

MODUL 4 Sistem Kendali PID Kasus P dan D



Mata Kuliah : Sistem Kendali

Kode Dosen : AJR

Kelas : D3TK-43-02

Anggota Kelompok :

1. Muhammad Yogi (6702194045)
2. M Rifki Arya Syahputra (6702190010)

**PROGRAM STUDI D3 TEKNOLOGI KOMPUTER
FAKULTAS ILMU TERAPAN
UNIVERSITAS TELKOM
BANDUNG
2021**

A. Tujuan

Maksud dan tujuan dari praktikum ini adalah :

1. Mahasiswa dapat memahami fungsi dan cara kerja PID pada motor DC
2. Mahasiswa dapat membuat program sistem kendali berbasis PID dengan error yang dihubungkan dengan konstanta proporsional dan derivatif

B. Alat dan Bahan

Alat dan Bahan :

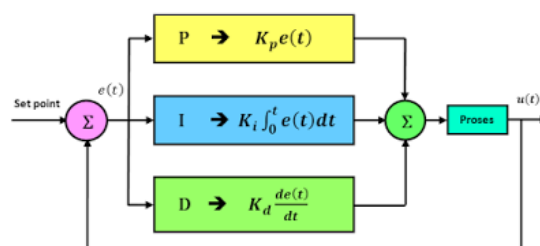
- a. Robot Kit Line Follower
- b. Baterai LiPo 2-Cell 1300 mAh
- c. Kabel Mini-USB
- d. Arduino Nano
- e. Battery Checker
- f. Battery Balancer

Perangkat Lunak :

- a. Software IDE Arduino
- b. Software Proteus (untuk simulasi)

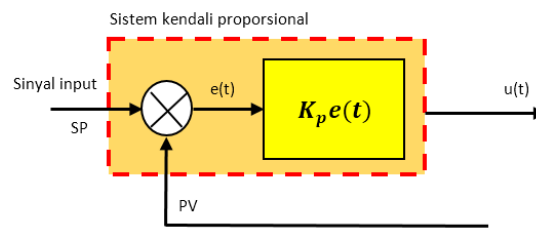
C. Teori dasar

Teknik kendali PID adalah pengendali yang merupakan gabungan antara aksi kendali proporsional ditambah aksi kendali integral ditambah aksi kendali derivatif/turunan (Ogata, 1996). PID merupakan kependekan dari proportional integral derivative. Kombinasi ketiga jenis aksi kendali ini bertujuan untuk saling melengkapi kekurangan-kekurangan dari masing-masing aksi kendali. Untuk memudahkan dalam memahami konsep teknik kendali PID silakan menyermati diagram blok pengendali PID pada gambar 1 di bawah ini.

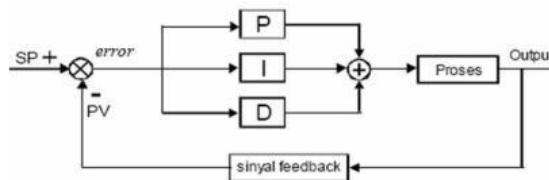


Dalam aksi kendali PID, ada beberapa parameter variabel (dapat diubah/berubah) yang dapat dimanipulasi untuk tujuan menghasilkan aksi kendali terbaik dalam aplikasinya. Cara manipulasi parameter ini sering dinamakan dengan Manipulated Variable (MV). Dalam notasi matematikanya dapat ditulis dengan $MV(t)$ atau $u(t)$.

Aksi kendali proporsional (P) adalah aksi kendali yang memiliki karakter dapat mengurangi waktu naik (rise time), tetapi tidak menghilangkan kesalahan keadaan tunak (steady state error).



Sistem kendali PID ini bertujuan untuk menentukan parameter aksi kendali Proportional, Integratif, Derivatif pada robot line follower. Proses ini dapat dilakukan dengan cara trial and error. Keunggulan cara ini plant tidak perlu diidentifikasi dan membuat model matematis plant. Hanya dengan cara mencoba memberikan konstanta P-I-D pada formula PID sehingga diperoleh hasil yang optimal, dengan mengacu pada karakteristik masing-masing kontrol P-I-D. Tujuan penggunaan sistem kendali PID adalah untuk mengolah suatu sinyal kesalahan atau error, nilai error tersebut diolah dengan formula PID untuk dijadikan suatu sinyal kendali atau sinyal kontrol yang akan diteruskan ke aktuator. Diagram blok sistem umpan balik loop tertutup pada perancangan sistem kendali PID pada robot line follower dapat dilihat pada gambar berikut ini:



Dari blok diagram di atas dapat dijelaskan sebagai berikut

1. SP = Set point, suatu parameter nilai acuan atau nilai yang diinginkan.
2. PV = Present Value, nilai bobot pembacaan sensor saat itu atau variabel terukur yang di umpan balik oleh sensor (sinyal feedback dari sensor).
3. Error = nilai kesalahan, deviasi atau simpangan antar variabel terukur atau bobot sensor (PV) dengan nilai acuan (SP)

D. Hasil Percobaan

A. Percobaan dalam praktikum

1. Kasus Percobaan



- a. Modifikasi program sistem kendali pada praktikum sebelumnya dengan menambahkan sebuah sub program untuk melakukan kalibrasi sensor secara otomatis (auto calibration). Kalibrasi ini berfungsi untuk menentukan nilai pembacaan sensor untuk warna hitam dan putih atau warna lainnya. Apabila push button yang terhubung dengan pin D2 (pin switch Left Down (Kiri bawah) pada Gambar 2) ditekan, eksekusi program berikut :

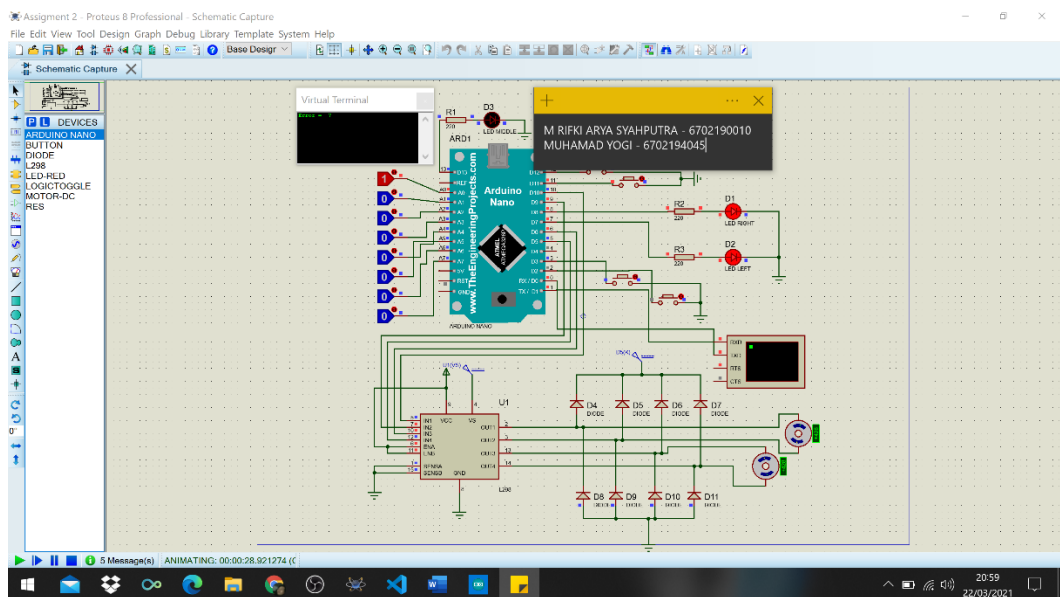
- Tambahkan variabel sensor[0] sampai sensor[7]
- Tambahkan variabel NilaiMaxSensor[0] sampai
- NilaiMaxSensor[7] dengan nilai awal 1023
- Tambahkan variabel NilaiMinSensor[0] sampai
- NilaiMinSensor[7] dengan nilai awal 0
- Tambahkan variabel NilaiTengahSensor[0] sampai
- NilaiTengahSensor[7]
- Baca kondisi sensor 1 sampai 8 dan simpan di variabel sensor[0]
- sampai sensor[7]. Untuk nilai i dari 0 hingga 7 :
 - Jika nilai variabel sensor[i] > NilaiMinSensor[i] maka
NilaiMinSensor[i]=sensor[i]
 - Jika nilai variabel sensor[i] < NilaiMaxSensor[i] maka
NilaiMaxSensor[i]=sensor[i]
 - $\text{NilaiTengahSensor}[i] = (\text{NilaiMinSensor}[i] + \text{NilaiMaxSensor}[i]) / 2$
 - Saat proses kalibrasi dilakukan, LED LEFT (Pin D7) dan LED RIGHT (Pin D8) akan berkedap-kedip (blinking) sebagai indicator bahwa proses kalibrasi sedang berjalan.

b. Apabila proses kalibrasi sudah selesai dan tombol push button yang terhubung dengan pin D3 (pin switch Left Up/kiri atas pada Gambar 2) ditekan, maka eksekusi program berikut ini :

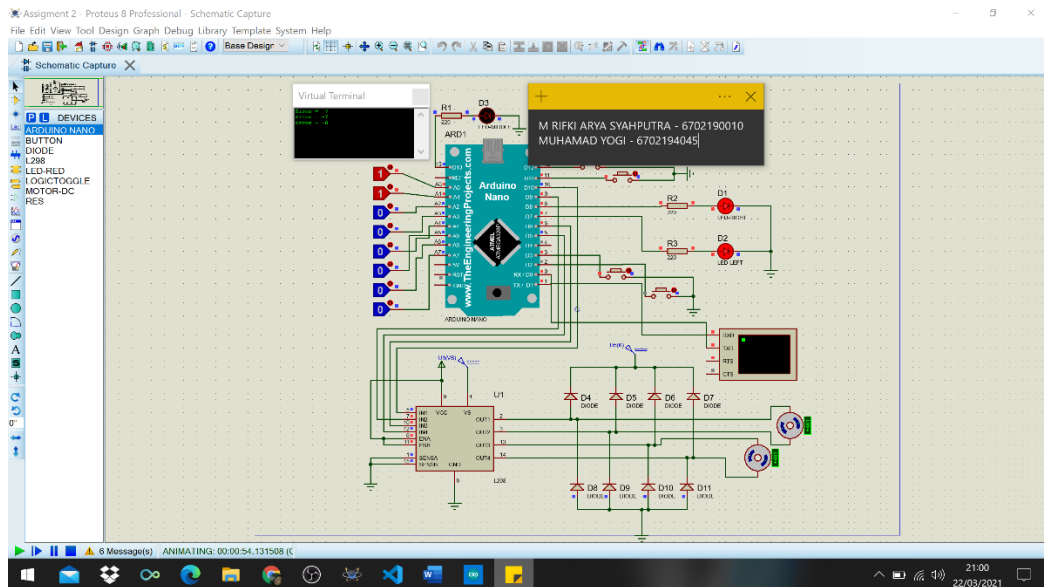
- Inisialisasi variabel dengan tipe int Kp dengan nilai awal 15, int Kd dengan nilai awal : 5, int moveControl dengan nilai awal 0, int error dengan nilai awal 0, int lastError dengan nilai awal 0. int kecepatanMotorKanan dengan nilai awal 0, int kecepatanMotorKiri dengan nilai awal 0, int kecepatanSetPoint dengan nilai awal 150 dan int rate (selisih error dengan lastError).

- Program harus dapat mendeteksi perubahan nilai pada sensor dan mengirimkannya ke serial monitor dengan ketentuan sebagai berikut.

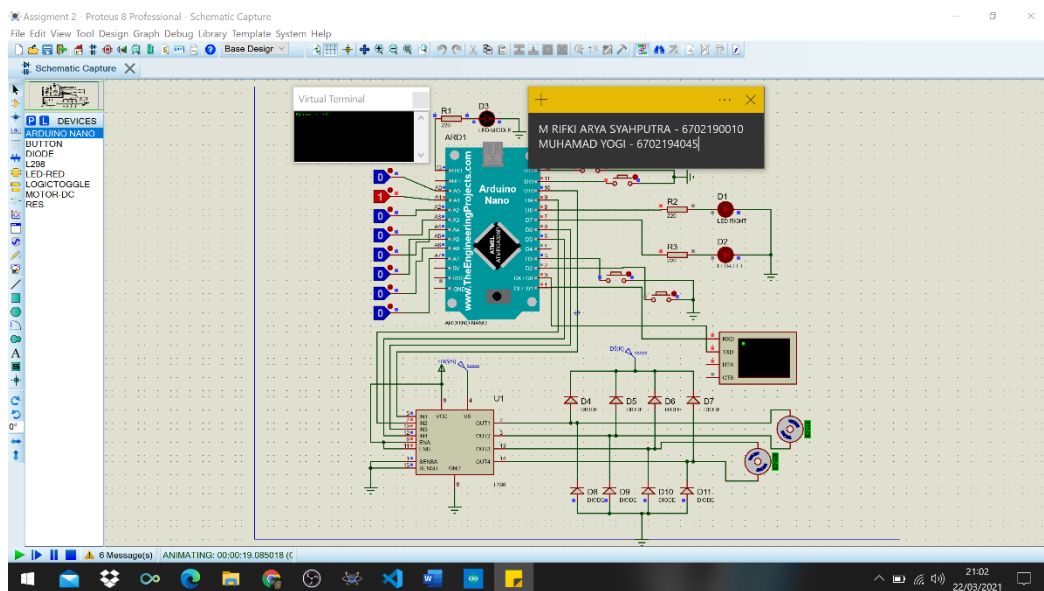
☐ Jika kondisi sensor "10000000", error = -7, print di serial monitor error = -7,



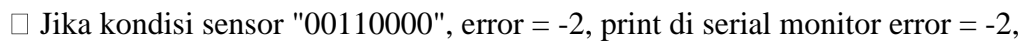
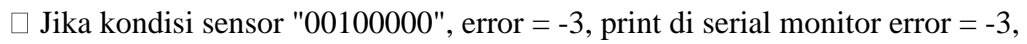
☐ Jika kondisi sensor "11000000", error = -6, print di serial monitor error = -6,

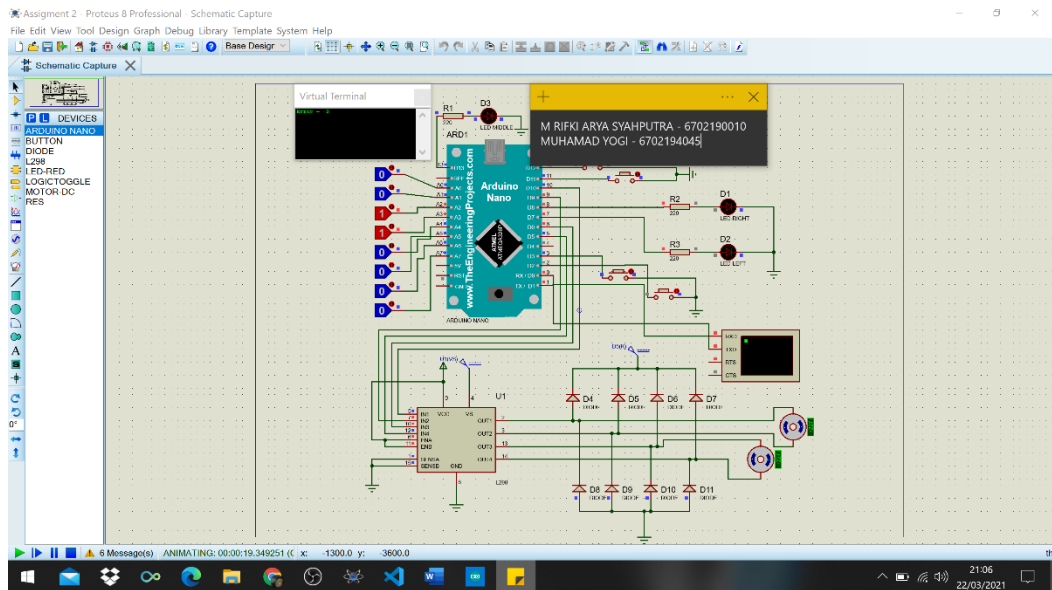


□ Jika kondisi sensor "01000000", error = -5, print di serial monitor error = -5,

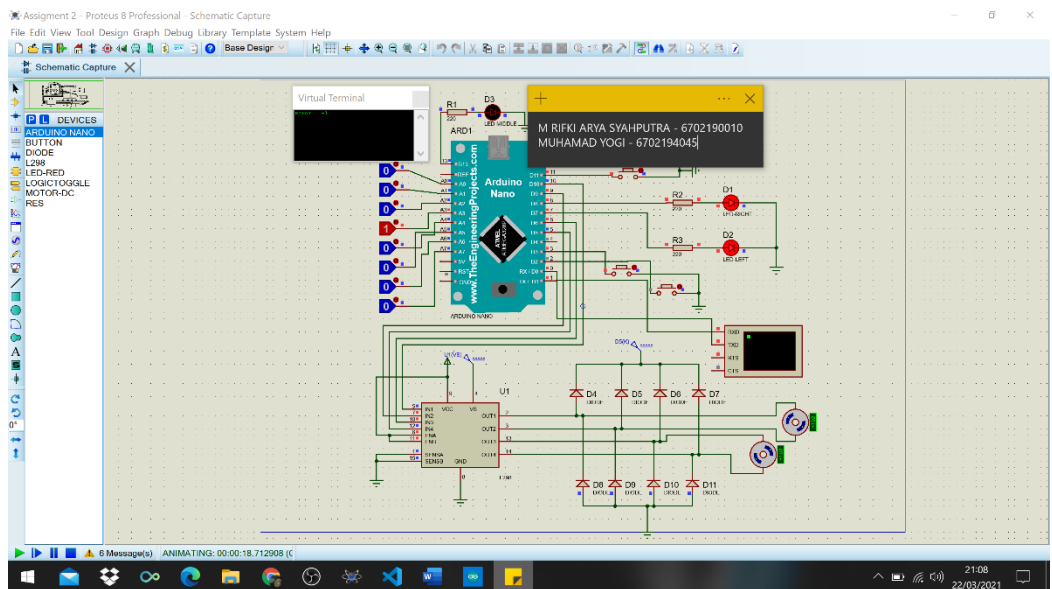


□ Jika kondisi sensor "01100000", error = -4, print di serial monitor error = -4,

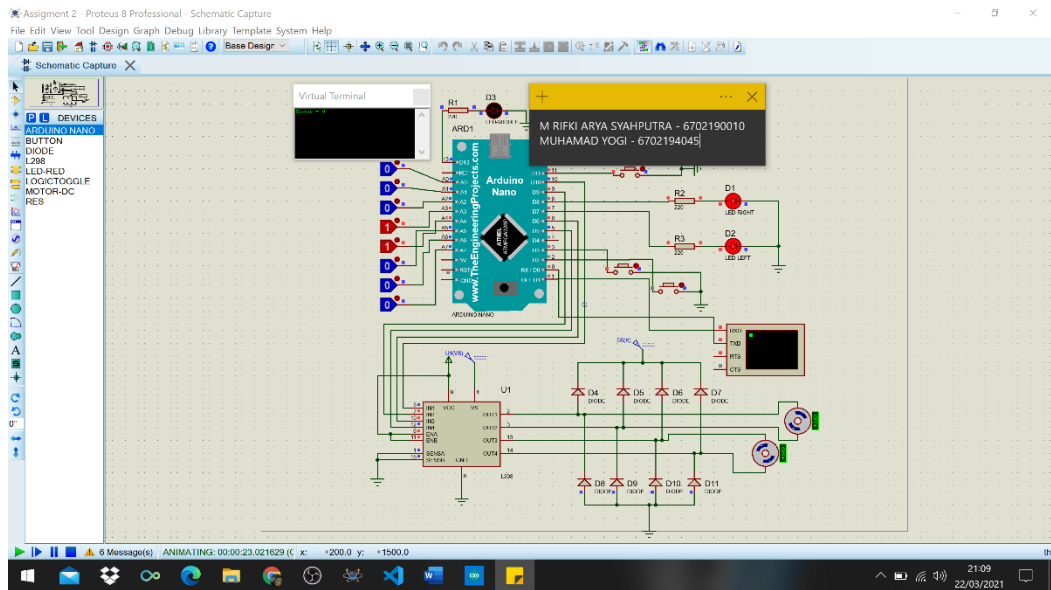




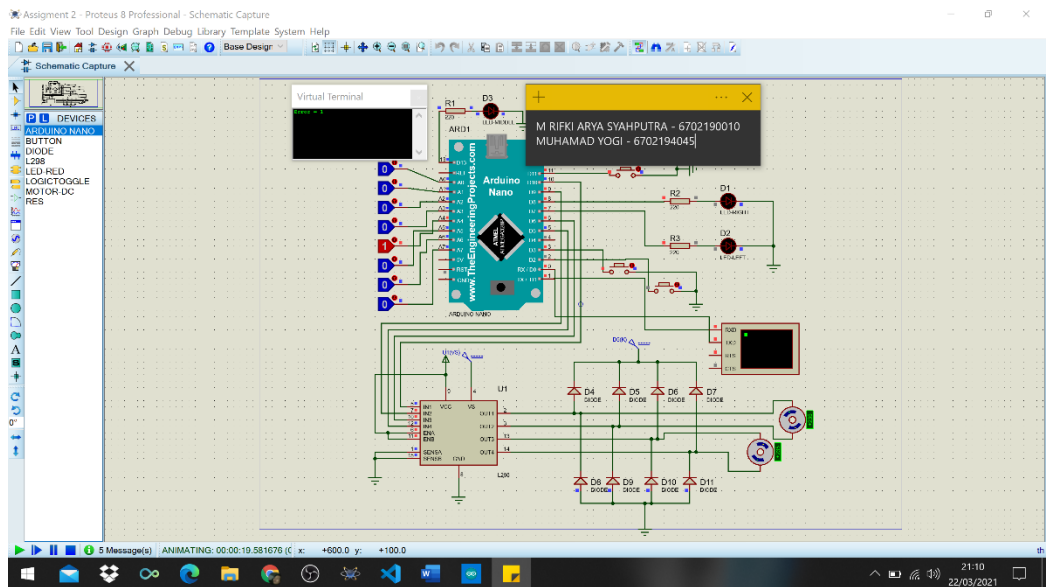
□ Jika kondisi sensor "00010000", error = -1, print di serial monitor error = -1,



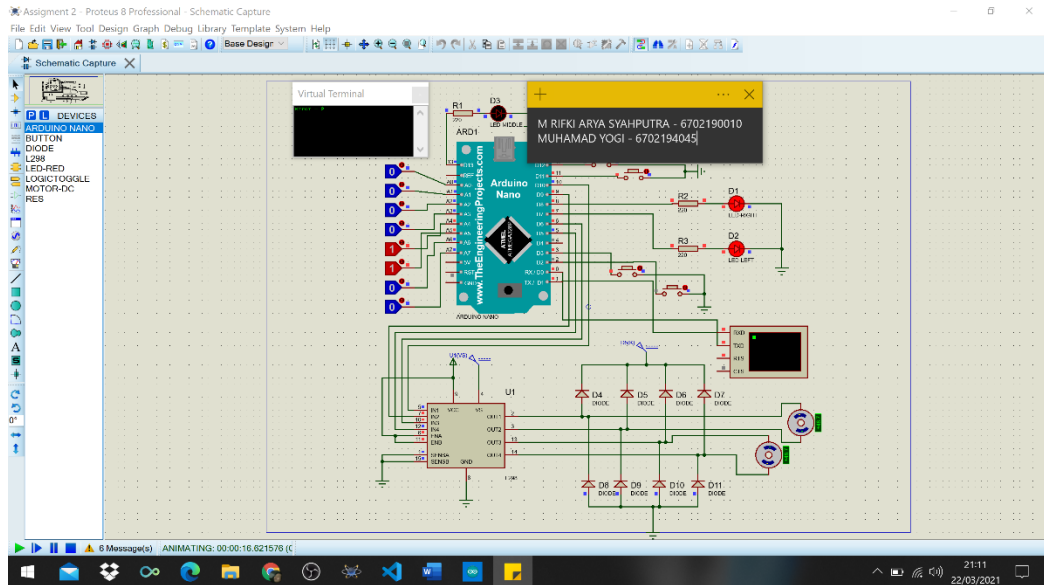
□ Jika kondisi sensor "00011000", error = 0, print di serial monitor error = 0,



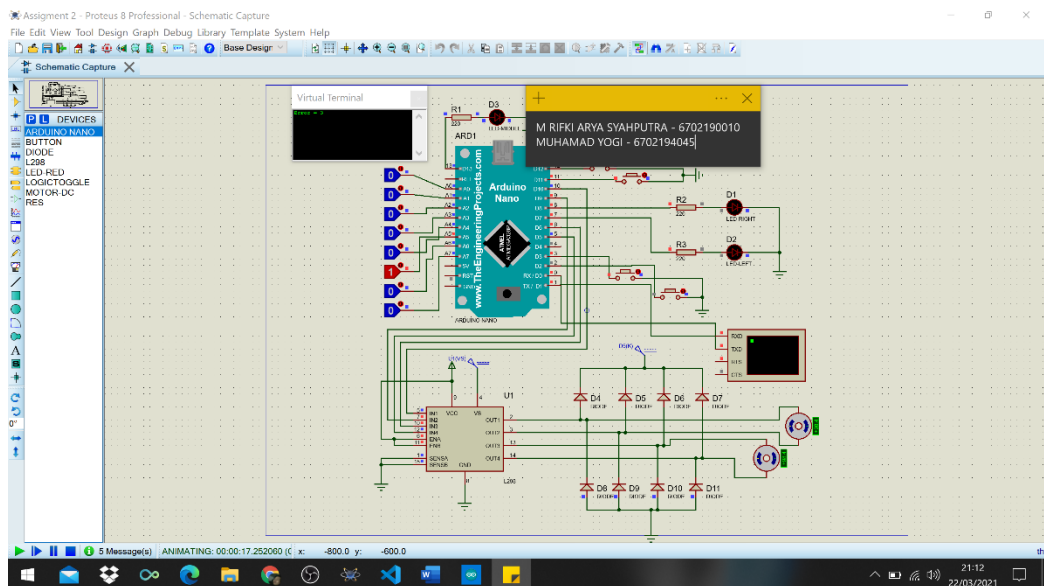
□ Jika kondisi sensor "00001000", error = 1, print di serial monitor error = 1,



□ Jika kondisi sensor "00001100", error = 2, print di serial monitor error = 2,



- Jika kondisi sensor "00000100", error = 3, print di serial monitor error = 3,



- Jika kondisi sensor "00000110", error = 4, print di serial monitor error = 4,
 - Jika kondisi sensor "00000010", error = 5, print di serial monitor error = 5,
 - Jika kondisi sensor "00000011", error = 6, print di serial monitor error = 6,
 - Jika kondisi sensor "00000001", error = 7, print di serial monitor error = 7,
- c. Kemudian tambahkan kode program dengan ketentuan sebagai berikut :

1. Simpan nilai error saat ini dengan variabel Error dan nilai error sebelumnya dengan variable lastError.
2. Simpan nilai selisih antara lastError dengan Error dalam variable rate.
3. Simpan nilai moveControl = perkalian Kp dan Error ditambah hasil perkalian Kd dan rate.

4. Simpan nilai kecepatanMotorKanan = kecepatanSetPoint dikurang moveControl
5. Simpan nilai kecepatanMotorKiri = kecepatanSetPoint ditambah moveControl
6. Kecepatan Motor Kiri dengan nilai analog sebesar kecepatanMotorKiri
7. Kecepatan Motor Kanan dengan nilai analog sebesar kecepatanMotorKanan

Variasikan nilai konstanta Kp dan Kd pada program kemudian ujicoba dan amati hasil perubahan kedua konstanta tersebut!

Jelaskan fungsi dari Kp dan Kd pada kode program di atas!

Jawab : Kp mampu untuk memperbaiki respon transien khususnya rise time dan settling time. Kd hanya berubah saat ada perubahan error sehingga saat error statis kontrol ini tidak akan bereaksi.

- d. Screenshoot keluaran serial monitor untuk setiap kondisi. Cetak dan tempelkan pada buku jurnal praktikum. **Jawab** : Screenshoot ada pada poin B.
- e. Isi tabel kebenaran dari sistem pada Tabel 2 dan tuliskan pada buku jurnal praktikum.

Sensor								Error	lastError	Rate	Analog Value	
0	1	2	3	4	5	6	7				Motor Kiri	Motor Kanan
1	0	0	0	0	0	0	0	-7	-7	0	115	185
1	1	0	0	0	0	0	0	-6	-6	0	120	180
0	1	0	0	0	0	0	0	-5	-5	0	125	175
0	1	1	0	0	0	0	0	-4	-4	0	130	170
0	0	1	0	0	0	0	0	-3	-3	0	135	165
0	0	1	1	0	0	0	0	-2	-2	0	140	160
0	0	0	1	0	0	0	0	-1	-1	0	145	155
0	0	0	1	1	0	0	0	0	0	0	150	150
0	0	0	0	1	0	0	0	1	1	0	155	145
0	0	0	0	1	1	0	0	2	2	0	160	140
0	0	0	0	0	1	0	0	3	3	0	165	135
0	0	0	0	0	1	1	0	4	4	0	170	130
0	0	0	0	0	0	1	0	5	5	0	175	125
0	0	0	0	0	0	1	1	6	6	0	180	120
0	0	0	0	0	0	0	1	7	7	0	185	115

E. Kesimpulan

Kesimpulan pada praktikum ini kita mengerti Sistem kendali PID ini bertujuan untuk menentukan paramater aksi kendali Proportional, Integratif, Derivatif pada robot line follower, membuat rangkaian dengan memunculkan error. Kp mampu

untuk memperbaiki respon transien khususnya rise time dan settling time. Kd hanya berubah saat ada perubahan error sehingga saat error statis kontrol ini tidak akan bereaksi,

F. Link

Link github : https://github.com/rifkiaryas/Kelompok_M-Yogi-M-Rifki-Arya.git

Link Youtube : https://youtu.be/pOztaQB_QYQ