

# Robot *Hybrid* pada Kontes Robot ABU Indonesia 2016 (*Climbing Task*)

Alex Nico Simanjuntak<sup>1</sup>, Hendawan Soebhakti<sup>\*</sup>

<sup>1</sup>Teknik Mekatronika, Politeknik Negeri Batam, Batam, Indonesia

<sup>\*</sup>Email: hendawan@polibatam.ac.id

**Abstrak**— Pada Kontes Robot ABU Indonesia tahun 2016 terdapat *task* terakhir yaitu *climbing task* yang harus diselesaikan oleh robot *hybrid* untuk memenangkan pertandingan. Pada bagian ini, robot *hybrid* harus memanjat tiang dan memasang *propeller* pada bagian *engine*. Untuk menyelesaikan *task* tersebut maka pada penelitian ini dikembangkan sistem pemanjat dan pemasangan *propeller* yang menggunakan 4 buah roda polyurethane dan 3 buah motor *power window* sebagai pemanjat. Kemudian akan dikontrol menggunakan metode Proporsional Derivatif (PD) dengan menggunakan EDF dan *rotary encoder* sebagai *feedback*. Lalu setelah berada di atas tiang, robot akan memasang *propeller* ke *engine*. Hasil percobaan menunjukkan robot dapat meluruskan diri dengan garis dalam waktu 0,3 detik dan memanjat tiang dengan rata – rata *error* sebesar 2,17°. Robot dapat memasang *propeller* dengan tingkat keberhasilan 65,0% dengan rata – rata waktu yang dibutuhkan adalah 6,46 detik untuk sukses memasang *propeller* pada *engine*. Kegagalan pemasangan *propeller* disebabkan oleh mekanik robot yang kurang kuat dan pemasangan roda yang tidak sejajar pada pipa.

**Kata kunci** : Kontroller PD, *climbing robot*, EDF, *propeller*

## I. PENDAHULUAN

PADA Kontes Robot ABU Indonesia tahun 2016 terdapat *task* terakhir yaitu *climbing task* yang harus dilakukan oleh robot *hybrid*. Pada bagian ini robot *hybrid* harus memanjat tiang dan memasang *propeller* pada bagian *engine*.

Penelitian sebelumnya tentang *pole climbing robot* yang dilakukan oleh Jean-Christophe Fauroux dan Joel Morillon [1] menggunakan beban robot sebagai tumpuan yang menekan pipa. Prinsip yang digunakan adalah meletakkan beban yang berat di bagian belakang dan menjepit pipa bagian atas dan bawah memiliki sudut tertentu. Namun pada sistem ini robot tidak bisa berputar di tiang dan bergerak naik turun karena tidak memiliki mekanisme pemutar di tiang dan penjepit lalu hanya memanfaatkan beban sebagai *self locking*.

Pada penelitian ini digunakan sistem pemanjat menggunakan 4 buah roda polyurethane dan 3 buah motor

*power window* sebagai pemanjat. 2 roda sebagai penggerak bagian atas dan 2 roda sebagai sebagai penggerak bagian bawah. Kemudian akan dikontrol menggunakan metode PD dengan menggunakan EDF dan *feedback* dari *rotary encoder*. Lalu setelah berada di atas, robot akan memasang *propeller* ke *engine*.

## II. DASAR TEORI

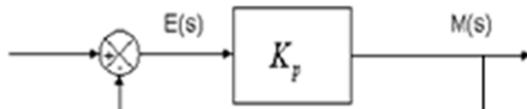
### A. Kontroler PD

Di dalam suatu sistem kontrol kita mengenal adanya beberapa macam aksi kontrol, diantaranya yaitu aksi kontrol proporsional, aksi kontrol integral dan aksi kontrol derivatif. Masing-masing aksi kontrol ini mempunyai keunggulan-keunggulan tertentu, dimana aksi kontrol proporsional mempunyai keunggulan *rise time* yang cepat, aksi kontrol integral mempunyai keunggulan untuk memperkecil *error*, dan aksi kontrol derivatif mempunyai keunggulan untuk memperkecil *error* atau meredam *overshot/undershot* [2]. Untuk itu agar kita dapat menghasilkan *output* dengan *rise time* yang cepat dan *error* yang kecil, kita dapat menggabungkan ketiga aksi kontrol ini menjadi aksi kontrol Proporsional Integral Derivatif (PID).

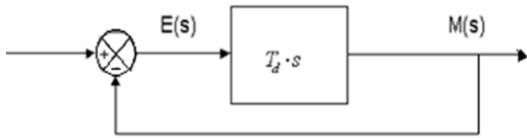
Pada penelitian ini kontrol yang akan digunakan adalah kontrol PD. Hal tersebut dilakukan berdasarkan penelitian dari Swati Singh dan Ankita Kosti [3] bahwa kontrol PD sudah dapat membuat sistem stabil meskipun ada kemungkinan sistem memiliki *steady state error*.

#### 1) Kontroler Proporsional

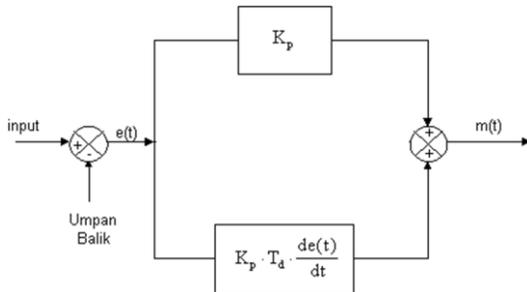
Kontroler proporsional [2] memiliki keluaran sinyal kontrol yang sebanding/proposional dengan besarnya perubahan nilai *error* (selisih antara nilai *set point* sistem dengan nilai aktual pembacaan sensor). Dapat dikatakan, bahwa keluaran kontroler proporsional merupakan perkalian antara nilai konstanta proporsional dengan nilai *error*. Sehingga kontroler proporsional memberi pengaruh langsung (sebanding) pada *error*, yaitu semakin besar *error*, maka semakin besar sinyal kendali yang dihasilkan kontroler proporsional. Gambar 1 menunjukkan hubungan antara sinyal *error* dan keluaran kontroler.



Gambar 1. Diagram blok kontroler proporsional



Gambar 2. Diagram blok kontroler diferensial



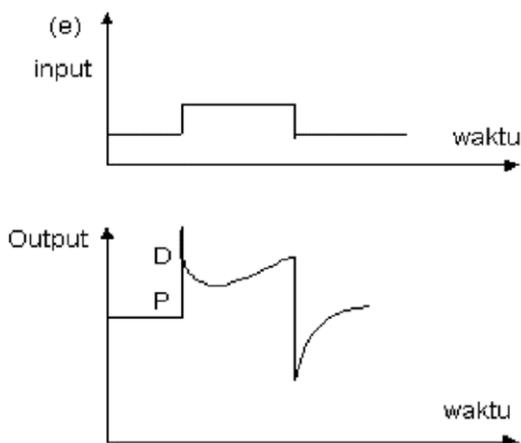
Gambar 3. Diagram blok kontroler PD

## 2) Kontroler Diferensial

Besarnya sinyal kontrol sebanding dengan perubahan *error* ( $d_e$ ). Semakin cepat *error* berubah, semakin besar aksi kontrol yang ditimbulkan. Keluaran kontroler diferensial memiliki sifat seperti halnya suatu operasi derivatif. Perubahan yang mendadak pada masukan kontroler, akan mengakibatkan perubahan yang sangat besar dan cepat. Gambar 2 menunjukkan diagram blok yang menggambarkan hubungan antara sinyal kesalahan dengan keluaran kontroler.

## 3) Kontroler Proporsional Diferensial

Setiap kekurangan dan kelebihan dari masing-masing kontroler P dan D dapat saling menutupi dengan menggabungkan keduanya secara paralel menjadi kontroler PD [2]. Elemen-elemen kontroler P dan D masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi



Gambar 4. Fungsi waktu pada kontroler PD

sebuah sistem, menghilangkan *overshoot* dan menghasilkan perubahan awal yang besar. Gambar 3 menunjukkan diagram blok kontroler PD.

Keluaran kontroler PD merupakan jumlahan dari keluaran kontroler proporsional dan keluaran kontroler diferensial. Gambar 4 menunjukkan hubungan tersebut.

Karakteristik kontroler PD sangat dipengaruhi oleh kontribusi besar dari kedua parameter P dan D. Penyetelan konstanta  $K_p$  dan  $T_d$  akan mengakibatkan penonjolan sifat dari masing-masing elemen. Salah satu konstanta tersebut dapat disetel lebih menonjol dibanding yang lain. Konstanta yang menonjol itulah akan memberikan kontribusi pengaruh pada respon sistem secara keseluruhan.

## III. PERANCANGAN SISTEM

### A. Perancangan Perangkat Keras

Pada penelitian ini akan dibuat robot dengan diagram blok seperti pada Gambar 5. Pada penelitian ini menggunakan *rotary encoder*, sensor garis, *limit switch* dan 2 potensiometer dan 2 motor *power window* sebagai pemanjat, dan EDF sebagai *steering*.

Sensor garis akan memberikan data *digital* kepada *microncontroller master* yang kemudian akan diolah oleh *microcontroller master* dengan menggunakan kontroler PD lalu akan memberikan *output* kepada *microcontroller slave* berupa kecepatan tiap motor supaya robot dapat mencapai *setpoint*. Pengiriman data dari *microcontroller master* ke *microcontroller slave* menggunakan sistem *broadcast* dimana master akan mengirimkan paket data ke semua *slave* secara bersamaan dalam interval waktu tertentu.

### B. Perancangan Mekanik

Desain robot menggunakan 4 roda dan EDF sebagai *steering* yang terlihat pada Gambar 6. Pada bagian depan terdapat *rotary encoder* yang digunakan untuk mengetahui posisi robot ketika memanjat tiang.

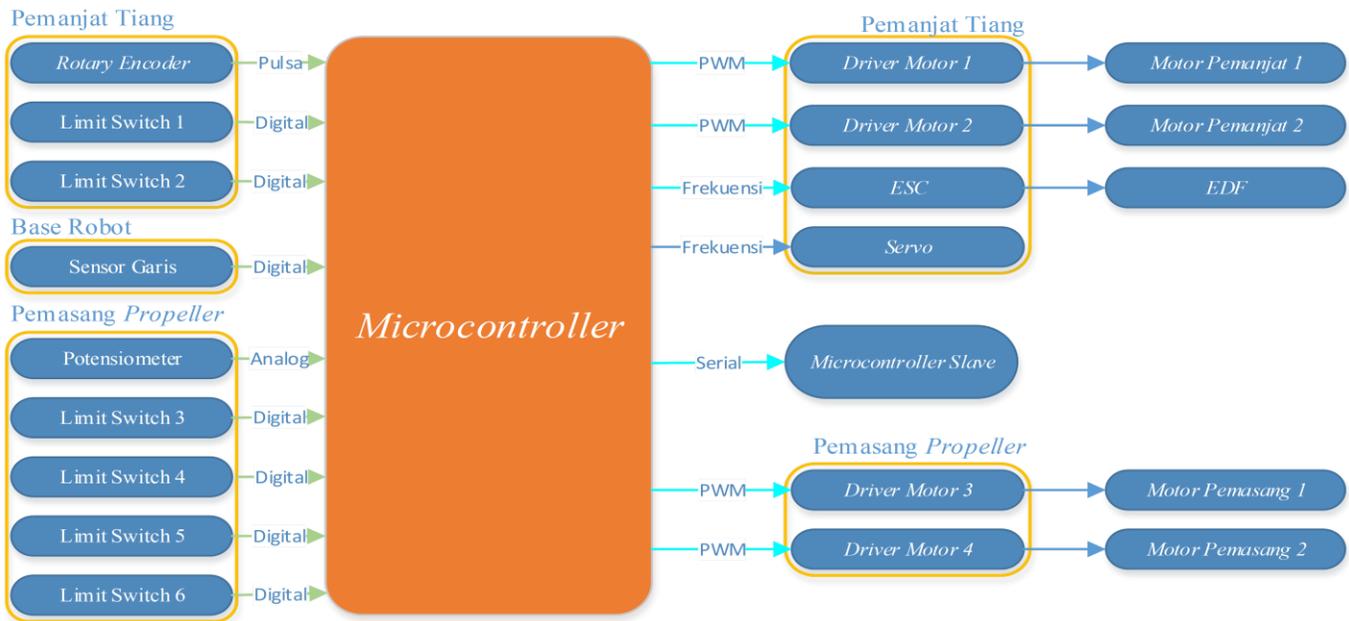
### C. Perancangan Perangkat Lunak

Posisi awal robot akan dipengaruhi oleh pembacaan sensor garis yang terletak di bagian depan robot. Sensor garis akan menghasilkan sejumlah data sehingga robot mengetahui nilai *error* posisi dari robot dan garis.

Pergerakan robot selanjutnya bergantung pada pembacaan pulsa *rotary encoder* untuk mendapatkan nilai sudut pergerakan robot pada tiang. Dengan nilai tersebut akan diketahui arah pergerakan robot yang dibentuk.

### D. Instrumen Penelitian

Untuk menguji robot yang telah dibuat, dilakukan skenario pengujian sebagai berikut: Skenario pertama yang akan dilakukan adalah meletakkan robot di posisi A. Kemudian robot akan bergerak menuju posisi B. Skenario selanjutnya adalah robot akan memanjat tiang dari titik B menuju titik C. Setelah mencapai posisi C maka robot akan memasang *propeller* pada *engine*. Perhatikan Gambar 7.



Gambar 5. Diagram blok robot

IV. PENGUJIAN DAN ANALISIS

A. Pengujian Sensor Posisi dengan Potensiometer

Pengujian dilakukan menggerakkan slider ke posisi kelipatan 2 cm. Kemudian diambil data tegangan dari output potensiometer setiap posisi yang ditentukan. Tabel I menampilkan data hasil pengujian yang telah dilakukan.

Posisi nol dari lengan robot ada pada tegangan 3,5 V, sedangkan posisi akhir dari lengan robot adalah 2,31 V. Rata-rata selisih tegangan setiap 2 cm adalah 0,09 V. Hasil pengujian menunjukkan bahwa output potensiometer cukup linear.

B. Pengujian Sensor Posisi dengan Rotary Encoder

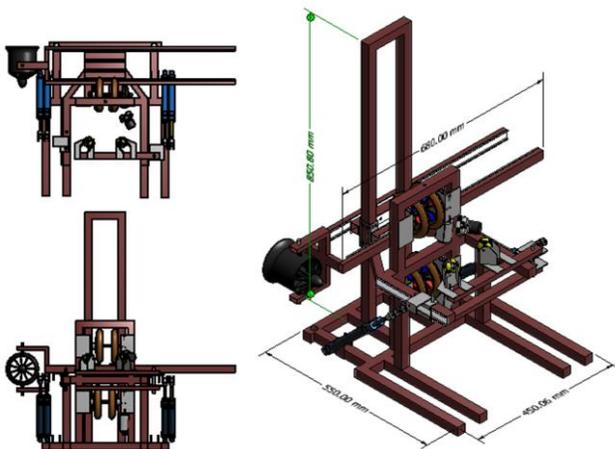
Pengujian pada rotary encoder dilakukan dengan memberi tanda berupa garis untuk mengetahui posisi sudut dari robot. Gambar 8 menunjukkan posisi tanda pada pipa yang diputar dan garis pada bagian robot.

Tabel II menunjukkan data rotary encoder yang telah

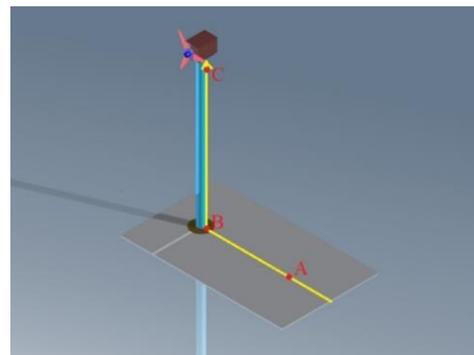
dikonversi menjadi sudut setiap 10°. Pulsa yang dihasilkan oleh rotary encoder dikonversi menjadi sudut (derajat). Dari data pembacaan yang telah diambil terdapat rata-rata error pembacaan sebesar 0,530.

C. Tuning Kontroler PD

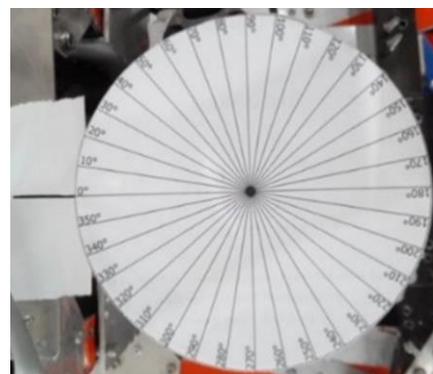
Untuk mencari konstanta proporsional dan konstanta diferensial yang dapat menghasilkan waktu steady state tercepat perlu dilakukan tuning PD. Proses tuning dilakukan



Gambar 6. Desain mekanik pemanjat



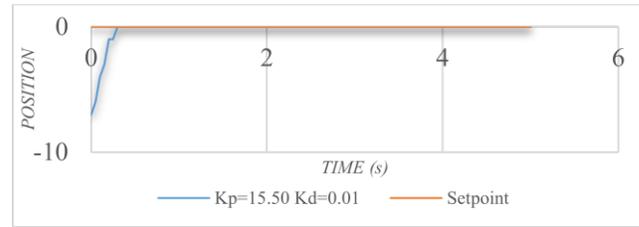
Gambar 7. Skenario posisi tujuan



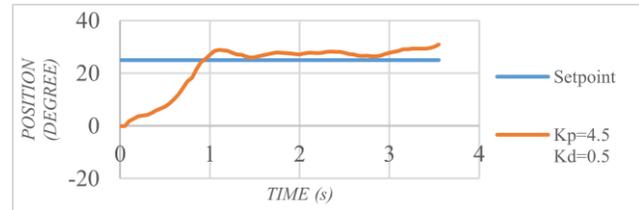
Gambar 8. Pengujian rotary encoder

TABEL I  
HASIL PENGUJIAN POTENSIO METER MULTITUNE

| Posisi (cm) | Tegangan (V) | Selisih Tegangan (V) |
|-------------|--------------|----------------------|
| 0           | 3,5          | 0                    |
| 2           | 3,4          | 0,1                  |
| 4           | 3,32         | 0,08                 |
| 6           | 3,2          | 0,12                 |
| 8           | 3,12         | 0,08                 |
| 10          | 3,03         | 0,09                 |
| 12          | 2,9          | 0,13                 |
| 14          | 2,81         | 0,09                 |
| 16          | 2,7          | 0,11                 |
| 18          | 2,63         | 0,07                 |
| 20          | 2,5          | 0,13                 |
| 22          | 2,4          | 0,1                  |
| 24          | 2,31         | 0,09                 |
| Rata - rata |              | 0,099166667          |



Gambar 9. Grafik tuning line follower terbaik



Gambar 10. Grafik tuning climbing pole terbaik

TABEL II  
DATA PENGUJIAN PULSA ENCODER

| Sudut Aktual (deg) | Sudut Pengukuran (deg) | Error (deg) |
|--------------------|------------------------|-------------|
| 0                  | 0                      | 0           |
| 30                 | 31,076                 | 1,076       |
| 60                 | 59,346                 | 0,654       |
| 90                 | 89,896                 | 0,104       |
| 120                | 119,29                 | 0,71        |
| 150                | 151,88                 | 1,88        |
| 180                | 179,47                 | 0,53        |
| 210                | 209,88                 | 0,12        |
| 240                | 240,1                  | 0,1         |
| 270                | 270,64                 | 0,64        |
| 300                | 299,6                  | 0,4         |
| 330                | 330,9                  | 0,9         |
| 360                | 360,38                 | 0,38        |
| Rata-rata          |                        | 0,53        |

TABEL III  
WAKTU PENGUJIAN LINE FOLLOWER

| Percobaan ke- | Waktu (s) |
|---------------|-----------|
| 1             | 0,5       |
| 2             | 0,4       |
| 3             | 0,4       |
| 4             | 0,4       |
| 5             | 0,4       |
| 6             | 0,4       |
| 7             | 0,4       |
| 8             | 0,35      |
| 9             | 0,35      |
| 10            | 0,3       |
| Rata - rata   | 0,39      |

dengan mengirimkan konstanta melalui komunikasi serial. Robot bergerak dan mengirimkan posisi aktual robot ke PC melalui komunikasi serial setiap 0,05 detik. Pengujian *tuning* PD dilakukan pada proses *line follower* dan *climbing*. Pada pengujian *tuning* PD, diketahui pengaruh dari setiap kontroler yaitu:

1. Semakin tinggi konstanta  $K_p$ , maka respon akan bertambah cepat begitu pula sebaliknya. Tetapi apabila konstanta  $K_p$  terlalu tinggi dapat menyebabkan osilasi dan *overshoot* berlebih.
2. Semakin tinggi konstanta  $K_d$  dapat mengurangi

TABEL IV  
PENGUJIAN CLIMBING POLE

| Percobaan ke- | Error (degree) | Waktu (s) | Pemasangan Propeller |
|---------------|----------------|-----------|----------------------|
| 1             | 2,43           | 6,35      | Berhasil Memasang    |
| 2             | 1,36           | 6,30      | Berhasil Memasang    |
| 3             | 6,97           | 6,25      | Gagal Memasang       |
| 4             | 7,73           | 7,05      | Gagal Memasang       |
| 5             | 5,91           | 7,35      | Gagal Memasang       |
| 6             | 1,36           | 6,15      | Berhasil Memasang    |
| 7             | 2,27           | 6,35      | Berhasil Memasang    |
| 8             | 11,07          | 6,70      | Gagal Memasang       |
| 9             | 0,91           | 6,30      | Berhasil Memasang    |
| 10            | 14,4           | 6,35      | Gagal Memasang       |
| 11            | 1,97           | 6,40      | Berhasil Memasang    |
| 12            | 4,85           | 6,35      | Gagal Memasang       |
| 13            | 3,33           | 6,15      | Berhasil Memasang    |
| 14            | 3,49           | 6,05      | Berhasil Memasang    |
| 15            | 2,85           | 6,15      | Berhasil Memasang    |
| 16            | 2,88           | 6,33      | Berhasil Memasang    |
| 17            | 4,85           | 6,25      | Gagal Memasang       |
| 18            | 3,33           | 6,30      | Berhasil Memasang    |
| 19            | 0,91           | 7,55      | Berhasil Memasang    |
| 20            | 1,21           | 7,60      | Berhasil Memasang    |
| Rata - rata   | 4,204          | 6,514     |                      |

*overshoot*, begitu pula sebaliknya. Tetapi apabila konstanta  $K_d$  terlalu tinggi dapat menyebabkan sistem menjadi tidak stabil.

Dari hasil pengujian *tuning* PD didapat nilai konstanta proporsional dan diferensial yang terbaik yang ditampilkan pada Gambar 9 dan Gambar 10.

#### D. Pengujian Line follower

Untuk menguji bagian *line follower* pada robot dilakukan percobaan dengan memposisikan robot pada posisi *error*. Kemudian robot dijalankan dengan menggunakan konstanta yang telah didapat pada saat percobaan *tuning* PD.

Dari pengujian yang telah dilakukan dapat diketahui bahwa waktu yang dibutuhkan untuk meluruskan diri dengan garis tidak selalu sama. waktu tercepat untuk mencapai *set point* adalah 0,3 detik dan waktu terlama adalah 0,5 detik. Rata-rata waktu yang dibutuhkan untuk memposisikan diri dengan garis adalah 0,39 detik.

TABEL V  
RATA-RATA WAKTU SUKSES MEMASANG *PROPELLER*

| Percobaan ke- | <i>Error (degree)</i> | Waktu (s) | Pemasangan <i>Propeller</i> |
|---------------|-----------------------|-----------|-----------------------------|
| 1             | 2,43                  | 6,35      | Berhasil Memasang           |
| 2             | 1,36                  | 6,3       | Berhasil Memasang           |
| 6             | 1,36                  | 6,15      | Berhasil Memasang           |
| 7             | 2,27                  | 6,35      | Berhasil Memasang           |
| 9             | 0,91                  | 6,3       | Berhasil Memasang           |
| 11            | 1,97                  | 6,4       | Berhasil Memasang           |
| 13            | 3,33                  | 6,15      | Berhasil Memasang           |
| 14            | 3,49                  | 6,05      | Berhasil Memasang           |
| 15            | 2,85                  | 6,15      | Berhasil Memasang           |
| 16            | 2,88                  | 6,33      | Berhasil Memasang           |
| 18            | 3,33                  | 6,3       | Berhasil Memasang           |
| 19            | 0,91                  | 7,55      | Berhasil Memasang           |
| 20            | 1,21                  | 7,6       | Berhasil Memasang           |
| Rata - rata   | 2,17                  | 6,46      |                             |

### E. Pengujian climbing Pole

Pengujian ini dilakukan dengan meletakkan robot di tiang lalu robot dijalankan pada *set point*  $0^\circ$ . Kemudian diambil data nilai *error* dari posisi robot dan waktu yang dibutuhkan untuk menyelesaikan tugasnya beserta keberhasilan robot dalam memasang *propeller*.

Untuk mengetahui tingkat keberhasilan robot dalam memasang *propeller*, Tabel V dapat dihasilkan berdasarkan Tabel IV.

Rata-rata waktu yang dibutuhkan untuk memasang *propeller* dengan benar adalah 6,46 detik. Rata-rata *error* posisi robot ketika berada diatas tiang adalah  $2,17^\circ$ . Persentase keberhasilan robot dalam memasang *propeller* adalah 65,0%.

## V. KESIMPULAN

Berdasarkan hasil analisis dan pembahasan maka dapat disimpulkan sebagai berikut:

1. Waktu tercepat untuk meluruskan diri dengan garis adalah 0.3 detik dengan konstanta  $K_p=15,5$  dan konstanta  $K_d=0,01$ .
2. Robot dapat memanjat tiang dengan rata-rata *error*  $2,17^\circ$  dengan konstanta  $K_p=4,5$  dan konstanta  $K_d=0,5$ .
3. Persentase tingkat keberhasilan robot dalam memasang *propeller* adalah 65,0% dan rata-rata waktu yang dibutuhkan untuk sukses memasang *propeller* adalah 6,46 detik.

## REFERENSI

- [1] Fauroux, J. C. and Morillon, J., "Design of a climbing Robot for Cylindro-conic Poles based on rolling self-locking", *Industrial Robot: An International Journal*, vol. 37(3), pp. 287-292, 2010.
- [2] Wicaksono, G., "Kontrol PID Pada Robot Bareleng 3.1", *Buku Tugas Akhir Diploma III, Jurusan Teknik Elektro, Politeknik Negeri Batam*, 2012.
- [3] Singh, S. and and Kosti, A., "Comparative Study Of Integer Order Pi-Pd Controller and Fractional Order Pi-Pd Controller of A DC Motor for Speed and Position Control", *International Journal of Electrical and Electronic Engineering & Telecommunications*, vol. 4(2), pp. 22-26, 2015.