

Nama : Christian Riovaldo Labina

NIM : 24/533483/PA/22581

### Extra Jawaban

1. Open-loop system itu merupakan sistem kendali yang output saat ini hanya bergantung dengan input saat ini. Sedangkan close-loop system itu berarti output saat ini bergantung dengan input saat ini dan output sebelumnya. Close-loop system bisa mengoreksi eror untuk mencapai nilai yang diinginkan, sedangkan open-loop system tidak bisa mengoreksi eror.
2. Sistem yang aku buat termasuk ke dalam close-loop system. Dari segi komponen, servo itu salah satu komponen yang menggunakan sistem kendali close-loop. Misal kita ingin servo bergerak sesuai sudut yang kita inginkan, maka servo akan terus mengoreksi output sebelumnya berdasarkan sinyal PWM yang masuk hingga mencapai sudut yang diinginkan. Dari segi kode program, terdapat perintah:

```
roll += g.gyro.x*dt*180/PI;  
pitch += g.gyro.y*dt*180/PI;
```

yang berarti output roll dan pitch saat ini bergantung dengan input sensor saat ini (g.gyro.x dan g.gyro.y) dan output sebelumnya sehingga hasilnya lebih akurat dan sesuai dengan sistem close-loop dari servo.

3. Fungsi sensor:
  - MPU6050 berfungsi untuk membaca akselerometer dan giroskop. Akselerometer yaitu percepatan yang dideteksi sensor terhadap sumbu x, y, dan z. Sedangkan giroskop yaitu kecepatan rotasi yang dideteksi sensor secara roll, pitch, dan yaw.
  - Sensor PIR berfungsi untuk mendeteksi gerakan berdasarkan perubahan sinar infra merah pada objek yang bergerak.
4. Pinout:
  - Pin servo 1-5 masing-masing yaitu GPIO 19, 18, 17, 16, dan 4, alasannya karena pin-pin tersebut mendukung sinyal PWM dan juga berdekatan untuk memudahkan wiring.
  - Pin sensor PIR yaitu 26, alasannya karena mendukung pin out dari PIR dan aku bedakan di sisi kiri ESP32 untuk sensor agar memudahkan konfigurasi.
  - Pin SDA = 21 dan SCL = 21 karena default untuk komunikasi I2C pada ESP32.
  - Pin AD0 = GND, alasannya karena library Adafruit\_MPU6050.h umumnya pakai alamat 0x68 dan ketika AD0 dipasang ke GND akan otomatis merujuk alamat 0x68. Jika pakai dua MPU dalam 1 mikon, maka AD0 yang satunya bisa dipasang ke 3,3V yang akan merujuk ke alamat 0x69.
  - Pin INT = 27 karena mendukung pinout nya saja. INT dipasang agar bisa memberi tahu ESP32 ketika ada data baru sehingga tidak terus-terusan membaca sensor. Namun, di kode aku belum tambahkan interrupt nya karena tidak sempat.
5. Sistem ini bertujuan untuk mendeteksi gerakan serta orientasi menggunakan sensor PIR dan MPU6050, lalu menggerakkan lima servo motor berdasarkan data yang diterima. Mikrokontroler yang digunakan adalah ESP32, yang berfungsi sebagai pusat kendali untuk memproses sinyal dari sensor dan mengatur aktuator. Saat sistem pertama kali dijalankan, ESP32 melakukan inisialisasi dengan mengaktifkan komunikasi serial dan I<sup>2</sup>C pada pin SDA (21) dan SCL (22) untuk berkomunikasi dengan sensor MPU6050. Servo 1-5 masing-masing dihubungkan ke pin 19, 18, 17,

16, dan 4, lalu diinisialisasi ke 90 derajat sebagai posisi awal. Sensor PIR diatur pada pin 26 sebagai input untuk mendeteksi adanya gerakan di sekitar area sensor. Setiap loop, ESP32 membaca data dari MPU6050 yaitu acceleration (percepatan translasi) dan gyroscope (kecepatan sudut) yang kemudian diolah menjadi roll, pitch, dan yaw. Servo 1 dan 2 digerakkan berdasarkan rotasi roll dengan arah yang berlawanan dengan arah roll, servo 3 dan 4 digerakkan berdasarkan rotasi pitch dengan arah yang sama dengan arah pitch, dan servo 5 digerakkan berdasarkan rotasi yaw dengan arah yang sama dengan arah yaw dan akan kembali ke posisi awal 90 derajat setelah 1 detik. ESP32 juga membaca data dari sensor PIR yang berupa HIGH atau LOW. Jika HIGH artinya terdapat gerakan eksternal sehingga kelima sensor akan bergerak acak sesuai sudut yang aku tentukan kemudian kembali ke posisi awal 90 derajat.