



## INTEGRASI MATH DAN CAD TOOL UNTUK MERANCANG KINEMATIKA MANIPULATOR SERI ROBOT INDUSTRI

**Roche Alimin**

Jurusan Teknik Mesin Universitas Kristen Petra  
Jalan. Siwalankerto 121-131, Surabaya 60236. Indonesia  
Phone: +62-31-8439040, Fax: +62-31-8417658  
E-mail : ralimin@peter.petra.ac.id

### ABSTRAK

*Makalah ini membahas integrasi antara beberapa perangkat lunak untuk memudahkan proses desain mekanisme sebuah serial robotic manipulator, khususnya pada desain kinematikanya. Hal ini dilatarbelakangi dengan kenyataan bahwa pada saat ini sudah banyak perangkat lunak CAD (solid modeling) yang mempunyai kemampuan untuk menganalisa dan memverifikasi kinematika atau dinamika sebuah mekanisme, baik yang built-in maupun add-in. Akan tetapi ada keterbatasan apabila digunakan untuk menganalisa atau memverifikasi mekanisme yang melibatkan invers-kinematic yang cukup kompleks.*

*Beranjak dari hal tersebut maka dilakukan integrasi antara dua perangkat lunak yang sudah cukup populer, yaitu Matlab yang mempunyai kemampuan perhitungan matrik untuk invers-kinematic dan Solidworks yang mempunyai kemampuan yang baik di dalam virtual prototype. Prosesnya berawal dari desain part dan assembly oleh Solidworks yang menghasilkan data dimensi dan geometry constrains, lalu data tersebut dikalkulasi (Denavit-Hartenberg methodology) oleh Matlab dengan keluaran dalam bentuk csv file. Selanjutnya data csv file tersebut secara langsung digunakan sebagai input pada joint (rotary/prismatic) pada Solidworks. Untuk studi kasus pada makalah ini digunakan konfigurasi SCARA manipulator robot.*

*Dari hasil integrasi ini, proses rancang bangun dari mekanisme serial robotic manipulator dapat dibuat lebih otomatis, dimana perancang dapat dengan mudah merancang-ulang mekanisme robot yang kurang sesuai dan melihat dengan cepat hasil analisa kinematikanya. Tentunya ini akan berkontribusi di dalam skala rapid prototyping.*

*Kata kunci: kinematika, manipulator robot, simechanics, rapid prototyping.*

### 1. Pendahuluan

Pada proses perancangan manipulator robot industri, analisa kinematika seringkali merupakan proses awal yang harus dilakukan (setidaknya pada saat *preliminary design*). Proses ini dilakukan sebelum dilakukan analisa dinamika dan perancangan sistem kontrolnya. Sesuai dengan fungsi dari sebuah manipulator robot yang dirancang untuk aplikasi tertentu, maka penentuan konfigurasi, dimensi serta analisa kinematika seringkali harus dilakukan secara bersamaan. Berkaitan dengan kebutuhan ini maka proses simulasi berbasis komputer sangat diperlukan, baik simulasi model fisik maupun simulasi kinematika. Dengan adanya simulasi maka kemampuan dan keterbatasan konfigurasi, daerah kerja (*workspace*), respon sistem kontrol dari manipulator robot dapat diperkirakan lebih awal.

*Robotic toolbox* (dibuat oleh P.I. Corke) maupun *SimMechanics toolbox* dari Matlab mampu memberikan simulasi analisa kinematika manipulator robot dengan hasil yang cukup memuaskan, tetapi dari segi dimensi masih jauh dari mewakili model yang sesungguhnya. Salah satu ketidak-terwakilan model adalah karena pada kenyataannya hampir setiap *revolute joint* tidak berputar 360° atau bahkan mendekatinya. Juga penentuan posisi titik berat masing-masing *link* juga masih dalam perkiraan kasar. Dan yang lebih penting lagi, detail desain dari *link/robot*, termasuk motor servo, sulit untuk diikuti dalam simulasi kinematikanya. Kekurangan-kekurangan ini dapat teratasi apabila di dalam proses pemodelan *rigid body* digunakan perangkat lunak CAD, khususnya kemampuan *solid modeling*-nya. Dengan *solid modeling*, part (*link* manipulator robot) dapat dengan mudah diketahui posisi titik beratnya, dan apabila terjadi *interference* antar *link* juga dapat dengan mudah diketahui.

Jeong dkk (2012) melakukan studi kinematika manipulator robot las (*Faraman AMI*) dengan menggunakan model yang dibangun dengan perangkat lunak CATIA V5 untuk dibandingkan dengan hasil dari persamaan *inverse kinematic*. Hasilnya menunjukkan *angle value error* yang cukup kecil (0,05-0,15 seconds) [1]. Clark dan Lin (2007) juga melakukan integrasi antara Matlab dan Pro/Mechanica untuk mensimulasi kinematika dari sebuah *six degree-of-freedom PUMA Industrial Robot*. Matlab digunakan untuk membuat file .tab, dimana masing-masing file .tab ini berisi data posisi fungsi waktu dari masing-masing *joint angle* dari robot [2].

Beranjak dari kebutuhan akan perancangan manipulator robot yang lebih cepat, lebih terintegrasi serta simulasi kinematika yang lebih akurat, maka untuk tujuan tersebut dilakukan integrasi dua buah perangkat lunak yang telah populer di pasaran, yaitu Matlab (dengan *SimMechanics-toolbox*-nya) dan Solidworks. *SimMechanics* (*Simulink-based*)

mempunyai kemampuan untuk mensimulasi kinematika dan dinamika dari sebuah mekanisme, akan tetapi di dalam membangun model mekaniknya hanya terbatas pada bentuk-bentuk penyederhanaan, seperti *convex hulls* atau *ellipsoids*. Dengan bersinergi dengan *Solidworks*, maka model manipulator robot dapat dibuat semirip mungkin dengan rancangan nyatanya.

Pada makalah ini akan dibahas salah satu studi kasus manipulator robot yang cukup populer digunakan di industri, yaitu *SCARA* robot. *SCARA* manipulator robot mempunyai 4 *degree-of-freedom*, yaitu 3 *revolute joint* dan 1 *prismatic joint*. Aplikasi manipulator robot ini banyak digunakan pada proses pemindahan bahan dan perakitan industri elektronika.

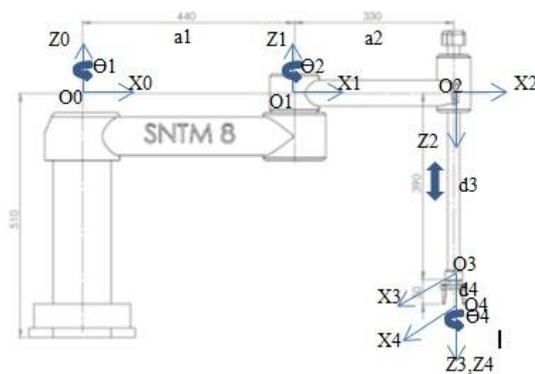
## 2. Metodologi

Garis besar langkah-langkah perancangan simulasi kinematika *SCARA* manipulator robot adalah sebagai berikut:

- Penentuan persamaan *forward kinematic* dan *inverse kinematic* dari *SCARA* manipulator robot.
- Membuat program *M-file Matlab* untuk persamaan *forward kinematic* dan *inverse kinematic* tersebut.
- Membuat *solid model* dengan *Solidworks* untuk setiap komponen/*part* dari *SCARA* manipulator robot, dilanjutkan dengan proses *assembling*-nya.
- Kasus pertama: melakukan simulasi kinematika dengan *motion analysis Solidworks*. Data berasal dari *file csv Matlab*.
- Kasus kedua: membuat *file xml* dari model, kemudian dengan bantuan *SimMechanics (Matlab toolbox)* dilakukan simulasi kinematika di lingkungan *Simulink*.

### 2.1 Kinematika *SCARA* Manipulator Robot

Aturan yang digunakan untuk mendapatkan persamaan *forward kinematics* adalah *D-H (Denavit-Hartenberg) convention*. Gambar di bawah ini adalah *home position* dari *SCARA* manipulator robot, dimana penempatan *coordinate frames* dan penentuan parameternya mengikuti *D-H convention*.



Gambar 1. Diagram *Joint-coordinate SCARA* Manipulator Robot

Tabel 1. *D-H parameters* dari *SCARA* Manipulator Robot

Joint	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
1	$a_1 = 440$	0	0	$\theta_1^\#$
2	$a_2 = 330$	180	0	$\theta_2^\#$
3	0	0	$d_3^\#$	0
4	0	0	$d_4 = 50$	$\theta_4^\#$

# adalah *joint variable*

Pada *D-H convention*, masing-masing homogeneous transformation  $A_i$  dinyatakan sebagai produk dari empat dasar transformasi matrik.

$$A_i = \text{Rot}_{z,0i} \text{Trans}_{z,di} \text{Trans}_{x,ai} \text{Rot}_{x,ai} \dots (1)$$

$$= \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & \mathbf{d} \\ 0 & 1 \end{bmatrix} = H$$

Dengan demikian matrik transformasi antara *end effector frame* dengan *base frame* adalah:

$$T_0^4 = A_1 A_2 A_3 A_4 = \begin{bmatrix} c_{12}c_4 + s_{12}s_4 & -c_{12}s_4 + s_{12}c_4 & 0 & a_1c_1 + a_2c_{12} \\ s_{12}c_4 - c_{12}s_4 & -s_{12}s_4 - c_{12}c_4 & 0 & a_1s_1 + a_2s_{12} \\ 0 & 0 & -1 & -d_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots (2)$$

Melalui matrik transformasi di atas, maka dengan mengetahui *joint variable* ( $\theta_1, \theta_2, \theta_4, d_3$ ) maka posisi dan orientasi *end-effector* manipulator robot dapat ditemukan. Sebagai kebalikan dari ini disebut *inverse kinematic*, yaitu mencari nilai *joint variable* ( $\theta_1, \theta_2, \theta_4, d_3$ ) didasarkan dari posisi dan orientasi tertentu dari *end-effector*. Oleh karena SCARA manipulator robot hanya mempunyai 4 *degree-of-freedom*, maka tidak semua elemen  $H$  mempunyai penyelesaian [3]. Penyelesaiannya ada hanya apabila  $R$  dalam bentuk:

$$R = \begin{bmatrix} c_\alpha & s_\alpha & 0 \\ s_\alpha & -c_\alpha & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$\text{dan jika jumlah } \theta_1 + \theta_2 - \theta_4 = \alpha = \text{Atan}[r_{12}, r_{11}] \dots (3)$$

Dengan memproyeksikan manipulator robot pada bidang  $x_0y_0$ , maka didapatkan:

$$\theta_2 = \text{Atan}(\pm\sqrt{1-r^2}, r)$$

$$\text{dimana: } r^2 = \frac{d_x^2 + d_y^2 - a_1^2 - a_2^2}{2a_1a_2}$$

$$\theta_1 = \text{Atan}(d_x, d_y) - \text{Atan}(a_1 + a_2c_2, a_2s_2)$$

Dari persamaan (3) didapatkan:

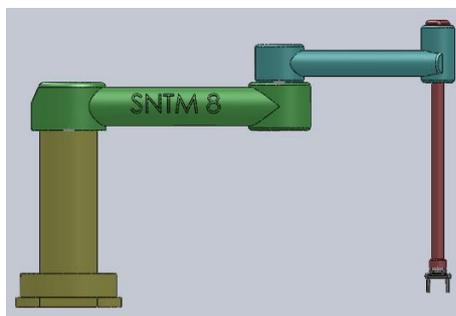
$$\theta_4 = \theta_1 + \theta_2 - \text{Atan}(r_{12}, r_{11})$$

Akhirnya  $d_3$  didapatkan dari:

$$d_3 = d_z + d_4$$

## 2.2 Model Simulasi

*Solidworks* digunakan untuk membuat solid model dari SCARA manipulator robot. Bagian-per-bagian/*part* dari manipulator dibuat, kemudian di-*assembling* dengan metode *bottom-up design*. Kombinasi dengan *top-down design* dapat juga dilakukan. Hal yang perlu diperhatikan adalah letak *coordinate frame* pada masing-masing *part* (dalam hal ini adalah *link* dari manipulator robot). Baik *base coordinate frame* maupun *follower coordinate frame* masing-masing *link*, penempatannya harus sama dengan penempatan *coordinate frame* pada *D-H coordinates*. *Geometry* dan *dimension constraint* juga harus diberikan agar posisi manipulator robot sama dengan posisi *home (zero)* pada *D-H coordinate*. Hasil dari proses *assembly* SCARA manipulator robot dapat dilihat pada gambar di bawah ini.

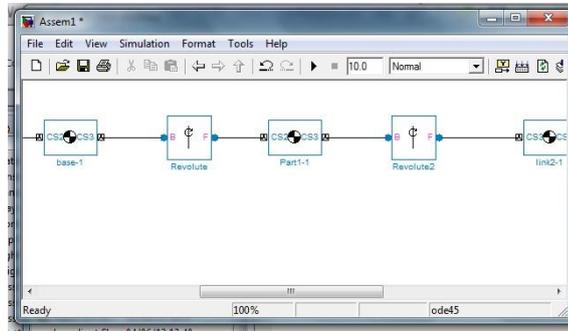


Gambar 2. Solid Model dari SCARA Manipulator Robot

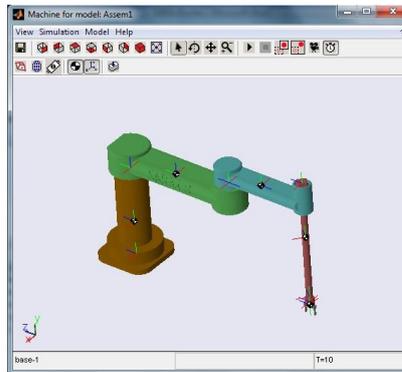
*Solidworks* mempunyai fasilitas *motion study*, baik berupa *motion analysis* maupun hanya berupa animasi saja. *Motion Analysis* merupakan *add-on program* untuk *Solidworks*. Pada saat membuat *motion study* baru maka secara otomatis posisi manipulator robot pada *assembly* dianggap sebagai *home position* dari manipulator robot tersebut. Untuk analisa kinematika (dimana melibatkan analisa perpindahan, kecepatan, percepatan, tetapi tidak melibatkan gaya) sebagai sumber aktuator pada masing-masing *link* manipulator robot dapat digunakan motor, baik untuk gerakan rotasi maupun

linier. Metode input yang digunakan adalah *interpolated linear* dan *cubic* berupa data yang ditabulasi. Data ini berupa *file csv* yang dihasilkan dari *Matlab*.

Pada kasus kedua, assembly model dari *SCARA* manipulator robot di *save-as* menjadi *file xml*. Selanjutnya *SimMechanics* akan meng-*import file* tersebut dan digunakan sebagai solid model di lingkungan *Simulink*. Secara otomatis *SimMechanics* akan mengkonversi setiap *link* dari manipulator menjadi *rigid body-rigid body* beserta *joint-joint*-nya sesuai dengan *constraint* yang diberikan di lingkungan *assembly Solidworks*.



Gambar 3. Model Block *SimMechanics* SCARA (terlihat sebagian)



Gambar 4. Simulasi model *SimMechanics* SCARA

### 3. Hasil dan Pembahasan

Analisa kinematika (*forward kinematic*) dengan *Motion Analysis* dari *Solidworks* menunjukkan hasil yang cukup memuaskan, tetapi membutuhkan perangkat keras komputer yang diatas rata-rata. Terutama apabila melibatkan desain detail yang mendeteksi kontak atau tumbukan antar *link*. Pada *Motion Analysis*, *Matlab* hanya berkontribusi di dalam perhitungan *forward kinematics*-nya saja.

Analisa *inverse kinematics* hanya dapat dilakukan melalui integrasi antara *Solidworks* dengan *Matlab*. Model block *SimMechanics* yang dihasilkan secara otomatis sangat membantu jika dibandingkan harus membuat secara terpisah sesuai dengan konfigurasi manipulator robot. Dimensi dan letak titik berat setiap *link* juga secara otomatis dihasilkan. Untuk analisa kinematika, *block* yang masih harus ditambahkan adalah *block joint sensor* dan *joint actuator*, serta *block-block* dari *Simulink* yang dibuat untuk menghasilkan data bagi model.

### 4. Kesimpulan

Integrasi antara *Matlab* dan *Solidworks* masih menunjukkan adanya keterbatasan, terutama dalam hal analisa kontak. Adapun demikian, melalui integrasi ini memungkinkan untuk melakukan analisa *inverse kinematics* pada sebuah manipulator robot. Simulasi gerakan kinematika ini memungkinkan bagi para designer untuk mencoba kemampuan kinematika dari rancangan sebelum dibuat prototipenya. Ini akan memudahkan dan memperpendek waktu rancang dari sebuah produk, khususnya yang melibatkan analisa kinematika.

### 5. Daftar Pustaka

1. Clark, S., dan Lin, Y., J., "CAD Tool Integration for Robot Kinematics Design Assurance with Studies On PUMA Robots", *Industrial Robot: An International Journal*, Emerland Group Publishing Limited, 34/3, 2007, hal 240-248.



2. Jeong, J., W., dkk., “A Study On Simulation Model and Kinematic Model of Welding Robot”, Journal of Achievements in Materials and Manufacturing Engineering, Vol. 55, Issue 1, International COSCO World Press, Nopember 2012, hal 66-73.
3. Spong, M.W., dan Vidyasagar M., Robot Dynamics and Control, John Wiley & Sons, New York, 1989.