

PEMBUATAN *LINE FOLLOWER ROBOT* DENGAN KONTROL PID (*PROPORTIONAL INTEGRAL DERIVATIVE*)

Muhammad Shidiq Abdul Wahid¹

Teknik Mesin, Sekolah Tinggi Teknologi Mandala Bandung

ABSTRAK

Line Follower Robot adalah suatu robot yang dapat mengikuti garis lintasan yang memiliki warna berbeda dengan *background* arena. Pada karya tulis ini, *line follower robot* yang dibuat menggunakan beberapa tambahan fitur dalam pengaplikasiannya. Karena robot ini dibuat untuk diaplikasikan pada mesin AGV(*automatic guide vehicle*), oleh karena itu robot ini dilengkapi modul *wireless* APC220 untuk berkomunikasi dengan server. dalam perancangannya terdapat beberapa masalah yang harus dipecahkan yaitu perancangan *hardware* dan perancangan *software* nya. Dalam perancangan *hardware* nya *Robot Line Follower* ini terdiri dari tiga bagian umum yaitu bagian mata dalam hal ini sensor cahaya yang menggunakan 8 sensor, bagian kaki yaitu bagian roda dan motor DC dan bagian otaknya yaitu bagian IC yang menggunakan mikrokontroler AVR ATmega 16. Karya tulis ini dibuat untuk merancang dan mengimplementasikan suatu robot yang dapat bergerak mengikuti garis menggunakan LED dan sensor *photodiode* serta dapat berkomunikasi serial antara robot dengan server menggunakan modul *wireless* APC220. Pada pelaksanaannya, *software* yang digunakan untuk membuat program *line follower robot* ini adalah *CodeVision AVR* yaitu *software* yang menggunakan bahasa *Codevision* sebagai bahasa pemrogramannya dan juga menggunakan *Terminal v1.9b* sebagai *software* untuk komunikasi serial antara *line follower robot* dengan server.

Keyword : *Robot, PID*

1. PENDAHULUAN

Kemajuan teknologi terus berjalan seiring perkembangan zaman. Terutama teknologi pada industri manufaktur yang berkembang semakin pesat, sehingga manusia dituntut untuk aktif mengikuti perkembangan yang terjadi dari era revolusi industri sampai dengan era globalisasi saat ini, perubahan pola pikir manusia sangat berbeda, banyak pemikiran dan perkembangan teknologi yang terjadi. Hal ini disebabkan karena rasa keingintahuan manusia dalam bidang teknologi tidak terkecuali dalam bidang teknologi robotik.

Robotika bukanlah hal yang baru saat ini, sehingga perkembangan dari bidang ilmu pengetahuan ini sudah banyak macam pengaplikasiannya. Dimana hampir disetiap kalangan meminati dan menggunakannya. Salah satunya adalah robot pengikutgaris (*Line follower robot*). Untuk itu, robot pengikut garis perlu dikembangkan. Karena robot ini merupakan ilmu dasar dari robot. Yang pada penulisan tugas akhir ini di aplikasikan untuk mesin AGV (*automatic guides vehicle*). Dimana mesin AGV ini adalah *mobile robot* yang membawa material dan mengikut lintasan yang di rancang

sedemikian rupa sesuai dengan kondisi ruangan dimana robot akan ditempatkan. Untuk itu judul tugas akhir ini adalah "**Pembuatan line follower robot dengan kontrol PID (proporsional integral derivatif)**". Kontrol robot ini berbasis mikrokontroler AT mega16 dengan menggunakan software Code Vision AVR yang dibuat dengan program bahasa *codevision* dan menggunakan sistem kendali server via radio frekuensi.

2. METODE PENELITIAN

Line follower robot adalah robot yang bisa bergerak mengikuti jalur panduan garis. Garis pandu yang digunakan dalam hal ini adalah garis hitam yang ditempatkan pada permukaan berwarna putih (cerah) atau sebaliknya, garis putih yang ditempatkan pada permukaan yang berwarna gelap. Prinsip kerja pendeteksian garis pandu dari robot tersebut adalah bahwa tiap-tiap warna permukaan memiliki kemampuan memantulkan cahaya yang berbeda-beda. Warna putih memiliki kemampuan memantulkan cahaya lebih banyak. Sebaliknya, warna-warna gelap memiliki lebih sedikit kemampuan memantulkan cahaya. Hal itulah yang digunakan untuk mendeteksi garis pandu tersebut. Berikut adalah beberapa komponen yang digunakan pada *line follower robot*:

Mikrokontroler ATmega 16

Mikrokontroler adalah sebuah sistem komputer lengkap dalam satu serpih (*chip*). Mikrokontroler lebih dari

sekedar sebuah mikroprosesor karena sudah terdapat atau berisikan ROM (*Read Only Memory*), RAM (*Random Access Memory*), beberapa bandar masukan maupun keluaran, dan beberapa *peripheral* seperti pencacah/pewaktu, ADC (*Analog to Digital converter*), DAC (*Digital to Analog converter*) dan serial komunikasi.

Motor Dc (Direct Current)

Motor DC (*Direct Current*) adalah motor listrik yang memerlukan suplai tegangan arus searah pada kumparan medan untuk diubah menjadi energi gerak mekanik. Kumparan medan pada motor DC disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar).

IC Motor Driver

L298 adalah IC yang dapat digunakan sebagai *driver* motor DC. IC ini menggunakan prinsip kerja *H-Bridge*. Tiap *H-Bridge* dikontrol menggunakan level tegangan TTL yang berasal dari *output* mikrokontroler. L298 dapat mengontrol 2 buah motor DC. Tegangan yang dapat digunakan untuk mengendalikan robot bisa mencapai tegangan 50V DC dan arus 2A untuk setiap kanalnya.

Line Tracking SFH 8 Sensor

Merupakan sebuah modul sensor yang dapat digunakan untuk mendeteksi jalur berwarna terang dengan latar belakang gelap atau jalur berwarna gelap dengan latar belakang terang. Modul ini terdiri dari 8 buah LED dan 8 buah NPN-Silizium-Fototransistor (SFH300). *Output* modul

sensor ini berupa tegangan analog yang berkisar dari 0VDC untuk jalur terang dan 4,9VDC untuk jalur gelap.

Wireless APC220

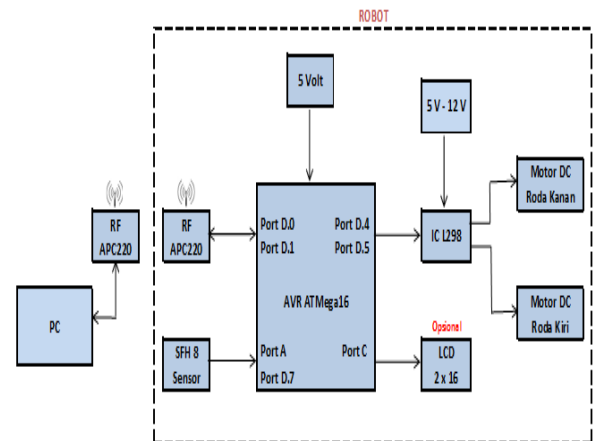
Wireless adalah teknologi yang menggunakan frekuensi dan transmisi radio sebagai media penghantarnya, pada area tertentu menggantikan fungsi kabel. APC220 bekerja pada frekuensi 418 MHz sampai 455 Mhz (*Ultra High Frequency*). Komunikasi bisa mencapai 1000 meter dalam keadaan *Line of sight* (transmisi antara kedua sinyal tanpa ada penghalang) dan 2400 bps *air rate*. *Air rate* berbeda dengan *baud rate*, *air rate* adalah laju data mengudara sedangkan *baud rate* adalah laju data UART.

1.1 PERANCANGAN SISTEM

perancangan sistem akan dibagi menjadi kedalam dua bagian, yaitu perancangan perangkat keras dan perancangan piranti lunak. Perancangan perangkat keras terdiri dari perancangan berbagai komponen elektrik yang digunakan dalam sistem. Perancangan piranti lunak terdiri dari perancangan program-program yang digunakan dalam sistem serta pembuatan *source code* menggunakan CodeVision AVR.

Perancangan Perangkat Keras

Perancangan perangkat keras digunakan untuk mendukung seluruh komponen atau modul elektronika yang digunakan pada *mobile robot*. Berikut ini adalah blok diagram sistem robot :



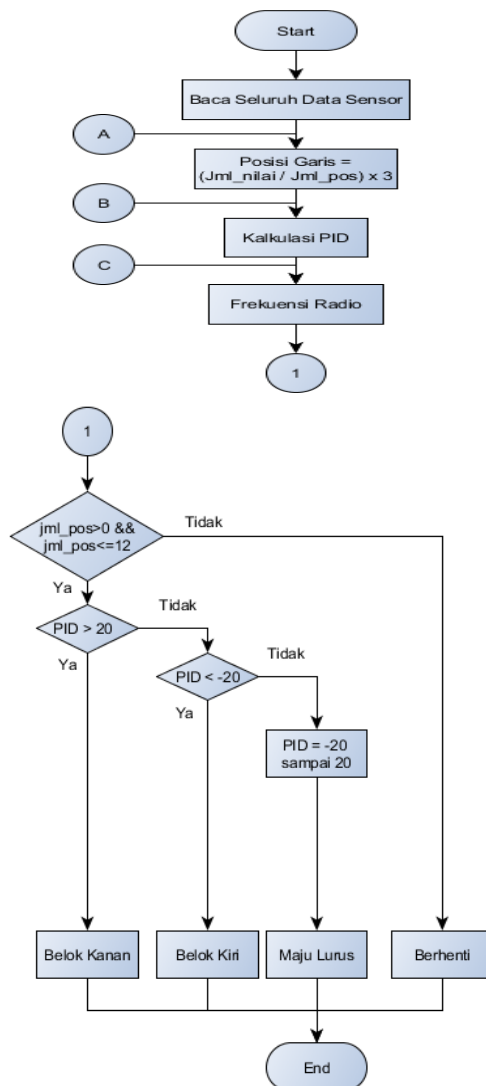
Gambar 1 Blok Diagram Sistem Robot

Pada Gambar 1 dapat dilihat bahwa robot menggunakan AVR ATmega16 untuk memproses semua pergerakan robot. Robot mendeteksi garis berwarna putih menggunakan sensor *phototransistor* SFH8 dan untuk dapat bergerak robot menggunakan motor DC sebagai aktuatornya, dimana pada robot terdapat dua motor yang difungsikan untuk roda kiri dan roda kanan, juga diperlukan *driver* motor untuk menggerakkan motor DC tersebut, dalam hal ini IC *driver* motor DC yang digunakan adalah IC L298 . Selain itu robot juga dilengkapi dengan modul LCD untuk menunjukkan posisi penyimpangan pada jalur yang terjadi pada robot namun modul LCD hanya sebagai opsional yaitu digunakan hanya saat awal untuk pengecekan apakah sensor bekerja dengan benar dan tidak digunakan lagi saat *line follower robot* dioperasikan sepenuhnya karena akan mengganggu kecepatan atau respon program terhadap robot.

Perancangan Piranti Lunak

Setelah perangkat keras dibuat maka langkah berikutnya adalah membuat piranti lunak atau *software*

yang akan mendukung kerja dari *hardware* karena jika tanpa *software* maka *hardware* tidak akan bekerja, dengan demikian piranti lunak merupakan pola pikir dari perangkat keras dan memegang peranan yang penting. Gambar diagram alir kerja robot dapat dilihat pada gambar 2 :

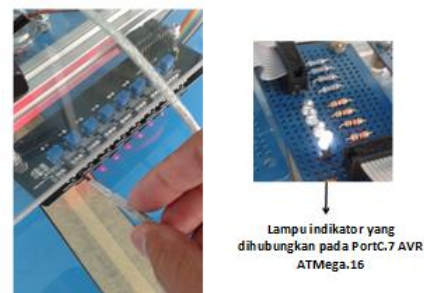


Gambar 2 Diagram Alir cara kerja robot

3. HASIL DAN PEMBAHASAN

Setelah merangkai semua komponen untuk model *line follower robot* maka hal berikut yang dilakukan adalah melakukan pengujian pada setiap komponen yang ada pada *line follower robot*. pengujian ini dilakukan untuk memastikan setiap komponen berperan dengan baik dan sesuai harapan agar robot dapat berjalan dengan baik. Berikut adalah beberapa hal yang dilakukan kepada beberapa komponen untuk di uji coba :

Kalibrasi Sensor



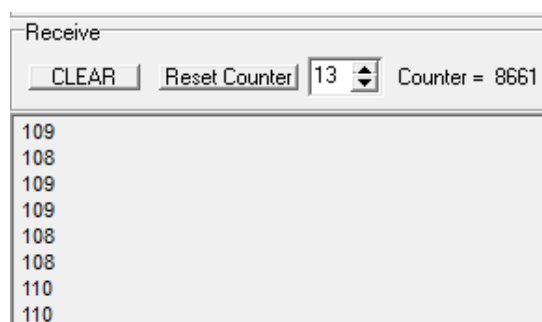
Gambar 3 Kalibrasi Sensor

Gambar 3 di atas menjelaskan bagaimana mengkalibrasi sensor LDR dengan memutar baut yang ada di atas sensor. Sensor yang digunakan adalah sensor tipe LDR (*light dependent resistor*) yang mendeteksi perubahan intensitas cahaya. Kalibrasi sensor ini diperuntukan agar sensor mendeteksi perubahan intensitas cahaya yang dipantulkan oleh warna lintasan dan medan disekitarnya dengan benar. Dalam hal ini warna garis yang akan di uji coba pada sensor adalah warna putih. Hal pertama yang harus di uji adalah nilai data sensor, nilai data sensor berkisar antara 0 – 1023 semakin kecil nilainya maka semakin terang warna arenanya

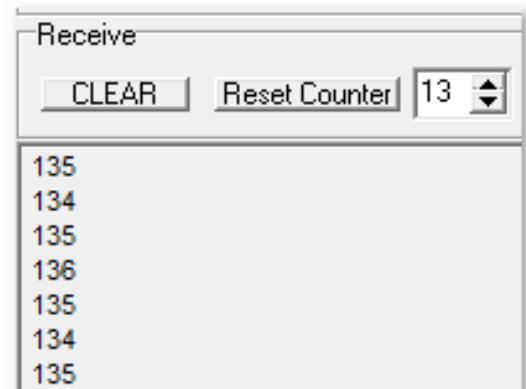
dan semakin besar nilainya maka semakin gelap warna arenanya. Pengujian nilai sensor menggunakan *software terminal v1.9b* dengan menyambungkan kabel komunikasi serial antara PC dan robot. Berikut adalah program dalam bahasa C pada *software CodeVision AVR* yang digunakan untuk pengujian nilai sensor :

```
PORTB.7=1;
delay_ms(5);
n=0;// pengujian dilakukan mulai dari bit 0
data_a[n]=read_adc(n);
PORTB.7=0;
Printf("%d \n", data_a[n]);
delay_ms(5);
```

Setelah program diatas di transfer ke robot maka hubungkan kabel komunikasi serial antara PC dan robot, kemudian jalankan *software terminal v1.9b* untuk menampilkan nilai data sensor antara warna garis dan warna luar garis. Berikut adalah nilai hasil pembacaan sensor menggunakan *software terminal v1.9b* :



Gambar 4 Nilai Data Sensor Warna Putih



Gambar 5 Nilai Data Sensor Warna Biru

Hasil Pengujian Kontrol PID

Hasil dari pengujian *tunning* PID dengan mengatur nilai KP, KD dan KI adalah sebagai berikut :

Tabel 1 Data Hasil Pengujian Tuning PID

| NILAI PID | | | NILAI PWM | | | Volt | MEMBACA GARIS | | HASIL |
|-----------|-----|----|-----------|-----|-----|------|---------------|--------|---|
| KP | KI | KD | MAX | MED | MIN | | GELAP | TERANG | |
| 10 | 0,1 | 0 | 250 | 200 | 0 | 15 V | — | √ | robot bergerak tidak stabil karena overshootnya tinggi dan robot bergerak terlalu cepat sehingga sering keluar garis |
| 10 | 5 | 10 | 250 | 200 | 0 | 10 V | — | √ | robot bergerak mulai stabil namun waktu untuk mencapai <i>set_point</i> nya terlalu lama sehingga mengurangi respon robot |
| 10 | 0,1 | 5 | 250 | 200 | 0 | 10 V | — | √ | robot bergerak mulai stabil namun robot bergerak masih terlalu cepat sehingga sering keluar garis |
| 15 | 0,1 | 7 | 210 | 170 | 0 | 9 V | — | √ | robot bergerak dengan stabil dan respon yang baik karena robot bergerak tidak terlalu cepat |

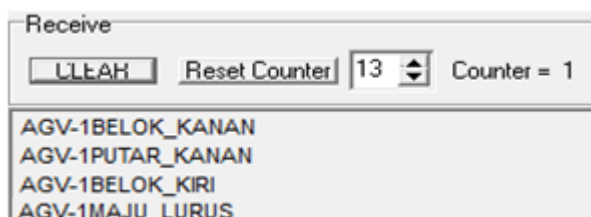
Dari tabel hasil uji coba di atas dapat disimpulkan bahwa nilai Kp yang terlalu besar akan membuat *overshoot*nya semakin besar karena waktu untuk mencapai *set point* nya semakin cepat sehingga membuat robot bergerak tidak stabil. Saat nilai Kp tidak dirubah dan menambahkan nilai Ki dan Kd maka akan memperkecil *overshoot* namun jika nilai Ki terlalu besar maka akan memperlambat waktu untuk mencapai *set point* sehingga respon robot menjadi lama. Oleh karena itu pada percobaan ini nilai racikan yang lebih baik yaitu merubah nilai Ki menjadi nol

dan nilai Kd lebih kecil dari nilai Kp dengan selisih yang cukup besar agar *overshoot*nya bisa ditekan dan respon robot tidak terlalu lambat karena waktu untuk mencapai *set point* nya tidak terganggu oleh nilai Ki.

Selain mengatur nilai Ki, Kd dan Kp, juga perlu memperhatikan nilai PWM robot dapat bergerak dengan stabil dan tidak terlalu cepat. Karena jika robot bergerak terlalu cepat maka saat berbelok atau saat ada

Uji Coba Komunikasi Serial

Setelah melakukan serangkaian uji coba diatas maka pengujian yang terakhir ialah pengujian komunikasi serial. Komunikasi serial yang digunakan adalah komunikasi dengan media frekuensi radio. Untuk mengatur komunikasi serial agar terhubung antara PC dan robot yaitu dengan mengatur menu yang ada pada *software terminal v1.9b* di atas sesuai dengan settingan robot. Gambar 6 berikut adalah hasil uji coba pada robot :



Gambar 6 Komunikasi Serial Robot dan PC

Pada gambar 6 diatas dijelaskan saat robot berada di persimpangan maka robot akan berhenti dan mengirimkan sinyal ke PC dengan kode robot masing-masing dan menunggu perintah dari PC. Pada gambar di atas robot diberi perintah dan setelah

persimpangan, sensor robot akan telat membaca garis karena kecepatan motor yang terlalu tinggi. Namun besarnya tegangan yang di alirkan kepada robot juga mempengaruhi kecepatan motor sehingga untuk mendapatkan gerak robot yang stabil diperlukan percobaan untuk mengatur nilai Kp, Ki, Kd, nilai PWM dan besarnya tegangan listrik arus DC yang diberikan kepada robot.

perintah dijalankan maka robot kembali mengirim sinyal pemberitahuan bahwa dia sudah melakukan perintah yaitu belok kanan, belok kiri, putar kanan atau maju lurus.

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

- Robot *line follower* dapat berfungsi dengan baik yaitu dengan mengikuti garis berwarna putih yang memantulkan cahaya kepada sensor yang membaca intensitas pantulan cahaya tersebut dan menghasilkan nilai referensi komparator yaitu putih bernilai 110 dan biru bernilai 134 yang diteruskan pada mikrokontroler untuk diproses.
- Pergerakan robot menggunakan program PID dengan *tunning* manual dengan melakukan beberapa kali uji coba yang sesuai dengan geometri robot. Setelah melakukan uji coba pada robot didapatkanlah hasil *tunning* untuk nilai Kp, Ki, Kd yang sesuai dengan geometri robot, nilai Kp = 15, Kd = 7 dan Ki = 0,1 dengan tegangan listrik yang diberikan pada robot adalah 9 volt dengan

max_speed = 210 , med_speed = 170 dan min_speed = 0.

- Uji coba komunikasi serial antara robot dengan PC berfungsi dengan benar yaitu dengan menggunakan *software terminal v1.9b* dengan menggunakan modul apc 220 yang dipasang pada PC dengan robot. saat robot berada di persimpangan maka robot akan mengirimkan sinyal berupa pesan kepada PC yaitu "AGV-1" saat pesan sudah diterima oleh PC maka PC akan mengirimkan perintah berupa symbol a, b, c atau d yang akan dikirimkan kembali pada robot. a yang berarti maju lurus, b = belok kanan, c = belok kiri dan d = putarkan ,saat robot sudah menerima pesan dari PC maka robot akan menjalankan perintah tersebut dan saa perintah tersebut sudah dilaksanakan maka robot akan mengirim kembali pesan yang menyatakan perintah tersebut sudah dijalankan.

4.2 Saran

- Gunakanlah lintasan dengan warna yang kontras agar perbedaan nilai komparator yang dihasilkan oleh sensor karena pantulan cahaya memiliki jarak yang besar, sehingga saat terjadi perubahan intensitas cahaya dari tempat uji coba tidak terlalu mempengaruhi pembacaan sensor terhadap garis dan membuat robot dapat berfungsi dengan baik saat pengujian.

- Karena sensor yang digunakan adalah sensor LDR atau sensor cahaya maka sensor ini sangat sensitive dengan perbedaan cahaya disuatu tempat, maka saat melakukan pengujian pada robot lakukanlah di tempat tertutup yang intensitas cahayanya relatif stabil. Karena jika dilakukan pada tempat yang intensitas cahayanya tidak stabil maka saat melakukan pengujian akan terganggu dengan keadaan cahaya yang terkadang terang atau redup.
- Untuk hasil yang lebih stabil dan lebih baik, maka gunakanlah sensor magnet karena tidak akan terpengaruh oleh perbedaan intensitas cahaya yang terjadi di dalam ruangan. Sehingga robot dapat bergerak dengan stabil walaupun cahaya di dalam ruangan berubah-ubah.

DAFTAR PUSTAKA

- [1] Winoto, Ardi.2010. *Mikrokontroler AVR atmega8/32/16/8535 dan Pemrogramannya dengan Bahasa C pada WinAVR*. Bandung : Informatika
- [2] Heri, Andrianto. 2013. *Pemrograman Mikrokontroler AVR AT mega 16 Menggunakan Bahasa C (CodeVisionAVR*. Bandung : Informatika
- [3] Rangkuti, Syahban. 2011. *Mikrokontroler ATMELE AVR Simulasi dan Praktek Menggunakan ISIS PROTEUS*

dan Code Vision AVR. Bandung
: Informatika

- [4] Suyadhi, Taufiq Dwi Septian. 2008. *buildyour own LINE FOLLOWER ROBOT*. Yogyakarta : Andi Yogyakarta
- [5] Sanjaya, Mada. 2013. *Membuat ROBOT bersama Profesor BOLABOT Simulasi Menggunakan Code Vision AVR dan Proteus*. Yogyakarta : Gava Media
- [6] Setiawan, Iwan. 2008. *Kontrol PID untuk Proses Industri*. Jakarta : PT. Elex Media Komputindo
- [7] Danny, Umar. 2014. *Contoh Pemrograman Bahasa C*.
<http://umar-danny.blogspot.com/2014/09/belajar-pemrograman-bahasa-c-download.html>, (diakses 10 Oktober 2014)
- [8] Appcon, T., 2008. *APC Series Transparent Transceiver Module APC220-43*.
http://www.famosastudio.com/download/datasheet/APC220_Datasheet.pdf,
(diakses 14 Desember 2014)