

APLIKASI PID PADA ROBOT LINE FOLLOWER BERBASIS MIKROKONTROLER AT-8535

Yani Prabowo, TW Wisjhnuadji, Andika Alie Wibowo

Sistem Komputer Universitas Budi Luhur

E-mail: [yan_crosser; wisnoex; andikaaliewibowo]@yahoo.com

Abstract—Currently, the development of technology in the developing world so quickly. In the world of robotics, many made a variety of useful robot to facilitate human work. For the robot control system is now also vary. In this research line follower robot by the method of proportional integral derivative (PID) is a robot that can follow a line which is supported by a series of electronic components and sensors fitted by a motor-driven wheel with the PID control system that works so that the robot can still run in accordance with data is read and processed by the sensor control settings P (Proportional), D (Derivative) and I (Integral). These robots are designed to navigate and move automatically and follow the flow lines are made of black or tracking.

This robot uses ATMEGA8535 microcontroller as a component controller, which controls the system microcontroller of the robot according to the desired program and is written in it. Software in the robot was created using artificial MCS Electronics BASCOM language with an IDE (Integrated Development Environment) BASCOM-AVR® made specifically for AVR-based microcontroller.

Key Words—Sensor, Sistem kendali, Proportional, Derivatif, Integral, Line Follower

Abstrak—Saat ini, perkembangan teknologi yang ada di dunia berkembang dengan begitu cepatnya. Dalam dunia robotik, banyak sekali dibuat bermacam-macam robot yang berguna untuk mempermudah pekerjaan manusia. Untuk sistem kendali robot saat ini juga bermacam-macam. Pada penelitian ini robot *line follower* dengan metode *proportional integral derivatif (PID)* adalah sebuah robot yang dapat mengikuti garis yang didukung oleh rangkaian komponen elektronik yang dilengkapi oleh sensor dan roda yang digerakkan oleh motor dengan sistem kontrol PID yang berfungsi supaya robot dapat tetap berjalan sesuai dengan data yang dibaca pada sensor dan diolah berdasarkan pengaturan Kontrol P (Proportional), D (Derivatif) dan I (Integral). Robot ini dirancang untuk bernavigasi dan bergerak secara otomatis dan mengikuti alur garis hitam atau tracking yang dibuat.

Robot ini menggunakan mikrokontroler ATMEGA8535 sebagai komponen pengendali, dimana mikrokontroler ini mengendalikan sistem dari robot sesuai dengan program yang diinginkan dan sudah dituliskan didalamnya. Perangkat lunak (*software*) didalam robot ini dibuat menggunakan bahasa BASCOM buatan MCS Electronics dengan IDE (*Integrated Development Environment*) BASCOM-AVR® yang dibuat khusus untuk mikrokontroler berbasis AVR.

Kata Kunci—Sensor, Sistem kendali, Proportional, Derivatif, Integral, Line Follower

I. PENDAHULUAN

Di zaman teknologi maju seperti sekarang, banyak robot-robot cerdas dengan bentuk dan fungsi yang beragam. Robot-robot mampu berkerja secara otomatis mengganti-tugas-tugas yang biasanya dilakukan manusia. Meskipun mempunyai fungsi yang sama, desain, cara kerja, dan kinerjanya tergantung pada kreatifitas pembuatnya. Dari sisi desain dan cara kerja, rata-rata robot menunjukkan kecanggihannya dengan strategi dan pemakaian komponen yang bermacam-macam. Sebagaimana telah kita ketahui bahwa banyak robot yang kinerjanya dikontrol dengan cara merubah program-program pada komputer. Urutan kerja yang harus dilakukan oleh robot telah diprogramkan kedalam perangkat lunak, sehingga komputer dapat melaksanakan program tersebut dan robot akan melaksanakan semua perintah yang terkandung pada program tersebut.

Namun dalam penggunaan komputer dalam mengontrol robot kurang efisien jika dilihat dari sisi pemakaian listrik dan mobilitas, maka digunakan mikrokontroler ATMEGA8535 sebagai pengendali robot. tersebut, pada penelitian ini menggunakan mikrokontroler

ATMEGA8535 keluaran ATMEL. Penggunaan mikrokontroler ATMEGA8535 sebagai pengontrol dari sistem robot ini dikarenakan kemampuannya dalam melakukan penanganan terhadap sistem dan mampu melakukan pengontrolan dengan baik, sehingga pergerakan atau tracking dari robot ini sesuai dengan program yang sudah ada. Mikrokontroler ini sangat mudah didapat dipasaran dan memiliki kapasitas karakteristik komponen yang mendukung untuk aplikasi kerja sistem yang dirancang. Untuk sistem gerak robot memakai roda dengan motor DC sebagai roda penggerak belakang sekaligus penentu arah belokan.

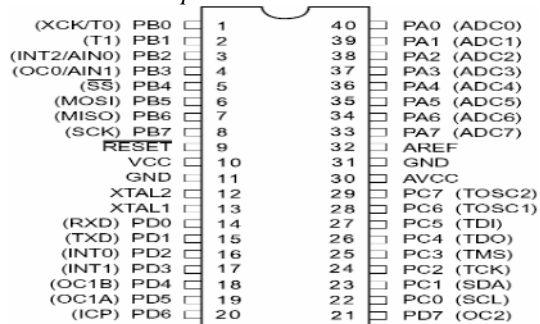
II. LANDASAN TEORI

2.1 Dasar Pemilihan Mikrokontroler

Mikrokontroler AVR (*Alf and Vegard's Risc Processor*) merupakan salah satu perkembangan produk mikroelektronika dari vendor Atmel. AVR merupakan teknologi yang memiliki kemampuan baik dengan biaya ekonomis yang cukup minimal. Mikrokontroler AVR memiliki arsitektur RISC 8 bit, dimana semua instruksi dikemas dalam kode 16 bit dan

sebagian besar instruksi dieksekusi dalam 1 (satu) siklus *clock*, berbeda dengan instruksi MCS51 yang membutuhkan 12 siklus *clock*. Tentu saja itu terjadi karena kedua jenis mikrokontroler tersebut memiliki arsitektur yang berbeda. AVR berteknologi RISC (*Reduced Instruction Set Computing*), sedang MCS 51 berteknologi CISC (*Complex Instruction Set Computing*). Secara umum AVR dapat dikelompokkan menjadi 4 kelas, yaitu keluarga ATiny, keluarga AT90Sxx, Keluarga ATMega, dan AT86RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, *peripheral*, dan fungsinya. Mikrokontroler ATMEGA8535 adalah Microcontroller CMOS 8 bit yang yang didasarkan pada AVR *Enhanced RISC Architecture* [1]. *Flash on-chip* memudahkan program memori untuk dapat diprogramkan dengan cepat dengan menggunakan *programmer memori* stabil, misalkan PG 302. Dengan mengkombinasikan suatu CPU 8-bit dengan *Flash* pada *chip* besar, maka ATMEGA8535 adalah mikrokomputer bertenaga yang memberikan fleksibilitas yang tinggi dan solusi hemat harga. Pada beberapa aplikasi kendali yang dapat dimasukkan ATMEGA8535 memberikan kemampuan sebagai berikut [2]:

1. 16 KByte *Flash*.
2. 1024 Byte RAM.
3. 32 jalur I/O.
4. Satu buah *timer/counter* 16 bit.
5. Satu Buah *timer/counter* 8 bit.
6. Delapan buah sumber interupsi.
7. *Port serial full duplex*.
8. *On chip oscillator* dan sirkuit jam.
9. 10 bit *Analog to Digital Converter*.
10. Mode *low-power idle* dan *Power-down*



Gambar 1. Blok Diagram Mikrokontroler ATMEGA8535

Dalam perancangan aplikasi ini ada beberapa hal yang harus diperhatikan, seperti pemilihan komponen, rangkaian yang akan dibuat dan bahan atau material dari alat sampai harga komponen dan ketersediaan dipasaran. Pemilihan ini dilakukan dengan berdasarkan atas kebutuhan spesifikasi dengan menekan biaya pembuatan tanpa mengurangi kualitas dari sistem yang nantinya akan dibuat, agar dapat dibuat

suatu alat yang dapat bekerja dengan baik dan efisien.

2.2 Dasar Sistem Kontrol

PID (dari singkatan bahasa Inggris: *Proportional-Integral-Derivative controller*) merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut. Komponen kontrol PID ini terdiri dari tiga jenis yaitu Proportional, Integratif dan Derivatif. Ketiganya dapat dipakai bersamaan maupun sendiri-sendiri tergantung dari respon yang kita inginkan terhadap suatu plant[3].

1. Kontrol Proporsional

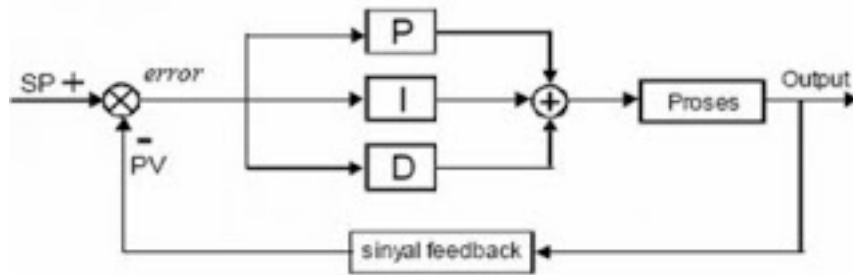
Kontrol P jika $G(s) = k_p$, dengan k adalah konstanta. Jika $u = G(s) \cdot e$ maka $u = K_p \cdot e$ dengan K_p adalah Konstanta Proporsional. K_p berlaku sebagai *Gain* (penguat) saja tanpa memberikan efek dinamik kepada kinerja kontroler. Penggunaan kontrol P memiliki berbagai keterbatasan karena sifat kontrol yang tidak dinamik ini. Walaupun demikian dalam aplikasi-aplikasi dasar yang sederhana kontrol P ini cukup mampu untuk memperbaiki respon *transien* khususnya *rise time* dan *settling time*.

2. Kontrol Integratif

Jika $G(s)$ adalah kontrol I maka u dapat dinyatakan sebagai $u(t) = \int e(t) dt$ K_i dengan K_i adalah konstanta Integral, dan dari persamaan di atas, $G(s)$ dapat dinyatakan sebagai $u = K_i \cdot [\Delta e / \Delta t]$ Jika $e(T)$ mendekati konstan (bukan nol) maka $u(t)$ akan menjadi sangat besar sehingga diharapkan dapat memperbaiki error. Jika $e(T)$ mendekati nol maka efek kontrol I ini semakin kecil. Kontrol I dapat memperbaiki sekaligus menghilangkan respon *steady-state*, namun pemilihan K_i yang tidak tepat dapat menyebabkan respon *transien* yang tinggi sehingga dapat menyebabkan ketidakstabilan sistem. Pemilihan K_i yang sangat tinggi justru dapat menyebabkan output berosilasi karena menambah orde sistem

3. Kontrol Derivatif

Sinyal kontrol yang dihasilkan oleh kontrol D dapat dinyatakan sebagai $G(s) = s \cdot K_d$ Dari persamaan di atas, nampak bahwa sifat dari kontrol D ini dalam konteks "kecepatan" atau *rate dari error*. Dengan sifat ini ia dapat digunakan untuk memperbaiki respon *transien* dengan memprediksi error yang akan terjadi. Kontrol Derivative hanya berubah saat ada perubahan error sehingga saat error statis kontrol ini tidak akan bereaksi, hal ini pula yang menyebabkan



Gambar 2. Diagram blok sistem kontrol PID

kontroler Derivative tidak dapat dipakai sendiri

Mode kontrol proposional integral derivative (PID) adalah aksi kontrol PID yang pada dasarnya bertujuan untuk menggabungkan kelebihan komponen-komponen dasar kontrol PID.[4]:

- Kontrol Proporsional : berfungsi untuk mempercepat respon.
- Kontrol Integral : berfungsi untuk menghilangkan *error steady*.
- Kontrol derivatif : berfungsi untuk memperbaiki sekaligus mempercepat respon *transsien*.

Pada gambar 2 adalah diagram blok sistem kontrol PID yang merupakan penggabungan dari fungsi kontrol proporsional, integral,derivatif.

Masing-masing aksi kontrol ini mempunyai keunggulan-keunggulan tertentu, dimana aksi kontrol proporsional mempunyai keunggulan *risetime* yang cepat, aksi kontrol integral mempunyai keunggulan untuk memperkecil *error* , dan aksi kontrol derivatif mempunyai keunggulan untuk memperkecil *derror* atau meredam *overshot/ undershot*. Untuk itu agar kita dapat menghasilkan output dengan risetime yang tinggi dan *error* yang kecil kita dapat

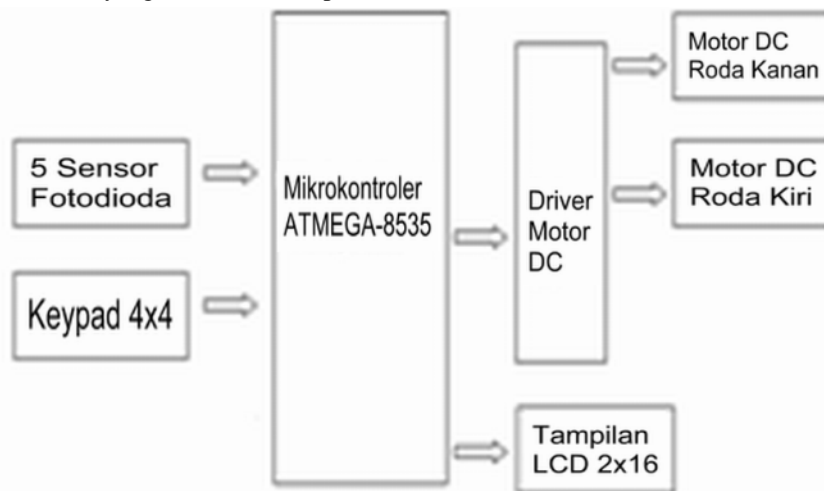
menggabungkan ketiga aksi kontrol ini menjadi aksi kontrol PID, dan pada penulisan ini sistem kendali yang digunakan adalah sistem kendali PID digital. Dalam kasus robot *line follower* kita dapat menggunakan kendali PID untuk menjalankan robot agar lebih halus dalam menyelesaikan lintasan.

Pada dasarnya sistem PID merupakan sistem yang kompleks, sistem yang bekerja berdasarkan nilai yang memiliki range yang terbatas dan konstan. Sehingga untuk aplikasinya sistem PID sangat cocok untuk robot yang memiliki arena yang terbatas seperti arena yang menggunakan garis atau lorong untuk penyelesaian masalahnya . Sistem PID seperti ini sering digunakan untuk algoritma *Line follower*[5].

III. ANALISA DAN PERANCANGAN

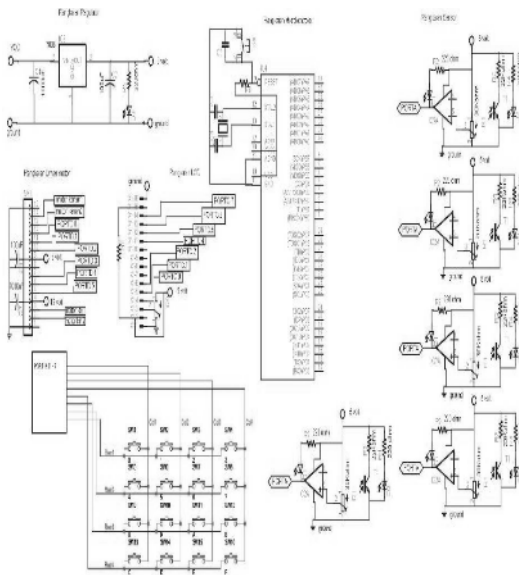
3. 1 Diagram Blok

Perancangan dilakukan ber-dasarkan blok per blok dari setiap rangkaian, dimana tiap-tiap blok mempunyai fungsi masing-masing dan blok rangkaian yang satu dengan blok rangkaian yang lain merupakan satu kesatuan yang saling terkait dan berhubungan serta membentuk satu kesatuan yang saling menunjang kerja dari sistem. Blok



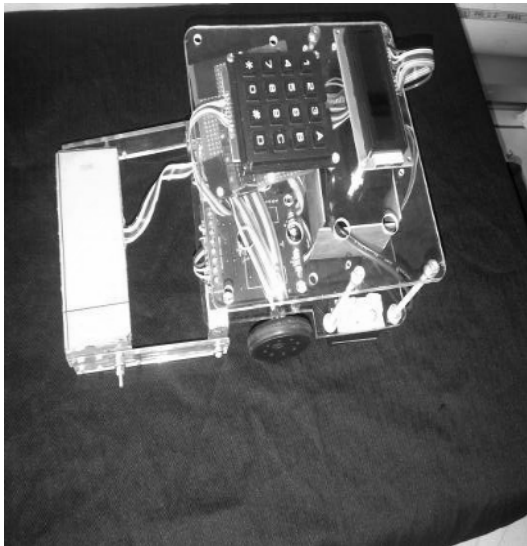
Gambar 3. Blok rangkaian dari Robot

rangkaian dari Robot ini dapat dilihat selengkapnya pada Gambar 3.



Gambar 4. Rangkaian Lengkap

Setelah perancangan sistem secara blok per blok ditentukan, maka perancangan terakhir akan digambarkan secara keseluruhan yang dapat dilihat pada gambar 4. Rangkaian keseluruhan sistem ini akan memperlihatkan keterkaitan seluruh sistem yang ada, mulai dari dua mikrokontroler ATMEGA8535 sebagai pusat dari pengendali sampai sensor-sensor sebagai *input* dan motor DC sebagai *output*-nya. Pada Gambar 4 merupakan konstruksi dari robot setelah semua komponen dirakit.



Gambar 5. Konstruksi robot setelah dirakit

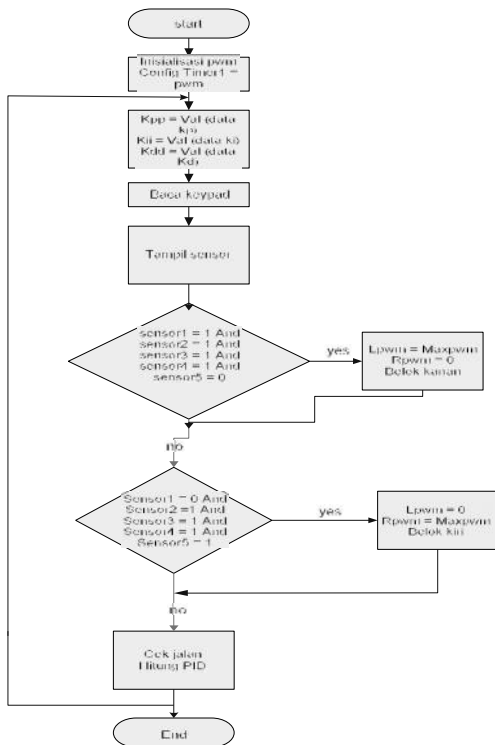
Mikrokontroler ATMEGA8535 ini merupakan keluarga dari AVR yang memiliki pin berjumlah 40, tetapi tidak semua di gunakan di bawah ini akan dijelaskan fungsi dari masing-masing *port* mikrokontroller yang digunakan, ditampilkan pada Tabel 1.

3.2 Perancangan software

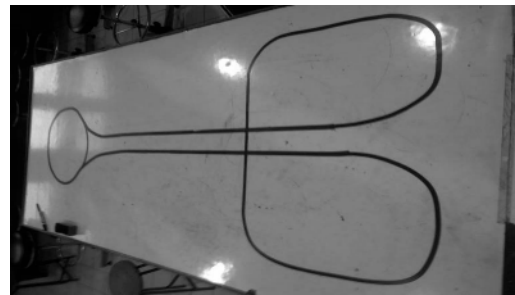
Dalam pembuatan software, listing program dibuat dengan sistem prosedur dimana setiap prosedur mempunyai fungsi tertentu dan secara keseluruhan prosedur membentuk sistem dari program yang diharapkan. Flowchart program utama rancangan alat dapat dilihat pada Gambar 6.

Tabel 1. Penggunaan Port mikrokontroler

NoPort mikrokontroler	penggunaan
Port a.0	sensor Photodiode sebagai sensor tracking kiri.
Port a.1	sensor Photodiode sebagai sensor tracking kiri tengah
Port a.2	sensor Photodiode sebagai sensor tracking tengah kiri
Port a.3	sensor Photodiode sebagai sensor tracking tengah kanan
Port a.4	sensor Photodiode sebagai sensor tracking kanan tengah
Port a.5	sensor Photodiode sebagai sensor tracking kanan
Port b.0 sd port b.7	input keypad ke mikrokontroler
Port c.0	Reques Send LCD
Port c.1	Enabel LCD
Port c.4 Sd c.7	input data LCD
Port d.2	mengaktifkan motor maju mundur
Port d.3	mengaktifkan motor kanan maju
Port d.4	mengaktifkan motor kanan mundur
Port d.5	mengaktifkan motor kiri
Port d.6	motor kiri mundur
Port d.7	motor kiri maju.



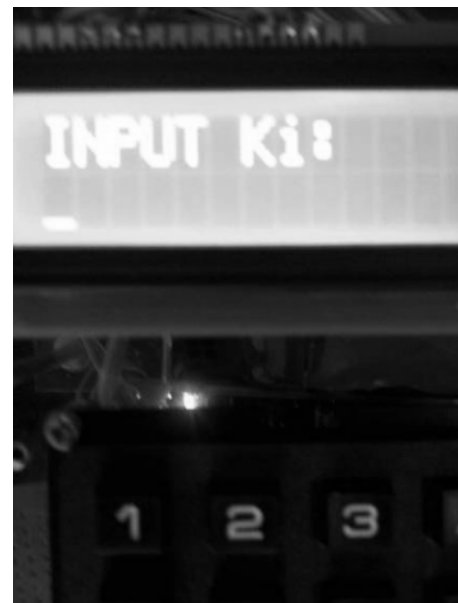
Gambar 6. Flowchart program



Gambar 7. Lintasan tracking



Gambar 8. Nilai KP



Gambar 9. Nilai Ki

IV. PEMBAHASAN

Robot ini dirancang untuk membaca atau melintasi lintasan hitam dengan lebar 1,5 cm. Pada prinsipnya robot ini merupakan robot tracking yang melakukan pergerakan berdasarkan pembacaan 5 buah sensor photodiode atau dalam selanjutnya disebut sensor tracking, yang diletakan pada bagian depan robot. Sensor tracking yang membaca lintasan untuk kemudian data tersebut akan dikirim ke mikrokontroler. Pada mikrokontroler ini yang berfungsi sebagai pusat pengendali dari robot dan keluaran dari hasil pembacaan pada mikro digunakan untuk menggerakkan motor DC dengankonsep PID. Sehingga robot tersebut dapat melewati seluruh lintasan tanpa mengalami kegagalan atau keluar lintasan. Pada Gambar 7 adalah gambar dari lintasan untuk tracking robot.

Sebelum menjalankan robot tersebut, terlebih dahulu menset nilai beberapa variable pid dengan menggunakan bantuan keypad. Seperti ditampilkan pada menu yang muncul di LCD pada robot. Set Nilai Proportional disebut nilai Kp, Set Nilai Integral disebut nilai Ki, dan Set Nilai Derivatif disebut nilai Kd. Set Nilai MAX PWM disebut nilai kecepatan maximum motor

Robot menggunakan 5 sensor di depan untuk mengikuti garis. Lebar garis yang ideal untuk diikuti adalah 1.5 cm dengan kemungkinan 1 - 2 sensor dapat mengenai garis. Langkah selanjutnya adalah melakukan mapping nilai sensor untuk mendapatkan *process variable* (PV). Kurang lebih seperti berikut (misal nilai 0 merepresentasikan sensor mengenai garis):



Gambar 10. Nilai Kd



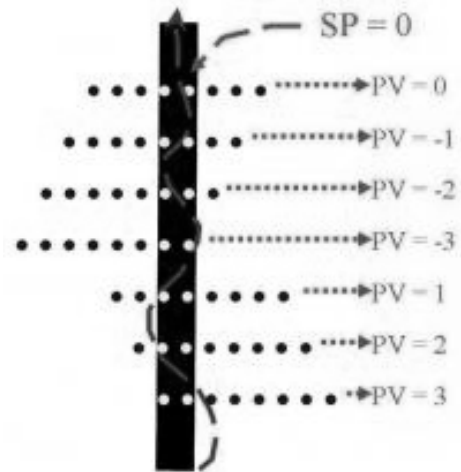
Gambar 11. Nilai Max PWM

Tabel 2. Pembacaan nilai sensor

No	Tabel	Nilai sensor	keterangan
1	11110	4	'ujung kanan
2	11100	3	
3	11101	2	
4	11000	1	
5	11001	1	
6	11011	0	tengah
7	10011	-1	
8	00011	-1	
9	10111	-2	
10	00111	-2	
11	01111	-4	'ujung kiri

Kondisi yang diharapkan pada robot adalah bergerak maju lurus mengikuti garis, dengan kata lain $PV = 0$ (nilai sensor = 11011). Dari sini *Set Point (SP)* / kondisi ideal, ditetapkan bernilai = 0. Nilai sensor yang dibaca oleh sensor disebut *Process Variable (PV)* / nilai aktual pembacaan. Menyimpangnya posisi robot dari garis disebut sebagai error (e), yang didapat dari $e = SP - PV$. Dengan mengetahui besar error, maka microcontroller dapat memberikan nilai untuk PWM motor kanan dan kiri yang sesuai agar dapat menuju ke posisi ideal ($SP = 0$). Besar PWM nilai ini didapat dengan menggunakan

kontrol Proportional (**P**), dimana $P = e * Kp$ (Kp adalah konstanta proportional yang nilainya kita set sendiri dari hasil tuning). Misalkan nilai PWM didefinisikan dari 0 - 255 dengan nilai 0 berarti berhenti dan 255 berarti kecepatan penuh kondisi real dimisalkan saat sepelan-pelannya motor adalah PWM 30 dan secepat-cepatnya (maju lurus) adalah 255. Saat $PV = 4$ atau -4 itu tergantung dari kondisi PV sebelumnya, jika PV lebih besar dari 0 maka, nilai PV adalah 4 dan jika PV kurang dari 0 maka nilai PV adalah -4 . Gambar 12 menunjukkan kondisi dari penyimpangan antara SP dikurangi PV ($e = SP - PV$).



Gambar 12. Penyimpangan antara SP dengan PV

Dengan mengukur seberapa jauh robot menyimpang dari kondisi ideal, sistem kontrol P sudah diterapkan. Output (berupa nilai PWM) didapat dari perhitungan yang melibatkan hanya variabel $P = e * Kp$. Jika pergerakan robot masih belum sesuai dengan garis (berbelok-belok), bisa ditambahkan kontrol *Derivative (D)*. Kontrol D digunakan untuk mengukur seberapa jauh robot bergerak dari kiri ke kanan atau dari kanan ke kiri. Semakin cepat bergerak dari satu sisi ke sisi lainnya, maka semakin besar nilai D. Konstanta D (Kd) digunakan untuk menambah atau mengurangi imbas dari derivative. Dengan mendapatkan nilai Kd yang tepat pergerakan sisi ke sisi yang berbelok-belok akibat dari proportional PWM bisa diminimalisasi. Nilai D didapat dari: $D = Kd * rate$, dimana $rate = e(n) - e(n-1)$. Dalam program nilai error ($SP - PV$) saat itu menjadi nilai *last_error*, sehingga *rate* didapat dari $error - last_error$ Untuk menambahkan kontrol D.

Jika dengan Proportional dan Derivatif sudah dapat membuat pergerakan robot cukup baik, maka penambahan Integral menjadi opsional. Dengan menambahkan Integral (**I**). I digunakan untuk mengakumulasi error dan mengetahui durasi error. Dengan menjumlahkan

Tabel 3. Hasil pengujian konsep PID

No	Kp	Ki	Kd	Letak Robot	Nilai PID	Pwm Kiri	Pwm Kanan
1	1	1	5	Tengah Garis	0	255	255
2	1	1	5	Kiri Garis	0	255	255
3	1	1	5	Kanan Garis	0	255	255
4	6	1,5	9	Tengah Garis	0	255	255
5	6	1,5	9	Kiri Garis	-1	255	0
6	6	1,5	9	Kanan Garis	1	0	255

error disetiap pembacaan PV akan memberikan akumulasi offset yang harus diperbaiki sebelumnya. Saat robot bergerak menjauhi garis, maka nilai error akan bertambah. Semakin lama tidak mendapatkan SP, maka semakin besar nilai I. Dengan mendapatkan nilai Ki yang tepat, imbas dari Integral bisa dikurangi. Nilai akumulasi error didapat dari: $I = I + \text{error}$. Nilai I sendiri : $I = I * Ki$. Jika diinginkan nilai $MV = P + I + D$.

Pengujian rangkaian ini dilakukan untuk memeriksa apakah konsep PID berjalan dengan semestinya yang nantinya dimasukkan kedalam inputan data PID. Sehingga memenuhi seperti yang diharapkan. Pengujian konsep ini dapat dilihat pada Tabel 3.

V. PENGUJIAN DAN ANALISA RANGKAIAN SENSOR PHOTODIODA

Tegangan referensi (V_{ref}) dari komparator ditentukan oleh tegangan potensiometer. Sedangkan tegangan input (V_{in}) komparator diperoleh dari bagi tegangan sumber (+5 V) antara Photodiode dengan R330.

Saat Photodiode terkena pantulan cahaya maka hambatan dari Photodiode menjadi sangat kecil sehingga V_{in} menjadi kecil, jika tegangan $V_{in} < V_{ref}$, maka output komparator akan menjadi high (+ 5V). Sedangkan saat dalam keadaan gelap hambatan dari Photodiode menjadi sangat besar sehingga V_{in} menjadi besar dan jika $V_{in} > V_{ref}$ maka output dari komparator akan menjadi low (0 V). Dengan prinsip kerja ini rangkaian mampu mendeteksi kondisi garis berwarna hitam dengan output low (+0 V) dan putih dengan output high (+5 V).

Karena kerja dari komparator hanya membandingkan V_{in} dengan V_{ref} -nya maka dengan mengatur V_{ref} , kita sudah mengatur kepekaan sensor terhadap perubahan tingkat intensitas cahaya yang terjadi. Dimana semakin

rendah V_{ref} semakin sensitive komparator terhadap perubahan tegangan V_{in} yang diakibatkan oleh perubahan intensitas cahaya.

Pengujian rangkaian Sensor Photodiode dilakukan dengan cara meletakkan sensor di garis tracking berwarna hitam dan dilapangan tracking berwarna putih. Rangkaian sensor Photodiode ini diuji sudah menggunakan penutup cahaya dan ketinggian 5 mm. Tabel 4. menunjukkan pengujian sensor photodiode terhadap pembacaan track warna hitam dan putih.

Tabel 4. Hasil pengujian sensor Photodiode

Pengujian	Kondisi pada warna putih (Nilai Logika)	Kondisi pada warna hitam (Nilai Logika)
1	1	0
2	1	0
3	1	0
4	1	0
5	1	0

VI. KESIMPULAN

Setelah melakukan perencanaan dan pembuatan sistem kemudian dilakukan pengujian dan analisa dari robot, maka diperoleh beberapa kesimpulan.

1. Dengan menggunakan mikro-kontroler ATMEGA8535, jumlah parameter Input dan dan parameter output lebih banyak, sehingga pergerakan robot lebih tepat. Dan mikrokontroler ATMEGA8535 tersebut saling terhubung dan berkomunikasi sesuai dengan program yang dibuat.

2. Pergerakan robot sudah sesuai dengan konsep PID, dimana robot dapat berjalan pada track yang sudah ditetapkan. Pada jalur hitam dengan lebar 1,5 cm dan jika track berwarna hitam akan menghasilkan nilai logik 0 dan jika track berwarna putih akan menghasilkan nilai logik 1.
3. Sensor Photodiode digunakan sebagai sensor tracking dalam penggunaannya di robot ini harus dipasang secara baik dan benar, letak ketinggian antara sensor dan lantai dapat mempengaruhi kinerja sensor tersebut.
4. Sensor harus diletakkan secara tepat dan sejajar atau tegak lurus dengan lantai untuk memberikan hasil pembacaan yang baik, kesalahan peletakan pada sensor dapat mempengaruhi kinerja sensor.
5. PID (*Proportional-Integral-Derivative*) merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem.
6. Nilai error pada formula PID akan diolah untuk mendapatkan nilai yang ideal. Berdasarkan perbandingan antara set point dan sinyal umpan balik.

DAFTAR PUSTAKA

- [1] www.atmel.com
- [2] Pratomo, Andi. 2005 *panduan praktis pemrograman AVR Mikrokontroler Atmega8535/16*. YOGYAKARTA: Penerbit Andi, ISBN : 979-731-464-7
- [3] <http://id.wikipedia.org/wiki/PID> diakses hari Selasa, 12/4/2012 2:50 PM
- [4] Setiawan, 2008. Kontrol PID untuk proses industri. Jakarta: Penerbit Elekmedia.987-979-27-4100-1
- [5] <http://amryagus.blogspot.com/2011/04/sistem-kontrol.html> diakses hari Senin, 10/12/2012