

# IDENTIFIKASI SISTEM BERBASIS DATA KE DALAM MODEL NARX-FNN DENGAN *EXTENDED KALMAN FILTER* DALAM KONDISI PENGUKURAN YANG *NOISY* UNTUK SISTEM KOLOM DISTILASI

## *DATA-DRIVEN SYSTEM IDENTIFICATION INTO NARX-FNN MODEL WITH EXTENDED KALMAN FILTER IN NOISY MEASUREMENT CONDITIONS FOR DISTILLATION COLUMN SYSTEM*

Muhammad Alifsyah Putra Nasution<sup>1</sup>, Dimitri Mahayana<sup>1</sup>, Pranoto Hidayat Rusmin<sup>1</sup>, Hayati Amalia<sup>1,2</sup>, Hasan Zidni<sup>1</sup>

<sup>1</sup> Institut Teknologi Bandung, Bandung

<sup>2</sup> Badan Standardisasi Nasional, Tangerang Selatan

Email: malifsyahputranasution@gmail.com

### ABSTRAK

Pada penelitian ini, digunakan *plant* sistem kolom distilasi tipe *batch* untuk memisahkan larutan biner campuran etanol dan air. Agar dapat mengimplementasikan skema sistem kendali, diperlukan pemodelan sistem kolom distilasi. Namun, masalahnya *plant* sistem kolom distilasi tersebut adalah sistem yang *highly nonlinear* dengan terbatasnya jumlah sensor yang dimiliki untuk memperoleh semua informasi *states*. Akibatnya, sistem kolom distilasi diidentifikasi menggunakan pendekatan *black-box modeling* berbasis data masukan-keluaran eksperimen memanfaatkan *Artificial Neural Network* (ANN) untuk memperoleh model *Nonlinear Autoregressive with exogenous inputs* (NARX). Hasil dengan *Mean Squared Error* (MSE) terbaik dicapai pada kasus tanpa *noise* pengukuran menggunakan algoritma *Extended Kalman Filter* (EKF) dengan hiperparameter  $R_{EKF} = 0,05$  yang menghasilkan MSE sebesar  $6,9896e-05$ . Selanjutnya, dapat dilihat pula pada kasus dengan *noise* ringan, algoritma EKF dengan hiperparameter  $R_{EKF} = 0,05$  mampu konvergen dengan paling cepat, yaitu pada sekitar *instance* ke-6. Untuk setiap kasus, secara umum, semakin kecil nilai  $R_{EKF}$  maka semakin cepat identifikasi sistemnya konvergen, namun dengan *trade-off* di mana MSE yang dihasilkan menjadi sedikit memburuk.

**Kata Kunci:** sistem kolom distilasi, *highly nonlinear*, *Artificial Neural Network*, model NARX, algoritma EKF

## **ABSTRACT**

*A batch-type distillation column system is used to separate a binary mixture of ethanol and water. To implement a control system scheme, it is necessary to model the distillation column system. The problem is that the plant is highly nonlinear, and there are a limited number of sensors available to obtain all the states information. Therefore, the plant is identified using a black-box modeling approach based on experiment input-output data, utilizing Artificial Neural Network (ANN) to obtain a Nonlinear Autoregressive with Exogenous inputs (NARX) model. The best results in terms of Mean Squared Error (MSE) were achieved in the case without measurement noise using the Extended Kalman Filter (EKF) algorithm with a hyperparameter  $R_{EKF} = 0,05$ , resulting in an MSE of  $6.9896e-05$ . Furthermore, it can be observed that in the case with light noise, the EKF algorithm with a hyperparameter  $R_{EKF} = 0,05$  converges the fastest, around the 6th instance. In general, for each case, a smaller value of  $R_{EKF}$  typically leads to faster convergence in system identification. However, this comes with a trade-off where the resulting Mean Squared Error (MSE) may degrade slightly.*

**Keywords:** *distillation column system, highly nonlinear, Artificial Neural Network, NARX model, EKF algorithm*

## 1. PENDAHULUAN

Industri kimia di Indonesia sampai saat ini terus berkembang dengan pesat, di mana tujuan dari industri kimia utamanya adalah untuk mengolah bahan mentah menjadi produk akhir (Kementerian Perindustrian, 2023). Salah satu proses yang hampir selalu ada dalam industri kimia adalah proses distilasi, yang merupakan salah satu metode tertua dan paling penting untuk memisahkan larutan dari komponen-komponen penyusunnya, seperti larutan biner campuran etanol dan air (Putri, 2021). Ada dua tipe instrumen yang digunakan dalam proses distilasi, yaitu tipe *continuous* dan tipe *batch*.

Penelitian ini menggunakan *plant* sistem kolom distilasi tipe *batch* di Laboratorium Sistem Kendali Honeywell ITB untuk memisahkan larutan biner campuran etanol dan air. Agar dapat mengimplementasikan skema sistem kendali secara efektif, diperlukan pemodelan sistem kolom distilasi. Namun, tantangannya terletak pada fakta di lapangan bahwa *plant* sistem kolom distilasi di laboratorium ITB adalah sistem yang *highly nonlinear* dan *multivariable* berdasarkan hukum-hukum fisiknya seperti pada model yang diturunkan oleh (Maulidda dkk., 2018, dan Rohman dkk., 2018). Selain itu, sensor yang tersedia untuk memperoleh

semua informasi *states* juga terbatas di mana hanya terdapat 1 *state*, yang juga merupakan keluaran, yang dapat diukur. *State* ini adalah konsentrasi etanol fasa cair di tangki distilat. Akibatnya, sistem kolom distilasi lebih mudah diidentifikasi dengan pendekatan *Single-Input Single-Output* (SISO). Salah satu cara untuk melakukan identifikasi sistem berbasis data pada sistem kolom distilasi secara SISO tersebut adalah dengan menggunakan pendekatan *black-box modeling* berdasarkan data masukan-keluaran dari eksperimen memanfaatkan jaringan saraf tiruan atau *Artificial Neural Network* (ANN) untuk memperoleh model seperti *Nonlinear Autoregressive with eXogenous inputs* (NARX) seperti pada penelitian (Putri dkk., 2021).

Berdasarkan pada banyak literatur, seperti yang dibahas dalam (Math Insight, Tanpa Tahun), model yang *highly nonlinear* dan *multivariable* bisa diaproksimasi dengan deret Taylor. Hal ini dilakukan dengan mengambil tiga suku pertama dalam ekspansi deret Taylor di sekitar satu titik operasi atau titik kesetimbangan tertentu yang dipilih, dapat diperoleh model NARX. Pendekatan ini dapat diandalkan jika sistem dioperasikan dalam *range* yang sangat sempit, yang artinya hanya dilakukan di sekitar satu titik operasi

tertentu yang dipilih untuk memperoleh model NARX tersebut. Jika digunakan untuk seluruh titik operasi, model NARX yang digunakan tidak dapat merupakan model yang *time-invariant*, tetapi harus merupakan model yang *time-varying*, sehingga perlu disesuaikan pada setiap titik operasi.

Jaringan saraf tiruan atau *Artificial Neural Network* (ANN) telah banyak digunakan dalam berbagai bidang, termasuk klasifikasi, pengenalan gambar, serta kendali dan identifikasi sistem. ANN adalah model matematika yang meniru neuron-neuron dalam otak manusia dan dapat digunakan untuk mengidentifikasi fungsi-fungsi yang tidak diketahui dengan memetakan data masukan-keluaran (Tutunji, 2016). Pada (Putri, 2021), telah dijabarkan hubungan yang jelas antara bobot-bobot dalam *Feedforward Neural Network* (FNN) dan parameter-parameter dalam model NARX. Pada (Putri, 2021), telah digunakan model NARX-FNN untuk *plant* sistem kolom distilasi di laboratorium ITB dengan algoritma pembelajaran atau *learning algorithm* yang digunakan adalah Levenberg-Marquadt.

Penelitian dalam makalah ini pada dasarnya adalah penelitian lanjutan dari yang sudah dilakukan dan dipublikasikan pada (Nasution dkk.,

2023) yang juga menggunakan model NARX-FNN untuk mengidentifikasi *plant* sistem kolom distilasi tipe *batch* di laboratorium ITB. Pada penelitian ini, identifikasi sistem dilakukan secara *online* dengan fokus pada meneliti penggunaan algoritma *Extended Kalman Filter* sebagai *learning algorithm* dengan mengubah-ubah hiperparameter-hiperparameter pada algoritma EKF tersebut yang digunakan untuk mengidentifikasi sistem kolom distilasi dalam beberapa kondisi *noise* pengukuran yang berbeda, yaitu: tanpa *noise* pengukuran, *noise* pengukuran ringan, *noise* pengukuran sedang, dan *noise* pengukuran parah. Tujuan pengujian dibagi ke dalam 4 kasus tersebut adalah untuk melihat identifikasi sistem berbasis data yang dilakukan dengan menggunakan algoritma EKF sangat efisien dan superior meskipun dalam kondisi data pengukuran dengan *noise* yang sangat tinggi (Sahoo dkk., 2013). Kenyataannya data-data yang diperoleh dari sensor hampir pasti mengandung *noise* termasuk sensor pada sistem kolom distilasi (Nasution, 2023).

Identifikasi sistem dilakukan secara *online* karena data masukan-keluaran dari sistem kolom distilasi diambil untuk seluruh titik operasi, sehingga perlu diaproksimasi ke dalam model NARX yang bersifat berubah

terhadap waktu (*time-varying*). Pada (Haykin, 2001), algoritma EKF bekerja secara sekuensial atau *instance-by-instance* (sebagai lawan dari mode *batch*) dan dapat dianggap sebagai ‘metode orde 2’ karena selain menggunakan informasi turunan parsial pertama, juga menggunakan informasi matriks kovarians dari *error* estimasi sehingga diharapkan memiliki performa yang baik dan konvergensi yang cepat meskipun digunakan untuk mengidentifikasi sistem dengan kondisi data masukan-keluaran yang *noisy*.

Makalah ini diatur sebagai berikut: pada bagian 2, dijelaskan tentang model NARX-FNN waktu diskrit. Pada bagian 3 dijabarkan tentang algoritma EKF. Pada bagian 4 dipaparkan hasil dan pembahasan mengenai implementasi identifikasi sistem secara *online* untuk sistem kolom distilasi. Terakhir, pada bagian 5 disajikan kesimpulan. Sementara, tujuan penelitian ini adalah untuk melihat keunggulan dari algoritma EKF yang digunakan untuk identifikasi sistem berbasis data masukan-keluaran *plant* sistem kolom distilasi secara *online* meskipun dalam kondisi data pengukuran dengan *noise* yang sangat tinggi.

## 2. MODEL NARX-FNN WAKTU DISKRET

Putri dkk., 2021 menjabarkan pendekatan dengan struktur model *Nonlinear Autoregressive with eXogenous inputs* (NARX) yang digunakan untuk memodelkan *plant* sistem kolom distilasi tipe *batch* di laboratorium ITB. Struktur model NARX dapat direpresentasikan dalam bentuk polinomial atau dapat juga direpresentasikan seperti pada persamaan [1] berikut:

$$\begin{aligned}
 y(k) &= \sum_{i=0}^m a_i u(k-i) \\
 &+ \sum_{j=1}^n b_j y(k-j) \\
 &+ \sum_{i=0}^m \sum_{j=0}^m a_{ij} u(k-i) u(k-j) \\
 &+ \sum_{i=1}^n \sum_{j=1}^n b_{ij} y(k-i) y(k-j) \\
 &+ \sum_{i=0}^m \sum_{j=1}^n c_{ij} u(k-i) y(k-j)
 \end{aligned} \quad [1]$$

dengan,

$u(k)$  dan  $y(k)$  masing-masing adalah sinyal masukan dan keluaran model pada sampel ke- $k$ ,  $m$  dan  $n$  masing-masing adalah *delay* tertinggi pada masukan dan keluaran,  $a_i$ ,  $b_j$ ,  $a_{ij}$ ,  $b_{ij}$ , dan  $c_{ij}$  adalah parameter-parameter model.

Masukan dan keluaran model NARX tersebut dapat disusun dalam bentuk vektor sebagai berikut:

$$u = [u(k) \quad u(k-1) \quad \dots \quad u(k-m)]^T \quad [2]$$

$$y = [y(k-1) \quad y(k-2) \quad \dots \quad y(k-n)]^T \quad [2]$$

Sementara parameter-parameter model  $a_i$ ,  $b_j$ ,  $a_{ij}$ ,  $b_{ij}$ , dan  $c_{ij}$  dapat juga dituliskan dalam bentuk vektor dan matriks sebagai berikut:

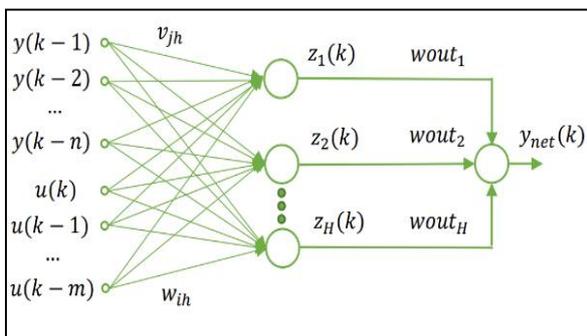
$$a = [a_0 \quad a_1 \quad \dots \quad a_m]^T \dots\dots\dots [3]$$

$$b = [b_1 \quad b_2 \quad \dots \quad b_n]^T \dots\dots\dots [4]$$

$$A = \begin{bmatrix} a_{00} & a_{01} & \dots & a_{0m} \\ a_{10} & a_{11} & \ddots & a_{1m} \\ \vdots & \ddots & \ddots & \vdots \\ a_{m0} & a_{m1} & \dots & a_{mm} \end{bmatrix} \dots [5]$$

$$B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \ddots & b_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix} \dots\dots [6]$$

$$C = \begin{bmatrix} c_{01} & c_{02} & \dots & c_{0n} \\ c_{11} & c_{12} & \ddots & c_{1n} \\ \vdots & \ddots & \ddots & \vdots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{bmatrix} \dots [7]$$



**Gambar 1. Arsitektur FNN yang Digunakan (Tutunji, 2016)**

Setelah itu, persamaan [1] dapat direpresentasikan ke dalam bentuk persamaan [8] berikut:

$$y(k) = a^T u + b^T y + u^T A u + y^T B y + u^T C y \quad [8]$$

Kemudian, menggunakan *neural network* seperti pada Gambar 1 yang terdiri dari 1 *input layer*, 1 *hidden layer* dan 1 *output layer*, akan diestimasi nilai-nilai dari  $a^T$ ,  $b^T$ ,  $A$ ,  $B$ , dan  $C$ .

*Neural network* pada Gambar 1 tersebut dijabarkan dengan persamaan-persamaan dalam bentuk vektor, tidak dalam bentuk sigma. Untuk *output layer*, fungsi aktivasi yang digunakan adalah penjumlahan linier. Sekarang, nilai  $y_{net}(k)$  dapat dituliskan dalam persamaan [9] berikut:

$$y_{net}(k) = W^T Z(k) \dots\dots\dots [9]$$

dengan,

$W$  adalah vektor yang elemen-elemennya adalah bobot-bobot keluaran dari *neural network*  $wout_h$  dengan  $h = 1, 2, \dots, H$ ,

$H$  adalah jumlah dari *hidden nodes* pada *hidden layer*.

Nilai  $W$  dapat dilihat pada persamaan [10] berikut:

$$W = [wout_1 \quad wout_2 \quad \dots \quad wout_H]^T \quad [10]$$

dimana,

$Z(k)$  adalah vektor yang elemen-elemennya adalah keluaran

perhitungan fungsi aktivasi pada setiap *node* yang ada di *hidden layer* seperti yang dituliskan pada persamaan [11] berikut:

$$Z(k) = [z_1(k) \quad z_2(k) \quad \dots \quad z_H(k)] \quad [11]$$

Padahal,  $z_h(k)$  adalah

$$z_h(k) = f_h(w_{ih}^T u + v_{jh}^T y) \quad [12]$$

dengan,

$f_h$  adalah fungsi aktivasi yang digunakan pada setiap *node* yang ada di *hidden layer*,

Nilai  $w_{ih}$  dan  $v_{jh}$  dapat dituliskan dalam bentuk vektor seperti pada persamaan [13] dan [14] sebagai berikut:

$$w_{ih} = [w_{0h} \quad w_{1h} \quad \dots \quad w_{mh}]^T \quad [13]$$

$$v_{jh} = [v_{1h} \quad v_{2h} \quad \dots \quad v_{nh}]^T \quad [14]$$

Sementara nilai masukan  $u$  dan keluaran  $y$  masing-masing telah dituliskan pada persamaan **Error! Reference source not found.** dan [2]. Fungsi aktivasi  $f_h(x)$  pada *hidden layer* dapat didekati dengan ekspansi deret Taylor untuk 3 suku pertama pada  $x_0 = 0$  dengan mengabaikan pembagian  $2!$  pada suku ketiga, seperti pada persamaan [15] berikut:

$$f_h(x) = f_h(0) + f_h^{(1)}(0)x + f_h^{(2)}(0)x^2 \quad [15]$$

Kemudian, dengan mensubstitusi persamaan [11]-[15] ke dalam persamaan

[9] diperoleh persamaan [16] sebagai berikut:

$$\begin{aligned} y_{net}(k) &= W^T [f_1(0) \quad f_2(0) \quad \dots \quad f_H(0)]^T \\ &+ (wout_1 f_1^{(1)}(0) w_{i1}^T + \dots \\ &+ wout_H f_H^{(1)}(0) w_{iH}^T) u \\ &+ (wout_1 f_1^{(1)}(0) v_{j1}^T + \dots \\ &+ wout_H f_H^{(1)}(0) v_{jH}^T) y \\ &+ u^T (wout_1 f_1^{(2)}(0) w_{i1} w_{i1}^T + \dots \\ &+ wout_H f_H^{(2)}(0) w_{iH} w_{iH}^T) u \\ &+ y^T (wout_1 f_1^{(2)}(0) v_{j1} v_{j1}^T + \dots \\ &+ wout_H f_H^{(2)}(0) v_{jH} v_{jH}^T) y \\ &+ 2u^T (wout_1 f_1^{(2)}(0) w_{i1} v_{j1}^T + \dots \\ &+ wout_H f_H^{(2)}(0) w_{iH} v_{jH}^T) y \end{aligned} \quad [16]$$

Persamaan [16] tersebut dapat digunakan untuk merepresentasikan sistem non-linier dalam bentuk model NARX seperti di persamaan [1] setelah proses *training* selesai dilakukan yang artinya *weights* ANN yang konvergen telah diperoleh. Namun, pada persamaan [16] tersebut, untuk suku pertama dari ekspansi deret Taylor atau yang merupakan orde nol dari ekspansi deret Taylor dapat diabaikan saja. Melalui cara tersebut, dengan membandingkan persamaan [8] dan persamaan [16], parameter-parameter model NARX dapat diestimasi sebagai berikut:

$$\begin{aligned} \hat{a}^T &= wout_1 f_1^{(1)}(0) w_{i1}^T + \dots \\ &+ wout_H f_H^{(1)}(0) w_{iH}^T \end{aligned} \quad [17]$$

$$\begin{aligned} \hat{b}^T &= w_{out_1} f_1^{(1)}(0) v_{j_1}^T + \dots \quad [18] \\ &+ w_{out_H} f_H^{(1)}(0) v_{j_H}^T \end{aligned}$$

$$\begin{aligned} \hat{A} &= w_{out_1} f_1^{(2)}(0) w_{i_1} w_{i_1}^T + \dots \quad [19] \\ &+ w_{out_H} f_H^{(2)}(0) w_{i_H} w_{i_H}^T \end{aligned}$$

$$\begin{aligned} \hat{B} &= w_{out_1} f_1^{(2)}(0) v_{j_1} v_{j_1}^T + \dots \quad [20] \\ &+ w_{out_H} f_H^{(2)}(0) v_{j_H} v_{j_H}^T \end{aligned}$$

$$\begin{aligned} \hat{C} &= 2 \left( w_{out_1} f_1^{(2)}(0) w_{i_1} v_{j_1}^T + \dots \quad [21] \right. \\ &\left. + w_{out_H} f_H^{(2)}(0) w_{i_H} v_{j_H}^T \right) \end{aligned}$$

ANN untuk pemodelan NARX-FNN ini kemudian di-*training* dengan data eksperimen dari *plant* sistem kolom distilasi tipe *batch* di laboratorium ITB. Ada beberapa algoritma *learning* untuk melakukan *training* ANN tersebut guna menjalankan identifikasi sistem kolom distilasi secara *online*, namun pada penelitian ini digunakan algoritma EKF. Seperti yang diketahui pada (Nasution, 2023) bahwa, masukan *plant* adalah rasio bukaan katup refluks  $\theta$  dan keluaran *plant* adalah konsentrasi distilat etanol fasa cair  $X_D$ . Namun, perlu diperhatikan bahwa sebelum menjalankan proses *training*, *dataset* harus dinormalisasikan terlebih dahulu ke dalam rentang nilai 0-1 sesuai penelitian (Putri, 2021).

### 3. ALGORITMA EKF

Algoritma EKF digunakan untuk mengestimasi *weights* pada FNN. Algoritma EKF bisa digunakan karena data masukan-keluaran *plant* dapat dianggap berasal dari perhitungan *feedforward* yang melalui FNN seperti pada Gambar 1 dan menggunakan persamaan [16] tersebut dengan semua *weights* pada FNN yang optimal di mana pada dasarnya juga dapat dituliskan ke dalam model *state space* diskret *Multiple-Input Single-Output* (MISO) sebagai berikut (Luttmann dan Mercorelli, 2021 dan Haykin, 2001):

$$W(k+1) = W(k) + d(k) \quad [22]$$

$$\begin{aligned} y_d(k) &= h(W(k), u_d(k)) \\ &+ n(k) \end{aligned} \quad [23]$$

dengan FNN *desired output* (target FNN)  $y_d(k)$  didefinisikan sebagai  $y_d(k) = y_s(k)$  di mana  $y_s(k)$  adalah *sensor output plant*, FNN *desired inputs* (fitur-fitur FNN)  $u_d(k)$  didefinisikan sebagai  $u_d(k) = [u^T \ y_s^T]^T$  dengan  $u$  adalah vektor yang terdiri dari masukan *plant* pada sampel waktu sekarang maupun pada sampel-sampel waktu sebelumnya ( $u = [u(k) \ u(k-1) \ \dots \ u(k-m)]^T$ ) dan  $y_s$  adalah vektor yang terdiri dari *sensor output plant* pada sampel-sampel waktu sebelumnya ( $y_s =$

$[y_s(k-1) \ y_s(k-2) \ \dots \ y_s(k-n)]^T$ , dan  $W(k)$  adalah vektor FNN *true states* yang merupakan vektor yang terdiri dari semua *weights* pada FNN ( $w_{out_h}$ ,  $w_{ih}$ , dan  $v_{jh}$ ) yang optimal. Sementara  $d(k)$  dan  $n(k)$  masing-masing adalah vektor *process disturbance* pada persamaan *state* dan vektor *measurement noise* pada persamaan keluaran. Adapun fungsi linier  $h(\cdot)$  dapat dipahami merupakan perhitungan *feedforward* yang melalui FNN seperti pada Gambar 1 dan menggunakan persamaan [16] tersebut dengan semua *weights* pada FNN yang optimal  $W(k)$  namun tanpa ditambahkan dengan *measurement noise*  $n(k)$ . Akibatnya, sekarang dapat dipahami bahwa  $y_d(k)$  bisa juga dipandang sebagai perhitungan *feedforward* seperti pada  $h(\cdot)$  yang ditambahkan dengan *measurement noise*  $n(k)$ .

Sekarang, tujuannya adalah untuk membuat nilai-nilai vektor FNN *estimate states* yang merupakan *estimate weights*  $\hat{W}(k)$  menuju nilai-nilai vektor FNN *true states* yang merupakan *true weights*  $W(k)$  menggunakan FNN yang harus di-*training* sepanjang waktu pada setiap *instance* atau waktu sampel yang mana sesuai dengan prinsip identifikasi sistem secara *online*. FNN yang harus di-*training* tersebut akan menghasilkan FNN *estimate output*  $\hat{y}_{net}(k)$  dari fungsi

linier  $h(\cdot)$  yang masukannya adalah *estimate weights* pada 1 *time step* sebelumnya  $\hat{W}(k-1)$  dan vektor FNN *desired inputs* (fitur-fitur FNN)  $u_d(k)$  sebagai berikut:

$$\hat{y}_{net}(k) = h(\hat{W}(k-1), u_d(k)) \quad [24]$$

*Training* FNN dilakukan agar nilai  $\hat{y}_{net}(k)$  menuju nilai  $y_d(k)$  sehingga nilai-nilai  $\hat{W}(k)$  konvergen menuju nilai-nilai  $W(k)$ . Jadi, pada dasarnya, algoritma EKF mengerjakan prosedur optimisasi *online*, yang artinya dikerjakan secara *instance-by-instance* dan mencoba menemukan *weights* FNN yang meminimalkan kuadrat dari *error* atau *Sum Squared Error* (SSE) antara ‘*sensor output*’ yang merupakan FNN *desired output* (target FNN)  $y_d(k) = y_s(k)$  dan ‘*estimate output*’ yang merupakan FNN *estimate output*  $\hat{y}_{net}(k) = h(\hat{W}(k-1), u_d(k))$  secara implisit (Luttmann dan Mercorelli, 2021). Lalu, karena ini adalah kasus *instance-by-instance*, maka indeks performa SSE tersebut didekati sebagai fungsi untuk setiap 1 pasangan data fitur-fitur FNN dan target FNN ( $u_d(k)$  dan  $y_d(k)$ ) yang membuat dalam kasus ini nilai SSE yang dihasilkan akan sama dengan nilai *Mean-Squared Error* (MSE).

Berdasarkan model FNN dalam bentuk *state space* diskret seperti pada persamaan [22] dan [23] yang dimiliki, *weight updation* FNN dapat dilakukan dengan menggunakan algoritma EKF sebagai berikut (Luttmann dan Mercorelli, 2021, Haykin, 2001, Sen dkk., 2020, dan Jacquot, 1995):

Untuk prediksi:

$$\begin{aligned}\widehat{W}^-(k) &= f\left(\widehat{W}(k-1), u_d(k-1)\right) \\ &= \widehat{W}(k-1)\end{aligned}\quad [25]$$

$$\begin{aligned}P^-(k) &= (A(k-1))(P(k-1))(A(k-1))^T + Q_{EKF} \\ &= P(k-1) + Q_{EKF}\end{aligned}\quad [26]$$

Untuk koreksi:

$$\begin{aligned}L(k) &= (P^-(k))(C(k))^T \left( (C(k))(P^-(k))(C(k))^T + R_{EKF} \right)^{-1}\end{aligned}\quad [27]$$

$$\begin{aligned}\widehat{W}(k) &= \widehat{W}^-(k) + L(k) \left( y_d(k) - h\left(\widehat{W}^-(k), u_d(k)\right) \right)\end{aligned}\quad [28]$$

$$P(k) = (I - (L(k))(C(k)))P^-(k)\quad [29]$$

dengan  $Q_{EKF}$  adalah matriks kovarians dari  $d(k)$ ,  $R_{EKF}$  adalah matriks kovarians dari  $n(k)$ ,  $P^-(k)$  adalah matriks kovarians *error* estimasi hasil prediksi,  $P(k)$  adalah matriks kovarians *error* estimasi hasil koreksi, dan  $L(k)$  adalah *Kalman gain*. Adapun  $A(k-1)$

merupakan matriks identitas  $I$  berdimensi  $(O \times O)$  di mana elemen-elemen pada diagonal utamanya bernilai 1, sedangkan elemen-elemen lainnya bernilai 0. Untuk  $C(k)$ , dapat dihitung nilainya dengan menggunakan algoritma *standard backpropagation* di mana algoritma *standard backpropagation* pada bagian 3 ini digunakan untuk melakukan turunan parsial pertama dari  $h(\cdot)$  terhadap setiap *weights* pada titik  $W(k) = \widehat{W}^-(k) = \widehat{W}(k-1)$  atau  $C(k) = \frac{\partial h(\widehat{W}(k-1), u_d(k))}{\partial W} = \frac{\partial \hat{y}_{net}(k)}{\partial W}$ .

Untuk menghitung turunan parsial pertama dari  $\hat{y}_{net}(k)$  terhadap setiap *weights* dengan menggunakan FNN seperti Gambar 1 dengan persamaan [16], aturan rantai dapat diterapkan sebagai berikut:

Untuk *weights* antara *hidden layer* dan *output layer* ( $w_{out_h}$ ):

$$\frac{\partial \hat{y}_{net}(k)}{\partial w_{out_h}} = z_h(k)\quad [30]$$

Untuk *weights* antara *input layer* dan *hidden layer* ( $w_{ih}$  dan  $v_{jh}$ ):

$$\begin{aligned}\frac{\partial \hat{y}_{net}(k)}{\partial w_{ih}} &= \frac{\partial \hat{y}_{net}(k)}{\partial z_h(k)} \frac{\partial z_h(k)}{\partial w_{ih}}\end{aligned}\quad [31]$$

$$\begin{aligned} & \frac{\partial \hat{y}_{net}(k)}{\partial w_{ih}} \\ & = (w_{out_h}(k-1))(0,5u(k \\ & - i)) \end{aligned} \quad [32]$$

$$\begin{aligned} & \frac{\partial \hat{y}_{net}(k)}{\partial v_{jh}} \\ & = \frac{\partial \hat{y}_{net}(k)}{\partial z_h(k)} \frac{\partial z_h(k)}{\partial v_{jh}} \end{aligned} \quad [33]$$

$$\begin{aligned} & \frac{\partial \hat{y}_{net}(k)}{\partial v_{jh}} \\ & = (w_{out_h}(k \\ & - 1))(0,5y_s(k-j)) \end{aligned} \quad [34]$$

di mana dapat dilihat untuk persamaan [34], yang digunakan adalah *sensor output*  $y_s$  bukan *true output*  $y$  (Nasution, 2023). Nilai  $y_s$  akan sama dengan  $y$  jika tidak terdapat *measurement noise*. Selain itu, perlu diperhatikan bahwa algoritma *standard backpropagation* pada bagian 3 ini dilakukan untuk semua *weights* secara keseluruhan terlebih dahulu agar mendapatkan matriks Jacobian  $C(k)$  sebelum melakukan *weight updation* untuk semua *weights* secara keseluruhan dengan menggunakan algoritma EKF pada persamaan [25]-[29]. Kemudian, matriks Jacobian  $C(k)$  disusun sebagai berikut:

$$C(k) = [C_1(k) \quad C_2(k) \quad C_3(k)] \quad [35]$$

$$\begin{aligned} & C_1(k) \\ & = \left[ \frac{\partial \hat{y}_{net}(k)}{\partial w_{01}} \quad \frac{\partial \hat{y}_{net}(k)}{\partial w_{11}} \quad \dots \quad \frac{\partial \hat{y}_{net}(k)}{\partial w_{mH}} \right] \end{aligned} \quad [36]$$

$$\begin{aligned} & C_2(k) \\ & = \left[ \frac{\partial \hat{y}_{net}(k)}{\partial v_{11}} \quad \frac{\partial \hat{y}_{net}(k)}{\partial v_{21}} \quad \dots \quad \frac{\partial \hat{y}_{net}(k)}{\partial v_{nH}} \right] \end{aligned} \quad [37]$$

$$\begin{aligned} & C_3(k) \\ & = \left[ \frac{\partial \hat{y}_{net}(k)}{\partial w_{out_1}} \quad \frac{\partial \hat{y}_{net}(k)}{\partial w_{out_2}} \quad \dots \quad \frac{\partial \hat{y}_{net}(k)}{\partial w_{out_H}} \right] \end{aligned} \quad [38]$$

Setelah memiliki matriks Jacobian  $C(k)$ , algoritma EKF untuk *weight updation* FNN seperti pada persamaan [25]-[29] dapat dikerjakan jika terdapat nilai-nilai dari  $u_d(k)$ ,  $y_d(k)$ ,  $\hat{W}(0)$ ,  $P(0)$ ,  $Q_{EKF}$ , dan  $R_{EKF}$  (Luttmann dan Mercorelli, 2021, Haykin, 2001, Sen dkk., 2020, dan Jacquot, 1995). Untuk nilai  $u_d(k)$  dan  $y_d(k)$  tentu tersedia karena merupakan gabungan dari data masukan dan *sensor output* sebagai fitur-fitur FNN untuk  $u_d(k)$  dan merupakan data *sensor output* sebagai target FNN untuk  $y_d(k)$ . Untuk nilai  $\hat{W}(0)$  tentu dapat ditebak saja nilainya, namun perlu diperhatikan bahwa nilai  $\hat{W}(0)$  ini terkait dengan nilai  $P(0)$ . Adapun penjelasan mengenai pemilihan nilai dari  $P(0)$ ,  $Q_{EKF}$ , dan  $R_{EKF}$  dalam praktiknya untuk implementasi algoritma EKF dapat dilihat referensinya pada (Luttmann dan Mercorelli, 2021, Haykin, 2001, Sen dkk., 2020, dan Jacquot, 1995).

#### 4. METODOLOGI

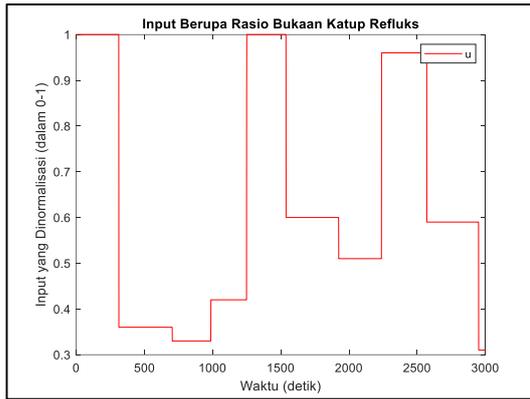
Implementasi identifikasi sistem berbasis data untuk *plant* sistem kolom distilasi tipe *batch* di laboratorium ITB pada penelitian ini tidak berfokus pada struktur atau arsitektur FNN yang digunakan. Tetapi, penelitian ini lebih berfokus pada meneliti penggunaan algoritma *Extended Kalman Filter* sebagai *learning algorithm* dengan mengubah-ubah hiperparameter-hiperparameter pada algoritma EKF tersebut yang digunakan untuk mengidentifikasi sistem kolom distilasi dalam beberapa kondisi *noise* pengukuran yang berbeda, yaitu: tanpa *noise* pengukuran, *noise* pengukuran ringan, *noise* pengukuran sedang, dan *noise* pengukuran parah.

Hasil implementasi identifikasi sistem yang dilakukan diharapkan memiliki MSE dengan nilai yang sekecil mungkin dan ditargetkan minimal besarnya sekitar  $1e-03$  agar lebih baik dari hasil identifikasi sistem kolom distilasi pada penelitian sebelumnya seperti pada (Putri, 2021). Selain itu, dilihat pula kecepatan konvergensi dari *weights* pada saat identifikasi sistem secara *online* dengan melihat banyaknya *instance* yang diperlukan untuk mencapai kondisi konvergen. Adapun *source code* MATLAB untuk

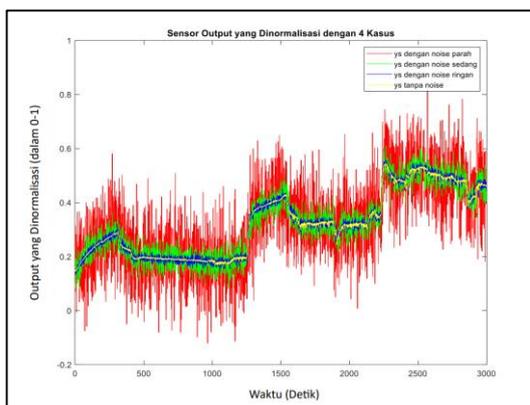
implementasi identifikasi sistem secara *online* untuk *plant* sistem kolom distilasi tipe *batch* di laboratorium ITB tersebut seperti yang dapat dilihat pada *link* akun ‘github.com’ peneliti, yaitu ‘klik: [https://github.com/alifsyah222/Paper\\_Jurnal-Instrumentasi\\_oleh-BSN\\_2023.git](https://github.com/alifsyah222/Paper_Jurnal-Instrumentasi_oleh-BSN_2023.git)’.

Data masukan-keluaran dari *plant* yang telah dinormalisasi dalam rentang 0-1 dan digunakan untuk implementasi identifikasi sistem secara *online* itu dapat dilihat pula pada *link* akun ‘github.com’ peneliti tersebut. Data masukan-keluaran tersebut di-*capture* selama 3000 sekon dengan waktu cuplik sebesar 1 sekon (yang artinya terdapat 3000 *instances* pasangan data masukan-keluaran pada data tersebut). Khusus untuk *plotting* data masukan *plant* (rasio bukaan katup refluks) yang dinormalisasi dapat dilihat pada Gambar 2.

*Plotting* data keluaran *plant* (konsentrasi distilat etanol fasa cair) yang dinormalisasi dengan 4 kasus, yaitu: tanpa *noise* pengukuran (*variance* 0) sebagai kasus 1, *noise* pengukuran ringan (*variance* 0,0001) sebagai kasus 2, *noise* pengukuran sedang (*variance* 0,001) sebagai kasus 3, dan *noise* pengukuran parah (*variance* 0,01) sebagai kasus 4, dapat dilihat pada Gambar 3.



**Gambar 2. Masukan yang Dinormalisasi**



**Gambar 3. Keluaran yang Dinormalisasi dengan 4 kasus**

Selanjutnya, arsitektur FNN dapat diatur dengan banyak *hidden nodes* adalah 5 dan orde model adalah 5 seperti pada penelitian (Nasution dkk., 2023). Sementara untuk inialisasi *weights* pada dasarnya dapat ditentukan saja nilainya secara acak sehingga pada penelitian ini untuk kemudahan ditentukan saja nilainya sebesar 0,05 untuk setiap *weights*. Lalu, beberapa percobaan/pengujian dilakukan dengan mengatur hiperparameter-hiperparameter dari algoritma EKF seperti pada Tabel 1 berikut:

**Tabel 1. Kombinasi Pengujian yang Dilakukan dengan Beberapa Variasi Hiperparameter-Hiperparameter dari Algoritma EKF**

No.	Hiperparameter-Hiperparameter Algoritma EKF
1	$R_{EKF} = 2, P(0) = 50I_{O \times O}$ $Q_{EKF} = 0,000001I_{O \times O}$
2	$R_{EKF} = 1, P(0) = 50I_{O \times O}$ $Q_{EKF} = 0,000001I_{O \times O}$
3	$R_{EKF} = 0,05, P(0) = 50I_{O \times O}$ $Q_{EKF} = 0,000001I_{O \times O}$

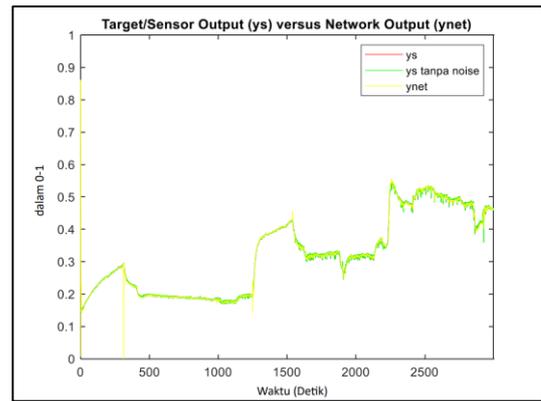
Tabel 1 dapat dilihat  $I_{O \times O}$  merupakan matriks identitas berukuran ( $O \times O$ ) dengan  $O$  adalah banyaknya *weights* pada FNN. Semua kombinasi pada Tabel 1 kemudian masing-masing diuji pada setiap kasus *noise* pengukuran, yaitu: tanpa *noise* pengukuran (*variance* 0) sebagai kasus 1, *noise* pengukuran ringan (*variance* 0,0001) sebagai kasus 2, *noise* pengukuran sedang (*variance* 0,001) sebagai kasus 3, dan *noise* pengukuran parah (*variance* 0,01) sebagai kasus 4.

## 5. HASIL DAN PEMBAHASAN

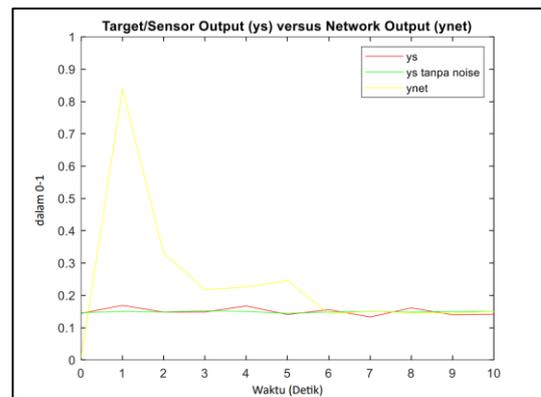
Gambar 7 dan Gambar 8 disajikan hasil identifikasi sistem secara *online* pada *plant* sistem kolom distilasi tipe *batch* di laboratorium ITB. Berdasarkan hasil pada Gambar 7, dapat dilihat bahwa untuk MSE terbaik dicapai oleh kasus 1 no. 3 (artinya kasus 1 dengan

hiperparameter-hiperparameter seperti pada Tabel 1 no. 3) dengan algoritma EKF, yaitu sebesar  $6,9896e-05$ . Pada Gambar 4 dapat dilihat *plotting* respons keluaran untuk kasus 1 no. 3 tersebut. Meskipun, dapat dilihat hampir semua kombinasi pada Gambar 7 memiliki performa yang baik karena mampu mencapai MSE yang ditargetkan, yaitu minimal besarnya sekitar  $1e-03$  (hanya kasus 4 no. 2 dan no. 3 yang gagal mencapai target tersebut).

Selanjutnya, dapat dilihat pula pada Gambar 8 kasus 2 no. 3, algoritma EKF mampu konvergen dengan paling cepat, yaitu pada sekitar *instance* ke-6, meskipun dengan *trade-off* di mana hasil MSE yang diperoleh menjadi sedikit memburuk karena cukup besarnya riak respons *network output* yang dihasilkan pada *instance-instance* awal sebelum konvergen. Gambar 5 dapat dilihat *plotting* respons keluaran untuk kasus 2 nomor 3 tersebut. Namun, kemampuan algoritma EKF untuk mampu konvergen dengan hanya menggunakan sedikit data ini menjadi sangat menguntungkan apabila dihadapkan pada kenyataan di lapangan di mana hanya terdapat sedikit *instances* yang terdapat dalam *dataset* (Nasution dkk., 2023).



**Gambar 4. Sensor Output vs Network Output untuk Kasus 1 (Tanpa Noise) No. 3 ( $R_{EKF} = 0,05$ )**

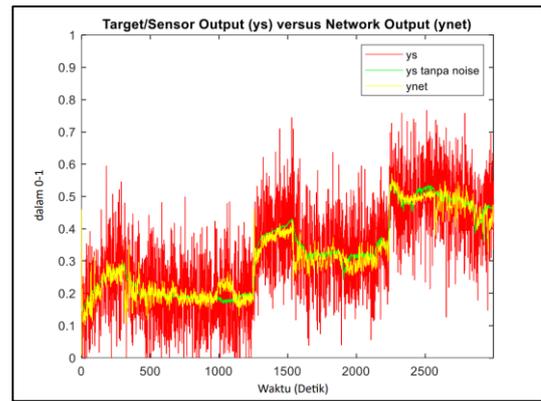


**Gambar 5. Sensor Output vs Sensor Output Tanpa Noise vs Network Output untuk Kasus 2 (Noise Ringan) No. 3 ( $R_{EKF} = 0,05$ )**

Terakhir, dari hasil yang ditampilkan Gambar 7 dan Gambar 8 dapat dipahami bahwa algoritma EKF mampu melakukan identifikasi sistem secara *online* meskipun menggunakan data ber-*noise* tinggi. Penurunan nilai MSE yang dihasilkan tidak terlalu signifikan dari kasus 1 hingga menuju kasus 4. Pada Gambar 6 dapat dilihat *plotting* respons keluaran untuk kasus 4 no. 1 di mana dapat dipahami bahwa

*network output* mampu mendekati nilai *sensor output* asli meskipun menggunakan data keluaran ber-*noise* parah. Untuk setiap kasus, secara umum dapat dipahami pula semakin kecil nilai  $R_{EKF}$  maka semakin cepat identifikasi sistemnya konvergen, namun dengan *trade-off* di mana MSE-nya menjadi sedikit memburuk.

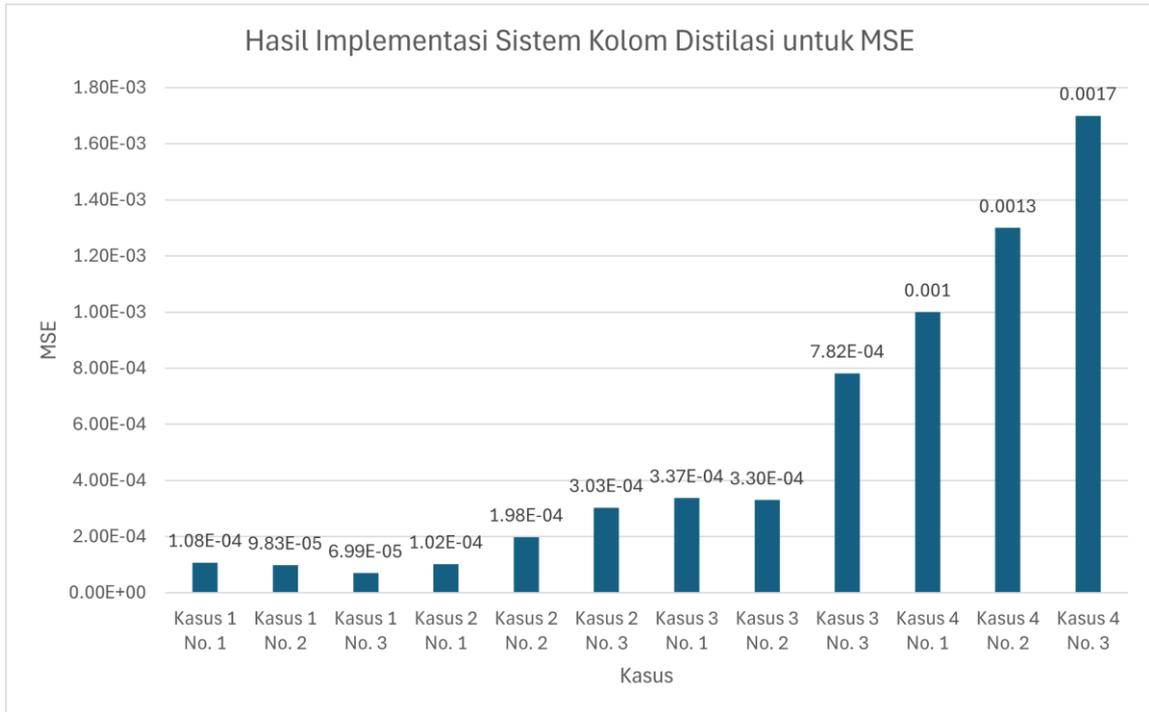
Kontribusi ilmiah penelitian ini adalah meneliti penggunaan algoritma *Extended Kalman Filter* (EKF) sebagai *learning algorithm* dengan mengubah-ubah hiperparameter-hiperparameter pada algoritma EKF tersebut yang digunakan untuk mengidentifikasi sistem kolom distilasi dalam beberapa kondisi *noise* pengukuran yang berbeda. Seperti yang diketahui, pada kenyataannya data-data yang diperoleh dari sensor hampir pasti mengandung *noise* termasuk sensor pada sistem kolom distilasi (Nasution, 2023). Penelitian dalam makalah ini pada dasarnya adalah penelitian lanjutan dari yang sudah dilakukan dan dipublikasikan pada (Nasution dkk., 2023) yang juga menggunakan model NARX-FNN untuk mengidentifikasi *plant* sistem kolom distilasi tipe *batch* di laboratorium ITB. Tetapi, pada (Nasution dkk., 2023), data-data yang digunakan untuk melakukan identifikasi sistem kolom distilasi di laboratorium ITB itu masih diasumsikan tanpa *noise* pengukuran saja.



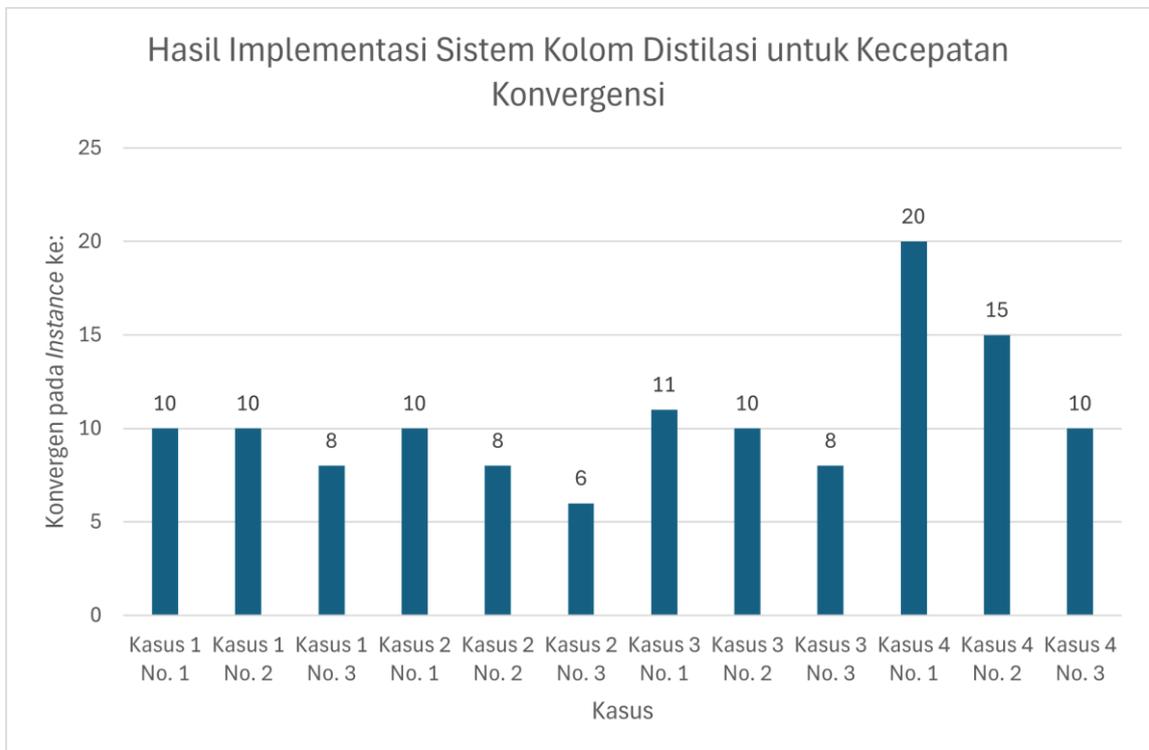
**Gambar 6. Sensor Output vs Sensor Output Tanpa Noise vs Network Output untuk Kasus 4 (Noise Parah) No. 1 ( $R_{EKF} = 2$ )**

## 6. KESIMPULAN

Hasil dengan MSE terbaik dicapai pada kasus 1 (tanpa *noise* pengukuran) menggunakan algoritma EKF dengan hiperparameter  $R_{EKF} = 0,05$  yang menghasilkan MSE sebesar  $6,9896e-05$ . Namun, dapat dipahami bahwa penurunan nilai MSE yang dihasilkan tidak terlalu signifikan dari kasus 1 hingga menuju kasus 4 yang mana menggunakan data keluaran ber-*noise* yang semakin parah menuju kasus 4. Selanjutnya, dapat dilihat pula pada kasus 2 (*noise* ringan), algoritma EKF dengan hiperparameter  $R_{EKF} = 0,05$  mampu konvergen dengan paling cepat, yaitu pada sekitar *instance* ke-6. Untuk setiap kasus, secara umum, semakin kecil nilai  $R_{EKF}$  maka semakin cepat identifikasi sistemnya konvergen, namun dengan *trade-off* di mana MSE yang dihasilkan menjadi sedikit memburuk.



**Gambar 7. Hasil Implementasi Identifikasi Sistem Secara *Online* Sistem Kolom Distilasi Tipe *Batch* di Laboratorium ITB untuk MSE**



**Gambar 8. Hasil Implementasi Identifikasi Sistem Secara *Online* Sistem Kolom Distilasi Tipe *Batch* di Laboratorium ITB untuk Kecepatan Konvergensi**

## 7. UCAPAN TERIMA KASIH

Ucapan terima kasih ditujukan kepada Bapak Dzuhri Radityo Utomo, S.T, M.E., Ph.D. yang merupakan dosen pembimbing skripsi penulis pertama dan telah membimbing penulis pertama mengenai identifikasi sistem dan studi tentang *noise* pengukuran selama masa studi S1 sehingga menginspirasi pengerjaan penelitian pada makalah ini.

## 8. DAFTAR PUSTAKA

- Haykin, S. (2004). *Kalman filtering and neural networks*. John Wiley & Sons.
- Jacquot, R. G. (2019). *Modern digital control systems*. Routledge.
- Kementerian Perindustrian. (2023). *IKI Januari 2023 Meningkatkan Tajam, Industri Optimis Tumbuh Lebih Tinggi di Tahun 2023*. <https://kemenperin.go.id/artikel/23844/IKI-Januari-2023-Meningkat-Tajam,-Industri-Optimis-Tumbuh-Lebih-Tinggi-di-Tahun-2023>
- Luttmann, L., & Mercorelli, P. (2021, October). Comparison of backpropagation and Kalman filter-based training for neural networks. In *2021 25th International Conference on System Theory, Control and Computing (ICSTCC)* (pp. 234-241). IEEE. <https://doi.org/10.1109/ICSTCC52150.2021.9607274>
- Math Insight. (Tanpa Tahun). *Introduction to Taylor's theorem for multivariable functions*. Retrieved July 27, 2023, from [https://mathinsight.org/taylors\\_theorem\\_multivariable\\_introduction](https://mathinsight.org/taylors_theorem_multivariable_introduction)
- Maulidda, R., Rusmin, P. H., Rohman, A. S., Idris Hidayat, E. M., & Mahayana, D. (2018). Modeling and Simulation of Mini Batch Distillation Column. *Proceedings of 2017 5th International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering, ICICI-BME 2017, November*, 62–67. <https://doi.org/10.1109/ICICI-BME.2017.8537724>
- Nasution, M. A. P., Mahayana, D., Rusmin, P. H., & Zidni, H. (2023). Comparison of Stochastic Steepest Gradient Descent and Extended Kalman Filter as ARMA-FNN Learning Algorithms for Data-Driven System Identification of Batch Distillation Column. *2023 IEEE 13th International Conference*

- on *System Engineering and Technology (ICSET)*, October, 233–238.  
<https://doi.org/10.1109/ICSET59111.2023.10295131>
- Nasution, M. A. P. (2023). *Identifikasi Sistem Berbasis Data Menggunakan Neural Network dengan Extended Kalman Filter untuk Sistem Kolom Distilasi*, Tesis Program Magister, Institut Teknologi Bandung. 23221086.
- Putri, A. N. (2021). *Pemodelan dan Perancangan Kendali Optimal Sistem Kolom Distilasi Tipe Batch dengan Pendekatan Penjejakan Kuadratis Linier*, Tesis Program Magister, Institut Teknologi Bandung.
- Putri, A. N., MacHbub, C., & Hidayat, E. M. I. (2021). NARX Neural Network Modeling of Batch Distillation Process. *2021 IEEE 11th International Conference on System Engineering and Technology, ICSET 2021 - Proceedings, November*, 252–257. <https://doi.org/10.1109/ICSET53708.2021.9612562>
- Rohman, A. S., Rusmin, P. H., Maulidda, R., Hidayat, E. M. I., Machbub, C., & Mahayana, D. (2018). Modelling of the mini batch distillation column. *International Journal on Electrical Engineering and Informatics*, 10(2), 350–368. <https://doi.org/10.15676/ijeei.2018.10.2.11>
- Sahoo, H. K., Dash, P. K., & Rath, N. P. (2013). NARX model based nonlinear dynamic system identification using low complexity neural networks and robust  $H_{\infty}$  filter. *Applied Soft Computing Journal*, 13(7), 3324–3334. <https://doi.org/10.1016/j.asoc.2013.02.007>
- Sen, G. D., Gunel, G. O., & Guzelkaya, M. (2020). Extended Kalman Filter Based Modified Elman-Jordan Neural Network for Control and Identification of Nonlinear Systems. *Proceedings - 2020 Innovations in Intelligent Systems and Applications Conference, ASYU 2020*, 1. <https://doi.org/10.1109/ASYU50717.2020.9259812>
- Tutunji, T. A. (2016). Parametric system identification using neural networks. *Applied Soft Computing Journal*, 47, 251–261. <https://doi.org/10.1016/j.asoc.2016.05.012>