

Nama : Kevin Bayu Pradana  
NIM : 224308009  
Kelas : TKA-6A

### Analisis

Pada tugas praktikum kali ini yaitu membuat sebuah sistem untuk mengontrol kecepatan motor DC dengan menggunakan PID berbasis IoT. Dalam praktikum ini alat dan bahan yang digunakan yaitu motor DC RF 300-EA, ESP32, motor driver, MQTT, Arduino IDE, dan Matlab. Langkah pertama yang dilakukan yaitu mencari spesifikasi dari motor DC yang digunakan, untuk spesifikasinya sebagai berikut:

Tegangan : 3,9V  
Kecepatan tanpa beban : 4400 r/min  
Tahanan armatur R : 10 Ohm  
Induktansi armatur L : 0.0002 Henry  
Momen inersia rotor J :  $3e-7 \text{ kg.m}^2$   
Koefisien redaman viskosa B :  $2e-6 \text{ Nm.s/rad}$   
Konstanta torsi Kt : 0.003 Nm/A  
Konstanta ggl balik Ke : 0.003 V.s/rad

Setelah itu, Motor dimodelkan dengan menggabungkan persamaan kelistrikan dan mekanik sebagai berikut:

- Persamaan kelistrikan (hukum Kirchhoff):
$$V(s) = LsI(s) + RI(s) + Ke\omega(s)$$
- Persamaan mekanik (hukum Newton):
$$J s\omega(s) + B\omega(s) = KtI(s)$$

Dengan eliminasi variabel  $I(s)$  diperoleh fungsi alih kecepatan sudut terhadap tegangan input sebagai:

$$G(s) = \omega(s) / V(s) = Kt / LsJ + (LB + RJ)s + (RB + KeKt)$$

Dengan substitusi nilai parameter, fungsi alih menjadi:

$$G(s) = 0,003 / 6 \times 10^{-11}s^2 + 6,06 \times 10^{-6}s + 6,09 \times 10^{-5}$$

Langkah selanjutnya setelah memperoleh fungsi alih sistem adalah melakukan simulasi respons dinamis motor serta merancang pengendali PID menggunakan MATLAB. Fungsi alih yang diperoleh, yaitu:

$$G(s) = 0,003 / 6 \times 10^{-11}s^2 + 6,06 \times 10^{-6}s + 6,09 \times 10^{-5}$$

Lalu diimplementasikan ke dalam MATLAB dalam bentuk fungsi transfer orde dua. Fungsi ini merepresentasikan hubungan antara tegangan masukan dan kecepatan sudut motor DC, yang menjadi dasar dalam proses desain pengendali. Perancangan pengendali PID dilakukan dengan menggunakan fitur *PID Tuning* di MATLAB. Melalui fitur ini, parameter kontroler Kp, Ki, dan Kd disesuaikan secara otomatis untuk mencapai performa sistem yang diinginkan, berdasarkan kriteria seperti waktu naik, waktu tunak, dan *overshoot*. Nilai parameter PID hasil tuning kemudian diimplementasikan ke dalam lingkungan *Simulink*.

Model Simulink terdiri atas blok *transfer function* untuk motor DC, blok *PID Controller*, dan sumber referensi kecepatan. Simulasi sistem tertutup dilakukan untuk mengevaluasi kinerja pengendali terhadap perubahan referensi serta kestabilan sistem terhadap gangguan. Integrasi MATLAB dan Simulink memungkinkan penyesuaian parameter secara simultan dan interaktif, sehingga proses tuning menjadi lebih efisien dan sistematis.

Setelah desain pengendali berhasil disimulasikan, tahap berikutnya adalah mengintegrasikan sistem dengan perangkat keras menggunakan mikrokontroler ESP32. ESP32 dikonfigurasi untuk mengendalikan motor DC melalui *motor driver* serta membaca kecepatan motor secara real-time menggunakan sensor seperti *encoder*. Komunikasi antara ESP32 dan MATLAB dilakukan melalui protokol MQTT, di mana data kecepatan dikirim dari ESP32 ke MATLAB untuk diproses dengan algoritma PID, lalu sinyal kendali (dalam bentuk nilai PWM) dikirim kembali ke ESP32.

Dengan pendekatan ini, MATLAB dan Simulink berperan sebagai platform utama dalam perancangan dan simulasi pengendali, sementara ESP32 bertindak sebagai unit eksekusi di sisi perangkat keras. Sistem ini menunjukkan penerapan pengendalian kecepatan motor DC berbasis PID secara real-time yang terintegrasi dengan teknologi IoT.