)$  adalah nilai dari state vektor hasil perkiraan pada iterasi ke  $k+1$ . Fungsi aktivasi yang digunakan pada jaringan adalah  $\sigma = [\sigma_1 \dots \sigma_m]^T$ ,  $\sigma \in \mathcal{R}^m$  dan  $\phi \in \mathcal{R}^{m \times m}$ . Nilai  $\sigma$  dan  $\phi$  dihitung dengan menggunakan fungsi sigmoid. Fungsi sigmoid yang digunakan adalah fungsi pada persamaan (3) dan (4).

$$\sigma(g(x)) = \frac{2}{1 + e^{-2g(x)}} - 0.5 \quad (3)$$

$$\phi(g(x)) = \frac{0.2}{1 + e^{-0.2g(x)}} - 0.05 \quad (4)$$



Gambar 1. Arsitektur recurrent neural network.

Keluaran jaringan dapat dihitung dengan persamaan berikut :

$$\begin{aligned} y(k) &= x(k+1) - Ax(k) - V_1^0 \sigma[W_1^0 x(k)] \\ &- V_2^0 \phi[W_2^0 x(k)]u(k). \end{aligned}$$

## 3. Metode

### 3.1 Pembelajaran dengan Algoritma Backpropagation

BP digunakan dalam hal melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa. Namun BP masih memiliki kelemahan antara lain proses konvergensi yang masih lambat, beban komputasional yang meningkat seiring bertambahnya iterasi dan proses pembelajaran sensitif terhadap *noise*. Pembelajaran bobot dengan algoritma BP dapat dihitung dengan persamaan sebagai berikut :

$$V_{1,k+1} = V_{1,k} - \lambda \sigma[W_{1,k}x(k)]e^T(k) \quad (5)$$

$$W_{1,k+1} = W_{1,k} - \lambda \sigma'[W_{1,k}x(k)]V_{1,k}^T x(k)e^T(k)$$

dimana,  $\lambda$  adalah learning rate dari jaringan.

### 3.2 Pembelajaran dengan Algoritma Extended Kalman Filter

Teknik Kalman filter digunakan untuk melakukan pembelajaran pada RNN (2) dengan nilai

identifikasi error  $e_i(k)$  antara plant dengan hasil estimasi JST yang dapat dibatasi.

$$e_i(k) = \hat{x}_i(k) - x_i(k)$$

Persamaan-persamaan yang digunakan EKF untuk pembelajaran RNN antara lain :

$$\begin{aligned} \hat{\theta}_i(k+1) &= \hat{\theta}_i(k) - K_k e_i(k), \\ \hat{y}_i(k) &= B_k^T \hat{\theta}_i(k), \\ P_{k+1} &= R_1 + [I - K_k B_k^T] P_k, \\ K_k &= P_k B_k (R_2 + B_k^T P_k B_k)^{-1}, \end{aligned} \quad (6)$$

dimana

$$i = 1, \dots, n$$

$P_k \in R^{4m \times 4m}$ ,  $P_k$  adalah matrik semidefinit.

$$B_k = [\sigma V_{1,k}^T x, \phi u, \phi' \text{diag}(u) V_{2,k}^T x(k)]^T$$

$$[\theta_1(k), \dots, \theta_n(k)] = [V_{1,k}, W_{1,k}^T, V_{2,k}, W_{2,k}^T]^T$$

$\theta_i(k) \in R^{4m \times 1}$ ,  $\theta_i(k)$  adalah state vektor yang merupakan bobot tiap sub sistem.

$R_1$  adalah konstanta positif yang kecil.  $R_2$  adalah konstanta positif.

### 3.3 Pembelajaran dengan Algoritma Dead-Zone Kalman Filter

Algoritma ini merupakan modifikasi dari algoritma EKF. Proses identifikasi sistem nonlinier dengan menggunakan algoritma pembelajaran ini dapat menghasilkan pendekatan yang lebih stabil dengan kecepatan konvergensi yang lebih cepat.

Teori kestabilan *Lyapunov* digunakan untuk membuktikan bahwa algoritma *Kalman filter* dengan *dead-zone* stabil untuk identifikasi sistem.

Algoritma EKF (6) akan dimodifikasi menjadi DKF sebagai berikut

$$\begin{aligned} \hat{\theta}_i(k+1) &= \hat{\theta}_i(k) - R_k^{-1} P_k B_k s_k, \\ \hat{y}_i(k) &= B_k^T \hat{\theta}_i(k), \end{aligned} \quad (7)$$

dimana  $R_k = R_2 + B_k^T P_k B_k$  adalah skalar,

$$s_k = \begin{cases} e_i(k), \frac{1}{R_k} e_i^2(k) \geq \frac{3}{R_2} \bar{\zeta}_i^2 \text{ dan } \frac{1}{R_k} \|P_k B_k B_k^T P_k\| \geq \|R_1\| \\ 0, \frac{1}{R_k} e_i^2(k) < \frac{3}{R_2} \bar{\zeta}_i^2 \text{ atau } \frac{1}{R_k} \|P_k B_k B_k^T P_k\| < \|R_1\| \end{cases}$$

$R_1$  adalah konstanta positif yang kecil.  $R_2$  adalah konstanta positif.  $\bar{\zeta}_i$  adalah parameter *dead-zone*

dan sebagai batas tertinggi dari nilai  $\zeta_{i,k}$  yang tidak pasti,  $|\zeta_{i,k}| < \bar{\zeta}_i$ .

Secara keseluruhan penerapan algoritma DKF dalam RNN dapat didesain dengan langkah-langkah sebagai berikut :

- Inisialisasi nilai untuk bobot  $\hat{\theta}_i(0)$  dan  $P_0 > 0$ .
- Membangun model dari RNN (2) untuk mengidentifikasi suatu sistem nonlinier yang tidak diketahui (1). Matrik A adalah matrik stabil.
- Tulis JST ke dalam bentuk linier  $y(k) = B_k^T \Theta_k + \zeta(k)$  dengan  $\Theta_k = [V_{1,k}, W_{1,k}^T, V_{2,k}, W_{2,k}^T]^T = [\theta_1(k), \dots, \theta_n(k)]$   
 $B_k = [\sigma V_{1,k}^T x, \phi u, \phi' \text{diag}(u) V_{2,k}^T x(k)]^T$
- Pembelajaran bobot dengan memperbaharui nilai bobot lama dengan menggunakan persamaan (7).
- Nilai  $P_k$  diperbaharui sesuai algoritma *Kalman filter* yaitu  $P_{k+1} = R_1 + [I - K_k B_k^T] P_k$ ,  
 $K_k = P_k B_k (R_2 + B_k^T P_k B_k)^{-1}$ .

### 4 Uji Coba dan Analisis

Uji coba dilakukan pada sebuah PC dengan prosesor Intel Pentium 4 2.26GHz, memori 512 MB RAM. Sistem operasi yang digunakan adalah Windows XP Professional Version 2002 SP2. Bahasa komputasi yang digunakan untuk implementasi metode adalah Matlab 7.0.4.

Uji coba dilakukan dengan beberapa sistem nonlinier dan skenario yang melibatkan jumlah neuron pada lapisan tersembunyi dan iterasi yang digunakan. Uji coba dilakukan dengan 5 Skenario antara lain :

- Skenario 1 : lapisan tersembunyi 1 neuron dengan 200 iterasi
- Skenario 2 : lapisan tersembunyi 1 neuron dengan 600 iterasi
- Skenario 3 : lapisan tersembunyi 10 neuron dengan 200 iterasi
- Skenario 4 : lapisan tersembunyi 10 neuron dengan 600 iterasi

- Skenario 5 : lapisan tersembunyi 20 neuron dengan 200 iterasi

Hasil estimasi yang paling mendekati plant didapatkan pada skenario yang jumlah neuron lapisan tersembunyinya adalah 10 dan jumlah iterasi jaringan sebesar 600 kali.

Sistem nonlinier 1 adalah sebagai berikut :

$$x_1(k+1) = \frac{x_1(k)x_2(k)x_3(k)}{1+x_1(k)^2+x_2(k)^2+x_3(k)^2} + 2u(k)$$

$$x_2(k+1) = x_1(k)$$

$$x_3(k+1) = x_2(k)$$

$$u(k) = 0.3\sin(3\pi kTs) + 0.1\sin(4\pi kTs) + 0.6\sin(\pi kTs)$$

Beberapa nilai parameter yang digunakan untuk uji coba sistem nonlinier 1 yaitu nilai

$A = \text{diag}(0.1)$ ,  $A \in \mathcal{R}^{n \times n}$  merupakan matrik diagonal stabil dan nilai awal bobot

$W_{1,0}, W_{2,0} \in \mathcal{R}^{m \times n}$ ,  $V_{1,0}, V_{2,0} \in \mathcal{R}^{n \times m}$  diinisialisasi antara [0,1]. Nilai parameter untuk algoritma EKF

yang digunakan antara lain nilai matrik semidefinit  $P = \text{diag}(100) \in \mathcal{R}^{n \times n}$  pada saat inisialisasi

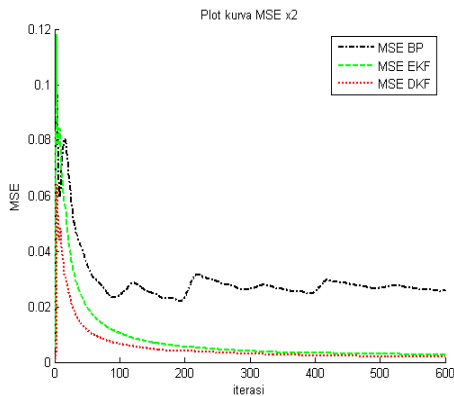
atau iterasi ke 0, nilai  $R_1 = \text{diag}(0.1) \in \mathcal{R}^{n \times n}$  dan nilai  $R_2 = 1.2$ . Parameter pada algoritma DKF

yang digunakan sama pada algoritma EKF hanya ditambah dengan nilai parameter dead-zone

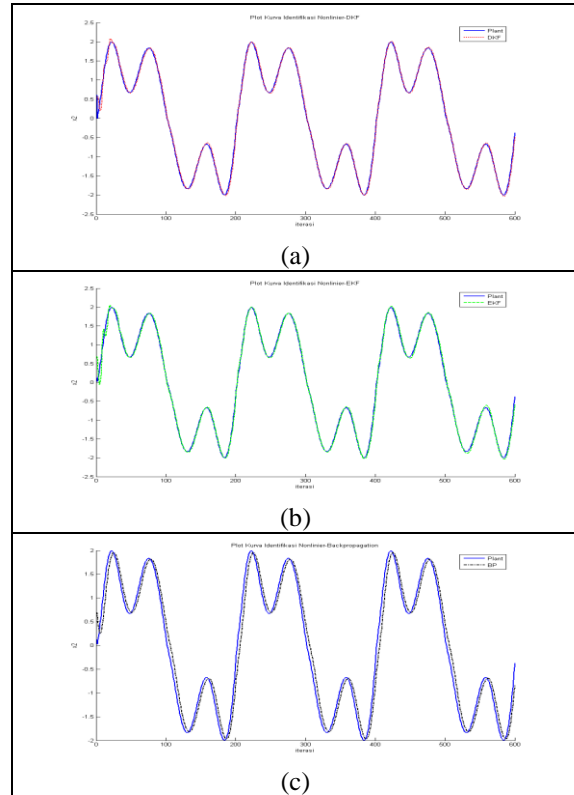
$\bar{\zeta}_i = 0.01$ . Sedangkan untuk algoritma BP hanya memerlukan parameter *learning rate*  $\lambda = 0.1$ .

Hasil identifikasi sistem nonlinier 1 dengan menggunakan 10 neuron pada lapisan tersembunyi dan 600 kali iterasi ditampilkan pada gambar 3

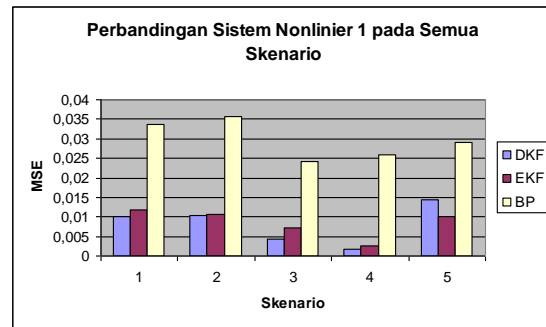
sedangkan grafik perbandingan MSE identifikasi sistem nonlinier 1 dari semua algoritma ditampilkan pada gambar 2.



Gambar 2. Kurva perbandingan nilai MSE identifikasi sistem nonlinier 1 untuk semua algoritma



Gambar 3. Kurva hasil identifikasi sistem nonlinier 1 (a) Identifikasi dengan DKF ; (b) EKF ; (c) BP



Gambar 4. Diagram perbandingan nilai MSE identifikasi sistem nonlinier 1 untuk semua skenario

Gambar 4 merupakan diagram perbandingan MSE hasil uji coba seluruh skenario sistem nonlinier 1. Dari diagram diatas dapat diketahui bahwa nilai MSE yang paling kecil terdapat pada skenario 4.

Sistem nonlinier 2 adalah sebagai berikut :

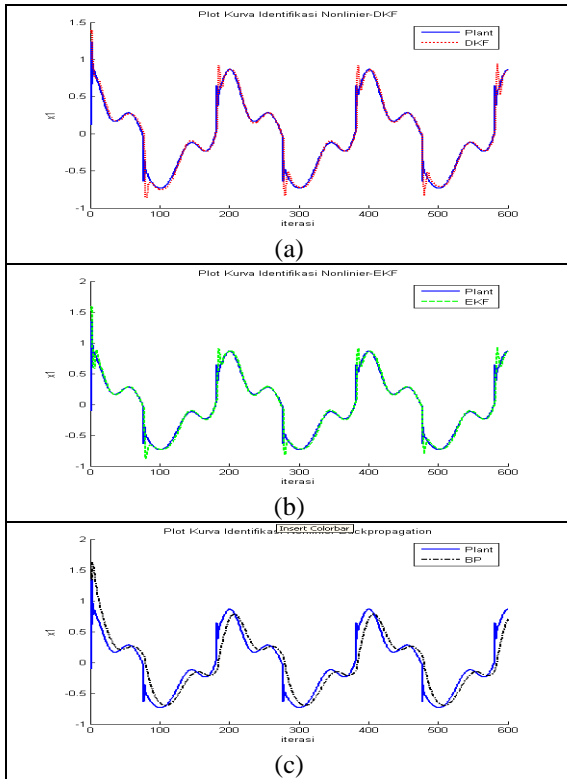
$$\begin{bmatrix} x_1(k+1) \\ x_2(k+2) \end{bmatrix} = \begin{bmatrix} -a_1 x_1 \\ -a_2 x_2 \end{bmatrix} + \begin{bmatrix} \beta_1 \text{sign}(x_2) \\ \beta_2 \text{sign}(x_1) \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

dengan kontrol *input* yaitu

$$u_1 = u_2 = 0.3\cos(3\pi kTs) + 0.1\cos(4\pi kTs) + 0.6\cos(\pi kTs),$$

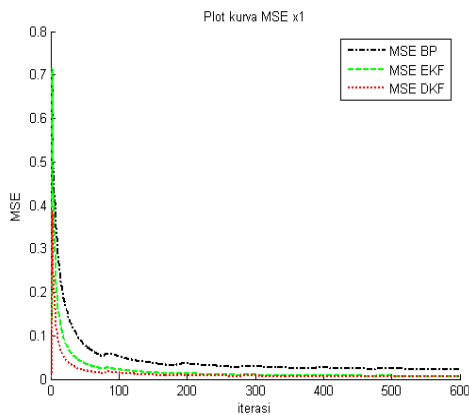
$$a_1 = 0.5, a_2 = 0.1, \beta_1 = 0.3, \beta_2 = 0.5.$$

Inisialisasi nilai parameter sama dengan sistem nonlinier 1.

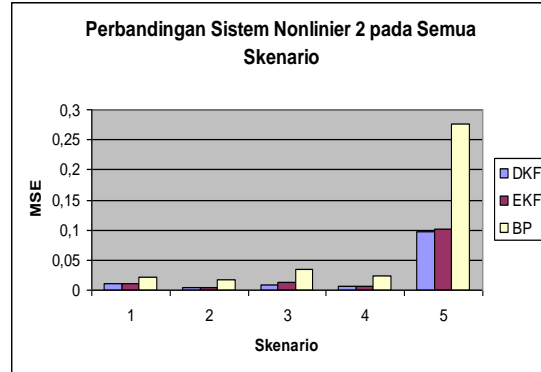


Gambar 5. Kurva hasil identifikasi sistem nonlinier 2  
 (a) Identifikasi dengan DKF ; (b) EKF ; (c) BP

Gambar 5 adalah kurva hasil uji coba identifikasi sistem nonlinier 2 dengan menggunakan 10 neuron pada lapisan tersembunyi dan 600 kali iterasi sedangkan grafik perbandingan MSE identifikasi sistem nonlinier 2 dari semua algoritma ditampilkan pada gambar 6.



Gambar 6. Kurva perbandingan nilai MSE identifikasi sistem nonlinier 2 untuk semua algoritma



Gambar 7. Diagram perbandingan nilai MSE identifikasi sistem nonlinier 2 untuk semua skenario

Gambar 7 merupakan diagram perbandingan MSE hasil uji coba seluruh skenario sistem nonlinier 2. Dari diagram diatas dapat diketahui bahwa nilai MSE dari algoritma DKF paling kecil dibandingkan dengan algoritma EKF dan BP untuk seluruh skenario.

Sistem nonlinier 3 adalah sebagai berikut :

$$x_1(k+1) = mx_1(k) + nx_2(k)x_3(k) + 0.2u(k)$$

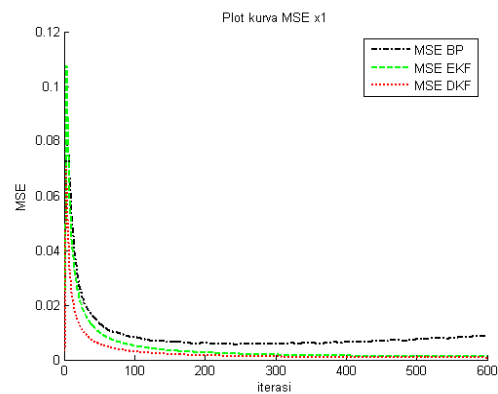
$$x_2(k+1) = mx_2(k) + ox_1(k)x_3(k) + 0.5u(k)$$

$$x_3(k+1) = mx_3(k) + ox_1(k)x_2(k) + 0.3u(k)$$

$$u = 0.3\sin(\pi k/30) + 0.5\sin(k/400)$$

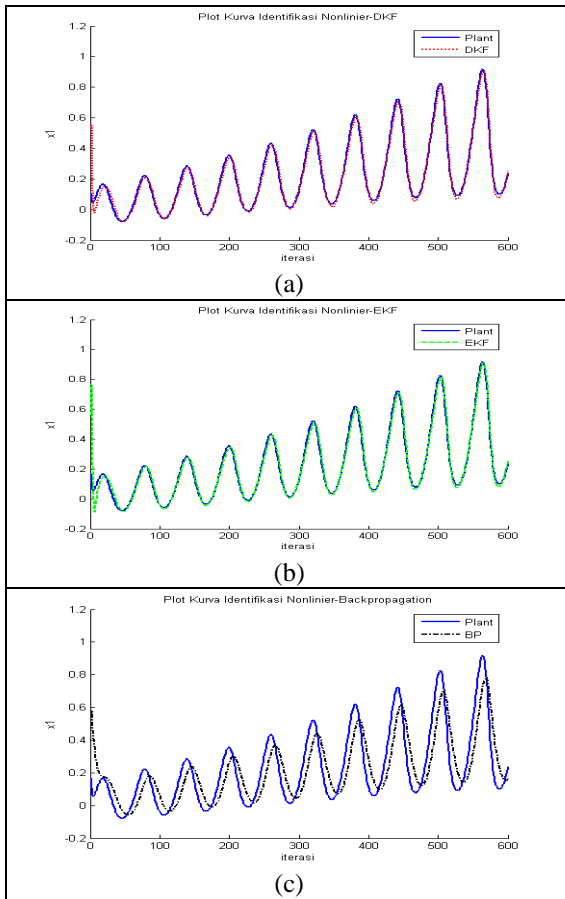
dengan  $m = 0.5, n = 0.3, o = 0.25$ .

Nilai inisialisasi parameter disamakan dengan inisialisasi pada sistem nonlinier 1. Grafik perbandingan MSE identifikasi sistem nonlinier 3 dari semua algoritma ditampilkan pada gambar 8.

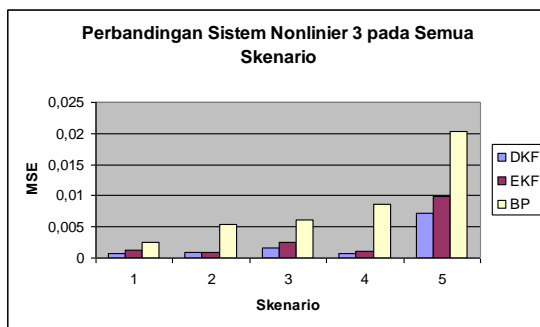


Gambar 8. Kurva perbandingan nilai MSE identifikasi sistem nonlinier 3 untuk semua algoritma

Hasil identifikasi sistem nonlinier 3 dengan menggunakan 10 neuron pada lapisan tersembunyi dan 600 kali iterasi ditampilkan pada gambar 9.



Gambar 9. Kurva hasil identifikasi sistem nonlinier 3  
(a) Identifikasi dengan DKF ; (b) EKF ; (c) BP



Gambar 10. Diagram perbandingan nilai MSE identifikasi sistem nonlinier 3 untuk semua skenario

Dari gambar 10 diatas dapat diketahui bahwa nilai MSE dari algoritma DKF paling kecil dibandingkan dengan algoritma EKF dan BP untuk seluruh skenario.

## 5. Kesimpulan

Setelah dilakukan uji coba dan analisis terhadap perangkat lunak yang dibuat, maka dapat diambil kesimpulan sebagai berikut:

1. Aplikasi RNN dengan algoritma pembelajaran DKF dapat melakukan identifikasi terhadap sistem nonlinier. Algoritma ini dapat dijadikan alternatif untuk menyelesaikan persoalan sistem nonlinier. Sistem nonlinier yang digunakan adalah persoalan nonlinier yang mempunyai kontrol input.
2. Hasil identifikasi sistem nonlinier dengan menggunakan algoritma *Kalman filter* lebih akurat dibandingkan dengan algoritma BP. Hal ini dapat disimpulkan berdasarkan hasil analisa dari kasus yang diujicobakan.
3. Algoritma DKF memiliki kemampuan identifikasi yang cepat. Hal ini dapat dilihat dari state awal yang dipilih merupakan nilai sembarang namun pengenalan selanjutnya bisa dilakukan dengan mengidentifikasi plant dengan cepat.
4. Algoritma DKF mampu bekerja dengan baik dalam melakukan pencocokkan kurva identifikasi (proses *training* jaringan), terlihat dari hasil minimum MSE sebesar 0.00180 pada kasus pertama, 0.00546 pada kasus yang kedua serta 0.0007 pada kasus yang ketiga.
5. Hasil identifikasi sistem nonlinier yang paling mendekati dicapai pada saat jaringan menggunakan 10 buah neuron pada lapisan tersembunyi dan 600 kali iterasi
6. Algoritma DKF memiliki kemampuan konvergensi yang cepat karena semakin lama hasil identifikasi semakin mendekati plant.

## 6. Daftar Pustaka

- [1] Haykin, Simon (1999). **Neural Network A Comprehensive Foundation Second Edition**. New Jersey: Prentice-Hall, Inc.
- [2] Rubio, Jose´ de Jesu´s, Yu, dan Wen. (2007). **Nonlinier system identification with recurrent neural network and dead-zone Kalman filter algorithm**. ScienceDirect, Neurocomputing 70, 2460-2466.
- [3] Rubio, Jose´ de Jesu´s, Yu, Wen dan Li, Xiaou. (2005). **Recurrent Neural Networks Training with Stable Risk-Sensitive Kalman Filter Algorithm**. IEEE.
- [4] Rubio, Jose´ de Jesu´s, dan Yu, Wen.(2007). **Recurrent neural networks training with optimal bounded ellipsoid algorithm**, Proc.of the 2007 American Control Conference Marriot Marquis Hotel at Times Square New York City,USA.
- [5] Yu, Wen.(2005). **State-Space Recurrent Fuzzy Neural Networks for Nonlinier**

**System Identification.** Neural Processing  
letter 22:391-404

- [6] W Yu dan Xiaou Li.(2003). **Discrete-time Nonlinear System Identification Using Recurrent Neural Networks.** IEEE International Conference on Decision and Control Maui, Hawaii USA, December 2003.