

SISTEM KONTROL PADA ARTICULATED MANIPULATOR ROBOT MENGUNAKAN KONTROL PID

Yusie Rizal¹, Muhammad Wahyu², Sarifudin³
Jurusan Teknik Elektro, Politeknik Negeri Banjarmasin^{1,2,3}
yusie.rizal@poliban.ac.id¹, m.wahyu@poliban.ac.id², sarif.poliban@gmail.com³

ABSTRACT

The control and its implementation to manipulator robot with 6 degrees of freedom is quite difficult to explain in the class. One of the reason is because it is difficult to access the real manipulator robot since the price is expensive and the robot takes amount of space. Generally to solve this problem is by using the numerical based simulation in the form of mathematical models using Matlab/Simulink. However, this numerical simulation is not attractive because it shows only some figures of signals. Other solution is by developing 3D simulation program so that the movement of robot can be observed and more attractive. This article discusses the PID (Proportional-Integral-Derivative) control system on an articulated robot manipulator. This study aims to implement a control system, develop a robot simulator, and implement a control system on the simulator. The Open-Dynamic Engine (ODE) based robot simulator was created using the Marilou AnyKode program and Visual C++ 2008. The 3D of robotic simulation was then used as a test tool in implementing the PID control system design. Based on the results of the experiment, it was found that each joint robot controlled using PID control can provide a response in accordance with what is expected where the controller is able to control the movement of the robot to follow the setpoint.

Keywords: PID control, robot simulation, anyKode marilou, manipulator robot, articulated robot

ABSTRAK

Pengontrolan dan implementasi kontrol pada manipulator robot dengan 6 derajat kebebasan cukup sulit untuk dijelaskan dalam pembelajaran di kelas. Hal ini disebabkan oleh sulitnya mengakses real manipulator robot karena harganya cukup mahal serta memakan ruang yang cukup besar. Solusinya yang umum dipakai adalah dengan simulasi berbasis numerik dalam bentuk model matematis dengan menggunakan program seperti Matlab/Simulink. Namun demikian, gambaran dari hasil simulasi ini tidak atraktif karena hanya berbentuk grafik sinyal. Solusi lainnya adalah dengan membuat program simulasi 3D manipulator robot sehingga gerakannya tampak terlihat dan lebih atraktif. Dalam artikel ini dibahas sistem kontrol PID (*Proporsional-Integral-Derivative*) pada manipulator *articulated* robot. Penelitian ini bertujuan untuk menerapkan sistem kontrol, mengembangkan simulator robot, serta mengimplementasikan sistem kontrol pada simulator tersebut. Simulator robot berbasis Open-Dynamic Engine (ODE) ini dibuat menggunakan program AnyKode Marilou dan Visual C++ 2008. Simulasi ini kemudian digunakan sebagai alat uji dalam mengimplementasikan rancangan sistem kendali PID ini. Berdasarkan dari hasil percobaan diperoleh bahwa setiap joint robot yang dikontrol menggunakan kontrol PID dapat memberikan respon yang sesuai dengan apa yang diharapkan dimana controller mampu mengendalikan gerakan robot untuk mengikuti setpointnya.

Kata Kunci: kontrol PID, simulasi robot, anyKode marilou, robot manipulator, articulated robot

PENDAHULUAN

Manipulator robot secara umum mengacu kepada sebuah sistem robot yang terdiri dari beberapa link dan joint dimana pada bagian ujungnya terdapat *end-effector* yang bisa dipasang dengan alat pengambil (*pick-place*), alat pengebor, alat pengecat, dan sebagainya. Manipulator jenis *articulated* robot adalah manipulator robot dimana joint-jointnya memiliki *revolute* joint yang bisa berputar. Bentuk real dari manipulator articulated robot dengan 6 joint robot dapat ditunjukkan pada gambar 1 dengan *end-effector* tanpa terpasang dengan alat apapun. Sistem robot manipulator ini memiliki 6 derajat kebebasan sehingga pada bagian ujungnya mampu bergerak pada 3 sumbu x , y , z dan berputar relatif terhadap ketiga sumbu tersebut.



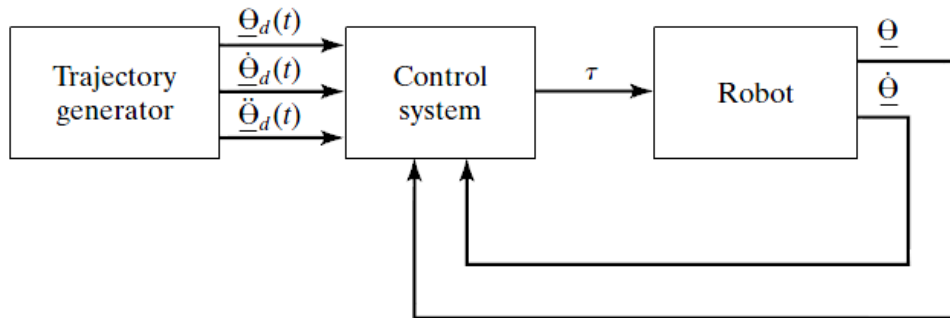
Gambar 1. Arm Manipulator Robot (Kehui, 2019)

Persoalan yang umumnya dihadapi dalam pembelajaran manipulator robot dan bagaimana cara mengendalikannya adalah dalam hal sulitnya mengakses manipulator robot sebagai bagian dari alat praktek di kelas/laboratorium. Meskipun jika robot ini dibuat sendiri akan memerlukan waktu lama dengan hasil yang belum tentu sempurna. Dilain pihak apabila robot tersebut dibeli dari perusahaan penyedia, harga untuk satu unit manipulator robot berstandar industri ini sangat mahal. Lebih jauh lagi, jika pun mampu dibeli maka memerlukan ruangan cukup besar untuk bisa menyimpan dan mengoperasikannya. Berdasarkan dari uraian persoalan tersebut maka diperlukan solusi alternatif untuk mengatasi masalah ini.

Simulasi berbasis *Open-Dynamic Engine* (ODE) merupakan jenis simulasi komputer dengan tampilan 3 dimensi dimana gerakan benda pada layar tampak mirip seperti aslinya. Pengembangan simulasi 3D ini dapat dikatakan sebagai sebuah simulator virtual. Simulasi berbasis ODE merupakan simulasi yang banyak dipakai dalam simulator untuk riset-riset maupun game. Contoh nyata penggunaan simulasi berbasis ODE dapat dilihat dari penelitian untuk sistem kontrol pada *reaction wheel pendulum* (Rizal et al., 2018) ataupun penelitian disertasi yang dibuat oleh Agus Naba (Naba, 2007) dimana ia mengembangkan sistem plant berupa *cart-pole* menggunakan simulasi berbasis ODE sebagai media untuk menguji algoritma kontrol adaptive yang dibuatnya.

Contoh lain dalam penggunaan simulasi berbasis ODE diberikan oleh Haris dan Conrad dalam artikelnya dimana ia melakukan survey terhadap sejumlah simulator robot yang ada dipasaran menggunakan Webots, Microsoft Robotics Developer Studio, AnyKode Marilou, robotSim Pro (Harris and Conrad, 2011). AnyKode Marilou sebagai salah satu simulator yang menjadi bagian dari surveynya disebutkan sebagai simulator yang memiliki sensor yang dapat dikonfigurasi sehingga cukup fleksibel. Dalam penelitian ini dikembangkan simulator manipulator robot menggunakan AnyKode Marilou.

Dalam literatur terdapat banyak pembahasan tentang bagaimana mengendalikan manipulator robot menggunakan beragam metode kontrol termasuk khususnya dengan metode PID (*Proportional-Integral-Derivative*) ataupun kombinasi dari kontrol PID dengan metode lainnya. Diantaranya seperti penggunaan approximate jacobian PID control law (Cheah et al., 1999), saturated nonlinear PID control (Santibañez et al., 2010), adaptive PID learning control (Kuc and Han, 2000), H_{∞} and Predictive Control (Iqbal, 2019), ataupun intelligent PID control (iPID) (Abouaïssa and Chouraqui, 2019). Penelitian tentang metode kontrol PID dan kombinasinya dengan metode lain dalam pengontrolan manipulator robot ini telah banyak memberikan pilihan yang banyak bagi peneliti. Ini belum termasuk metode-metode lainnya menggunakan nonlinear control.



Gambar 2. Diagram blok sistem kontrol robot.

Secara umum, prosedur dalam perancangan sistem kontrol pada manipulator robot adalah melalui penggunaan model dinamis dari manipulator robot dengan *torque* sebagai kontrol inputnya (Islam et al., 2014). Kemudian persamaan error dari tiap joint dirumuskan dan dicari persamaan dinamis dari errornya. Setelah itu baru kontrol PID digunakan dalam sistem tersebut karena komponen kontrol PID melibatkan persamaan dari error, turunan error, dan integrasi dari error sebelumnya. Selanjutnya tuning dilakukan terhadap masing-masing kontrol gain sehingga diperoleh performa respon yang diharapkan. Skema dari sistem kontrol ini adalah menggunakan umpan balik sebagaimana diberikan pada gambar 2. Simbol τ adalah input kontrol yang diberikan pada sistem plant (robot), θ dan $\dot{\theta}$ masing-masing adalah posisi sudut joint dan kecepatan sudut dari joint robot. Perlu dicatat bahwa semua variabel dari $\tau, \theta, \dot{\theta}$ adalah vektor larik yang mewakili masing-masing joints. Karena sistem manipulator yang digunakan disini ada 6 joint yang terdiri dari 5 revolute joint $[\tau_0 \ \tau_1 \ \tau_2 \ \tau_3 \ \tau_4]$ dan 1 prismatic joint. Demikian pula dengan variabel $[\theta_0 \ \theta_1 \ \theta_2 \ \theta_3 \ \theta_4]$ dan $[\dot{\theta}_0 \ \dot{\theta}_1 \ \dot{\theta}_2 \ \dot{\theta}_3 \ \dot{\theta}_4]$.

Dalam penelitian ini, penulis menggunakan metode kontrol PID yang cukup standar dalam pengontrolan manipulator robot. Hal ini disebabkan karena harapan yang ingin dicapai adalah kemampuan mengembangkan simulator robot manipulator articulated robot dengan software komersial seperti AnyKode Marilou. Selain dari itu agar penelitian ini dapat memberikan manfaat bagi mahasiswa dalam pengalamannya menerapkan konsep kontrol PID pada manipulator robot. Simulator dari manipulator robot ini dibuat dan dikembangkan menggunakan simulasi berbasis ODE sehingga diperoleh virtual robot yang dapat dikontrol melalui sebuah program komputer. Program ini dibuat menggunakan program C/C++ sebagai “otak” dari sistem manipulator tersebut. Program ini analoginya seperti isi mikrokontroller yang mampu mengontrol dan mengendalikan keseluruhan joint-joint robot.

METODE PENELITIAN

Terdapat 4 alat bantu (*tools*) yang diperlukan dalam penelitian yang mencakup proses perhitungan, pengembangan simulator, dan pengujian sistem. Keempat tools tersebut adalah:

1. Software Mathematica
2. Software AnyKode Marilou
3. Software Microsoft Visual C++
4. Software Matlab

Masing-masing dari tools tersebut memiliki fungsi sebagai berikut: (a) Perangkat software Mathematica 8 digunakan untuk melakukan komputasi simbolik, yaitu untuk mengecek penurunan model sistem agar tetap akurat saat diturunkan; (b) AnyKode Marilou dan (c) Visual C++ masing-masing digunakan saat pengembangan model 3D simulator manipulator robot dan pengontrolan virtual robot tersebut. Selanjutnya, (d) Matlab digunakan untuk melakukan plotting atau grafik dari respon masing-masing joint robot sehingga diketahui seberapa bagus responnya.

Selain dari itu, software Matlab ini digunakan untuk perhitungan numerik nilai performa indeks menggunakan Integral Square Error (ISE) dan Integral Absolute Error (IAE).

Perancangan Kontrol PID

Persamaan sistem dinamis dari manipulator robot diberikan oleh (Santibañez et al., 2010) sebagaimana persamaan berikut:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (1)$$

dimana $M(q)$ adalah matrik inersia dari manipulator robot, $C(q, \dot{q})$ adalah matrik torsi sentripetal dan torsi coriolis, q adalah vektor posisi joint robot, sedangkan τ merupakan vektor dari torsi, dan $g(q)$ adalah vektor torsi gravitasi. Persamaan (1) ini merupakan persamaan dinamis yang diturunkan dari persamaan model dinamis dari manipulator robot. Isi dari masing-masing variabel vektor dan matriks tersebut dapat diperoleh dari penurunan persamaan sebagaimana penelitian yang sebelumnya telah dilakukan oleh (Islam et al., 2014) dengan menggunakan metode Euler-Lagrange.

Dengan menggunakan kontrol PID ini diharapkan agar error yang diperoleh bisa diminimalis sedemikian rupa sehingga pada waktu t tak terhingga errornya menuju ke 0 atau paling tidak disekitar nilai 0. Persamaan error ini dapat dituliskan:

$$e(t) = x_a(t) - x_d(t) \quad (2)$$

dengan $e(t)$ adalah error tracking dari manipulator robot, $x_a(t)$ adalah posisi sudut dari joint robot, sedangkan $x_d(t)$ adalah posisi sudut yang diinginkan atau diperintahkan. Persamaan (2) ini sejatinya adalah vektor persamaan untuk keseluruhan joint robot sehingga total ada 5 buah (revolute joint) persamaan (2) yang merepresentasikan persamaan dinamis dari error. Selain dari itu, jika dikaitkan dengan persamaan sebelumnya diketahui bahwa:

$$\begin{aligned} x_a &= \theta_i \\ x_d &= \theta_d \end{aligned} \quad (3)$$

Apabila persamaan (2) diturunkan, maka diperoleh sebagai berikut:

$$\begin{aligned} \dot{e} &= \dot{x}_a - \dot{x}_d \\ \ddot{e} &= \ddot{x}_a - \ddot{x}_d \end{aligned} \quad (4)$$

Apabila dituliskan dalam persamaan *state space*-nya diperoleh:

$$\frac{d}{dt} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} U \quad (5)$$

Maka *control law* U dari persamaan (5) ini diberikan oleh:

$$U = -K_p e - K_d \dot{e} + v \quad (6)$$

$$\dot{v} = -K_i e \quad (7)$$

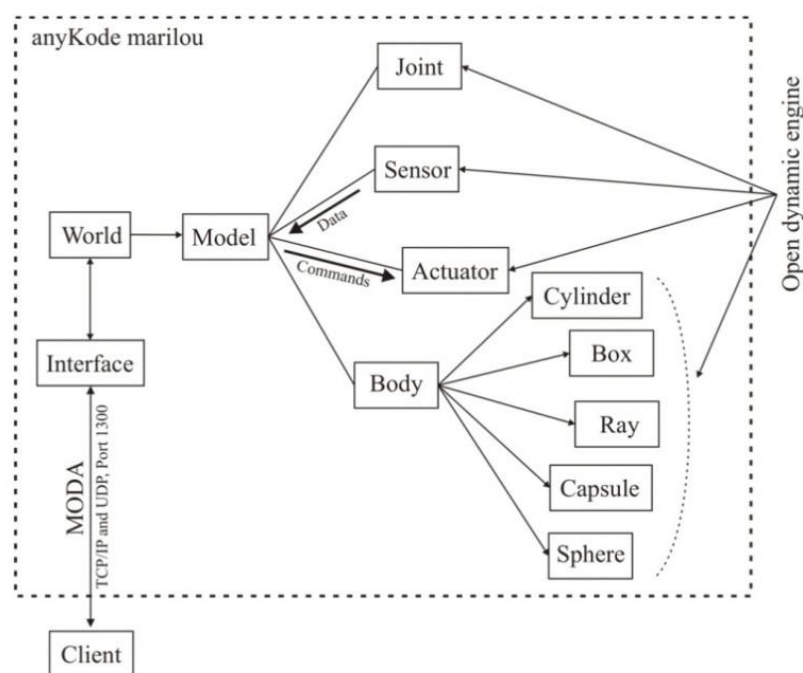
Persamaan (6)-(7) ini merupakan persamaan kontrol PID. Kontrol gain K_p, K_d dan K_i dimana $K_p, K_d, K_i > 0$ perlu dituning atau ditentukan nilainya sehingga dapat membuat sistem pada persamaan (5) menjadi stabil.

Perancangan Simulator Manipulator Robot

Simulator manipulator robot dibuat dengan menggunakan software AnyKode Marilou versi 4.8.2. Langkah-langkah yang dilakukan adalah dengan merancang model manipulaternya dengan menggunakan sejumlah link berbentuk silinder sedangkan pada bagian alas dan tangannya menggunakan balok. Masing-masing link ini kemudian direkatkan menggunakan joint yang bersifat revolute (bisa berputar). Ada 6 joint yang digunakan, 5 diantaranya menggunakan joint revolute sedangkan joint sisanya adalah prismatic/linear. Pada kelima joint tadi dipasang motor

sebagai aktuatornya sedangkan joint lainnya menggunakan pneumatic. Sensor yang digunakan dalam sistem ini adalah encoder motor yang bisa dikaitkan langsung dengan motornya.

Komponen-komponen utama dari struktur program AnyKode Marilou dapat dijabarkan pada gambar 3. Sebuah sistem model terdiri dari 4 komponen utama, yaitu joint sebagai penghubung/pengikat, sensor yang tertanam (*embedded*), aktuator sebagai penggerak/pendorong, dan benda tegar (*body*) sebagai pembentuk struktur robot atau sistem mekanik. Program anyKode ini dapat dihubungkan dengan program eksternal melalui interface menggunakan MODA yang diakses melalui port TCP IP atau Port 1300. Microsoft Visual C++ digunakan sebagai program pengendali dari robot yang dikoneksikan melalui MODA tadi. MODA disini sebenarnya adalah sejumlah library C/C++ yang telah disediakan oleh developernya (anyKode Marilou) dan bisa dipanggil dan diakses melalui program C++ yang kita buat. Sebagai contoh, apabila sensor motor encoder digunakan maka pada saat program anyKode Marilou ini dijalankan maka pada program C/C++ yang telah dieksekusi bisa dimunculkan hasil pembacaan sudut dari sensor tersebut.



Gambar 3. Struktur umum AnyKode Marilou (Rizal, 2019).

Perhitungan Performa Indeks Menggunakan ISE dan IAE

Perhitungan performa indeks sebenarnya digunakan untuk membandingkan performa dari sistem apabila ada 2 jenis metode yang digunakan pada sistem yang sama. Berdasarkan dari perhitungan ini akan diperoleh metode mana yang terbaik dengan melihat seberapa kecil nilai dari ISE dan IAE yang diperoleh dari satu metode dengan metode lainnya. Meskipun dalam penelitian ini hanya digunakan satu metode saja, penulis menganggap perlu untuk melihat bagaimana perbandingan dari nilai ini apabila terdapat nilai setpoint yang berbeda-beda antara joint satu dengan joint lainnya.

Dalam melakukan perhitungan performa indeks ini menggunakan ISE dan IAE dapat dituliskan dalam dua persamaan (8)-(9) berikut:

$$IAE = \int |e(t)| dt \quad (8)$$

$$ISE = \int e^2(t) dt \quad (9)$$

Dalam persamaan (8)-(9), data-data yang digunakan dalam perhitungan adalah perbandingan antara nilai setpoint dan nilai aktualnya. Pengumpulan data masing-masing error ini

menggunakan waktu sampling yang sama, yaitu 10 ms. Perhitungan dari ISE dan IAE dilakukan terhadap masing-masing joint robot.

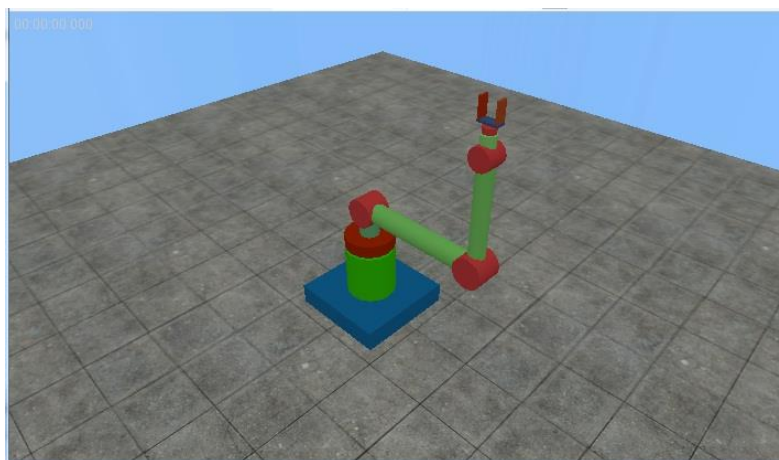
Plotting Respon Sistem dan Kestabilan

Plotting dari respon sinyal dari masing-masing *joint manipulator* robot berkaitan dengan pengumpulan data error dari penjelasan persamaan (8)-(9). Data yang disimpan adalah data aktual dari masing-masing joint dan data setpoint yang diberikan kepada sistem. Data-data ini disimpan kedalam file *.dat dan kemudian dibuka menggunakan program Matlab. Pengolahan data-data tersebut dilakukan dengan membuat program di Matlab sehingga data gambar bisa disajikan dari waktu ke waktu (mulai dari waktu ke-0 sampai seterusnya).

Kestabilan dari sistem dapat dilihat dari pengolahan data tersebut, yaitu dengan melihat apakah respon dari sistem dapat membuat posisi sudut aktual dari *joint* robot bisa menuju posisi sudut setpointnya. Hal ini berarti bahwa *error* dari sistem akan selalu menuju konvergen ke titik kesetimbangannya (dalam hal ini adalah 0). Apabila hal ini dicapai, maka dapat disimpulkan bahwa sistem kontrol PID yang telah dibuat dan diimplementasikan mampu mengontrol manipulator robot.

HASIL DAN PEMBAHASAN

Berdasarkan hasil dari pengembangan simulator manipulator robot menggunakan anyCode Marilou diperoleh hasil model desain sebagaimana diberikan pada gambar 4. Pada gambar tersebut terdapat 6 joint robot dengan komposisi 5 buah joint revolute (bisa berputar) dan 1 joint prismatic (bisa bergeser). Joint terakhir, yakni prismatic joint terdapat pada bagian *end-effector*nya. Model yang dibuat pada gambar 4 ini mengikuti model yang diberikan pada gambar 1. Perbedaan keduanya hanya terletak pada ujungnya (*end-effector*).



Gambar 4. Simulator arm manipulator robot

Dalam simulasi komputer yang dilakukan, diberikan situasi tertentu dimana diasumsikan bahwa manipulator robot memperoleh sejumlah perintah (*command*) kepada masing-masing joint robotnya dengan nilai setpoint yang berbeda sebagaimana diberikan pada tabel 1. Masing-masing setpoint bervariasi dan mempunyai nilai positif dan negatif sebagai perintah.

Tabel 1. *Setpoint* untuk masing-masing joint robot

Joint ke-i	Nilai Setpoint	Satuan
0	60.0	Derajat
1	-40.0	Derajat
2	140.0	Derajat
3	15.0	Derajat
4	40.0	Derajat

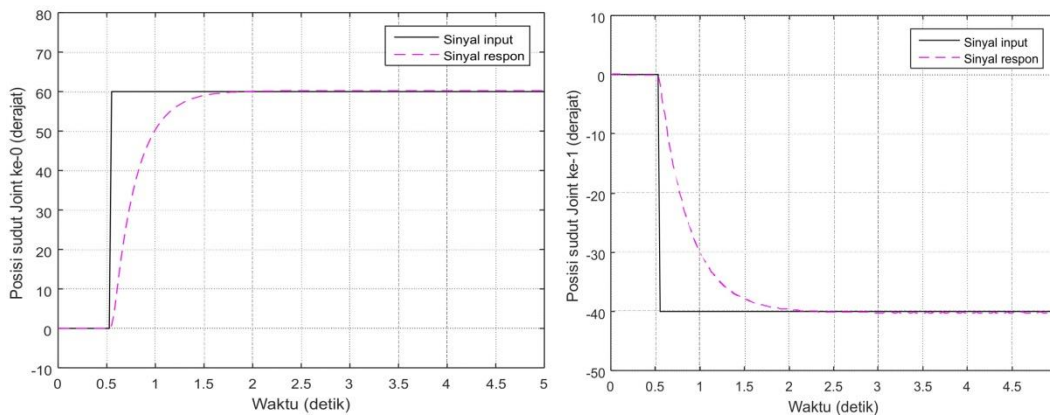
Setelah semua setpointnya diisikan kedalam program, kontrol gain dari masing-masing joint robot juga ditentukan sesuai dengan persamaan *control law* yang diberikan pada persamaan (6)-

(7). Dua persamaan tersebut adalah persamaan umum dan perlu diterapkan pada masing-masing joint robot, sehingga tiap joint memiliki kontrol gain (K_p, K_d, K_i) masing-masing yang berbeda. Dalam eksperimen yang dilakukan diperoleh nilai-nilai kontrol gain yang membuat sistem cukup stabil sebagaimana diberikan pada tabel 2. Penentuan nilai ini (*tuning*) dilakukan secara coba-coba (*trial and error*).

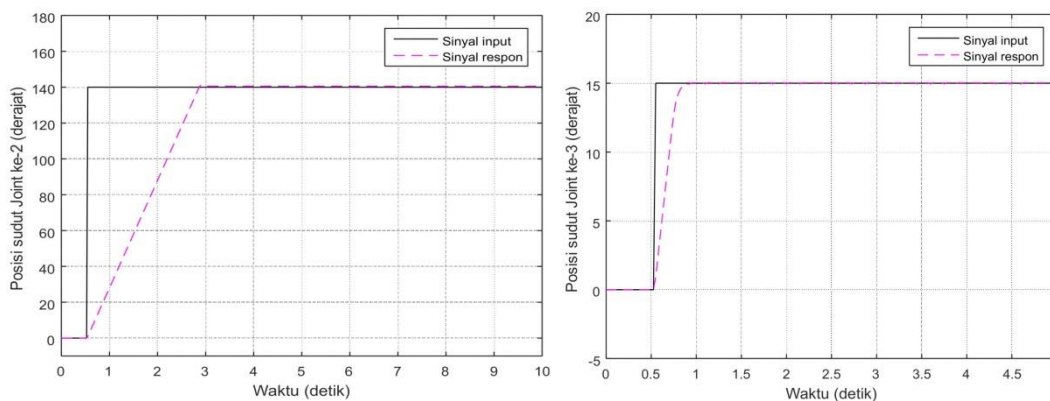
Tabel 2. Gain kontrol PID untuk manipulator robot

Joint ke-i	K_p	K_d	K_i
0	3.0	0.050	0.001
1	2.3	0.020	0.001
2	40.0	0.085	0.003
3	10.0	0.090	0.002
4	12.0	0.014	0.006

Gambar 5-7 adalah hasil dari uji percobaan dari sistem kontrol PID yang diimplementasikan pada manipulator robot dengan perintah sesuai dengan setpointnya (tabel 1). Pada gambar 5 diperoleh bahwa sistem dapat dibuat stabil dimana error yang ada bisa dibuat konvergen dan menuju titik 0 dalam waktu 1.5~2 detik pertama.

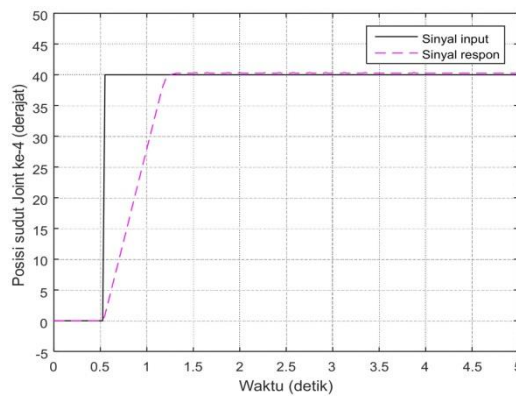


Gambar 5. Respon posisi sudut dari joint ke-0 (kiri) dan joint ke-1 (kanan) robot



Gambar 6. Respon posisi sudut joint ke-2 (kiri) dan joint ke-3 (kanan) robot

Pada gambar 6 dapat dilihat bahwa kedua joint mampu distabilkan namun memiliki waktu konvergenya tidak sama. Pada joint ke-2 diperoleh waktu konvergen lebih lama yakni sekitar 3 detik sedangkan waktu konvergen joint ke-3 adalah 0.5 detik. Demikian pula pada gambar 7, waktu konvergenya terjadi sekitar 1.25 detik. Ketiga joint tersebut (gambar 6 dan 7) pada akhirnya juga bisa distabilkan (ini terbukti dari nilai setpointnya sama dengan nilai joint robot).



Gambar 7. Respon posisi sudut dari joint ke-4 robot

Performa indeks dari kedua pendekatan ISE dan IAE untuk masing-masing joint diberikan pada tabel 3. Dari tabel tersebut ditunjukkan banyak variasi dari nilai masing-masing performa indeksnya. Secara umum nilai IAE lebih kecil dibandingkan nilai ISE. Ini logis karena berdasarkan persamaan (8)-(9) diketahui bahwa untuk kasus ISE menggunakan integrasi dari kuadrat errornya, sedangkan IAE menggunakan integrasi dari nilai absolut errornya.

Tabel 3. Perbandingan antara ISE dan IAE untuk tiap joint manipulator robot

Joint ke-i	Integral Square Error (ISE)	Integral Absolute Error (IAE)
0	517.5362	17.4242
1	270.6647	14.6879
2	15179	166.9824
3	18.0425	2.2319
4	350.7412	15.5714

Dari tabel 3 tersebut diketahui bahwa nilai ISE (15179) dan IAE (166.9824) yang terbesar terjadi pada joint ke-2. Jika dikaitkan dengan besaran setpointnya, pada joint ke-2 tersebut diketahui bahwa nilai setpointnya adalah juga yang terbesar, yakni 140 derajat. Sebaliknya nilai ISE dan IAE yang terkecil terdapat pada joint ke-3 dan jika dikaitkan dengan nilai setpointnya diperoleh pula bahwa nilainya juga terkecil (15 derajat).

KESIMPULAN DAN SARAN

Berdasarkan dari hasil penelitian ini diperoleh bahwa simulator robot berupa manipulator *articulated* robot dapat dibuat menggunakan program anyKode Marilou. Sistem kendali PID dapat dipakai untuk mengendalikan masing-masing joint apabila setpoint untuk masing-masing joint robot diberikan. Terakhir, dalam penelitian ini diketahui bahwa performa indeks yang digunakan sebagai metode untuk mengukur efektivitas metode kontrol dapat memberikan nilai yang berbeda-beda tergantung dari besaran setpointnya. Apabila nilai setpointnya besar sehingga errornya besar maka membuat nilai IAE dan ISE menjadi besar pula, demikian pula sebaliknya. Dalam penelitian selanjutnya, perlu dikembangkan metode kontrol lain untuk bisa diimplementasikan pada manipulator robot ini, misalnya menggunakan nonlinear kontrol. Selain dari itu pula, penelitian tentang metode bagaimana membuat *trajectory generation* bisa pula dilakukan agar perintah yang dikirimkan kepada masing-masing joint bisa menjadi efisien.

UCAPAN TERIMA KASIH

Para penulis mengucapkan terima kasih kepada Politeknik Negeri Banjarmasin melalui Unit P3M yang telah memberikan bantuan pembiayaan dalam penelitian yang telah dilaksanakan melalui nomor kontrak: 211/PL18/SPK-LT/2019.

DAFTAR PUSTAKA

- Abouaïssa, H. and Chouraqui, S. (2019), “On the control of robot manipulator: A model-free approach”, *Journal of Computational Science*, Vol. 31, pp. 6–16.
- Cheah, C.C., Kawamura, S., Arimoto, S. and Lee, K. (1999), “PID control of robotic manipulator with uncertain Jacobian matrix”, *IEEE International Conference on Robotics and Automation*, pp. 494–499.
- Harris, A. and Conrad, J.M. (2011), “Survey of popular robotics simulators, frameworks, and toolkits”, *Conference Proceedings - IEEE SOUTHEASTCON*, pp. 243–249.
- Iqbal, J. (2019), “Modern Control Laws for an Articulated Robotic Arm”, *Engineering, Technology & Applied Science Research*, Vol. 9 No. 2, pp. 4057–4061.
- Islam, R., Iqbal, J. and Khan, Q. (2014), “Design and Comparison of Two Control Strategies for Multi-DOF Articulated Robotic Arm Manipulator”, *Journal of Control Engineering and Applied Informatics*, Vol. 16 No. 2, pp. 28–39.
- Kehui. (2019), “Industrial Automation & Intelligent Solution”, available at: <https://www.zzkehui.com/cpzx/zdhzj/KH-N20-welding-robot.html> (accessed 17 April 2019).
- Kuc, T.Y. and Han, W.G. (2000), “Adaptive PID learning control of robot manipulators”, *Automatica*, Vol. 36 No. 5, pp. 717–725.
- Naba, A. (2007), *Adaptive Control Based on Adjustable Evaluation Function*, Tsukuba.
- Rizal, Y. (2019), “Computer Simulation of Human-Robot Collaboration in the Context of Industry Revolution 4.0”, *Future of Robotics - Becoming Human with Humanoid or Emotional Intelligence*, InTech, available at: <https://www.intechopen.com/online-first/computer-simulation-of-human-robot-collaboration-in-the-context-of-industry-revolution-4-0>.
- Rizal, Y., Mantala, R., Rachman, S. and Nurmahaludin, N. (2018), “Balance Control of Reaction Wheel Pendulum Based on Second-Order Sliding Mode Control”, *International Conference on Applied Science and Technology (ICAST)*, Manado, Indonesia, pp. 1–6.
- Santibañez, V., Camarillo, K., Moreno-Valenzuela, J. and Campa, R. (2010), “A practical PID regulator with bounded torques for robot manipulators”, *International Journal of Control, Automation and Systems*, Vol. 8 No. 3, pp. 544–555.