

Penerapan Decision Tree untuk Pemodelan Jawab Kinematika Langsung Manipulator Paralel Bidang 5R

An Application of Decision Tree for Modeling the Direct Kinematic Solution of 5R Planar Parallel

Adriyan

Program Studi Teknik Mesin, Sekolah Tinggi Teknologi Nasional Jambi
Jl. Kapten Patimura No. 100, Kel. Rawasari, Kec. Alam Barajo, Jambi, Indonesia

[doi.10.21063/jtm.2021.v11.i1.61-73](https://doi.org/10.21063/jtm.2021.v11.i1.61-73)

Correspondence should be addressed to adriyan0686@gmail.com

Copyright © 2021 Adriyan. This is an open access article distributed under the [CC BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Article Information

Received:
March 25, 2021
Revised:
April 26, 2021
Accepted:
April 27, 2021
Published:
April 30, 2021

Abstract

This article addressed a machine learning approach for determining a solution model for the direct kinematic problems of parallel manipulators. A 5R planar parallel manipulator was utilized for that approach because it had the solution in the closed form. Initially, a dataset was created from an inverse kinematic solution of the manipulator for one of its assembly modes. Then, this dataset was fed as the input (the joint space) and the output (the platform space) for modeling the direct kinematic solution of the manipulator using one of the machine learning algorithms, which was the decision tree. To avoid overfitting during the training, hyperparameter tuning was employed to find the best parameters for the decision tree model, which was later called the best model. Hence, the best model can be validated by using the closed form solution. If the best model failed to model the direct kinematic solution in the validation, remodeling had to be performed and executed the same steps again. For remodeling, the training dataset consisted of all assembly modes of the manipulator. Consequently, the best model after remodeling was able to present the direct kinematic solutions for all possible input domains. Unfortunately, around 5% of solutions shown a higher deviation which had to be investigated in the future.

Keywords: Direct kinematic solution, 5R planar parallel manipulator, closed-form solution, decision tree algorithm, hyperparameter tuning.

1. Pendahuluan

Permasalahan kinematika langsung didefinisikan dengan diberikannya nilai sambungan aktif sebagai *joint space*-nya atau disebut juga *actuator space*, tentukanlah posisi *end-effector space* atau *platform space*. Sementara itu, permasalahan kinematika invers merupakan persoalan sebaliknya. Secara umum, persoalan kinematika langsung dan invers merupakan pemetaan antara *joint space* dan *end-effector/platform space* atau sebaliknya.

Penentuan jawab kinematika langsung (*the solution of the direct/forward kinematic problems*) suatu manipulator paralel cukup rumit. Tidak semua rantai kinematika paralel manipulator memiliki jawab tertutup (*closed form solutions*) atau jawab matematika eksak untuk persoalan kinematika langsungnya. Ketika jawab tertutup suatu manipulator paralel tidak dapat ditemui, langkah seperti penggunaan metode numerik sering dijumpai [1]–[3]. Metode lainnya seperti metode heuristik juga sudah diimplementasikan seperti

continuous ant colony optimization [4] dan algoritma genetik [5].

Penerapan algoritma-algoritma pembelajaran mesin (*machine learning algorithms*) juga telah dilakukan untuk penentuan jawab kinematika langsung manipulator paralel. Algoritma-algoritma ini diterapkan dengan menggunakan data *joint space* dan *end-effector/platform space* manipulator paralel berdasarkan jawab kinematika inversnya. Penggunaan *artificial neural network* (jaringan syaraf tiruan) sebagai salah satu bagian dari pembelajaran mesin cukup banyak dilakukan dalam memodelkan jawab kinematika langsung manipulator paralel [6]–[9]. Sementara itu, penggunaan algoritma-algoritma pembelajaran mesin seperti *decision tree*, *support vector machines*, *random forest* belum ditemui untuk penentuan jawab kinematika langsung. Penggunaannya hanya untuk jawab kinematika invers manipulator paralel seperti yang dibahas oleh [10], [11].

Pembahasan yang cukup mendalam dilakukan oleh Lopez, dkk [6] menggunakan jaringan syaraf tiruan dalam memodelkan jawab kinematika langsung manipulator paralel bidang 5R. Dataset dari jawab kinematika invers diperoleh dengan mendiskritisasi *workspace* secara merata. Penggunaan jawab kinematika invers untuk berbagai modus rakitan manipulator paralel sebagai dataset untuk melatih dan menguji model tidak dibahas di dalam artikel penelitian mereka.

Selanjutnya, proses *training* dan *testing* jaringan syaraf yang dibangun tidak dijelaskan secara terperinci, karena proses *training* dan *testing* untuk pembangunan model dalam pembelajaran mesin cukup penting. Di dalam tulisannya hanya dideskripsikan proses *training* tanpa adanya proses *testing*. Penerapan proses *training* tanpa *testing* dikhawatirkan dapat membawa model menjadi *overfitting* [12]. Sementara itu, validasi hanya dipergunakan untuk beberapa titik saja yang mungkin tidak dapat merepresentasikan keseluruhan jawab kinematika langsung manipulator paralel 5R di dalam *workspace*-nya.

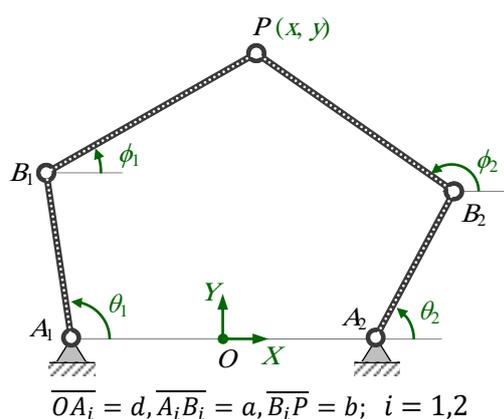
Tulisan ini akan menyajikan suatu kajian terkait penerapan pembelajaran mesin (*machine learning*) untuk memodelkan jawab kinematika langsung manipulator paralel berdasarkan kekurangan dan keterbatasan yang telah disajikan oleh penelitian sebelumnya. Untuk itu, manipulator paralel 5R [13], [14] dipilih dan digunakan dalam penelitian ini. Pemilihan dan penggunaan disebabkan oleh adanya jawab tertutup untuk persoalan kinematika langsung

manipulator paralel ini. Model untuk jawab kinematika langsung dengan pembelajaran mesin dapat dibuat dengan menggunakan dataset yang bersumber dari jawab kinematika invers manipulator ini. Model ini dibangun dengan menggunakan algoritma *decision tree*.

Decision tree merupakan salah satu algoritma yang digunakan untuk menyelesaikan persoalan tak-linier antara hubungan *input*/fitur/variabel bebas dan *output*/target/respons/variabel bergantung [12], [15]. Algoritma ini menerapkan pemisahan suatu dataset mejadi sub-dataset berdasarkan kriteria *if-then* [12]. Setelah proses latihan (*training*) selesai, model dari pembelajaran mesin ini dapat dibandingkan keakuratannya dengan jawab tertutup. Di samping itu, telaah terkait kemampuan model pembelajaran mesin dalam memprediksi jawaban tertutup manipulator juga akan dibahas.

2. Material dan Metode

Penelitian yang disajikan melalui tulisan ini dilaksanakan dengan menurunkan jawab analitik kinematika langsung manipulator paralel bidang 5R. Jawab analitik ini akan dijadikan sebagai acuan dari jawab pendekatan permasalahan kinematika langsung dengan menerapkan salah satu algoritma pembelajaran mesin, yaitu algoritma *decision tree*. Konfigurasi manipulator paralel bidang 5R yang digunakan ditunjukkan oleh Gambar 1. Nilai konstanta kinematik a , b , dan d yang digunakan masing-masingnya yaitu 170 mm, 320 mm, dan 110 mm.



Gambar 1. Manipulator paralel bidang 5R.

Pada manipulator ini θ_1 dan θ_2 merupakan sambungan aktif sedangkan ϕ_1 dan ϕ_2 merupakan sambungan pasif. Titik $P(x, y)$ menyatakan posisi dari *end-effector* manipulator ini. Dalam penerapan pembelajaran mesin dibutuhkan suatu dataset yang terdiri dari

variabel bebas (fitur) dan variabel bergantung (target). Untuk permasalahan kinematika langsung ini, fitur yang digunakan yaitu sambungan aktif sebagai *joint space*-nya, θ_1 dan θ_2 . Sementara, targetnya merupakan posisi titik $P(x, y)$ sebagai *end-effector space*.

Dataset ini dapat dengan mudah dihasilkan melalui jawab kinematika invers manipulator paralel bidang 5R di dalam *useful workspace*-nya yang berada di atas sumbu x . Pada tahap awal akan digunakan salah satu modus rakitan (*assembly mode*) dari manipulator ini, yaitu modus (+,-). Dengan demikian, untuk kasus ini akan dibuat tiga buah dataset dengan jumlah baris masing-masingnya berbeda, yaitu sebesar 1000, 10000, dan 100000 baris. Tentunya setiap baris di ketiga dataset dihasilkan dengan mengambil secara acak posisi titik P di dalam *useful workspace*-nya.

Ketiga dataset ini disusun masing-masingnya melalui pengurutan fitur dan target, yaitu θ_1 ; θ_2 ; x ; dan y , pada dataset. Untuk proses *training* (pelatihan) dan *testing* (pengujian), dataset ini dibagi dengan proporsi 90% : 10%. Proses *training* (pelatihan) dataset dilakukan dengan menerapkan *hyperparameter tuning* terhadap parameter algoritma *decision tree* melalui penggunaan *grid search cross validation* untuk 4-fold. Hal ini ditujukan untuk mencegah terjadi *overfitting* pada model jawab kinematika langsung yang dibangun dengan algoritma *decision tree*.

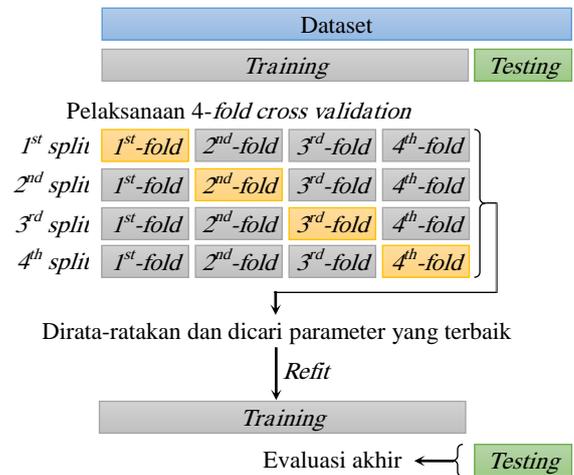
Hyperparameter algoritma *decision tree* yang dimaksud untuk proses *tuning* ini yaitu kedalaman maksimum tree (*maximum depth of the tree* disingkat *max_depth_tree*), jumlah minimum sampel cabang untuk dipisahkan (*minimum number of samples split* disingkat dengan *min_samples_split*), dan jumlah minimum daun diakhir pemisahan (*minimum number of samples leaf* yang disingkat *min_samples_leaf*). Penggunaan ketiga parameter ini untuk mengatasi *overfitting* model dikenal juga dengan nama *pre-pruning*. Nilai untuk ketiga parameter ini ditunjukkan oleh Tabel 1.

Tabel 1. Nilai-nilai yang digunakan dalam proses *pre-pruning* model jawab kinematika langsung manipulator paralel 5R dengan algoritma *decision tree*.

Parameter	Nilai
<i>max_depth_tree</i>	[8, 12, 16, 20, 24, 28, 32, None]
<i>min_samples_split</i>	[2, 4, 6, 8, 10]
<i>min_samples_leaf</i>	[1, 2, 3, 4, 5]

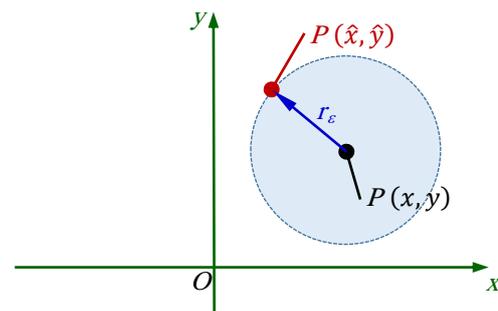
Penerapan 4-fold *cross validation* pada penggunaan *hyperparameter tuning* melalui

grid search diilustrasikan pengaplikasiannya seperti pada Gambar 2. Penerapan ini dilakukan untuk setiap dataset dari ketiga dataset yang digunakan. Setiap dataset akan dilatih melalui 4-fold *cross validation* ini dengan untuk setiap pasangan ketiga nilai *hyperparameter* seperti yang telah didefinisikan pada Tabel 1.



Gambar 2. Proses pembangunan model berdasarkan dataset yang digunakan dengan menerapkan 4-fold *cross validation* untuk setiap pasangan *hyperparameter* pada Tabel 1.

Berdasarkan nilai-nilai ketiga parameter pada Tabel 1 terdapat 200 pasangan nilai-nilai *hyperparameter* yang akan digunakan dalam *training*. Pasangan ketiga parameter terbaik dalam proses *training* nantinya yang akan digunakan untuk evaluasi akhir melalui bagian dataset untuk *testing*.



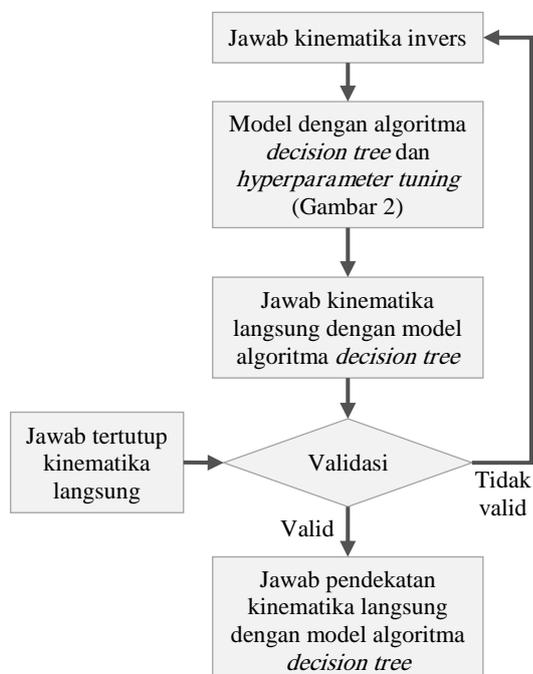
Gambar 3. *Radius error* sebagai metrik untuk menentukan sejauh mana posisi titik P yang diprediksi (\hat{x}, \hat{y}) terdeviasi dari posisi sebenarnya (x, y) .

Proses *training* yang dilakukan ditujukan dengan meminimumkan *loss* yang terjadi melalui penerapan nilai rerata dari galat radius (*radius error*) yang terjadi. *Radius error* ini diilustrasikan oleh Gambar 3. Penggunaan *radius error* ini disandarkan pada kenyataan bahwa nilai posisi P yang diprediksi (\hat{x}, \hat{y}) nantinya menyimpang dari nilai posisi P yang sebenarnya di dalam dataset, (x, y) . *Radius*

error ini, dalam lain kata, merupakan metrik untuk menentukan kepresisian hasil prediksi terhadap nilai sebenarnya posisi titik P manipulator. Secara matematis, *radius error* dideskripsikan oleh

$$r_\varepsilon = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2}. \quad (1)$$

Melalui proses *hyperparameter tuning* atau *pre-pruning decision tree* menggunakan dataset bagian *training* dan *testing* dapat diperoleh model terbaik. Model terbaik ini dapat disimpan untuk digunakan dalam menentukan jawab kinematika manipulator paralel bidang 5R dengan konstanta kinematik yang telah ditentukan. Selanjutnya, jawab pendekatan melalui model berbasis *decision tree* yang sudah disimpan dapat dibandingkan dengan jawab analitik. Jika ada terdapat kelemahan model maka akan dilakukan proses pemodelan ulang. Pemodelan ulang dilakukan dengan mengeksplorasi keseluruhan jawab invers kinematik di dalam *useful workspace* untuk berbagai modus rakitan manipulator ini. Modus rakitan yang dimaksud adalah (+, -); (-, +); (+, +); dan (-, -) [13], [14]. Keseluruhan proses yang dilakukan dapat dilihat alurnya secara ringkas melalui Gambar 4.



Gambar 4. Ilustrasi skematik penentuan jawab pendekatan manipulator paralel bidang 5R menggunakan pembelajaran mesin dengan algoritma *decision tree*.

Proses komputasi jawab kinematika invers, jawab kinematika langsung secara analitik, dan

jawab pendekatan dengan algoritma *decision tree* dilakukan dengan menggunakan python 3.8.8 dan pustaka python yang terkait. Pustaka yang dimaksud diantaranya adalah NumPy 1.19.5 [16], pandas 1.2.4 [17], [18], scikit-learn 0.23.2 [19], matplotlib 3.3.4 [20], seaborn 0.11.1 [21], dan JupyterLab 3.0.14 [22]. Python dan pustakanya ini dijalankan pada *notebook* PC dengan prosesor Intel i5 8250U 4 inti 8 untaian (*clock speed* maksimum 3.4 GHz) dan memori 16 GB. Seluruh proses komputasi menggunakan keseluruhan inti dan untaian prosesor.

3. Hasil dan Pembahasan

Bagian ini menyajikan jawab kinematika langsung mekanisme paralel bidang 5R melalui jawab analitik dan jawab pendekatan melalui algoritma *decision tree* pembelajaran mesin.

A. Jawab Analitik Kinematika Langsung Manipulator Paralel Bidang 5R

Penentuan jawab analitik manipulator paralel bidang 5R dilakukan dengan mengacu pada Gambar 1. Posisi titik P dari titik O untuk kedua kaki (*limb*) manipulator mulai dari kiri ke kanan dapat dinyatakan secara vektorial sebagai

$$\overrightarrow{OP} = \overrightarrow{OA_i} + \overrightarrow{A_iB_i} + \overrightarrow{B_iP}; \quad i = 1, 2; \quad (2)$$

dengan $\overrightarrow{OP} = (x \ y)^T$, $\overrightarrow{OA_i} = ((-d)^i \ 0)^T$, $\overrightarrow{A_iB_i} = (a \cdot c\theta_i \ a \cdot s\theta_i)^T$, dan $\overrightarrow{B_iP} = (a \cdot c\phi_i \ a \cdot s\phi_i)^T$, serta $c\theta_i, s\theta_i, c\phi_i, s\phi_i$ masing-masingnya merupakan cos dan sin untuk sudut θ_i dan ϕ_i . Dengan mengeliminasi sudut sambungan pasif ϕ_i diperoleh

$$(x + \zeta_i)^2 + (y - \eta_i)^2 = b^2; \quad i = 1, 2; \quad (3)$$

dengan $\zeta_i = -(-c)^i - a \cdot c\theta_i$ dan $\eta_i = a \cdot s\theta_i$.

Persamaan (3) diselesaikan untuk x dan y . Awalnya, persamaan (3) dinyatakan sebagai fungsi dari x , dengan y diperlakukan sebagai suatu konstanta. Dengan demikian diperoleh

$$x^2 + 2\zeta_i x + D_i = 0; \quad i = 1, 2; \quad (4)$$

dengan $D_i = \zeta_i^2 + (y - \eta_i)^2 - b^2$. Persamaan (4) dapat diselesaikan secara langsung untuk x yang menghasilkan

$$x = \frac{D_2 - D_1}{2(\zeta_1 - \zeta_2)} = \frac{\eta_1 - \eta_2}{\zeta_1 - \zeta_2} y + \frac{-\eta_1^2 + \eta_2^2 - \zeta_1^2 + \zeta_2^2}{2(\zeta_1 - \zeta_2)}. \quad (5)$$

Persamaan (5) menunjukkan bahwa x bergantung pada y .

Akhirnya, dengan menggunakan persamaan (4) dan menerapkan metode eliminasi dialitik Sylvester dapat dibentuk

$$\begin{bmatrix} 1 & 2\zeta_1 & D_1 & 0 \\ 1 & 2\zeta_2 & D_2 & 0 \\ 0 & 1 & 2\zeta_1 & D_1 \\ 0 & 1 & 2\zeta_2 & D_2 \end{bmatrix} \begin{Bmatrix} x^3 \\ x^2 \\ x^1 \\ x^0 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}. \quad (6)$$

Jawab trivial persamaan (6) diberikan oleh determinan matriks 4×4 pada ruas kanan persamaan (6), yaitu

$$E_2 y^2 + E_1 y + E_0 = 0, \quad (7)$$

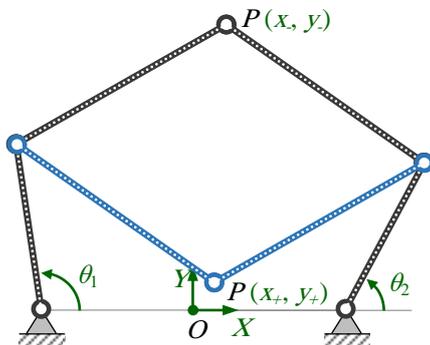
dengan

$$\begin{aligned} E_2 &= -4[(\eta_1 - \eta_2)^2 + (\zeta_1 - \zeta_2)^2], \\ E_1 &= 4(\eta_1 + \eta_2)[(\eta_1 - \eta_2)^2 + (\zeta_1 - \zeta_2)^2], \\ E_0 &= -(\eta_1^2 - \eta_2^2)^2 - 2(\eta_1^2 + \eta_2^2 - 2b^2)(\zeta_1 - \zeta_2)^2 - (\zeta_1 - \zeta_2)^4. \end{aligned}$$

Persamaan (7) ini dapat diselesaikan untuk y , yang menghasilkan

$$y = \frac{-E_1 \pm \sqrt{E_1^2 - 4E_0E_2}}{2E_2}. \quad (8)$$

Dengan demikian, persamaan (5) dan (8) merupakan jawab tertutup atau analitik permasalahan kinematika langsung mekanisme paralel bidang 5R.



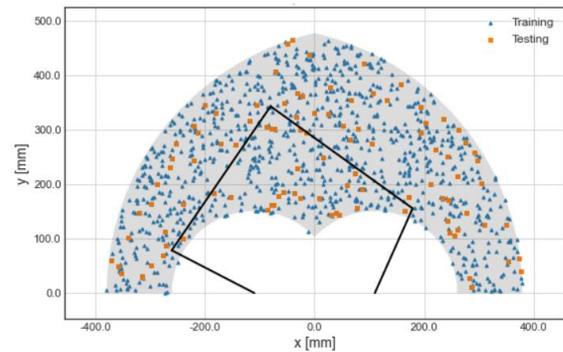
Gambar 5. Ilustrasi salah satu pose manipulator paralel bidang 5R untuk jawab kinematika langsung (x, y) dan (x_+, y_+) berdasarkan pasangan θ_1 dan θ_2 .

Untuk θ_1 dan θ_2 yang diberikan akan menghasilkan dua jawab kinematika langsung berupa (x, y) dan (x_+, y_+) seperti yang diilustrasikan oleh Gambar 5. Nilai y dan y_+

adalah jawab yang diberikan oleh persamaan (8) untuk nilai negatif dan positif nilai di bawah tanda akar kuadrat. Sementara itu, nilai x dan x_+ merupakan jawab dari persamaan (5) yang masing-masingnya setelah y dan y_+ disubstitusikan.

B. Jawab Pendekatan Kinematika Langsung dengan Decision Tree

Langkah pertama yang dilakukan adalah menyiapkan dataset yang dihasilkan dari jawab penyiapan dataset yang dihasilkan dari jawab invers kinematik di dalam *useful workspace* manipulator 5R ini. Jumlah titik-titik di dalam *useful workspace* yang digunakan berdasarkan dataset yang telah didefinisikan. Sebagai contoh pada Gambar 6 diperlihatkan sebaran dataset dengan 1000 titik pada *useful workspace* yang digunakan untuk proses *training* dan *testing* model jawab pendekatan menggunakan algoritma *decision tree*.



Gambar 6. Sebaran titik-titik di dalam *useful workspace* untuk dataset 1000 titik/baris dalam penentuan model jawab pendekatan menggunakan algoritma *decision tree*, termasuk proporsi bagian *training* dan *testing*. Konfigurasi manipulator 5R untuk salah satu posisi titik P juga ditunjukkan.

Penerapan *4-fold cross validation* (lihat Gambar 1) pada penggunaan *hyperparameter tuning* melalui *grid search* (lihat Tabel 1) menghasilkan nilai terbaik ketiga parameter yang dilakukan *tuning*. Nilai terbaik ini merupakan *score* terbaik dari pasangan nilai ketiga parameter melalui penerapan *4-fold cross validation*. *Score* ini merupakan nilai rerata \pm standar deviasi penerapan metrik *radius error* untuk dataset *training*. Ketiga parameter pada Tabel 1 yang optimum diberikan dalam Tabel 2 untuk masing-masing ketiga dataset yang digunakan.

Berdasarkan Tabel 2 ini dapat diketahui bahwa jumlah titik/baris data pada dataset mempengaruhi *radius error* dari model yang dibangun (lihat kolom *score*). Idealnya, model terbaik merupakan model yang dilatih dengan data tidak berhingga. Parameter *min_samples_split* dan *min_samples_leaf*

terbaik bernilai sama untuk keseluruhan model terbaik dari ketiga dataset yang digunakan. Sementara, untuk parameter *max_tree_depth* terbaik bervariasi dari 16 hingga 24 untuk model-model ketiga dataset tersebut. Model dengan dataset 1000, 5000, dan 10000 titik/baris memiliki nilai parameter *max_tree_depth* yang sama, yaitu 16.

Tabel 2. Nilai terbaik ketiga parameter yang dilakukan proses *pre-pruning* untuk setiap dataset, berikut dengan *score* bagian *training* dataset.

Dataset	Score	<i>max_</i> <i>depth</i> <i>_tree</i>	<i>min_</i> <i>samples</i> <i>_split</i>	<i>min_</i> <i>samples</i> <i>_leaf</i>
1000	13.602 ± 0.904	16	2	1
10000	4.475 ± 0.074	16	2	1
100000	1.404 ± 0.004	24	2	1

Struktur dari *decision tree* terbaik yang ditunjukkan oleh Tabel 2 tidak dapat disajikan di dalam tulisan ini mengingat pohon keputusan yang dihasilkan sangat dalam dan bercabang. Akan tetapi struktur dari *decision tree* terbaik ini disimpan ke dalam berkas *.dot graphviz. Dengan mengacu pada ketiga parameter terbaik pada Tabel 2 dapat dilakukan prediksi untuk evaluasi akhir menggunakan dataset bagian *testing* untuk masing-masing dataset yang ada. Hasil prediksi untuk dataset bagian *testing* ini ditampilkan dalam Tabel 3. Melalui Tabel 3 ini ditampilkan statistik deskriptif dari *radius error* prediksi dataset bagian *testing*.

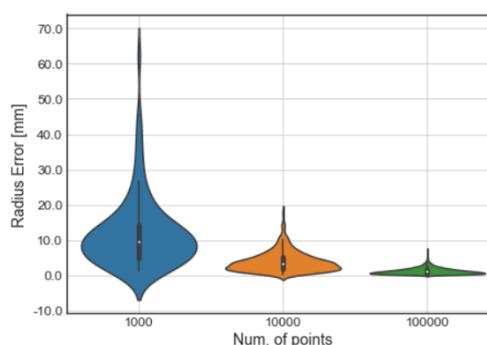
Tabel 3. Statistik deskriptif *radius error* [mm] hasil prediksi dataset bagian *testing*.

Dataset	Minimum [mm]	Maksimum [mm]	Rerata [mm]	Standar deviasi [mm]
1000	1.033	62.179	11.983	10.024
10000	0.127	18.264	3.915	2.826
100000	0.004	7.374	1.209	0.864

Untuk lebih memudahkan, nilai statistik deskriptif ini divisualisasikan ke dalam plot biola (*violin plot*). Melalui plot biola ini dapat dilihat plot Box-Whisker berikut dengan distribusi *radius error*-nya. Distribusi *radius error* ini diplotkan secara simetris terhadap sumbu vertikal. Distribusi yang dimaksud ini ditentukan melalui estimator kepadatan kernel (*kernel density estimator* yang disingkat *kde*) dari hasil perhitungan *radius error*.

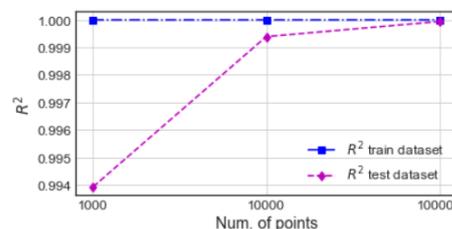
Gambar 7 dan Tabel 3 ini menunjukkan bahwa hasil prediksi yang dilakukan pada dataset bagian *testing* mendekati nol seiring dengan besarnya jumlah titik/baris dataset yang

digunakan dalam pelatihan. Untuk keseluruhan dataset terlihat bahwa mayoritas *radius error* berada disekitar nilai reratanya. Idealnya, rerata *radius error* adalah bernilai nol yang menunjukkan hasil prediksi identik dengan nilai sebenarnya. Ketiga dataset ini menunjukkan sebaran *radius error* hasil yang serupa yaitu nilai rerata yang lebih dekat ke nilai rerata ideal. Dengan demikian, *radius error* ini mengikuti distribusi Rayleigh.



Gambar 7. Plot biola nilai statistik deskriptif dan distribusi *radius error* hasil prediksi dataset bagian *testing* berdasarkan nilai terbaik *hyperparameter* (Tabel 2).

Selanjutnya, nilai koefisien determinasi regresi, R^2 , dapat digunakan untuk melihat seberapa baik *decision tree* memodelkan jawab kinematika langsung manipulator paralel bidang 5R. Plot kedua nilai R^2 untuk dataset bagian *training* dan *testing* diperlihatkan pada Gambar 8. Untuk masing-masing dataset dari ketiga dataset yang digunakan dapat diketahui bahwa model jawab kinematika langsung yang dibangun dengan *decision tree* memberikan hasil yang baik. Kondisi *overfit* maupun *underfit* tidak terjadi karena perbedaan nilai R^2 antara hasil *training* dan *testing* sangat kecil di setiap ketiga dataset (< 0.006).

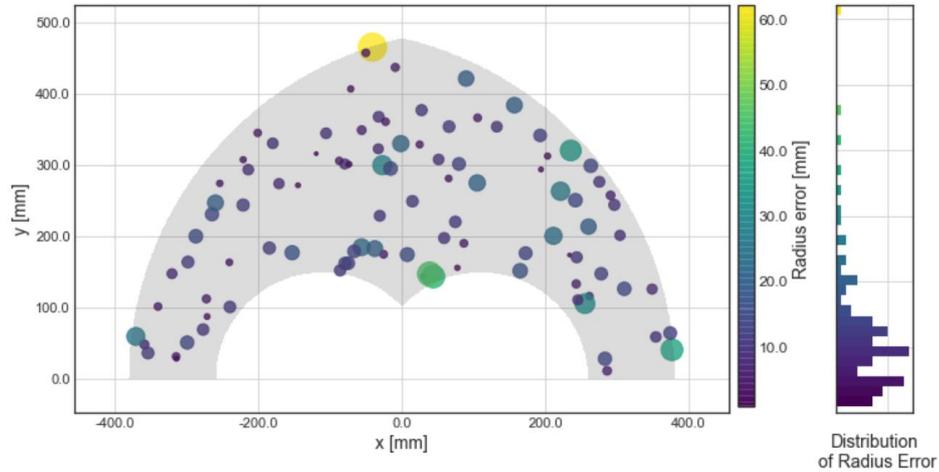


Gambar 8. Nilai koefisien determinasi R^2 untuk hasil *fitting* dan prediksi masing-masing dataset bagian *training* dan *testing*.

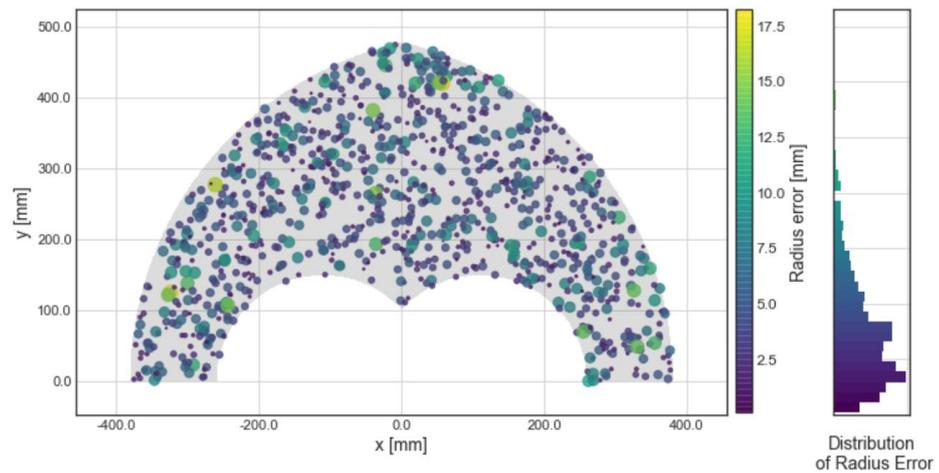
Kemudian, hasil prediksi dataset bagian *testing* untuk ketiga dataset dapat ditampilkan di dalam *useful workspace*-nya. Ketiga visualisasi ini ditunjukkan oleh Gambar 9-11 melalui *scatter plot* di dalam *useful workspace*. Untuk setiap dataset, besarnya *radius error* ditunjukkan dengan besarnya lingkaran pada

setiap gambar. Melalui ketiga gambar dapat diketahui di posisi mana titik P hasil prediksi memiliki *radius error* terkecil hingga terbesar. Secara umum, tidak ada bagian wilayah tertentu

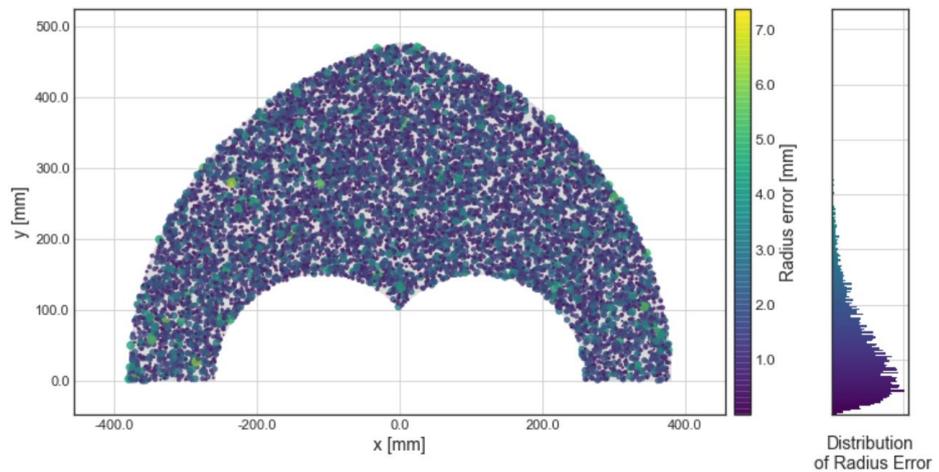
di dalam *useful workspace* yang menunjukkan adanya *radius error* yang terkelompok secara spesifik. Besar atau kecilnya *radius error* tersebar secara acak di dalam *useful workspace*.



Gambar 9. Prediksi posisi titik $P(\hat{x}, \hat{y})$ berdasarkan dataset bagian *testing* (10% dari dataset 1000 titik). Ukuran lingkaran dan warna menunjukkan besar atau kecilnya *radius error* setiap titik. Histogram sebelah kanan menunjukkan distribusi *radius error* dalam nilai jangkauannya yang diberikan pada sumbu vertikal.



Gambar 10. Prediksi posisi titik $P(\hat{x}, \hat{y})$ berdasarkan dataset bagian *testing* (10% dari dataset 10000 titik). Ukuran lingkaran dan warna menunjukkan besar atau kecilnya *radius error* setiap titik. Histogram sebelah kanan menunjukkan distribusi *radius error* dalam nilai jangkauannya yang diberikan pada sumbu vertikal.



Gambar 11. Prediksi posisi titik $P(\hat{x}, \hat{y})$ berdasarkan dataset bagian *testing* (10% dari dataset 100000 titik). Ukuran lingkaran dan warna menunjukkan besar atau kecilnya *radius error* setiap titik. Histogram sebelah kanan menunjukkan distribusi *radius error* dalam nilai jangkauannya yang diberikan pada sumbu vertikal.

Akhirnya, model jawab kinematika langsung manipulator paralel bidang 5R dengan algoritma *decision tree* telah berhasil diperoleh. Dari ketiga model jawab pendekatan dengan *decision tree* yang didasari pada jumlah dataset dapat diambil model terbaik yaitu model dengan nilai *radius error* yang terendah. Dengan demikian, model yang dilatih dan diuji dengan 100000 titik/baris data ini dapat disimpan untuk digunakan nantinya dalam menentukan jawab kinematika langsung manipulator paralel bidang 5R.

Model yang telah dibangun ini hanya terbatas untuk manipulator paralel bidang 5R dengan konstanta kinematik yang telah disebutkan sebelumnya, yaitu $a = 170$ mm, $b = 320$ mm, dan $d = 110$ mm. Untuk nilai konstanta kinematik lainnya perlu dilakukan kembali *training* dan *testing* untuk memperoleh model yang terbaik.

C. Komparasi Jawab Analitik dan Jawab Pendekatan untuk Kinematika Langsung Manipulator Paralel Bidang 5R

Berdasarkan jawab analitik yang telah diberikan oleh persamaan (5) dan (8) dapat dihasilkan sejumlah titik di dalam *useful workspace*. Untuk itu, jawab analitik dibuat untuk 5000 titik di *end-effector space* (target) berdasarkan nilai acak *joint space* (fitur). Komparasi ini ditujukan untuk memvalidasi jawab pendekatan yang telah dibangun dengan algoritma *decision tree*. Tentunya, komparasi ini dinyatakan kembali dalam metrik *radius error*.

Jika ditilik kembali bahwa jawab analitik memiliki 2 pasang posisi titik P di dalam *workspace*, yaitu (x, y) dan (x_+, y_+) . Sementara, jawab pendekatan yang dibangun dengan algoritma *decision tree* dilatih dengan salah satu pasang target berdasarkan hasil jawab kinematika invers. Untuk itu, jawab kinematika invers 10000 titik akan dicek terlebih dahulu manakah di antara titik-titik tersebut merupakan pasangan jawab kinematika langsung (x, y) atau (x_+, y_+) . Hasil pengecekan ini menunjukkan bahwa terdapat 99406 titik (99.4060 %) yang merupakan pasangan jawab (x, y) . Sementara itu, sisanya yaitu 594 titik (0.5940 %) merupakan pasangan jawab (x_+, y_+) .

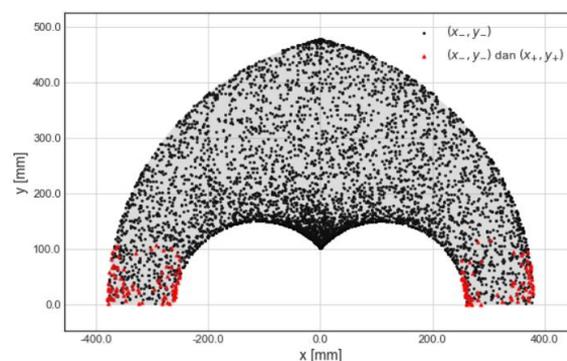
Hasil yang ditunjukkan ini memperlihatkan bahwa di dalam *useful workspace* yang berada di atas sumbu- x (Gambar 6) masih terdapat sekitar 0.5% jawab kinematik langsung di pasangan lainnya. Kondisi seperti dapat dicek kembali untuk ketiga dataset yang digunakan seperti yang diperlihatkan oleh Tabel 4. Secara

umum terlihat bahwa pasangan jawab kinematika langsung (x_+, y_+) hanya memiliki kontribusi kurang dari 1 % di dalam *useful workspace* untuk kasus ini.

Tabel 4. Jumlah (dan persentase) jawab kinematika langsung di dalam jawab kinematika invers yang digunakan dalam dataset untuk penentuan jawab pendekatan algoritma *decision tree*.

Dataset	Pasangan jawab (x, y)	Pasangan jawab (x_+, y_+)
1000	997 (99.7000 %)	3 (0.3000 %)
5000	4965 (99.3000 %)	35 (0.7000 %)
10000	9932 (99.3200 %)	68 (0.6800 %)
50000	49704 (99.4080 %)	296 (0.5920 %)
100000	99406 (99.4060 %)	594 (0.5940 %)

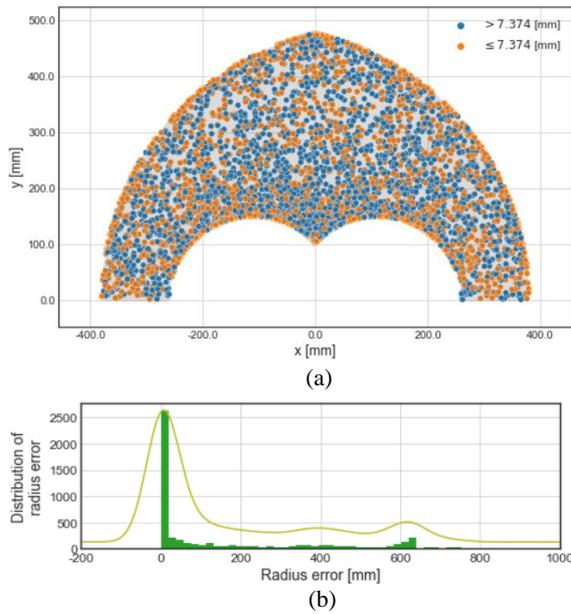
Penentuan 5000 titik jawab kinematika langsung berdasarkan persamaan (5) dan (8) menghasilkan pasangan jawab sebanyak 5000 titik untuk (x, y) dan 178 titik untuk (x_+, y_+) di dalam *useful workspace* di atas sumbu- x . Jadi, di dalam *useful workspace* ini ada 178 titik nilai *actuator space* yang memiliki jawab kinematika langsung untuk kedua pasang (x, y) dan (x_+, y_+) . Sebaran 5000 titik ini di dalam *useful workspace* diperlihatkan oleh Gambar 12.



Gambar 12. Sebaran titik-titik di dalam *useful workspace* untuk dataset 5000 titik/baris jawab tertutup kinematika langsung. Titik-titik ini digunakan untuk validasi model jawab pendekatan menggunakan algoritma *decision tree* yang telah dibangun.

Penggunaan model jawab kinematika langsung menggunakan algoritma *decision tree* untuk dataset 100000 menghasilkan hasil prediksi seperti yang ditunjukkan oleh Gambar 13. Hasil prediksi posisi titik $P(\hat{x}, \hat{y})$ dapat dikelompokkan atas dua, yaitu hasil prediksi posisi titik $P(\hat{x}, \hat{y})$ yang memiliki nilai *radius error*: (1) kurang atau sama dari nilai maksimum yang diberikan oleh Tabel 3 (7.374 mm); dan (2) melebihi nilai maksimum yang diberikan oleh Tabel 3 (7.374 mm). Melalui Gambar 12(a) ini ditunjukkan bahwa sebaran kedua

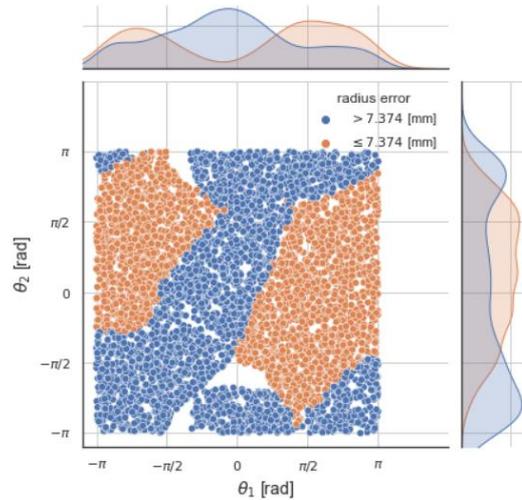
kelompok hasil prediksi berjumlah dan menyebar hampir berimbang.



Gambar 13. (a) Prediksi posisi titik $P(\hat{x}, \hat{y})$ berdasarkan dataset validasi (5000 titik) menggunakan pasangan jawab kinematika langsung (x, y) , yang dikelompokkan berdasarkan *radius error* dan (b) distribusi *radius error* hasil prediksi posisi titik $P(\hat{x}, \hat{y})$.

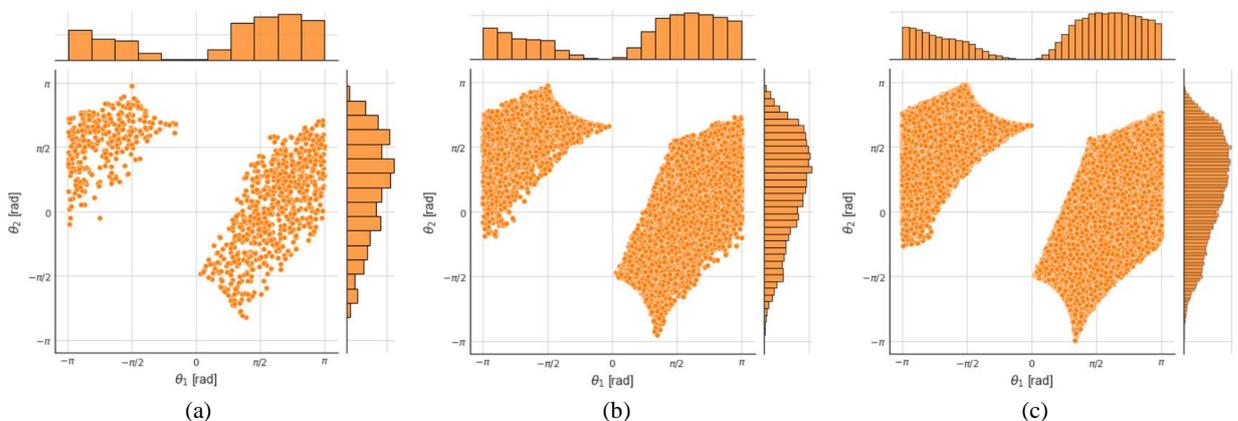
Sementara itu, distribusi *radius error* yang ditunjukkan oleh Gambar 13 (b) menunjukkan bahwa kelompok (1) memiliki jumlah hasil prediksi yang sedikit lebih banyak dibandingkan dengan kelompok (2). Secara umum, kondisi yang ditunjukkan oleh Gambar 13 memberikan jawaban bahwa algoritma yang telah dibangun tidak dapat dengan baik memodelkan jawaban kinematika langsung berdasarkan dataset jawab kinematika invers, walaupun dataset sudah memiliki jumlah titik yang cukup banyak (100000 titik). Untuk itu

diperlukan pemetaan dan distribusi data dari pasangan fitur (θ_1 dan θ_2) untuk dataset validasi ini yang kemudian dikelompokkan berdasarkan *radius error* (≤ 7.374 mm dan > 7.374 mm).



Gambar 14. Pemetaan pasangan fitur (θ_1 dan θ_2) dataset validasi berdasarkan *radius error* (≤ 7.374 mm dan > 7.374 mm) berikut dengan distribusinya.

Pemetaan pasangan fitur (θ_1 dan θ_2) dataset validasi berdasarkan *radius error* (≤ 7.374 mm dan > 7.374 mm) berikut dengan distribusi masing-masingnya ditunjukkan oleh Gambar 14. Di sini terlihat bahwa wilayah berwarna oranye yang memiliki *radius error* ≤ 7.374 mm. Wilayah ini merupakan wilayah dengan *radius error* maksimum hasil pemodelan yang telah dilakukan sebelumnya. Sementara, wilayah berwarna biru menunjukkan nilai *radius error* yang melebihi dari nilai maksimum *radius error* hasil pemodelan.

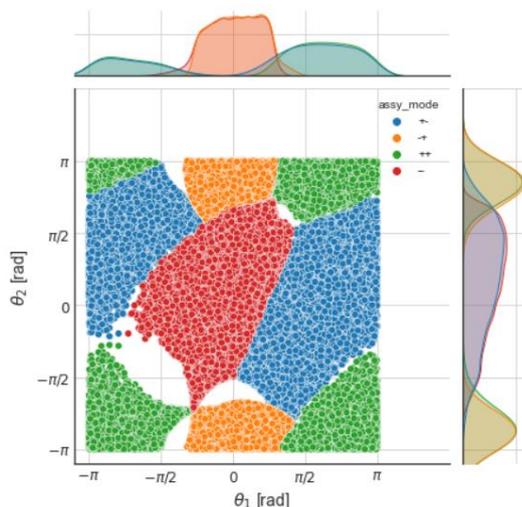


Gambar 15. Pemetaan pasangan fitur (θ_1 dan θ_2) dataset yang digunakan untuk membangun model jawab kinematika langsung (a) 1000 titik data, (b) 10000 titik data, dan (c) 100000 titik data.

Pengecekan pasangan fitur (θ_1 dan θ_2) untuk ketiga dataset jawab invers kinematik dapat menunjukkan kenapa terjadi *radius error* yang cukup besar di wilayah dengan pasangan fitur (θ_1 dan θ_2) yang berwarna biru. Untuk itu, pemetaan pasangan fitur (θ_1 dan θ_2) untuk ketiga dataset jawab invers kinematik ditampilkan pada Gambar 15. Pasangan fitur (θ_1 dan θ_2) di ketiga dataset yang ditunjukkan merupakan jawab kinematika invers untuk satu modulus rakitan saja, yaitu (+, -). Sementara, pasangan fitur (θ_1 dan θ_2) untuk dataset validasi terlihat bahwa hampir seluruh pasangan fitur berada dalam rentang $-\pi$ radian s/d π radian. Oleh sebab itu, pemodelan ulang akan dilakukan dengan tetap menggunakan dataset hasil jawab kinematika invers tetapi melibatkan keempat modulus rakitan manipulator paralel bidang 5R.

D. Pemodelan Ulang untuk Jawab Pendekatan Kinematika Langsung dengan *Decision Tree*

Untuk pemodelan ulang dilakukan dengan melibatkan jawaban kinematika invers untuk keempat modulus rakitan. Jumlah titik yang digunakan dalam dataset berjumlah 100000 titik dengan 25000 titik untuk setiap modulus rakitan. Pasangan fitur (θ_1 dan θ_2) dihasilkan secara acak di dalam *useful workspace*-nya. Pemetaan pasangan fitur ini berikut dengan distribusi masing-masing fitur yang dibedakan berdasarkan modulus rakitannya ditampilkan di dalam Gambar 16.



Gambar 16. Pemetaan pasangan fitur (θ_1 dan θ_2) berdasarkan jawab kinematika invers untuk keempat modulus rakitan untuk pemodelan ulang jawab kinematika langsung menggunakan algoritma *decision tree* untuk 100000 titik data.

Jika wilayah pasangan fitur (θ_1 dan θ_2) jawab kinematika invers (Gambar 16)

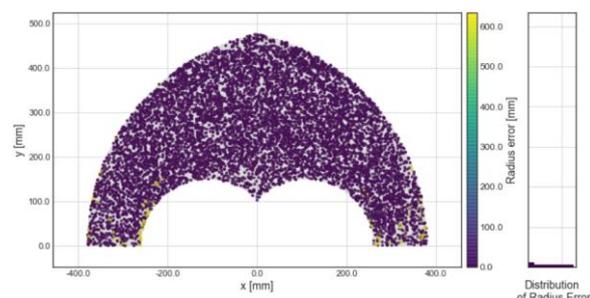
dibandingkan dengan pasangan fitur (θ_1 dan θ_2) untuk validasi menggunakan jawab tertutup kinematika langsung (Gambar 14) terdapat beberapa wilayah yang tidak terlindungi. Hal ini tentunya akan berakibat pada performansi model dalam validasi nantinya. Akan tetapi, wilayah fitur (θ_1 dan θ_2) dalam Gambar 16 telah hampir menyebar dalam rentang $-\pi$ radian s/d π radian yang mana ini lebih baik dibandingkan dengan sebaran fitur dataset untuk kondisi awal (Gambar 15).

Selanjutnya, proses untuk pemodelan dilakukan seperti pada bagian 3.B, yaitu penggunaan

- proporsi dataset untuk *training* : *testing* sebesar 90% : 10% (Gambar 2),
- 4-fold *cross validation* (Gambar 2),
- *pre-pruning* atau *hyperparameter tuning* sesuai dengan Tabel 1, dan
- metrik *radius error* (persamaan 1).

Proses ini diterapkan untuk algoritma *decision tree*.

Mengulangi langkah di dalam bagian 3.B yang ditunjukkan oleh Gambar 17 dapat divisualisasikan prediksi titik $P(\hat{x}, \hat{y})$ menggunakan dataset bagian *testing*. Secara umum *radius error* yang dihasilkan cukup kecil dengan nilai reratanya sebesar 9.793 mm. Namun, nilai rerata *radius error* dalam dataset untuk pemodelan ulang ini lebih besar dibandingkan dengan nilai maksimum *radius error* dataset untuk pemodelan awal (7.374 mm), lihat Tabel 3. Jika dicermati distribusi *radius error* ini melalui Gambar 17 terlihat bahwa mayoritas *radius error* berada di bawah 10 mm, atau dapat juga dinyatakan di bawah nilai reratanya.

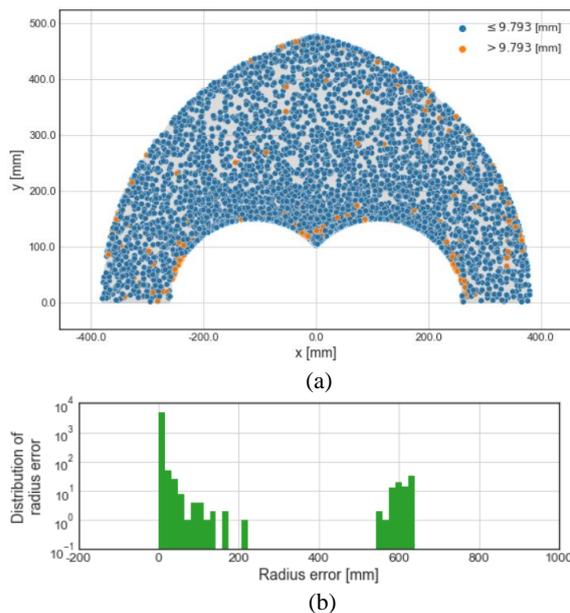


Gambar 17. Prediksi posisi titik $P(\hat{x}, \hat{y})$ berdasarkan dataset bagian *testing* (10% dari dataset 100000 titik). Histogram sebelah kanan menunjukkan distribusi *radius error* dalam nilai jangkauannya yang diberikan pada sumbu vertikal.

Inspeksi nilai prediksi titik $P(\hat{x}, \hat{y})$ dataset bagian *testing* dengan *radius error* lebih kecil atau sama dan lebih besar dibandingkan dengan reratanya diperoleh masing-masingnya sebesar 94.62% dan 5.38%. Adanya nilai *radius error*

yang sangat besar ini diakibatkan oleh adanya semacam lompatan pemisahan dalam algoritma *decision tree* ketika membuat percabangan. Kondisi ini dapat diakibatkan oleh penggunaan keempat modus rakitan di dalam dataset untuk *training* dan *testing*. Di samping itu, hal ini juga didukung dengan kenyataan yang diberikan oleh distribusi fitur θ_1 dan θ_2 pada Gambar 16 yang memperlihatkan adanya nilai yang tumpang tindih untuk modus rakitan yang berbeda.

Nilai *hyperparameter* terbaik untuk algoritma *decision tree* diberikan oleh *max_tree_depth* sebesar 28, serta nilai *min_samples_split* dan *min_samples_leaf* masing-masingnya sebesar 2 dan 1. Dua parameter terakhir masih sama nilainya dengan pemodelan awal (Tabel 2). Lebih besarnya nilai parameter *max_tree_depth* dalam proses pemodelan ulang ini merupakan dampak dari keadaan yang telah dipaparkan dalam paragraf sebelumnya. Selanjutnya, struktur dari *decision tree* terbaik ini disimpan ke dalam berkas *.dot graphviz mengingat ukuran strukturnya yang cukup besar.

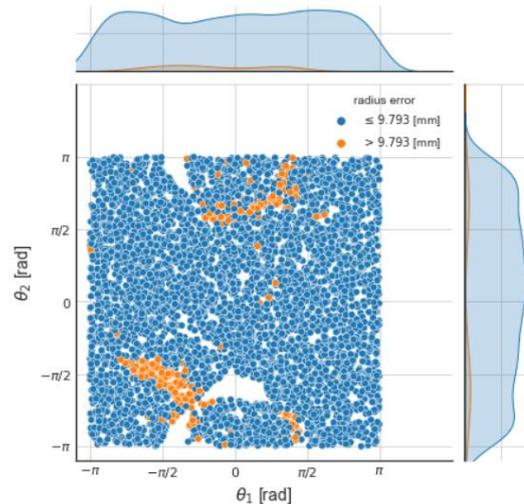


Gambar 18. (a) Prediksi posisi titik $P(\hat{x}, \hat{y})$ berdasarkan dataset validasi (5000 titik) menggunakan pasangan jawab kinematika langsung (x, y) , yang dikelompokkan berdasarkan *radius error* dan (b) distribusi *radius error* hasil prediksi posisi titik $P(\hat{x}, \hat{y})$.

Langkah selanjutnya yaitu melakukan validasi dengan menggunakan dataset yang masih sama, yaitu dataset jawab kinematika langsung seperti yang ditampilkan oleh Gambar 11. Hasil prediksi posisi titik $P(\hat{x}, \hat{y})$ dengan dataset validasi ini ditunjukkan oleh Gambar 18 (a). Dalam hal ini, nilai rerata *radius error* hasil *testing* (9.793 mm) dijadikan acuan untuk

mengelompokkan *radius error* hasil prediksi dataset validasi. Secara umum, algoritma *decision tree* yang telah dilatih dan diuji dengan dataset yang memiliki keempat modus rakitan menghasilkan mayoritas *radius error* hasil prediksi yang cukup kecil pada dataset validasi. Tentunya, pengamatan ini dilakukan dengan membandingkan dengan hasil prediksi dataset validasi yang hanya melibatkan satu modus rakitan saja yaitu (+, -).

Untuk hasil prediksi posisi titik $P(\hat{x}, \hat{y})$ berdasarkan dataset validasi terdapat 94.48% hasil prediksi dengan *radius error* berada di bawah nilai 9.793 mm. Sisanya yaitu 5.52% memiliki nilai *radius error* di atas 9.793 mm. Persentase kedua kelompok ini juga tergambar di dalam distribusi *radius error* melalui Gambar 18 (b), di sini sumbu vertikalnya memiliki skala logaritmik. Dengan demikian, pemodelan ulang ini membuktikan kekonsistenan model yang dibangun dengan mengacu pada hasil prediksi menggunakan dataset *testing* dan dataset validasi.



Gambar 19. Pemetaan pasangan fitur $(\theta_1$ dan $\theta_2)$ dataset validasi berdasarkan *radius error* (≤ 9.793 mm dan > 9.793 mm) berikut dengan distribusinya.

Gambar 19 memperlihatkan visualisasi pasangan fitur $(\theta_1$ dan $\theta_2)$ yang digunakan untuk menghasilkan jawab tertutup kinematika langsung (lihat Gambar 11) divisualisasikan berdasarkan kedua kelompok *radius error*. Wilayah yang ditandai dengan *radius error* > 9.793 mm (warna oranye) merupakan wilayah yang tidak memiliki pasangan θ_1 dan θ_2 ketika dilatih dan diuji (Gambar 16). Di samping itu, akibat tumpang tindihnya pasangan θ_1 dan θ_2 untuk keempat modus rakitan juga mengindikasikan kondisi yang serupa (*radius error* > 9.793 mm).

4. Simpulan

Telah dilakukan pemodelan jawab kinematika langsung manipulator paralel bidang 5R menggunakan algoritma *decision tree*. Pemodelan jawab pendekatan kinematika langsung dengan algoritma *decision tree* dilatih dan diuji menggunakan hasil jawab kinematika invers. Agar pemodelan jawab kinematika langsung dapat mencakup keseluruhan fitur atau *active joint space* dalam nilai rentang masing-masing fitur maka diperlukan jawab kinematika invers untuk seluruh modus rakitan. Pengecekan dapat dilakukan dengan melakukan visualisasi pasangan dua fitur melalui *scatter plot*.

Akan tetapi, penggunaan seluruh modus rakitan untuk dataset *training* dan *testing* masih menghasilkan *radius error* yang cukup besar untuk beberapa baris pasang fitur. Hal ini disebabkan masih adanya ruang kosong diantara pasangan fitur yang ditunjukkan melalui *scatter plot*. Penyebab dan penyelesaiannya akan diinvestigasi untuk penelitian mendatang. Penggunaan algoritma pembelajaran mesin lainnya seperti algoritma *ensemble* berupa *random forest*; dan *gradient boosting*, atau menggunakan jaringan syaraf tiruan *multi layer perceptron* juga akan dikaji dalam penelitian mendatang. Untuk permintaan kode yang digunakan dalam tulisan ini dapat menghubungi penulis melalui surel.

Ucapan Terima Kasih

Terima kasih diucapkan kepada mitra bebestari yang telah menelaah tulisan ini secara *blind review*.

Referensi

- [1] X. Wu and Z. Xie, "Forward kinematics analysis of a novel 3-DOF parallel manipulator," vol. 26, pp. 18–24, 2019, doi: 10.24200/sci.2018.20740.
- [2] K. Ibrahim, A. A. Ramadan, M. Fanni, Y. Kobayashi, A. Abo-ismail, and M. G. Fujie, "Development of a new 4-DOF endoscopic parallel manipulator based on screw theory for laparoscopic surgery," *Mechatronics*, vol. 28, pp. 4–17, 2015, doi: 10.1016/j.mechatronics.2015.02.006.
- [3] R. F. Abo-shanab, "An Efficient Method for Solving the Direct Kinematics of Parallel Manipulators Following a Trajectory," vol. 2, no. 3, pp. 228–233, 2014, doi: 10.12720/joace.2.3.228-233.
- [4] X. Zhijiang, L. Huan, and S. Daiping, "Forward kinematics of 3-RPS parallel mechanism based on a continuous ant colony algorithm," *China Mech. Eng.*, vol. 26, no. 6, pp. 799–803, 2015.
- [5] U. G. Algorithms *et al.*, "Optimal Forward Kinematics Modeling of Stewart Manipulator," *Jordan J. Mech. Ind. Eng.*, vol. 3, no. 4, pp. 280–293, 2009.
- [6] E. J. Lopez, D. Servin De La Mora-Pulido, R. Servin De La Mora-Pulido, F. J. Ochoa-Estrella, M. Acosta Flores, and G. Luna-Sandoval, "Modeling in Two Configurations of a 5R 2-DoF Planar Parallel Mechanism and Solution to the Inverse Kinematic Modeling Using Artificial Neural Network," *IEEE Access*, vol. 9, pp. 68583–68594, 2021, doi: 10.1109/ACCESS.2021.3073402.
- [7] A. H. Elsheikh, E. A. Showaib, and A. E. M. Asar, "Artificial Neural Network Based Forward Kinematics Solution for Planar Parallel Manipulators Passing through Singular Configuration," *Adv. Robot. Autom.*, vol. 2, no. 2, pp. 1–6, 2013, doi: 10.4172/2168-9695.1000106.
- [8] H. Faraji, "Solving the Forward Kinematics Problem in Parallel Manipulators Using Neural Network," in *Proceedings of the 2017 The 5th International Conference on Control, Mechatronics and Automation*, 2017, pp. 23–29.
- [9] H. Sadjadian, T. D, and A. Fatehi, "Neural Networks Approaches for Computing the Forward Kinematics of a Redundant Parallel Manipulator," *Int. J. Comput. Inf. Eng.*, vol. 2, no. 5, pp. 1664–1671, 2008.
- [10] M. J. Thomas *et al.*, "Determination of inverse kinematic solutions for a 3 degree of freedom parallel manipulator using machine learning," *2020 IEEE Students' Conf. Eng. Syst. SCES 2020*, no. November, pp. 0–5, 2020, doi: 10.1109/SCES50439.2020.9236725.
- [11] M. J. Thomas, M. M. Sanjeev, A. P. Sudheer, and J. M.L, "Comparative study of various machine learning algorithms and Denavit–Hartenberg approach for the inverse kinematic solutions in a 3-PPSS parallel manipulator," *Ind. Rob.*, vol. 47, no. 5,

- pp. 683–695, 2020, doi: 10.1108/IR-11-2019-0233.
- [12] P. Flach, *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*. New York: Cambridge University Press, 2012.
- [13] Adriyan, “Analisis Kinematika dan Singularitas Manipulator Paralel Bidang 2 DOF dengan Rantai Kinematik Paralelogram Simetris,” *J. Tek. Mesin*, vol. 10, no. 2, pp. 79–87, 2020.
- [14] X. J. Liu and J. J. Wang, *Parallel Kinematics: Type, Kinematics, and Optimal Design*. Heidelberg: Springer, 2014.
- [15] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge: Cambridge University Press, 2019.
- [16] C. R. Harris *et al.*, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, 2020, doi: 10.1038/s41586-020-2649-2.
- [17] T. pandas development team, “pandas-dev/pandas: Pandas.” Zenodo, Feb. 2020, doi: 10.5281/zenodo.3509134.
- [18] W. McKinney, “Data Structures for Statistical Computing in Python,” in *Proceedings of the 9th Python in Science Conference*, 2010, pp. 56–61, doi: 10.25080/Majora-92bf1922-00a.
- [19] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
- [20] J. D. Hunter, “Matplotlib: A 2D graphics environment,” *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.
- [21] M. L. Waskom, “seaborn: statistical data visualization,” *J. Open Source Softw.*, vol. 6, no. 60, p. 3021, 2021, doi: 10.21105/joss.03021.
- [22] T. Kluyver *et al.*, “Jupyter Notebooks—a publishing format for reproducible computational workflows,” in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, 2016, pp. 87–90, doi: 10.3233/978-1-61499-649-1-87.