

KENDALI JARAK JAUH POSISI MOTOR DC MENGGUNAKAN PID VIA NCS (NETWORKED CONTROL SYSTEM)

CONTROL OF DC MOTOR POSITION USING PID VIA NCS (NETWORKED CONTROL SYSTEM)

Andri Damar Hidayah¹, Agung Surya Wibowo², Erwin Susanto³

^{1,2,3} Universitas Telkom, Bandung

¹andridamarhd@student.telkomuniversity.ac.id, ²agungsw@tekomuniversity.ac.id,

³erwinelektro@telkomuniversity.ac.id

Abstrak

Penelitian ini bertujuan untuk membuat sistem kendali jarak jauh posisi motor DC menggunakan PID via NCS (*Networked Control System*). Dengan arsitektur NCS, alat master mengirimkan setpoint ke broker MQTT dan alat slave menerima setpoint dari broker MQTT yang bertugas untuk kendali posisi pada motor DC dengan PID dari setpoint yang diterima. Metode penelitian yang digunakan dalam penelitian ini adalah metode studi literatur, pengujian dan analisis terhadap kinerja sistem. Dari data hasil penelitian yang diperoleh bahwa alat yang dirancang memiliki rata-rata akurasi 1,73% derajat dan presisi rata-rata standar deviasi 0,56 derajat. Dengan nilai K_p 30 dan K_d 2,5 mendapatkan grafik respon sistem yang baik. Respon waktu delay dari jaringan internet untuk slave menerima setpoint membutuhkan waktu rata-rata 0,925 detik. Berdasarkan grafik respon hasil pengujian setpoint dan respon sistem closed loop bahwa error steady state 0%, settling time 0,2 detik dan tidak memiliki overshoot.

Keywords: NCS, MQTT, PID, Broker, Master, Slave

Abstract

This study aims to create a DC motor position remote control system using PID via NCS (Networked Control System). With the NCS architecture, the master tool sends set points to the MQTT broker and the slave device receives the set points from the MQTT broker who is in charge of position control on the DC motor with the PID of the received setpoints. The research method used in this research is the method of literature study, testing and analysis of system performance. From the research data obtained that the tool designed has an average accuracy of 1.73 degrees and an average precision of a standard deviation of 0.56 degrees. With the value of K_p 30 and K_d 2.5 get a good system response graph. The response time delay from the internet network for the slave receiving the setpoint takes an average of 0.925 seconds. Based

on the graph of the response of the setpoint test results and the response of the closed loop system that the steady state error is 0%, the settling time is 0.2 seconds and has no overshoot.

Keywords: NCS, MQTT, PID, Broker, Master, Slave

1. Pendahuluan

Networked Control System (NCS) adalah sistem yang tersebar secara spasial di mana komunikasi antara sensor, aktuator dan pengendali terjadi melalui jaringan komunikasi[9]. Penggunaan *multi-purpose shared network* untuk menghubungkan elemen yang didistribusikan secara spasial menghasilkan arsitektur yang fleksibel dan umumnya mengurangi biaya pemasangan dan perawatan. Sudah banyak alat yang mengaplikasikan NCS [1]. Dengan menggunakan *Internet of Things* (IoT) untuk menghubungkan perangkat dan sensor industri melalui internet, memungkinkan pengguna pada industri untuk mendapatkan dan menggunakan informasi dari perangkat yang terhubung. Dalam lingkungan industri, protokol MQTT merupakan protokol yang protokol tingkat aplikasi yang paling terkenal. Sehingga, pada industri yang berbasis IoT seseorang dapat mengendalikan dan survei dari mana saja di dunia[3].

Motor DC digunakan dalam berbagai macam aplikasi seperti pada industri, robotika karena kesederhanaannya, kemudahan aplikasi, keandalan dan hemat biaya. Kontrol posisi motor DC sangat penting dalam aplikasi untuk sistem kontrol presisi. Dengan kontrol posisi motor DC menggunakan pengontrol PID maka keluaran yang dihasilkan sangat tepat dengan sudut yang diinginkan. Stabilitas kinerja dengan menggunakan pengontrol PID dapat diterapkan untuk sistem kontrol posisi lengan robot dan aplikasi industri lainnya[2].

Pada tugas akhir ini, penulis akan membuat kendali jarak jauh posisi motor DC menggunakan PID via NCS. Dengan mengaplikasikan NCS pada kendali posisi motor DC menggunakan PID dan kendali posisi pada motor DC dapat dikendalikan dari mana saja di dunia selama perangkat terhubung

dengan jaringan internet. Menggunakan protokol MQTT untuk komunikasi pada jaringan internet.

2. Dasar Teori

2.1 Pengaturan Posisi pada Motor DC Close Loop

Untuk pengaturan posisi pada Motor DC close loop menggunakan kontrol PID. PID (Proportional-Integral-Derivative) merupakan controller untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut. Komponen dari PID terdiri dari 3 jenis yaitu Proportional, Integral dan Derivative. Dari ketiga jenis tersebut dapat digunakan secara bersamaan maupun sendiri tergantung dari kebutuhan respon yang kita inginkan pada plant tersebut[6].

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (1)$$

Keterangan:

u(t) = Keluaran controller

Kp = Gain proportional

Ki = Gain integral

Kd = Gain derivative

e = Error (Y_{sp} - Y_m)

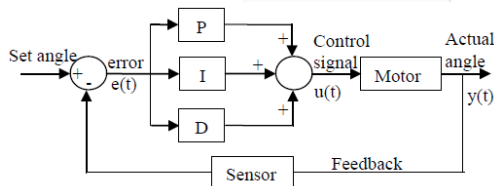
Y_{sp} = Setpoint

Y_m = Variabel proses

T = waktu

∫ = Variabel integrasi, nilai diambil dari waktu nol sampai t.

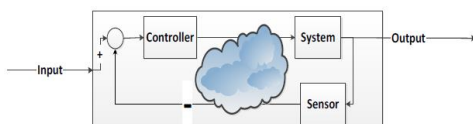
Gambar 1. Memperlihatkan desain PID controller untuk system control. Persamaan PID dihitung untuk mendapatkan sinyal control untuk motor DC[2].



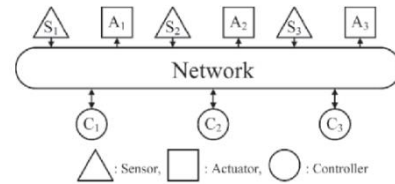
Gambar 1. Close Loop Kendali Posisi Motor DC menggunakan PID controller

2.2 Networked Control System (NCS)

Networked Control System (NCS) adalah sistem kendali umpan balik dimana elemen dari sistem yang tersambung satu sama lain akan melibatkan jaringan komunikasi. Misalnya antara sensor dan actuator, akan tersambung ke pengendali melalui jaringan komunikasi, berikut adalah skema dari NCS yang akan ditunjukkan pada gambar 2 dan struktur dari NCS pada gambar 3[8].

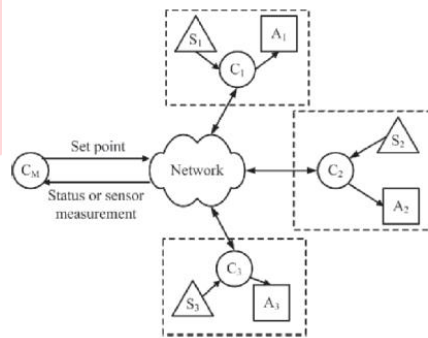


Gambar 2. Skematik NCS



Gambar 3. Struktur NCS

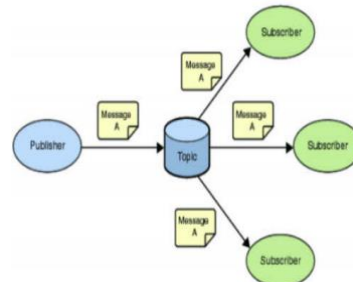
Hierarchical NCS dapat mempunyai banyak sub sistem yang berasal dari struktur hierarchical. Disetiap sub sistem diperlukan remote controller untuk mengendalikan setiap subsistem dan pengendali keseluruhan sistem. Ilustrasi ada gambar 4[8].



Gambar 4. Hierarchical NCS

2.3 Protokol MQTT

MQTT merupakan singkatan dari Message Queuing Telemetry Transport. MQTT merupakan protokol Internet of Things (IoT) yang berbasis open source dengan standar terbuka yang dirancang untuk perangkat terbatas dan bandwidth rendah, dengan latency tinggi atau berjalan pada jaringan yang tidak dapat diandalkan. MQTT merupakan salah satu protokol yang dibutuhkan sehingga NCS dapat bekerja pada alat[5].



Gambar 5. Prinsip Kerja Protokol MQTT

2.4 Arduino Uno

Arduino Uno adalah sebuah papan microcontroller yang berbasis ATmega328. Arduino mempunyai 14 pin input/output dimana 6 pin dapat digunakan sebagai output PWM, 6 input analog, crystal osilator 16 MHz, power jack, koneksi USB, kepala ICSP, dan tombol reset.

Microcontroller ini cukup sederhana dengan penggunaannya hanya menghubungkan ke komputer melalui koneksi USB[9].

2.5 ESP8266

ESP8266 merupakan modul wifi yang berfungsi sebagai perangkat tambahan mikrokontroler seperti arduino untuk bisa terhubung langsung dengan koneksi wifi dan membuat koneksi TCP/IP. Modul ini membutuhkan daya sekitar 3.3v dan memiliki 3 mode yaitu *station*, *access point* dan keduanya secara bersamaan. Dilengkapi dengan prosesor, memori dan GPIO[10].

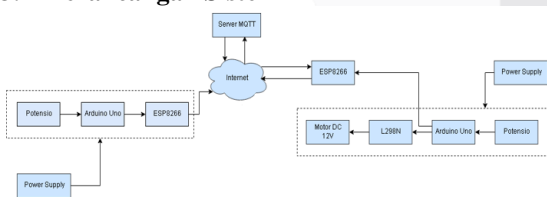
2.6 Motor Driver

Dengan menggunakan *Driver Motor* kecepatan motor DC dapat dikendalikan dengan menggunakan sinyal *Pulse Width Modulation (PWM)*. Dengan menggunakan sinyal PWM tegangan yang diaplikasikan ke kumparan motor DC dapat diubah. Perubahan tegangan menyebabkan perubahan kecepatan putaran motor DC. Pada banyak aplikasi regulasi tegangan sederhana akan menyebabkan banyak *power loss* pada rangkaian kendali. Oleh karena itu PWM banyak digunakan pada aplikasi pengendali motor DC. Tetapi, naik dan turun yang cepat memastikan bahwa perangkat daya semikonduktor dihidupkan atau dimatikan secepat mungkin untuk meminimalkan waktu transisi *switching*.

2.7 Potensiometer

Potensiometer adalah salah satu alat yang memanfaatkan prinsip pembagi tegangan. Pembagi tegangan secara sederhana dibentuk oleh rangkaian seri dari dua buah hambatan, dengan sebuah suplai tegangan. Diantara kedua hambatan tersebut, diambil sebuah jalur yang akan digunakan sesuai keperluan kita, misalnya sebagai inputan ke *microcontroller*.

3. Perancangan Sistem

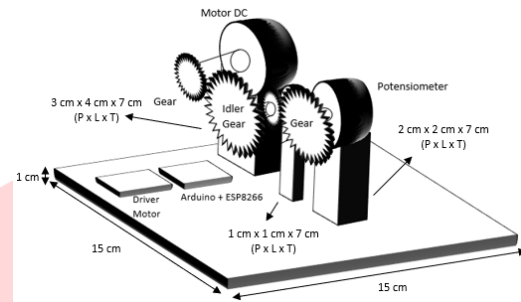


Gambar 6. Diagram Blok Sistem

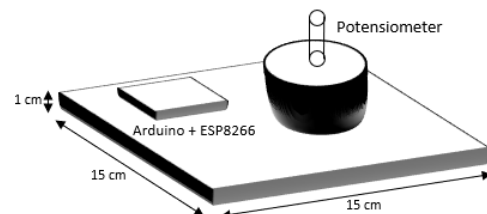
3.1 Perancangan Perangkat Keras

Pada kendali jarak jauh posisi motor DC menggunakan PID via NCS memiliki 2 desain mekanik yaitu desain mekanik *master* dan *slave*. Untuk desain mekanik *master* terdapat panel berbahan akrilik untuk tempat arduino, potensiometer dan esp8266. Pada desain mekanik *slave* terdapat panel berbahan akrilik. Pada *slave* memiliki tempat yang berfungsi untuk menahan motor DC, gear dengan 8 gigi gear dan motor driver L298N untuk mengendalikan motor DC. Potensiometer memiliki tempat untuk

mempertahankan potensiometer dan gear 40 gigi. Terdapat *idler gear* antara motor DC dan potensiometer yang berfungsi untuk mendapatkan rasio gear yang diinginkan dengan menggunakan gear yang ada. Gear 8 gigi pada motor DC ke *idler gear* 24 gigi dan *idler gear* 12 gigi ke gear 40 gigi pada potensiometer untuk mendapatkan rasio 1:10. Rasio 1:10 digunakan untuk *step down gear* sehingga kecepatan putaran motor DC untuk memutar potensiometer lebih lambat. ESP8266 pada *slave* berfungsi untuk memperoleh data dari server MQTT.



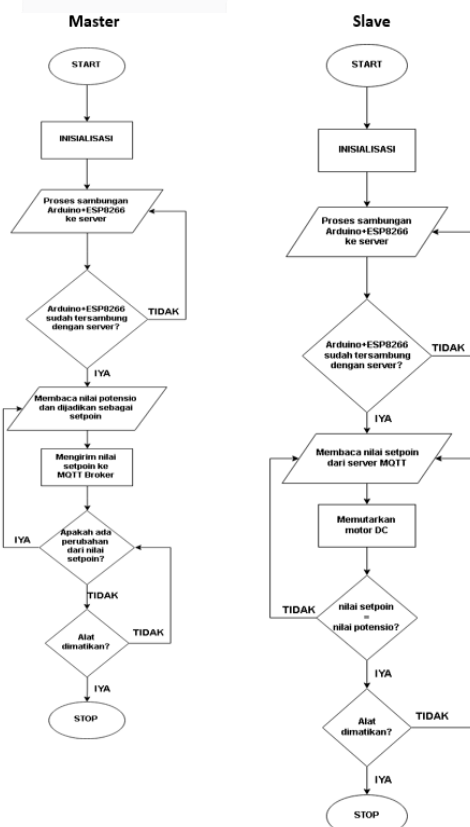
Gambar 7. Desain Perangkat Keras Slave



Gambar 8. Desain Perangkat Keras Master

3.2 Perancangan Perangkat Lunak

Berikut merupakan *flowchart* pada system:



Gambar 9. Flowchart Sistem

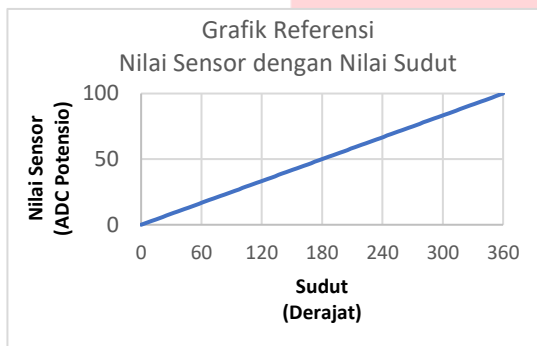
4. Hasil Pengujian dan Analisis

4.1 Pengujian Sensor Kontrol Posisi

Untuk pengujian sensor kontrol posisi motor DC ini dilakukan dengan pengujian akurasi dan pengujian presisi untuk mengetahui tingkat akurasi dan presisi dari *sensor* yang digunakan. *Sensor* yang digunakan untuk kontrol posisi motor DC adalah potensio.

4.1.1 Pengujian Akurasi

Pengujian akurasi dilakukan dengan membandingkan output sudut aktual yang dihasilkan oleh alat dan sudut acuan dengan sensor sesuai dengan grafik pada gambar 10. Pada alat ini saya mengubah nilai sensor yaitu nilai ADC potensio yang terbaca di arduino menjadi nilai sudut seperti di tabel berikut.



Gambar 10. Grafik Referensi Nilai Sensor dengan Nilai Sudut

Pada gambar 10 adalah referensi atau acuan untuk mengetahui berapa nilai sensor alat pada sudut tertentu. Nilai ADC potensio yang terbaca pada arduino yaitu 0-1023 ADC, karena menggunakan potensio 10 kali putaran penuh maka satu putaran penuh bernilai 0-102.3, tetapi untuk acuan nilai sensor yang akan dijadikan nilai sudut saya hanya menggunakan nilai ADC 0-100.

Sebelum saya memulai percobaan saya melakukan kalibrasi dimana sudut 0 derajat bernilai 500 ADC pada sensor. Setelah dilakukan pengujian untuk mendapatkan output sudut yang dihasilkan oleh alat, dari data tersebut dicari berapa persentase error yang didapat dengan membandingkan dengan sudut acuan. Berikut adalah rumus untuk mencari persentase error.

$$\%error = \frac{|approx - exact|}{exact} 100\% \quad (2)$$

Keterangan:

%error = Persentase Error (%)

Approx = nilai estimasi (Sudut Acuan)

Exact = nilai aktual (Sudut Aktual)

Setelah mendapatkan persentase error maka dihitung rata-rata error yang dihasilkan, semakin

kecil nilai error atau mendekati nol maka alat dapat terbilang akurat. Berikut adalah hasil data pengujian akurasi yang dihasilkan pada alat.

Tabel 1. Data Pengujian Akurasi

| Sudut (Derajat) | | Error (%) |
|--------------------|--------|-----------|
| Acuan | Aktual | |
| 18 | 19 | 5,56 |
| 36 | 37 | 2,78 |
| 54 | 55 | 1,85 |
| 72 | 73.5 | 2,08 |
| 90 | 91 | 1,11 |
| 108 | 111 | 2,78 |
| 126 | 129 | 2,38 |
| 144 | 147 | 2,08 |
| 162 | 167 | 3,09 |
| 180 | 183 | 1,67 |
| 198 | 201 | 1,52 |
| 216 | 218 | 0,93 |
| 234 | 234 | 0 |
| 252 | 251 | 0,4 |
| 270 | 268 | 0,74 |
| 288 | 284 | 1,39 |
| 306 | 302 | 1,31 |
| 324 | 320 | 1,23 |
| 342 | 339 | 0,88 |
| 360 | 357 | 0,83 |
| Rata-rata Error(%) | | 1,73 |

Berdasarkan tabel 1, hasil pengujian dari dua puluh sudut yang berbeda didapatkan persentase error dan dihitung rata-rata error yaitu 1,73%. Ada beberapa faktor yang mempengaruhi nilai error pada alat, yaitu poros pada jarum sudut yang tidak stabil dan penempatan jarum sudut pada poros.

4.1.2 Pengujian Presisi

Pengujian presisi dilakukan untuk mengetahui apakah output dari alat stabil. Pengujian presisi dilakukan dengan melakukan percobaan berkali-kali dengan memasukkan setpoint yang sama pada alat. Kemudian, dicari nilai rata-rata output yang dihasilkan dari semua percobaan dan dihitung nilai standar deviasi dan relatif standar deviasi. Untuk mencari standar deviasi digunakan rumus sebagai berikut[12]:

$$s = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N - 1}} \quad (3)$$

Keterangan:

s = standar deviasi

N = Jumlah data

x_i = data ke-i (i= 1,2,3...N)

\bar{x} = rata-rata sampel

Rumus relatif standar deviasi adalah:

(4)

$$RSD = \frac{s}{\bar{x}} \times 100\%$$

Keterangan:

RSD = Relatif Standar Deviasi

s = standar deviasi

\bar{x} = rata-rata sampel

Untuk pengujian ini dilakukan pada sudut 90, 180 dan 360 derajat. Masing-masing sudut akan dilakukan percobaan sebanyak 30 kali. Berikut adalah tabel data hasil pengujian presisi pada alat.

Tabel 2. Data Pengujian Presisi pada Sudut 90 Derajat

| No | Sudut Aktual (Derajat) | No | Sudut Aktual (Derajat) | No | Sudut Aktual (Derajat) |
|-----------------|------------------------|-------|------------------------|----|------------------------|
| 1 | 92 | 11 | 91 | 21 | 91 |
| 2 | 91 | 12 | 91 | 22 | 92 |
| 3 | 91 | 13 | 91 | 23 | 91 |
| 4 | 92 | 14 | 91 | 24 | 91 |
| 5 | 92 | 15 | 92 | 25 | 92 |
| 6 | 92 | 16 | 91 | 26 | 92 |
| 7 | 92 | 17 | 91 | 27 | 92 |
| 8 | 92 | 18 | 91 | 28 | 92 |
| 9 | 91 | 19 | 91 | 29 | 92 |
| 10 | 90 | 20 | 92 | 30 | 92 |
| Rata-Rata | | 91,47 | | | |
| Standar Deviasi | | 0,57 | | | |
| RDS | | 0,62% | | | |

Tabel 3. Data Pengujian Presisi pada Sudut 180 Derajat

| No | Sudut Aktual (Derajat) | No | Sudut Aktual (Derajat) | No | Sudut Aktual (Derajat) |
|-----------------|------------------------|--------|------------------------|----|------------------------|
| 1 | 179 | 11 | 180 | 21 | 180 |
| 2 | 180 | 12 | 180 | 22 | 180 |
| 3 | 179 | 13 | 180 | 23 | 180 |
| 4 | 181 | 14 | 180 | 24 | 180 |
| 5 | 180 | 15 | 180 | 25 | 180 |
| 6 | 180 | 16 | 180 | 26 | 179 |
| 7 | 180 | 17 | 179 | 27 | 179 |
| 8 | 181 | 18 | 180 | 28 | 180 |
| 9 | 180 | 19 | 181 | 29 | 180 |
| 10 | 179 | 20 | 181 | 30 | 180 |
| Rata-Rata | | 179,93 | | | |
| Standar Deviasi | | 0,58 | | | |
| RDS | | 0,32% | | | |

Tabel 4. Data Pengujian Presisi pada Sudut 360 Derajat

| No | Sudut Aktual (Derajat) | No | Sudut Aktual (Derajat) | No | Sudut Aktual (Derajat) |
|----|------------------------|----|------------------------|----|------------------------|
| 1 | 360 | 11 | 360 | 21 | 361 |
| 2 | 360 | 12 | 359 | 22 | 360 |
| 3 | 360 | 13 | 359 | 23 | 360 |
| 4 | 361 | 14 | 360 | 24 | 360 |
| 5 | 361 | 15 | 360 | 25 | 360 |
| 6 | 360 | 16 | 360 | 26 | 360 |

| | | | | | |
|-----------------|-----|--------|-----|----|-----|
| 7 | 360 | 17 | 360 | 27 | 360 |
| 8 | 360 | 18 | 359 | 28 | 360 |
| 9 | 360 | 19 | 361 | 29 | 360 |
| 10 | 360 | 20 | 361 | 30 | 360 |
| Rata-Rata | | 360,07 | | | |
| Standar Deviasi | | 0,52 | | | |
| RDS | | 0,14% | | | |

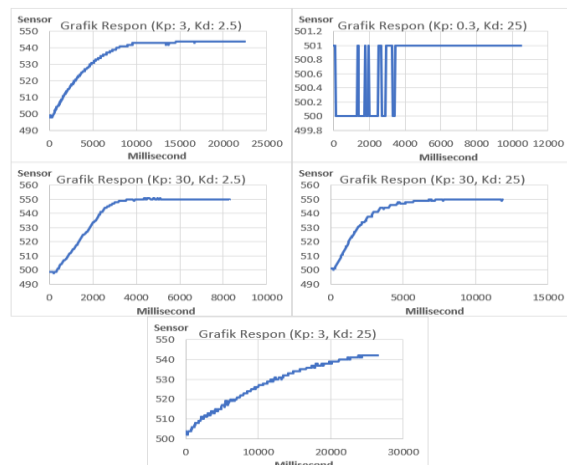
Berdasarkan tabel 2 dengan output sudut 90 derajat dengan pengujian sebanyak 30 kali menghasilkan data dengan rata-rata sudut aktual yaitu 91,47 derajat dan setelah dihitung standar deviasi yang didapat yaitu 0,57 derajat dan relatif standar deviasi 0,62%. Pada tabel 3 output sudut 180 derajat sebanyak 30 kali pengujian menghasilkan data dengan rata-rata sudut aktual yaitu 179,93 derajat dan standar deviasi 0,58 derajat dan relatif standar deviasi 0,32%. Pada tabel 4 output sudut 360 derajat sebanyak 30 kali pengujian menghasilkan data dengan rata-rata sudut aktual yaitu 360,07 derajat dan standar deviasi 0,52 derajat dan relatif standar deviasi 0,14%. Rata-rata standar deviasi dari ketiga pengujian di atas adalah 0,56 derajat. Bisa disimpulkan bahwa presisi pada alat lumayan baik karena nilai standar deviasi kurang dari 1 derajat dan relatif standar deviasi kurang dari 1%. Faktor yang mempengaruhi nilai presisi pada alat sama dengan faktor yang mempengaruhi nilai akurasi pada alat.

4.2 Pengujian Respon Sistem Closed Loop

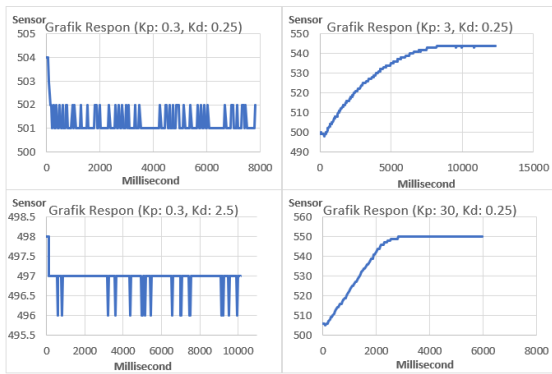
Pengujian ini dilakukan dengan menggunakan beberapa nilai Kp dan Kd kemudian dibandingkan hasil data yang didapatkan untuk mengetahui hasil respon melalui grafik respon. Setelah melakukan pengujian dengan metode *trial and error* untuk mendapatkan nilai Kp dan Kd yang stabil untuk alat. Memperoleh nilai Kp dan Kd dari yang kecil, sedang dan besar dari hasil metode *trial and error*. Berikut nilai Kp dan Kd yang akan digunakan untuk pengujian ini.

Tabel 5. Nilai Kp dan Kd

| Kp | Kd |
|-----|------|
| 0,3 | 0,25 |
| 3 | 2,5 |
| 30 | 25 |



Gambar 11. Grafik Respon Uji Closed-Loop 1



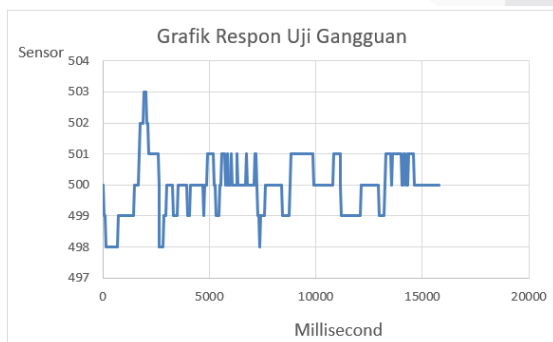
Gambar 12. Grafik Respon Uji Closed-Loop 2

Dari semua pengujian yang dilakukan dengan 9 kombinasi Kp dan Kd yang berbeda bisa terlihat grafik respon memperlihatkan hasil yang berbeda-beda. Semakin besar nilai Kp maka rise time dari grafik respon lebih cepat untuk mencapai setpoint. Pada nilai Kp 0.3 alat tidak dapat berjalan menuju setpoint dikarenakan nilai tersebut terlalu rendah dibawah nilai minimum PWM agar motor DC dapat berjalan pada alat tersebut. Semakin besar nilai Kd maka ripple pada grafik respon semakin banyak.

Kp dan Kd yang optimal untuk alat ini bisa terlihat pada pengujian ke-6 dengan nilai Kp yaitu 30 dan nilai Kd yaitu 2.5, alasan kenapa nilai Kp dan Kd tersebut optimal yaitu grafik respon rise timenya lebih cepat mencapai setpoint dan ripple tidak terlalu besar dan sedikit.

4.3 Pengujian Gangguan

Pengujian ini dilakukan untuk mengetahui bila diberi gangguan pada sensor kontrol posisi apakah alat tersebut akan kembali ke setpoint atau tidak. Pengujian dilakukan dengan nilai setpoint yaitu 500 dan sensor akan diberi gangguan dari luar seperti diputarakan sensor kontrol posisi dengan tangan, untuk menunjukkan hasil dari pengujian ini akan ditunjukkan grafik respon hasil uji gangguan sebagai berikut.

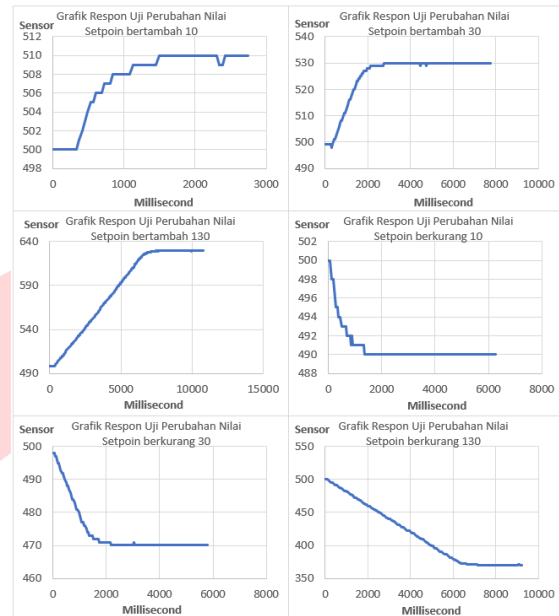


Gambar 13. Grafik Respon Uji Gangguan

4.4 Pengujian Setpoint

Pengujian setpoint dilakukan dengan menaikkan nilai setpoint dan menurunkan nilai setpoint.

Kenaikkan nilai setpoint dilakukan dengan menaikkan nilai setpoint dengan jarak yang berbeda-beda dari naiknya nilai yang kecil, sedang dan besar. Untuk penurunan nilai setpoint dilakukan hal yang sama dari jarak turun yang kecil, sedang dan besar. Pada pengujian ini nilai setpoint awal yaitu 500. Untuk perubahan setpoint yang naik yaitu ke 510, 530 dan 630. Perubahan setpoint turun yaitu ke 490, 470 dan 370.



Gambar 14. Grafik Respon Uji Setpoint

Terlihat dari grafik respon hasil pengujian perubahan setpoint, semakin besar jarak perubahan naik atau turunnya nilai setpoint maka semakin lama mencapai nilai setpoint. Karena semakin besar jarak dari setpoint awal ke setpoint yang diinginkan membutuhkan waktu yang lebih lama untuk mencapai setpoint yang diinginkan.

4.5 Pengujian Waktu Mengirim dan Menerima Data

Pengujian waktu mengirim dan menerima data dilakukan untuk mengetahui berapa lama waktu yang dibutuhkan alat master untuk mengirim setpoint ke server dan berapa lama waktu yang dibutuhkan untuk alat slave yaitu kontrol posisi motor DC untuk menerima setpoint dari server. Proses pengiriman dan penerimaan data ini memakai internet dengan koneksi wifi. Pengujian ini dilakukan dengan dua kondisi yaitu kondisi koneksi baik dan koneksi buruk.

Tabel 6. Data Pengujian Waktu Mengirim dan Menerima Data

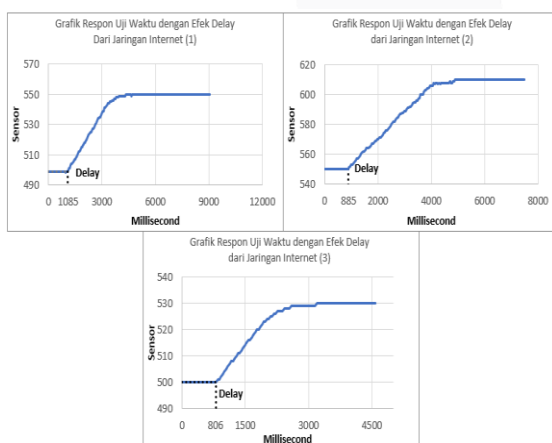
| Kondisi Baik | | | Kondisi Buruk | | |
|--------------|---------------|----------------|---------------|---------------|----------------|
| Uji | Kirim (detik) | Terima (detik) | Uji | Kirim (detik) | Terima (detik) |
| 1 | 0,574 | 0,459 | 1 | 1,550 | 2,017 |
| 2 | 0,519 | 0,919 | 2 | 1,332 | 1,417 |
| 3 | 0,612 | 0,965 | 3 | 2,652 | 1,406 |

| | | | | | |
|-----------|--------|--------|-----------|--------|--------|
| 4 | 0,576 | 0,857 | 4 | 2,145 | 2,690 |
| 5 | 0,588 | 0,688 | 5 | 3,154 | 3,504 |
| 6 | 0,622 | 0,447 | 6 | 1,436 | 2,440 |
| 7 | 0,549 | 0,969 | 7 | 1,642 | 5,234 |
| 8 | 0,517 | 1,325 | 8 | 0,742 | 1,201 |
| 9 | 0,554 | 0,450 | 9 | 0,945 | 0,647 |
| 10 | 0,518 | 1,518 | 10 | 3,023 | 2,641 |
| Rata-rata | 0,5629 | 0,8597 | Rata-rata | 1,8621 | 2,3197 |

Berdasarkan tabel 6 rata-rata waktu yang dibutuh untuk mengirim dan menerima data pada saat kondisi jaringan stabil kurang dari 1 detik. Sedangkan, pada saat kondisi jaringan buruk waktu yang dibutuhkan untuk mengirim dan menerima data kurang lebih 2 detik. Bisa disimpulkan bahwa pada saat kondisi internet lemah dan tidak stabil, proses pengiriman dan penerimaan data lumayan lambat, waktu tidak teratur dan pada saat penulis melakukan pengujian sering putus koneksi dari alat ke internet.

4.6 Pengujian Respon Waktu dengan Efek Delay dari Jaringan Internet

Pengujian ini dilakukan untuk mengetahui berapa lama waktu *delay* pada alat *slave* untuk mendapatkan nilai setpoint dari *broker* MQTT yang dikirim oleh *master*. Grafik respon pada pengujian ini dimulai dari mengubah setpoint dari *master* hingga *slave* mencapai setpoint akhir yang didapat dari *master*. Untuk mendapatkan waktu *delay* yang dibutuhkan *slave* untuk mendapatkan nilai setpoint dilihat dari grafik respon dari waktu 0 detik hingga grafik mulai naik ke setpoint akhir.



Gambar 15. Grafik Respon Uji Respon Waktu dengan Efek Delay dari Jaringan Internet

Dari grafik respon pada gambar 15, waktu respon dengan efek *delay* dari jaringan internet yang dibutuhkan *slave* untuk mendapatkan setpoint pada pengujian pertama adalah 1085 *millisecond* atau 1,085 detik, pengujian kedua adalah 885 *millisecond* atau 0,885 detik dan pengujian ketiga adalah 806 *millisecond* atau 0,806 detik. Rata-rata *delay* dari ketiga pengujian diatas yaitu 0,925 detik.

Dari ketiga pengujian tersebut bisa terbilang bahwa waktu respon dengan efek *delay* dari jaringan internet yang dibutuhkan *slave* untuk mendapatkan setpoint kurang lebih 1 detik. Faktor-faktor yang mempengaruhi waktu *delay* untuk *slave* mendapatkan nilai setpoint yaitu tergantung stabilitas koneksi dengan jaringan internet. Jika koneksi jaringan internet buruk maka waktu *delay* bisa lebih lama atau lebih dari 2 detik dan jika koneksi jaringan internet stabil dan lancar maka waktu *delay* bisa lebih cepat atau kurang dari 1 detik.

4. Kesimpulan dan Saran

4.1 Kesimpulan

Berdasarkan pengujian dan analisa yang telah dilakukan pada tugas akhir, maka didapatkan kesimpulan sebagai berikut:

1. Dengan alat *master* yang bertugas mengirimkan setpoint ke *broker* MQTT melalui jaringan internet dan alat *slave* menerima nilai setpoint dari *broker* MQTT melalui jaringan internet, kemudian melakukan kalkulasi PID untuk kontrol posisi motor DC. Maka, alat telah mengaplikasikan metode kendali posisi motor DC berbasis PID pada NCS menggunakan protokol MQTT dengan jaringan internet.
2. Melakukan beberapa pengujian untuk menguji kinerja sistem kendali yang menggunakan arsitektur NCS yaitu, pengujian akurasi dan presisi sensor, pengujian gangguan, pengujian setpoint, pengujian waktu mengirim dan menerima data, pengujian respon waktu dengan efek *delay* dari jaringan internet.
3. Berdasarkan hasil data pengujian, untuk pengujian akurasi sensor yaitu rata-rata error 1,73%. Pengujian presisi dengan standar deviasi dari tiga pengujian adalah 0,56 derajat. Pengujian respon sistem closed loop dengan K_p 30 dan K_d 2,5 menghasilkan grafik respon sistem yang bagus. Alat kembali ke setpoint awal jika diberi gangguan. Respon waktu *delay* dari jaringan internet untuk *slave* menerima setpoint berdasarkan tiga pengujian grafik respon waktu rata-rata membutuhkan 0,925 detik.

4.2 Saran

Berdasarkan saat proses pembuatan alat, pengujian alat dan analisa yang telah dilakukan, maka saran yang dapat diberikan adalah:

1. Menggunakan motor DC dengan RPM yang sangat kecil yaitu kurang lebih 10 RPM dan memiliki torsi yang sangat tinggi.
2. Poros jarum derajat harus stabil pada saat berputar sehingga pengukuran sudut pada jarum akan lebih akurat.
3. Menambahkan UI (User Interface) pada alat agar dapat mengubah nilai K_p , K_i , K_d dan setpoint pada alat sehingga tidak perlu

- mengubah nilai-nilai tersebut langsung pada program.
4. Untuk pengembangan alat selanjutnya, satu master dapat mengatur lebih dari satu alat slave.
 5. Kontroller langsung di master sehingga saat ingin mengubah program bisa langsung di master dan tidak perlu ke alat slave untuk mengubah program.

Referensi

- [1] J. Hespanha, P. Naghshtabrizi, dan Y. Xu, "A Survey of Recent Results in Networked Control Systems", *Proc. of the IEEE*, vol. 95, no. 1, January 2007.
- [2] M. M. Maun, M. M. Latt, C. M. Nwe, "DC Motor Angular Position Control using PID Controller with Friction Compensation", *IJSRP*, Vol. 8, Issue 11, November 2018.
- [3] H. Hasan, Bahaa. Mohammed, "Evaluation of MQTT Protocol for IoT Based Industrial Automation", *IJESC*, vol. 8, no. 12, December 2018.
- [4] D. Locke, "Introduction to MQTT", Webinar Presenters, 2013. [Online]. Available: <https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt>.
- [5] G. Yudha, A. Denha, F. Khusnu, F. Angga, F. Jati, "Penerapan Protokol MQTT pada Teknologi WAN (Studi Kasus Sistem Parkir Universitas Brawijaya)", *Jurnal Informatika Mulawarman*, vol. 12, no. 2, September 2017.
- [6] K. Ogata, "Modern Control Engineering 5th Edition". New Jersey: Prentice Hall, 2010.
- [7] N. S. Nise, "Control Systems Engineering 6th Edition". New Jersey: Wiley, 2011.
- [8] Fei-Yue Wang dan Derong Liu, "Networked Control Systems: Theory and Applications". US: Springer Publishing Company, 2008.
- [9] <https://www.arduino.cc/en/Main/ArduinoBoardUno> [Diakses 19 September 2019]
- [10] <https://www.electroschematics.com/wp-content/uploads/2015/02/esp8266-datasheet.pdf> [Diakses 23 September 2019]
- [11] <https://learn.sparkfun.com/tutorials/voltage-dividers> [Diakses 4 November 2019]
- [12] N. J. Salkind, "Exploring Research 8th Edition". London: Pearson, 2012.