

PERENCANAAN MANIPULATOR PARALEL MENGGUNAKAN KONTROL NEURAL NETWORK

Hans Natalius¹⁾, Roche Alimin¹⁾, Felix Pasila²⁾

Prodi Teknik Mesin Universitas Kristen Petra¹⁾, Prodi Teknik Elektro Universitas Kristen Petra²⁾

Jalan. Siwalankerto 121-131, Surabaya 60236. Indonesia

Phone: +62-31-8439040, Fax: 0062-31-8417658

E-mail : hans.natalius@gmail.com , ralimin@peter.petra.ac.id , felix@petra.ac.id

ABSTRAK

Dalam mendesain sistem kontrol manipulator paralel, diperlukan sebuah metode yang efisien untuk menyelesaikan masalah Inverse Static Analysis pada manipulator tersebut. Dalam hal ini, problem ISA adalah bagaimana cara untuk menentukan state tiap aktuator pada manipulator paralel apabila posisi atau gaya diketahui. Penulisan makalah ini melaporkan tentang perencanaan manipulator paralel dengan 16 aktuator dan implementasi neural network sebagai solusi Inverse Static Analysis (ISA) pada manipulator tersebut. Aktuator manipulator pada penelitian ini dikontrol secara diskrit.

Metode penelitian yang digunakan adalah dengan metode simulasi, pengujian perangkat keras berupa manipulator, dan training neural network tipe Elman Neural Network. Hasil penelitian menunjukkan bahwa neural network menunjukkan potensi yang baik untuk digunakan sebagai metode untuk menyelesaikan masalah ISA pada manipulator paralel.

Kata Kunci: Manipulator paralel, Inverse Static Analysis, Elman Neural Network

1. Pendahuluan

Manipulator paralel adalah manipulator yang terdiri dari sejumlah aktuator dan disusun secara paralel. Pada umumnya, mekanisme manipulator paralel terdiri atas kombinasi dari beberapa *joint*, dimana aktuator yang menggerakkan manipulator berfungsi sebagai *prismatic joint* bagi manipulator. Manipulator paralel banyak dikembangkan untuk berbagai macam aplikasi seperti untuk aplikasi *machine tool*, sebagai simulator, dan aplikasi *biomechanic*.

Untuk merancang sistem kontrol manipulator paralel, perlu diketahui beberapa hal, antara lain: data utama yaitu *state* dari setiap aktuator, posisi akhir titik tertentu dari pada manipulator paralel, dan total gaya yang diberikan oleh aktuator kepada *end-effector* manipulator paralel. Dua metode yang biasanya dilakukan untuk mendapatkan data-data diatas adalah metode *Static Analysis* dan metode *Inverse Static Analysis (ISA)*. Metode *Static Analysis* dilakukan dengan menentukan terlebih dahulu *state* dari setiap aktuator untuk dapat menentukan posisi akhir titik tertentu pada manipulator dan total gaya pada manipulator. Metode ISA, yang merupakan kebalikan dari metode *Static Analysis*, dilakukan dengan cara menentukan terlebih dahulu posisi akhir titik tertentu pada manipulator dan total gaya pada manipulator untuk dapat menentukan *state* yang sesuai untuk tiap aktuator.

Ada beberapa solusi yang dapat digunakan untuk menentukan solusi dari ISA, yaitu menentukan *state* aktuator yang sesuai. Solusi pertama, disebut sebagai solusi *Brute Force*, dilakukan dengan cara menghitung jumlah kombinasi *state* dari tiap-tiap aktuator yang memungkinkan dan memeriksa masing-masing kombinasi tersebut satu persatu hingga didapatkan

sejumlah hasil dengan *error* yang terkecil. Solusi ini tidak efisien dan membutuhkan banyak waktu untuk mendapatkan solusi *state* aktuator dari total *state* aktuator yang tersedia. Solusi yang kedua adalah dengan menggunakan metode Jacobian. Metode Jacobian ini sudah digunakan dalam aplikasi sistem kontrol manipulator paralel, seperti contohnya pada *Stewart Platform*. Meskipun demikian, metode ini memiliki kekurangan karena matriks Jacobian hanya dapat menghasilkan maksimum 6 output, sehingga hanya dapat digunakan untuk mengendalikan tidak lebih dari 6 aktuator. Solusi yang memungkinkan untuk dipakai adalah dengan menggunakan sistem kecerdasan buatan *Neural Network*. Dalam manipulator paralel, *Neural Network* digunakan sebagai aproksimator untuk menentukan parameter-parameter yang diperlukan.

Melihat perkembangan sekarang ini, belum ada manipulator paralel yang menggunakan lebih dari 6 aktuator yang dikontrol, yang diakibatkan penggunaan metode Jacobian yang hanya dapat menghasilkan paling banyak 6 output. Untuk itu, perlu didesain sebuah mekanisme manipulator paralel dengan jumlah aktuator lebih dari 6 yang dikontrol secara diskrit dengan menggunakan sistem kecerdasan buatan *Neural Network*.

Penelitian ini bertujuan untuk merancang manipulator paralel dengan aktuator berjumlah lebih dari 6 yang dikontrol secara diskrit dan dikendalikan dengan sistem kontrol *Neural Network*. Tujuan kedua adalah untuk mengimplementasikan sistem kontrol *Neural Network* untuk mengendalikan manipulator paralel yang sudah disebutkan. Tujuan ketiga adalah memperoleh aproksimasi *state* untuk setiap aktuator untuk mendapatkan hasil yang efisien dengan *Root Mean Square (RMS) Error* lebih kecil dari 10%. Sedangkan

manfaat yang dapat diperoleh dari penelitian ini adalah sebagai berikut:

- Memeroleh rancangan manipulator paralel yang menggunakan aktuator yang dikontrol secara diskrit dan dikendalikan dengan sistem kontrol kecerdasan buatan *Neural Network*.
- Mendapatkan metode yang dapat digunakan untuk mengimplementasikan sistem kontrol *Neural Network* untuk mengendalikan manipulator paralel.

Penelitian sebelumnya yang berkaitan dekat dengan pengontrolan manipulator paralel secara diskrit dengan menggunakan kecerdasan buatan dilakukan oleh Pasila [6]. Penelitian ini menitikberatkan pada pengontrolan manipulator paralel 6 DOF dengan menggunakan kecerdasan buatan *Neuro-fuzzy*. Manipulator paralel yang digunakan memiliki 10 aktuator prismatic dengan mekanisme 10-SPS-3D. SPS berarti *Spherical-Prismatic-Spherical*. Aktuator yang digunakan merupakan *double action pneumatic actuator* sehingga membutuhkan *directional control valve* sesuai dengan jumlah aktuatornya. Sistem kontrol yang digunakan merupakan sistem kecerdasan buatan *Takagi Sugeno Multi-Input Multi-Output Neuro-fuzzy* (NFTS) dan *Neuro Fuzzy Look Up Table* (NFLUT) yang merupakan versi *Neuro Fuzzy Takagi Sugeno* yang menggunakan *Look Up Table*. Input yang digunakan merupakan koordinat yang diinginkan serta momen total dari sistem, sedangkan output yang dihasilkan berupa *state* masing-masing aktuator. Pada fase *Fuzzyfication*, digunakan fungsi persamaan Gaussian. Hasil yang didapatkan dari penelitian ini adalah manipulator paralel yang digunakan mengalami *twist* atau terpelintir akibat susunan dari aktuator yang terpisah satu sama lainnya. Sedangkan dari segi implementasi sistem kontrolnya, didapatkan bahwa kedua jenis sistem kontrol *Neuro fuzzy* yang diuji dapat digunakan untuk mengendalikan manipulator paralel dan didapatkan bahwa NFLUT dapat menghasilkan generalisasi yang lebih baik meskipun membutuhkan waktu komputasi yang lebih lama [6].

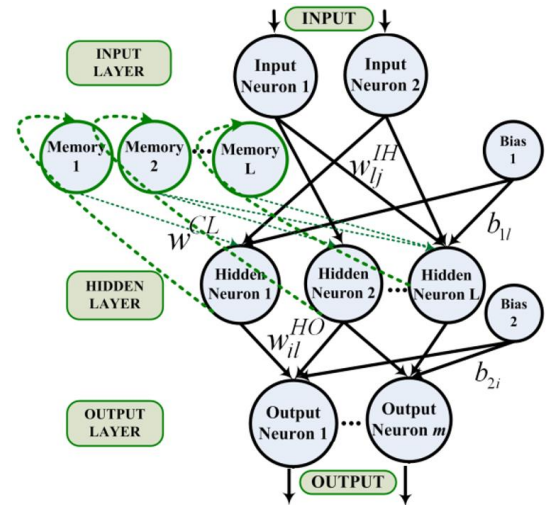
2. Metodologi

Model *Neural Network* yang akan dibahas disini didasarkan pada arsitektur *Recurrent Neural Network* (RNN) dengan fungsi aktivasi *hyperbolic-tangent*. Keduanya juga merupakan model dinamis yang memiliki fitur memori jangka pendek yang memungkinkan NN untuk dapat menghasilkan pemetaan yang berbasis waktu. Untuk input tertentu, $X(n) = [X_1(n), X_2(n)] = [s(n), F(n)]$ pada data ke n (dimana $n= 1, \dots, 819$), kedua model akan menentukan *state* dari aktuator $u_i(n)$ (dimana $i= 1, \dots, m$) sebagai berikut:

$$u_i(t) = \text{round}\left[G\left(b_{2i} + \sum_{l=1}^L w_{il}^{HO} G(\alpha_l(t))\right)\right] \quad (1)$$

Dengan fungsi aktivasi ($G(y)$)

$$G(y) = y/\sqrt{1 + y^2} \quad (2)$$



Gambar 1. RNN-Elmann
Sumber: Pasila (2013)

$$\alpha_l(n) = \sum_{j=1}^2 w_{lj}^{IH} X_j(n) + w^{CL} \alpha_l(n-1) + b_{l1} \quad (3)$$

Dengan $\alpha_l(0) = 0$.

Pada persamaan diatas, $\alpha_l(n)$ adalah total input dari L -*hidden neuron* dan b_{l1} , b_{2i} , w_{il}^{HO} , w_{lj}^{IH} , dan w^{CL} (for $i = 1, \dots, m; j = 1, 2; l = 1, \dots, L$) merupakan parameter model yang harus ditentukan. Dari persamaan ($u_i(n)$) dapat ditentukan apakah model *neural network* yang digunakan merupakan model MLP atau ERNN. ERNN didapatkan dengan memasukan nilai $w^{CL} = 1/L$. Jumlah parameter *network* (L) ditentukan melalui prosedur pembelajaran *neural network* untuk mendapatkan nilai L yang menghasilkan hasil yang optimal. Pada *neural network* yang dibahas, pembelajaran dilakukan dengan menggunakan *Back-Propagation Algorithm* yang dipercepat (BPA) [3,4].

Desain Manipulator Paralel

Desain manipulator paralel yang digunakan terdiri atas sepasang bodi, yaitu bodi atas yang berfungsi sebagai *moving platform* dan bodi bawah yang berfungsi sebagai *fixed body*, yang terhubung oleh 16 aktuator pneumatik. Bodi atas dan bodi bawah masing-masing berbentuk lingkaran dan memiliki diameter yang berbeda. Dalam menentukan dimensi *fixed body* dan *moving platform* untuk manipulator pada penelitian ini, sekaligus menentukan letak masing-masing aktuator, digunakan metode *trial and error*. Metode *trial and error* dilakukan dengan simulasi menggunakan bantuan *software Solidworks Motion Study*. Metode ini dilakukan untuk memperoleh dimensi *fixed body* dan *moving platform* untuk mengakomodasi susunan aktuator agar manipulator tidak mengalami *twist*. Dalam menentukan apakah manipulator akan mengalami *twist* atau tidak, dalam hal ini pada manipulator paralel dengan jumlah aktuator lebih dari 6, terdapat beberapa hal yang harus

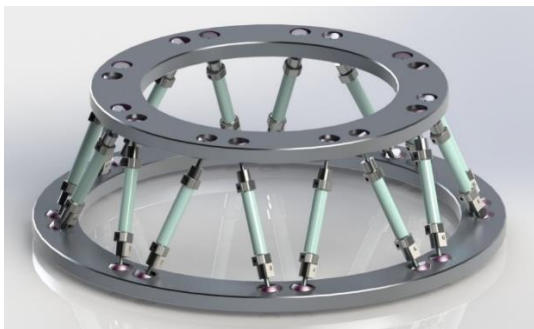
diperhatikan, yaitu jumlah aktuator dan posisi aktuator yang mempengaruhi dimensi dari manipulator. Jumlah aktuator minimum yang diperlukan agar manipulator tidak mengalami *twist* adalah 6 aktuator, dan jumlah aktuator maksimum yang dapat digunakan hanya dibatasi oleh dimensi yang ditetapkan untuk manipulator. Dalam penelitian ini, ditetapkan bahwa aktuator yang digunakan adalah sebanyak 16 aktuator. Dalam menentukan posisi masing-masing aktuator digunakan 2 desain manipulator paralel sebagai acuan, yaitu manipulator Pasila dan Stewart-Gough platform. Kedua desain ini diuji menggunakan bantuan *software* Solidworks Motion Study untuk menguji apakah manipulator akan mengalami *twist*. Dari hasil pengujian dengan Solidworks Motion Study didapatkan bahwa desain manipulator dengan 16 aktuator tidak mengalami *twist* apabila menggunakan susunan posisi aktuator yang berbasis pada Stewart-Gough platform. Spesifikasi bodi dapat dilihat pada Tabel 1 dan Tabel 2. Tampilan manipulator dapat dilihat pada Gambar 2 dan Gambar 3.

Tabel 1. Spesifikasi *Fixed Body* Manipulator

<i>Fixed Body</i>		
Bahan	Aluminium 6061	
Massa	8764.42	Gram
Volume	3246082.03	mm ³
Diameter Luar	740	Mm
Diameter Pusat Joint	660	Mm
Diameter Dalam	596	Mm

Tabel 2. Spesifikasi *Moving Platform* Manipulator

<i>Moving platform</i>		
Bahan	Aluminium 6061	
Massa	6758.56	Gram
Volume	2503170.76	mm ³
Diameter Luar	560	mm
Diameter Pusat Joint	500	mm
Diameter Dalam	400	mm



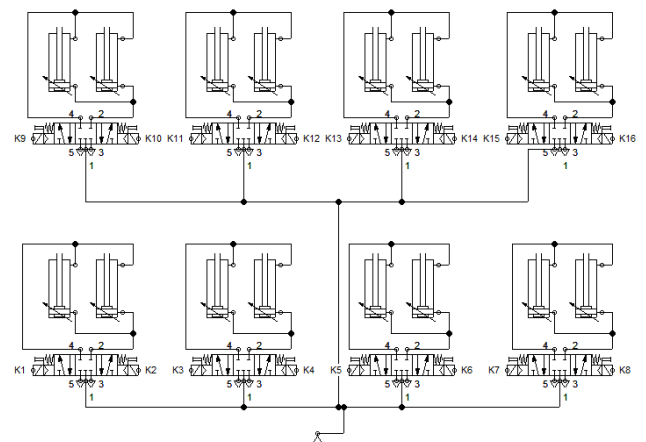
Gambar 2. Tampilan CAD manipulator



Gambar 3. Prototipe manipulator

Manipulator paralel yang digunakan memiliki 16 pasang *spherical joint* dan 16 aktuator pneumatik yang berfungsi sebagai *prismatic joint*. Aktuator dihubungkan dengan *moving platform* dan *fixed body* dengan menggunakan *spherical joint* sehingga terbentuk konstruksi *spherical-prismatic-spherical*. Aktuator yang digunakan merupakan aktuator pneumatik JELPC bertipe *dual action* dengan *stroke* sebesar 70 mm dan *bore* sebesar 12 mm dan dapat bekerja pada tekanan 1 – 9 kg/mm². Kedua ujung aktuator dihubungkan dengan *hub* berdiameter 25 mm dengan tinggi 21 mm yang terbuat dari baja ST60. *Hub* berfungsi untuk menghubungkan aktuator dengan *spherical joint*. Hubungan antara *hub* dan *spherical joint* ini dikunci dengan menggunakan sepasang pelat dengan tebal 1 mm berdiameter 30 mm yang terbuat dari baja ST42.

Sebagai sistem penggerak manipulator, digunakan sebuah kompresor, dan 8 buah *valve solenoid 5/3* yang dioperasikan dengan menggunakan 2 buah *Programmable Logic Controller (PLC)* Siemens S7-200 dimana masing-masing PLC mengendalikan 4 buah *valve*. Tiap *valve* mengendalikan sepasang aktuator yang bekerja bersamaan. Aktuator disetel agar dapat bekerja secara diskrit dengan 3 *state*, yaitu *retract*, *floating*, dan *extend*. Gambar 4 menunjukkan rangkaian sistem pneumatik manipulator yang dibuat dengan bantuan *software* Festo Pneumatic.



Gambar 4. Rangkaian sistem pneumatik manipulator

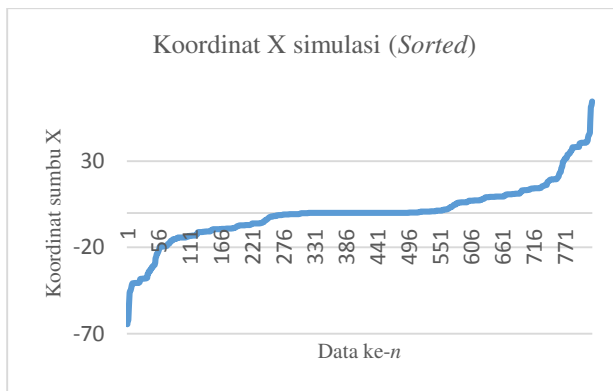
Pengumpulan Data

Pengumpulan data dilakukan dengan 2 cara, yaitu simulasi dengan menggunakan *software* Solidworks

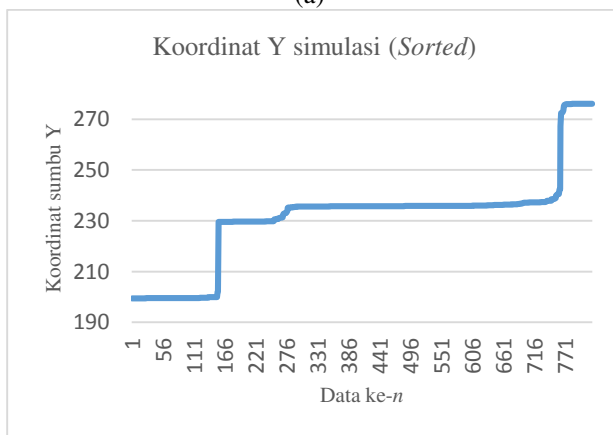
Motion Study dan pengujian mekanis manipulator dengan mengukur posisi sebuah titik pada *moving platform*. Simulasi dilakukan dengan menambahkan parameter 49 buah *Contact* yang berfungsi untuk mencegah sebuah bodi untuk bergerak menembus bodi lainnya. Pengukuran posisi titik *moving platform* dilakukan dengan bantuan jarum dan lampu untuk mempertegas posisi koordinat X dan Z titik *moving platform*. Untuk menentukan posisi koordinat Y titik *moving platform* digunakan alat ukur berupa penggaris L. Data yang didapat dari hasil pengukuran ini, disebut data hasil pengujian mekanis, digunakan untuk menghitung *error* mekanis yang muncul. Dalam pengukuran ini, diambil sampel sebanyak 105 dari 819 data.

3. Hasil dan Pembahasan

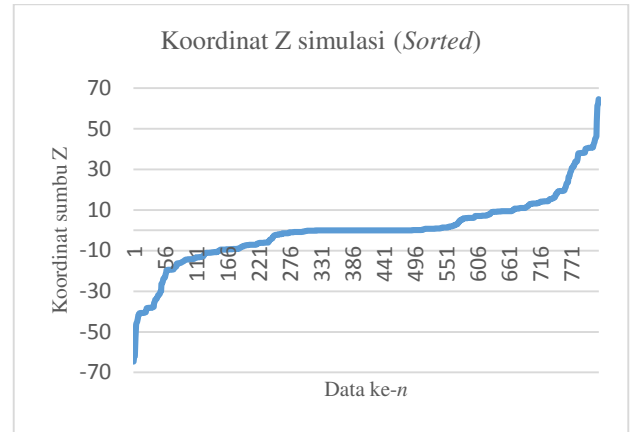
Gambar 5 menunjukkan grafik data hasil simulasi yang berjumlah 819 data yang sudah diurutkan dari nilai terkecil hingga nilai terbesar. Data hasil simulasi didapatkan menggunakan bantuan aplikasi Motion Study.



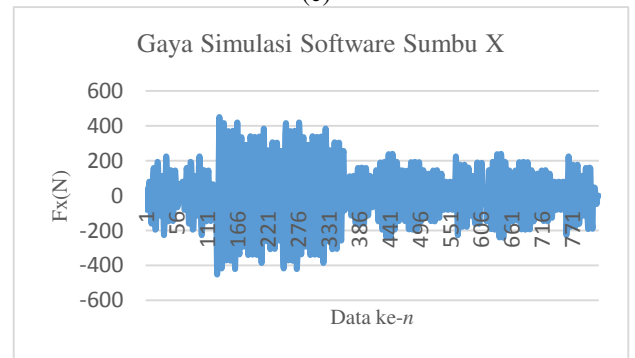
(a)



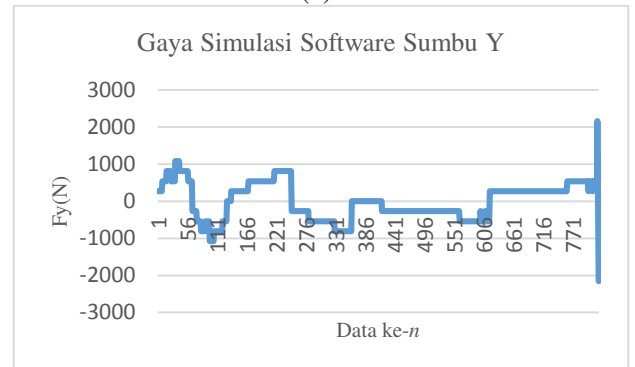
(b)



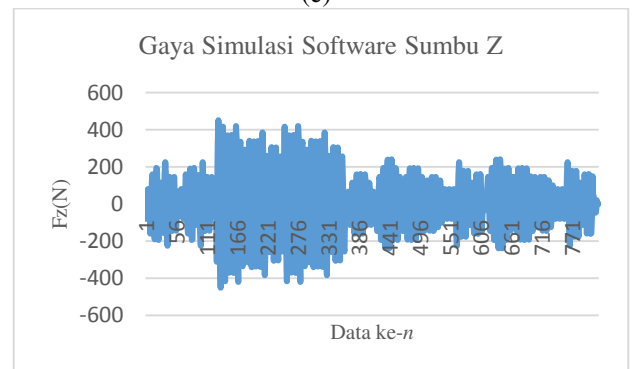
(c)



(d)



(e)



(f)

Gambar 5. Grafik Data Hasil Simulasi Software (a) Koordinat sumbu X hasil simulasi yang sudah diurutkan (b) Koordinat sumbu Y hasil simulasi yang sudah diurutkan (c) Koordinat sumbu Z hasil simulasi yang sudah diurutkan (d) Gaya pada sumbu X hasil simulasi (e) Gaya pada sumbu Y hasil simulasi (f) Gaya pada sumbu Z hasil simulasi.

Dari hasil simulasi didapatkan bahwa untuk posisi koordinat sumbu X nilai maksimalnya adalah 64.63 mm dan nilai minimumnya adalah -64.64 mm, koordinat sumbu Y nilai maksimalnya adalah 276.14 mm dan nilai minimumnya adalah 199.42 mm, dan koordinat sumbu Z nilai maksimalnya adalah 64.62 mm dan nilai minimumnya adalah -64.67mm. Pada sumbu X, gaya maksimalnya adalah 450.64 N, gaya minimumnya adalah -450.64 N. Pada sumbu Y, gaya maksimalnya adalah 2154.78 N, gaya minimumnya adalah -2154.78 N. Pada sumbu Z, gaya maksimalnya adalah 450.64 N, gaya minimumnya adalah -450.64 N. Grafik koordinat dan gaya sumbu Y berbeda dibandingkan grafik lainnya dan memiliki data yang nilainya tidak tersebar merata.

Data hasil pengujian mekanis dibandingkan dengan data hasil simulasi untuk mendapatkan *mechanism error* yang dinyatakan dalam bentuk *root mean square error* (RMSE). RMSE yang didapatkan apabila dinyatakan dalam satuan mm dan persen error dapat dilihat pada Tabel 3.

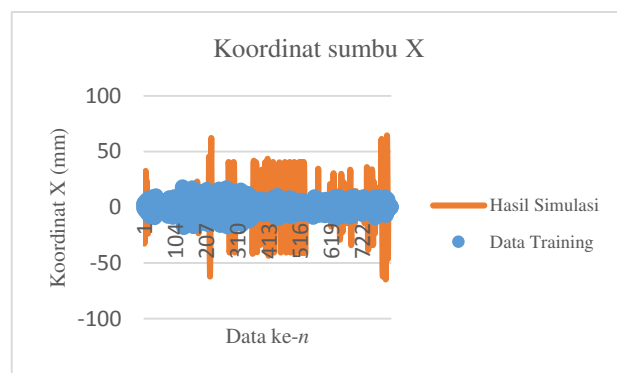
Tabel 3. RMSE perbandingan hasil pengukuran manipulator dan hasil simulasi dengan *software*

RMSE	mm	% Error
Sumbu X	0.57692	5.872%
Sumbu Y	1.0598	0.451%
Sumbu Z	0.47052	6.053%
Gabungan (n_t)	0.43171	2.815%

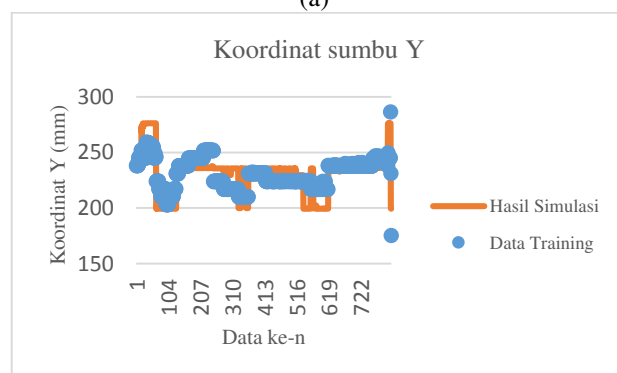
Pada beberapa sampel didapatkan bahwa *error* yang dihasilkan mencapai 100%. Apabila dianalisa, hal ini terjadi pada sampel dimana hasil simulasi menunjukkan titik pengukuran berada pada koordinat X atau Z diantara 1 dan 0, sedangkan pengukuran menunjukkan bahwa titik pengukuran berada pada koordinat X atau Z = 0. Dalam penelitian ini, alat ukur yang digunakan hanya memiliki ketelitian sebesar 1 mm, sehingga tidak memungkinkan untuk secara teliti menunjukkan posisi titik pengukuran saat berada pada kondisi tersebut. Akibatnya, data yang dicatatkan pada pengukuran adalah koordinat terdekat yang didapatkan pada saat pengukuran, dalam hal ini berada pada X dan Z = 0.

Dari hasil yang sudah didapatkan, beberapa sampel data pada sumbu X dan Z menghasilkan *error* yang lebih besar dari 100%. Hal ini diakibatkan konstruksi mekanisme yang masih kurang baik. Dari hasil perhitungan RMSE, didapatkan bahwa RMSE mekanisme pada sumbu X sebesar 5.87%, sumbu Y sebesar 0.45% dan sumbu Z sebesar 6.05%, sedangkan RMSE yang didapat dari perhitungan RMSE gabungan sumbu X, Y, dan Z sebesar 2.81%. Kecilnya nilai RMSE gabungan ini didapat karena nilai RMSE sumbu Y yang kecil, yaitu hanya sebesar 0.45%. Hal ini berarti *mechanism error* yang didapatkan relatif kecil meskipun *error* yang dihasilkan sumbu X dan Z melebihi 5%. Oleh karena itu, dapat dikatakan bahwa meskipun konstruksinya masih kurang baik, mekanisme manipulator paralel 16 aktuator ini bekerja dengan baik.

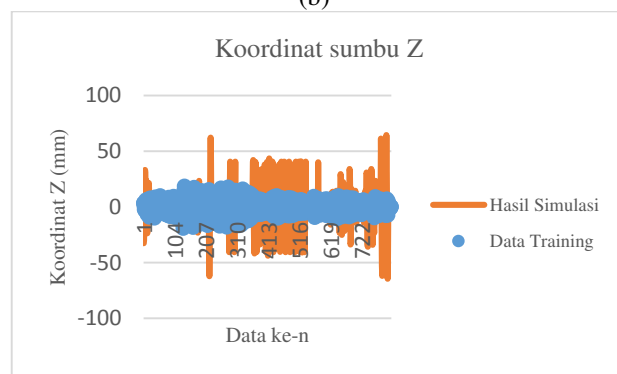
Proses *training* ERNN menghasilkan data training berjumlah 819 data yang berguna untuk memeriksa besarnya RMSE data *training* terhadap data hasil simulasi. Hasil perbandingan data *training* dengan data hasil simulasi dalam bentuk grafik dapat dilihat pada Gambar 6.



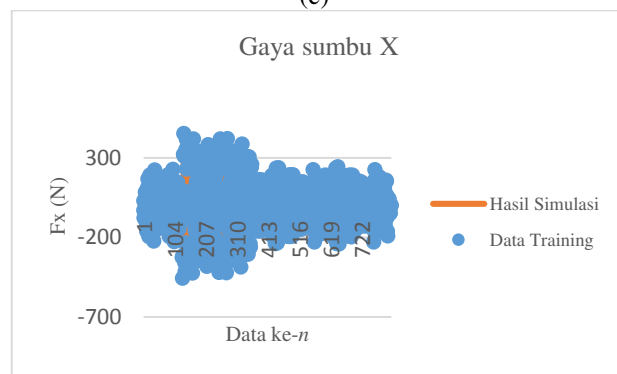
(a)



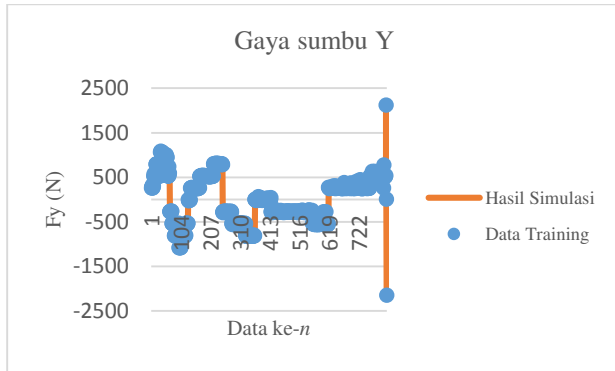
(b)



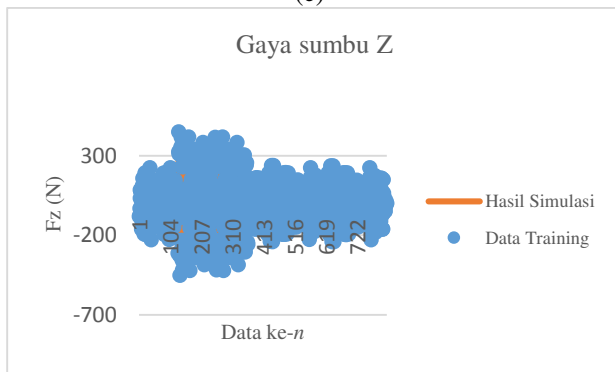
(c)



(d)



(e)



(f)

Gambar 6. Grafik Perbandingan Data Hasil Simulasi *Software* dan Data Hasil *Training Neural Network* (a) Koordinat sumbu X (b) Koordinat sumbu Y (c) Koordinat sumbu Z (d) Gaya sumbu X (e) Gaya sumbu Y (f) Gaya sumbu Z

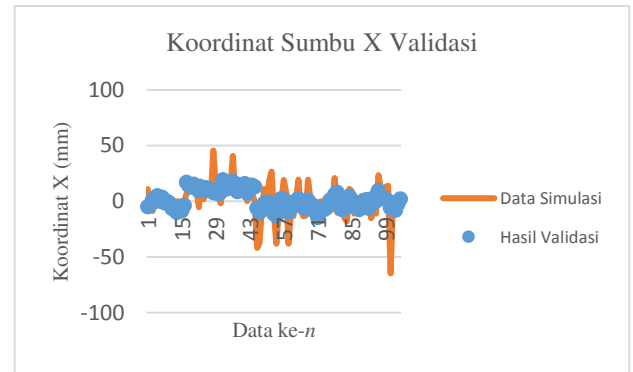
Parameter ERNN yang didapatkan dari hasil *training* yang dilakukan dapat dilihat pada Tabel 4.

Tabel 4. Parameter ERNN yang dihasilkan dari proses *training NN*

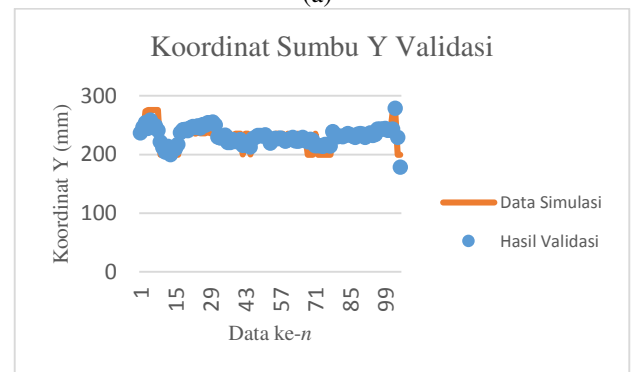
RMSE Training	9.73%
Waktu training	794 (s)
Jumlah hidden neuron	5
Jumlah Input network	8
Jumlah Output network	6
Feedback (w_i)	1

Dari perbandingan hasil *training* dan hasil simulasi, terlihat bahwa ERNN dapat menghasilkan data *training* yang mendekati data hasil simulasi. Dalam hal gaya, data *training* yang dihasilkan sangat mendekati gaya yang didapatkan dari hasil simulasi. Dalam hal koordinat, data *training* masih menunjukkan *error* yang cukup besar. Hal ini terjadi karena nilai data hasil simulasi *software*, yang tidak tersebar merata. Akibatnya, timbul *noise* yang menyebabkan gangguan pada proses *training* yang dapat memperbesar *error* yang terjadi. Dengan waktu *training* sebesar 794 sekon, didapatkan RMSE *training* sebesar 9.73%. ERNN masih dapat dioptimasi lagi untuk mendapatkan nilai RMSE yang lebih kecil, yaitu dengan meningkatkan lamanya waktu *training*.

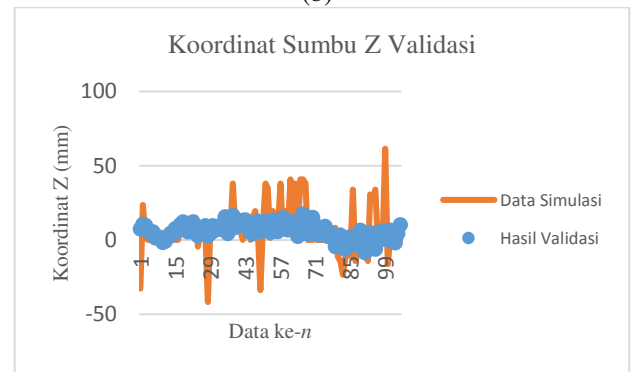
Parameter ERNN yang sudah didapat kemudian diuji lagi untuk mendapatkan data validasi. Pengujian ini menggunakan 105 data yang sama dengan yang digunakan untuk pengujian manipulator. Hasil perbandingan data validasi dengan data hasil simulasi dalam bentuk grafik dapat dilihat pada Gambar 7.



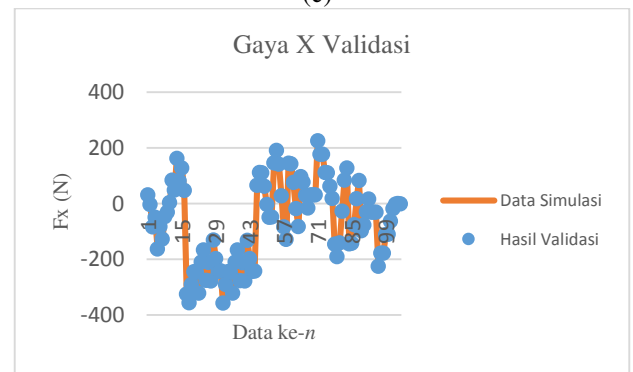
(a)



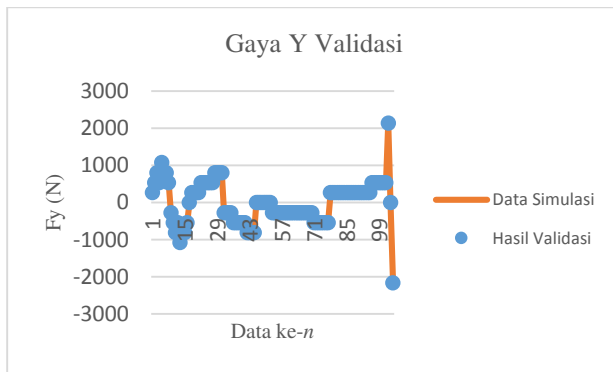
(b)



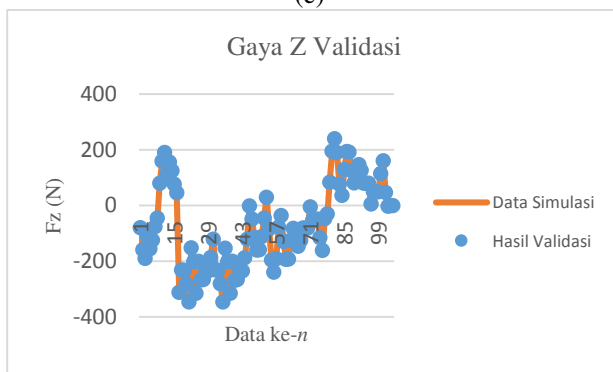
(c)



(d)



(e)



(f)

Gambar 7. Grafik Perbandingan Data Hasil Simulasi *Software* dan Data Hasil Validasi NN (a) Koordinat sumbu X (b) Koordinat sumbu Y (c) Koordinat sumbu Z (d) Gaya sumbu X (e) Gaya sumbu Y (f) Gaya sumbu Z

Dari perbandingan data validasi dan data simulasi, didapatkan nilai RMSE sebesar 16.85%. Dapat dilihat bahwa RMSE validasi lebih besar daripada RMSE *training* yang sebesar 9.73%. *Error* yang paling besar terjadi pada validasi koordinat posisi, seperti yang terjadi pada data hasil *training*. Dalam hal gaya, seperti pada data *training*, data validasi yang dihasilkan sangat mendekati gaya yang didapatkan dari hasil simulasi. Dari data hasil *training* ERNN dan hasil validasi parameter ERNN didapatkan bahwa ERNN dapat melakukan aproksimasi kembali gaya total dan posisi manipulator yang akan dihasilkan oleh suatu *state* aktuator tertentu, sehingga dapat dikatakan bahwa ERNN dapat menyelesaikan masalah ISA yang dalam hal ini merupakan masalah ISA pada manipulator paralel.

4. Kesimpulan

Sebagai kesimpulan, penulisan ini membahas mengenai implementasi *Elmann Recurrent Neural Network* (ERNN) sebagai metode untuk menyelesaikan permasalahan *Inverse Static Analysis* (ISA) pada manipulator paralel 16 aktuator. Manipulator paralel 16 aktuator yang dirancang dalam penulisan ini bekerja dengan baik, ditunjukkan oleh nilai RMSE gabungan sumbu X, Y, dan Z sebesar 2.81%. Pada uji coba yang dilakukan dalam penelitian ini, didapatkan bahwa ERNN menghasilkan RMSE *training* sebesar 9.73% dan RMSE validasi sebesar 16.85%. Nilai-nilai RMSE ini

didapatkan dengan waktu *training* selama 794 detik. Nilai-nilai RMSE tersebut dapat diperkecil lagi melalui optimasi waktu yang digunakan untuk proses *training* ERNN. Dalam penelitian ini, juga didapatkan bahwa ERNN mampu mengaproksimasi kembali besaran gaya dan posisi manipulator yang digunakan sebagai *input* dalam ISA. Oleh karena itu, dapat disimpulkan bahwa ERNN dapat digunakan sebagai metode untuk menyelesaikan masalah ISA pada manipulator paralel.

5. Daftar Pustaka

1. Palit AK, Babuška R, "Efficient training algorithm for Takagi-Sugeno type Neuro-Fuzzy network," Proc. Of FUZZ-IEEE, Melbourne, Australia, vol. 3: 1538-1543, 2001.
2. Palit AK., Popovic D., "Nonlinear combination of forecasts using ANN, FL and NF approaches," FUZZ-IEEE, 2:566-571, 2002a.
3. Pasila F., "Forecasting of Electrical Load using Takagi-Sugeno type MIMO Neuro-Fuzzy network", Master thesis, Bremen, 2006.
4. Palit AK., D. Popovic, "Computational Intelligence in Time Series Forecasting, Theory and Engineering Applications", Springer, 2005.
5. Pasila F., "Inverse Static Analysis of Massive Parallel Arrays of Three-State Actuators via Artificial Intelligence", PhD Dissertation, University of Bologna, 2013.
6. Pasila F. & Alimin R., 2013. "Designing the 6-DOF Massive Parallel Arrays with Artificial Intelligence Control." International Conference on Robotic Automation System (ICORAS 2013), Bali, n.d.