

PENGENDALIAN MOTOR DC ROBOT MIROSOT UPN VETERAN YOGYAKARTA MENGUNAKAN MIKROKONTROLER ATMEGA 8

by Dessyanto Boedi Prasetyo

Submission date: 03-Jul-2020 12:39AM (UTC+0700)

Submission ID: 1352674524

File name: 1051-3020-1-PB.pdf (552.75K)

Word count: 2687

Character count: 16612

² PENGENDALIAN MOTOR DC ROBOT MIROSOT UPN VETERAN YOGYAKARTA MENGGUNAKAN MIKROKONTROLER ATMEGA 8

Awang Hendrianto Pratomo¹⁾, Nuryono Setya Widodo²⁾, Dessyanto Boedi Prasetyo³⁾, & Muhammad Rifai Samekta Adi⁴⁾

³
^{1,3,4)} Program Studi Teknik Informatika, Fakultas Teknologi Industri, UPN Veteran Yogyakarta, Jl. Babarsari No. 2, Tambakbayan Yogyakarta, 55281.

²⁾ Program Studi Teknik Elektro, Fakultas Teknologi Industri, Universitas Ahmad Dahlan Yogyakarta, Jl. Prof. Dr. Soepomo, S.H., Janturan, Warungboto, Umbulharjo, Yogyakarta 55164

³
e-mail : awang@upnyk.ac.id, wiweet2@yahoo.com, dess95@gmail.com, & samektaadi17@gmail.com

Abstrak

Pada sistem robot UPN V Yogyakarta, mikrokontroler baru digunakan sebagai perantara bagi pusat pengendali seluruh sistem atau host komputer dan motor DC. Mikrokontroler akan menerima data dari sistem komunikasi dan kemudian langsung mengirimkan secara langsung data tersebut ke motor DC. Hal ini dikarenakan robot soccer mikro Teknik UPN V Yogyakarta adalah robot yang dibangun dari nol, maka pengendalian pergerakan dasar robot diperlukan untuk mendukung pengendalian pergerakan yang lebih kompleks. Sistem yang dirancang berkaitan dengan beberapa komponen hardware robot yaitu micro robot soccer menggunakan sistem penggerak motor DC yang memiliki pulsa berjumlah 512 encoder. Mikrokontroler ATmega8 digunakan sebagai sistem kontrol motor dan komunikasi robot. Bahasa yang digunakan adalah bahasa pemrograman C untuk mengimplementasikan kendali motor di dalam mikrokontroler. Untuk mengirimkan kecepatan digunakan aplikasi yang dibuat dengan bahasa pemrograman C#. Dengan membangun kendali motor DC pada mikrokontroler atmega8, pergerakan robot diharapkan mampu bergerak sesuai perintah yang dikirimkan oleh komputer host.

Kata Kunci : MiroSot, Motor DC, Mikrokontroler, ATmega8

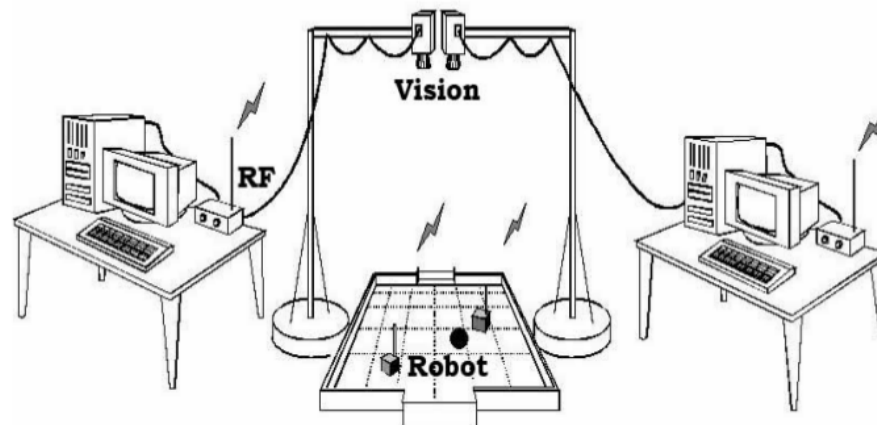
1. PENDAHULUAN

Mirosot merupakan robot yang menggunakan motor DC untuk bergerak diatas 2 roda dan membutuhkan *microcontroller* untuk menjalankan dan mengendalikan sumber daya hardware yang ada (Kim et al, 1997). *Microcontroller* merupakan perantara bagi pusat pengendali seluruh sistem dan motor DC. Singkatnya *microcontroller* akan menerima data dari sistem komunikasi dan kemudian langsung akan mengirimkan data yang diperolehnya tersebut ke motor DC (Huabin et al, 2004). Hal tersebut tentu saja akan sangat membantu apabila diterapkan sebagai ke² ampuan dasar untuk pergerakan robot Mirosot. Teknik Informatika UPN "Veteran" Yogyakarta merupakan salah satu perguruan tinggi yang ikut berpartisipasi dalam mengembangkan robot Mirosot. Mirosot yang dibangun dari nol belum memiliki pergerakan yang terkendali, oleh karenanya pergerakan robot tersebut menjadi kurang teratur dan strategi yang diterapkan tidak berjalan secara tepat dan akurat.

Penelitian ² membahas mengenai kendali motor DC pada *microcontroller* untuk mengendalikan pergerakan robot mirosot. Proses pengendalian diawali dengan diterimanya data dari RF (*Radio Frequention*). Selanjutnya *microcontroller* akan mengolah data tersebut dan membandingkannya dengan nilai *encoder* yang muncul berdasarkan perputaran mo³ DC. Nilai *encoder* akan dipaksa oleh sistem kendali untuk menghasilkan nilai yang sesuai dengan data dari RF. Tujuan dari penelitian ini adalah untuk membuat sistem kendali motor DC yang memastikan robot Mirosot mampu bergerak sesuai perintah dan mencapai posisi yang dituju dengan kecepatan yang diinginkan.

2. KAJIAN LITERATUR

Mirosot merupakan salah satu kategori permainan robot sepakbola internasional yang dinaungi oleh FIRA. Dibutuhkan kombinasi kontrol motor, vision system, dan komunikasi yang membuat permainan berjalan secara baik dan teratur (Kim et al, 1997; Huabin et al, 2004). Secara spesifikasi, robot mirosot memiliki ukuran 7,5 x 7,5 x 7,5 cm dengan ukuran lapangan 220 x 180 cm (FIRA, 2014) seperti ditunjukkan dalam Gambar 1.



Gambar 1 Sistem Mirobot

2.1. Vision System

Vision system digunakan untuk mengenali objek dan mentransformasi objek lapangan berdasarkan keadaan nyata. Salah satu teknik *vision system* dalam mengenali objek di lapangan menggunakan *Color Patch* yang berisi informasi tentang warna tim dan nomor id robot (Jiang H et al, 2004; Choi et al, 2009).

2.2. Sistem Komunikasi

Sistem komunikasi diperlukan untuk mengontrol robot. Secara aturan, sistem komunikasi yang diperbolehkan adalah sistem komunikasi nirkabel. Saat ini *Radio Frequency* masih menjadi pilihan terbaik untuk sistem komunikasi karena dapat digunakan untuk komunikasi jarak jauh dan mendukung *multichannel* (Kim et al, 1997; Li et al, 2004).

2.3. Motor DC

Motor DC sering diterapkan dalam menggerakkan mirosot, hal tersebut dikarenakan desain sistem yang ada mampu membuat mirosot bergerak cepat dengan dua roda dan sepasang motor. Pada implementasi untuk motor DC berdaya rendah, digunakan driver *chopper* atau *H-Bridge*. *H-Bridge* dapat berubah nyala atau mati tergantung pada input sinyal yang berupa PWM (*Pulse Width Modulation*) (Vieira et al, 2011).

Dari beberapa jenis motor, terdapat motor DC dimana didalamnya terfasilitasi encoder. Encoder berfungsi sebagai alat yang mendeteksi putaran motor. Kontrol perputaran motor membutuhkan informasi yang dimiliki encoder. Informasi encoder ini akan memberikan data tentang perputaran motor DC. Nilai yang datang dari encoder tersebut kemudian digunakan untuk menentukan nilai tegangan listrik yang dibutuhkan. Tegangan yang diperlukan untuk mengendalikan motor yang setara dengan sinyal PWM (Vieira et al, 2011)

2.4. Mikrokontroler

Mikrokontroler merupakan pusat pengendali bagi perangkat lain yang didalamnya telah diisi instruksi-instruksi atau program sebelum digunakan, sehingga mampu mengatur kerja agar sistem bekerja sesuai yang diinginkan (Nataliana et al, 2013; Rizki Setiawan et al, 2013; Vieira et al, 2011). ATmega8 adalah contoh produk AVR dari kelas ATmega yang dikeluarkan oleh Atmel Corp. Mikrokontroler ATmega8 mempunyai 28 pin untuk kemasan PDIP (*Plastic Dual Inline Package*) dan 32 pin untuk kemasan TQFP (*Thin Quad Flat Pack*) dengan fasilitas cukup lengkap yaitu 23 jalur Input / Output, 8KByte *In System Programmable Flash*, 512bytes EEPROM, 1Kbytes Internal SRAM, Internal ADC, *Timer/Counter*, SPI, dan USART (Harsono et al, 2009; Jesse, 2007; Huabin et al, 2004; Peter Kopacek, 2008).

Dalam mengendalikan kecepatan motor sering digunakan mikrokontroler sebagai sub sistem dalam memenuhi kecepatan yang diinginkan. Di dalam penelitian yang dilakukan oleh Nataliana (Nataliana et al, 2013), mikrokontroler digunakan untuk mengendalikan perputaran DC motor. Pada saat motor DC berputar maka sensor akan mendeteksi putaran yang terjadi dalam satuan RPM (*Rotation Per Minute*). Sensor kemudian mengirim data perputaran motor ke mikrokontroler. Kemudian mikrokontroler akan membandingkan data perputaran motor (dalam RPM) dengan *set point*. Apabila terjadi kesalahan, maka mikrokontroler akan memperbaiki kesalahan tersebut. Mikrokontroler akan memberikan sinyal PWM dengan jumlah tertentu untuk menggerakkan motor DC sesuai *set point*.

2.5. PID

PID merupakan pengendali yang menggabungkan Proportional (P), integral (I), dan derivative (D) yang menjadi standar control proses dan biasa digunakan pada sistem otomatisasi rumit yang menghilangkan kebutuhan operator untuk terus mengawasi jalannya proses. Sehingga dengan adanya PID, proses sistem akan otomatis terawasi. Jika terjadi error maka akan secara langsung diperbaiki hingga mencapai nilai error seminimal mungkin (Ruzita S, 2009; Hyalij et al, 2009; Malhotra et al, 2011).

Konsep kendali PID adalah dengan memberikan nilai pada masing-masing konstanta *proportional* (P), *integral* (I), dan *derivative* (D) dan menghitung nilai *error* terhadap nilai *set point* untuk memberikan umpan balik kepada sistem. Kendali PID akan terus melakukan perbaikan selama suatu sistem dijalankan dan memberikan kontrol terhadap sistem untuk mencapai dan memberikan nilai stabil terhadapnya (Hyalij et al, 2009). Kendali PID didisain untuk mengurangi nilai *error* yang muncul pada suatu sistem (Malhotra et al, 2011).

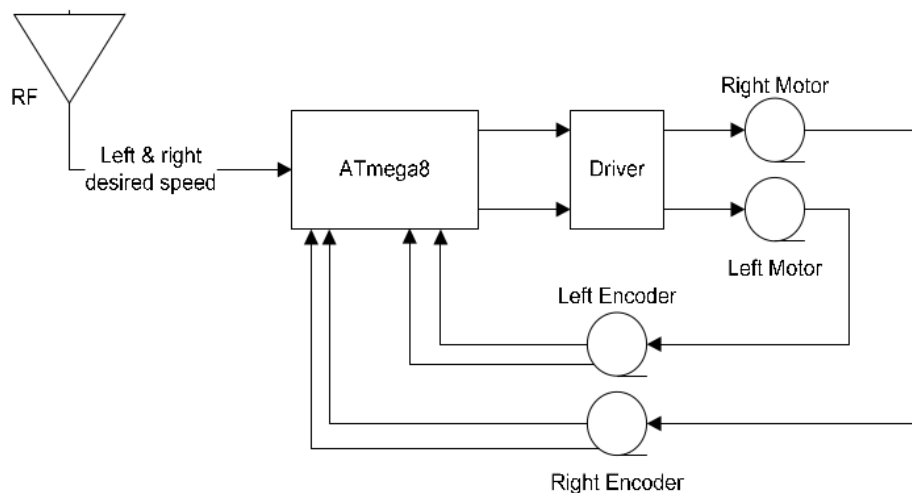
$$K_p + \frac{1}{T_i} \int e(t)dt + T_d \frac{de(t)}{dt} = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(t)dt + T_d \frac{de(t)}{dt} \right] \quad (1)$$

$$u = K_p e + K_i \int e dt + K_d \frac{de}{dt} \quad (2)$$

3. METODE PENELITIAN

3.1. Blok Diagram Sistem Robot Mirobot

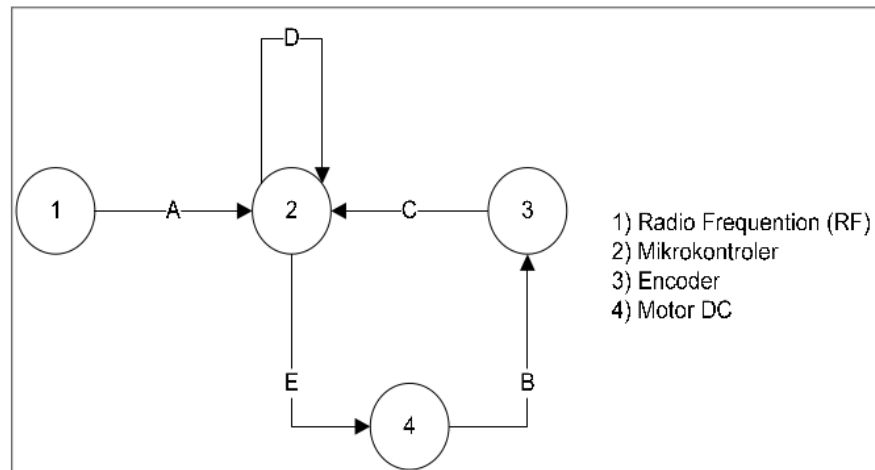
Untuk dapat bergerak, robot mirosot diperintah oleh komputer melalui RF *transmitter*. Kemudian data yang dikirim dari komputer akan diterima oleh *receiver* RF yang berada pada robot. Data tersebut berupa data kecepatan yang diinginkan. Selanjutnya mikrokontroler ATmega8 akan mengolah data tersebut untuk menggerakkan motor kanan dan kiri melalui driver motor. Dengan bergeraknya motor, encoder akan membaca putaran motor dan hasil pembacaanya dijadikan sebagai pembanding terhadap kecepatan yang diinginkan.



Gambar 2 Blok diagram sistem robot mirosot

3.2. Arsitektur Sistem

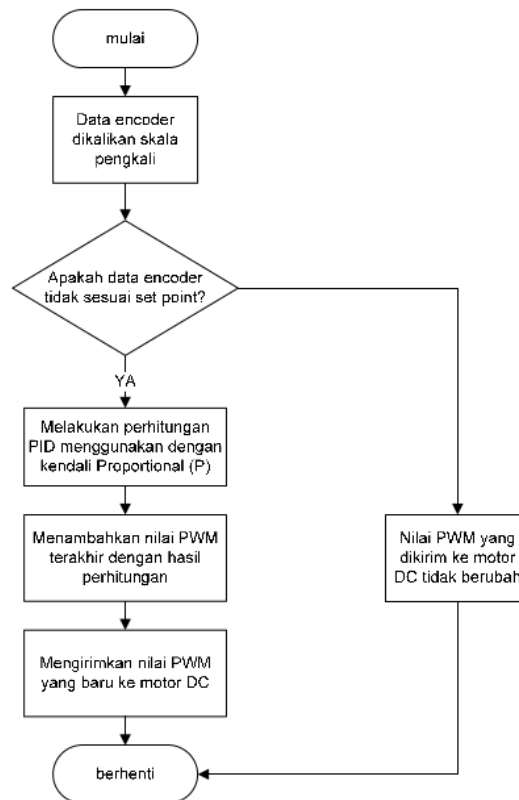
Sistem kendali robot mirosot UPN Veteran Yogyakarta merupakan sistem yang terdiri dari beberapa perangkat yaitu *Radio Frequention* (RF), mikrokontroler dan motor DC yang memiliki fasilitas *encoder*. Cara kerja sistem kendali (Gambar 3) diawali dengan perintah kecepatan yang dikirimkan melalui gelombang radio (Proses A). Perintah ini kemudian akan diterima oleh mikrokontroler untuk diolah bersama data pembacaan *encoder*. Data *encoder* didapatkan dari pergerakan motor DC. Data tersebut berupa angka yang menunjukkan munculnya sinyal *encoder* terhadap aktivitas motor DC (Proses B dan C). Proses selanjutnya yaitu mikrokontroler akan membandingkan perintah kecepatan dan data *encoder* (Proses D). Hasil dari proses membandingkan tersebut adalah suatu nilai yang harus dikirimkan ke motor DC untuk bergerak sesuai perintah kecepatan yang datang dari RF (Proses E). Proses mikrokontroler membandingkan perintah kecepatan dengan data *encoder* untuk menghasilkan nilai yang dikirim ke motor DC akan berulang sampai RF mengirimkan data kecepatan yang baru.



Gambar 3 Arsitektur Sistem Kendali

3.3. Flowchart kendali kecepatan atau fungsi mcontrol()

Fungsi `mcontrol()` merupakan inti dari sistem kendali perputaran motor DC. Fungsi ini dipanggil oleh fungsi *external timer interrupt 0* yang dijalankan setiap 10 *millisecond*. Pada fungsi ini data yang didapatkan melalui RF akan dibandingkan dengan jumlah perhitungan jumlah *encoder* yang dihasilkan oleh *external interrupt 0* dan *external interrupt 1*. Data yang berasal dari RF berupa data kecepatan dijadikan *set point* kecepatan. Jumlah *encoder* dikendalikan jumlahnya sesuai *set point* untuk menghasilkan kecepatan motor yang diinginkan dengan mengubah nilai yang dikirim ke motor DC sehingga robot berjalan dengan kecepatan yang sesuai. *Flowchart* fungsi kendali ditampilkan pada Gambar 4.



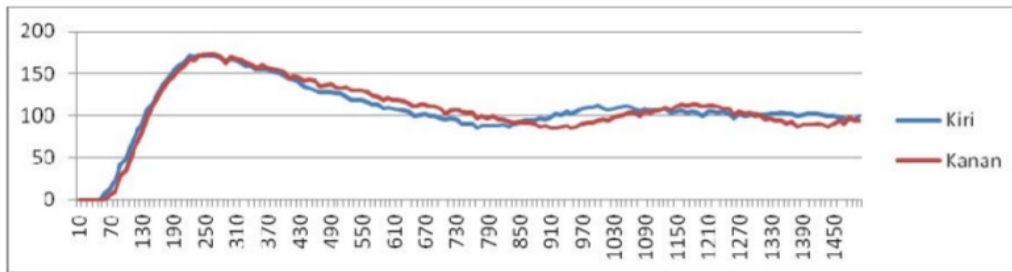
Gambar 4 Flowchart kendali kecepatan atau fungsi mcontrol()

Fungsi kendali diawali dengan membuat nilai encoder mempunyai rentang nilai yang sama dengan nilai PWM yang dikirimkan ke motor DC. Hal ini perlu dilakukan untuk memudahkan logika berpikir antara nilai *set point*, nilai encoder dan nilai PWM yang harus dikirim ke motor DC. Kemudian proses selanjutnya adalah pengecekan nilai encoder dengan nilai *set point*. Apabila sama maka tidak ada perbaikan terhadap nilai PWM yang dikirimkan, sedangkan jika berbeda maka sistem kendali akan menghitung perbaikan dengan konsep PID dimana dalam sistem ini hanya digunakan kendali *proprtional* (P) dalam mengendalikan perputaran motor DC. Setelah didapatkan hasil perhitungan, nilai PWM baru yang akan dikirim adalah PWM terakhir yang dikirim ditambah dengan hasil perhitungan kendali *proprtional* (P). Hasilnya dikirim ke motor DC untuk menggerakkan motor DC sesuai perintah. Sistem kendali ini akan mengendalikan perputaran motor DC setiap 10 *millisecond* sesuai dengan jalannya fungsi *external timer interrupt 0*.

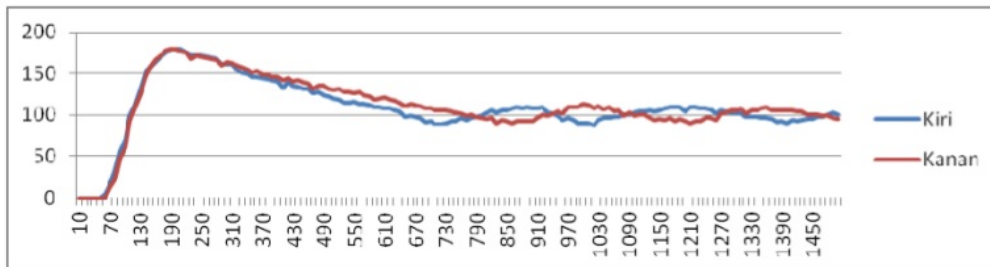
4. HASIL DAN PEMBAHASAN

Pengujian akan dilakukan untuk menentukan beberapa parameter yang berhubungan dengan kendali kecepatan pada mikrokontroler. Kendali kecepatan yang akan digunakan merupakan salah satu jenis dari kendali PID yaitu kendali *Proportional*. Untuk menerapkan kendali *Proportional* harus ditentukan konstanta *proportional* yang memiliki pengaruh terhadap sistem yang terbaik. Robot akan dijalankan dengan konstanta *proportional* yang berbeda-beda dan direkam datanya.

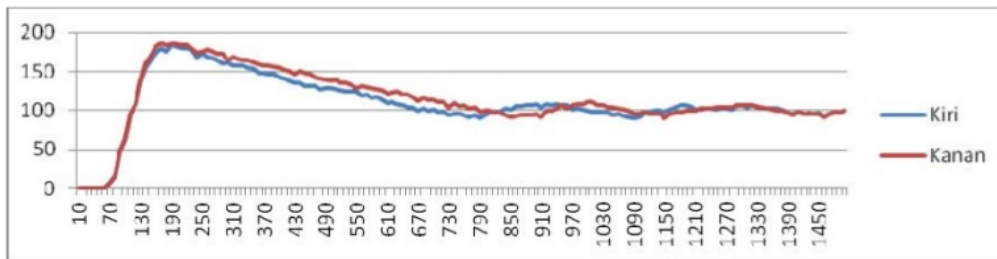
Dalam pengujian konstanta *proportional* grafik menunjukan adanya *overshoot* pada nilai kecepatan saat awal robot menerima kecepatan. Kemudian kecepatan menurun secara bertahap mendekati nilai *set point*. Hasil pengujian dengan nilai *set point* 100 tanpa beban robot ditunjukkan oleh gambar 5 - 14. Dalam grafik tersebut sumbu X merupakan waktu dalam *millisecond* dan sumbu Y adalah nilai *encoder* yang muncul.



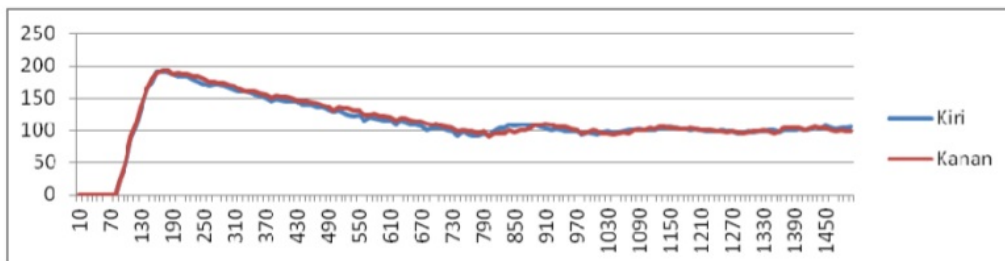
Gambar 5 grafik pengujian dengan *set point* 100 konstanta *proportional* 0.1



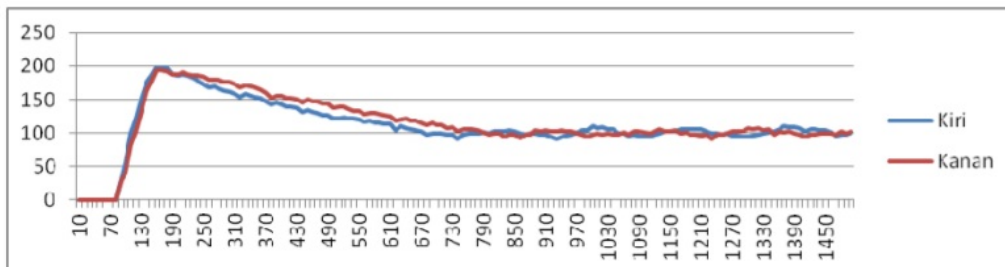
Gambar 6 grafik pengujian dengan *set point* 100 konstanta *proportional* 0.2



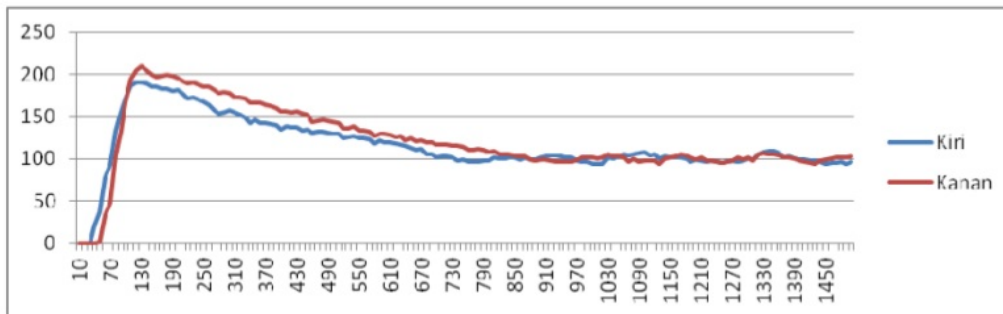
Gambar 7 grafik pengujian dengan *set point* 100 konstanta *proportional* 0.3



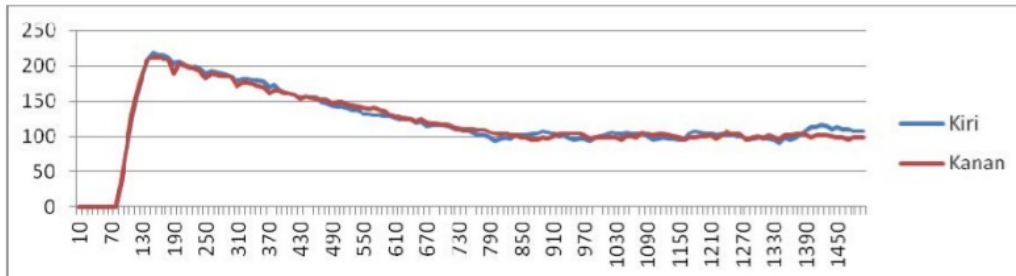
Gambar 8 grafik pengujian dengan *set point* 100 konstanta *proportional* 0.4



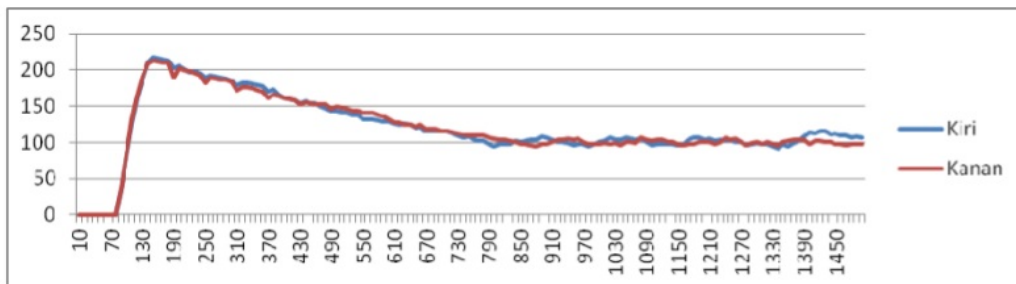
Gambar 9 grafik pengujian dengan *set point* 100 konstanta *proportional* 0.5



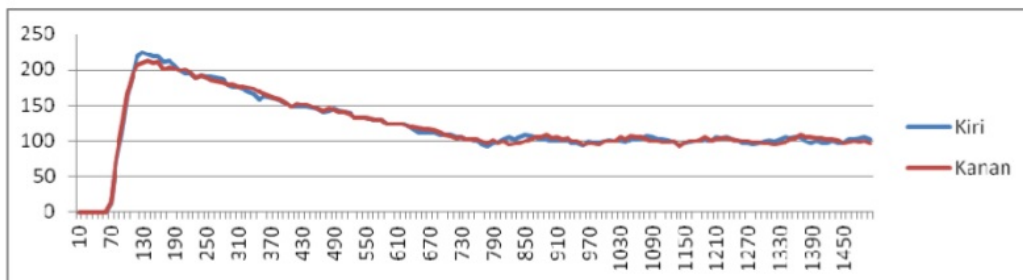
Gambar 10 grafik pengujian dengan *set point* 100 konstanta *proportional* 0.6



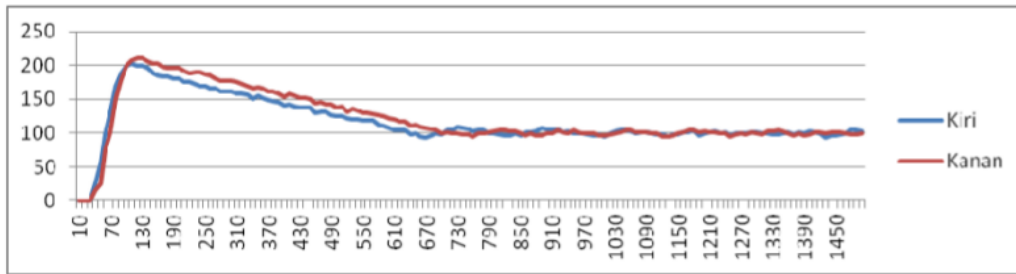
Gambar 11 grafik pengujian dengan *set point* 100 konstanta *proportional* 0.7



Gambar 12 grafik pengujian dengan *set point* 100 konstanta *proportional* 0.8

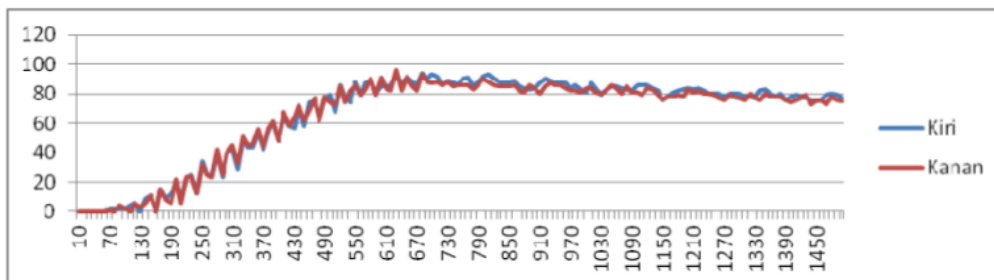


Gambar 13 grafik pengujian dengan *set point* 100 konstanta *proportional* 0.9

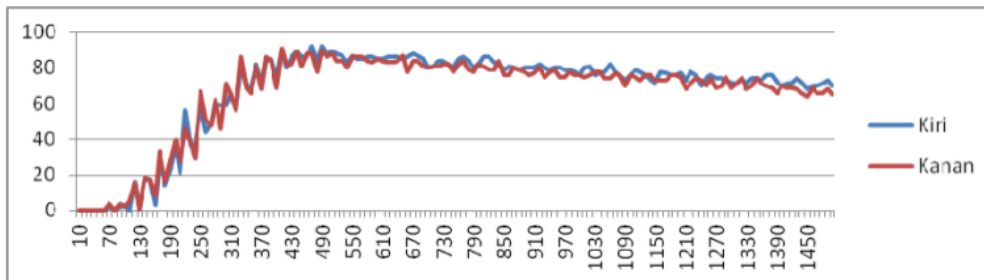


Gambar 143 grafik pengujian dengan set point 100 konstanta *proportional* 1.0

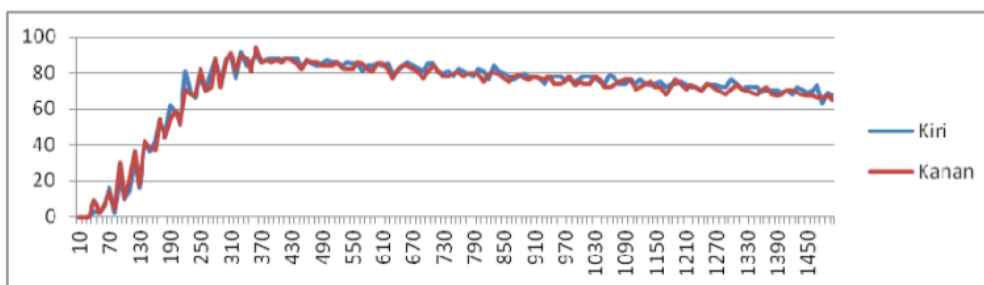
Selain dilakukan pengujian respon motor tanpa beban, dilakukan juga pengujian respon motor DC dengan beban robot. Nilai *set point* adalah 50 dengan konstanta *proportional* antara 0,1 – 1,0. Grafik respon motor DC dapat dilihat pada gambar 15 – 24.



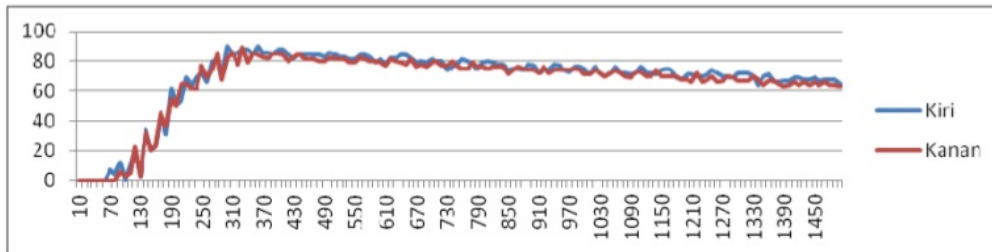
Gambar 15 grafik pengujian dengan set point 50 konstanta *proportional* 0.1



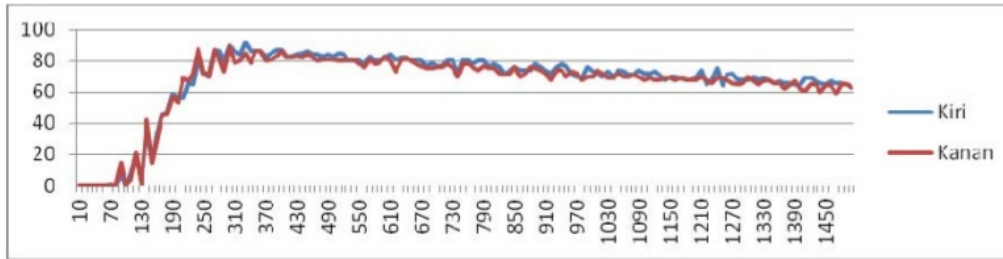
Gambar 16 grafik pengujian dengan set point 50 konstanta *proportional* 0.2



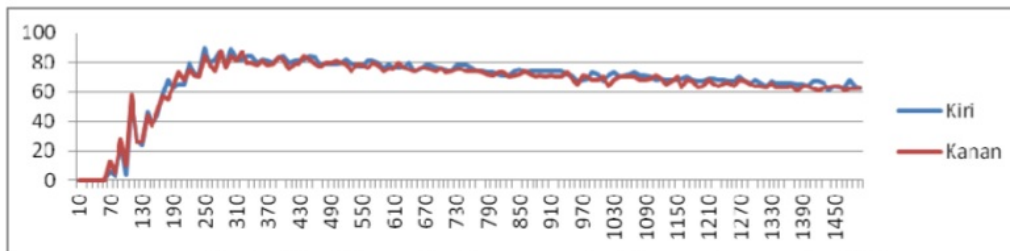
Gambar 17 grafik pengujian dengan set point 50 konstanta *proportional* 0.3



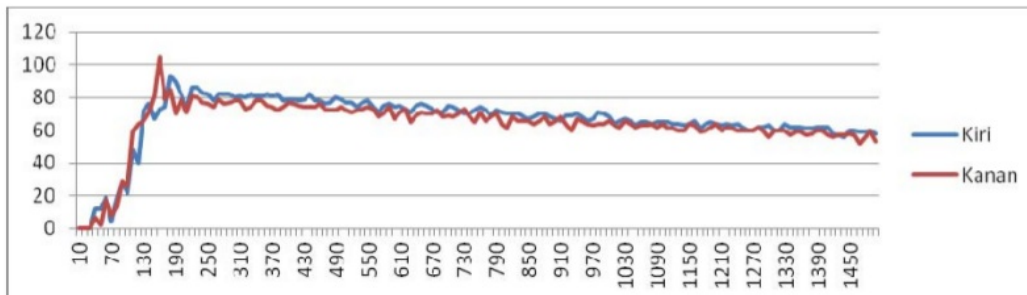
Gambar 18 grafik pengujian dengan set point 50 konstanta *proportional* 0.4



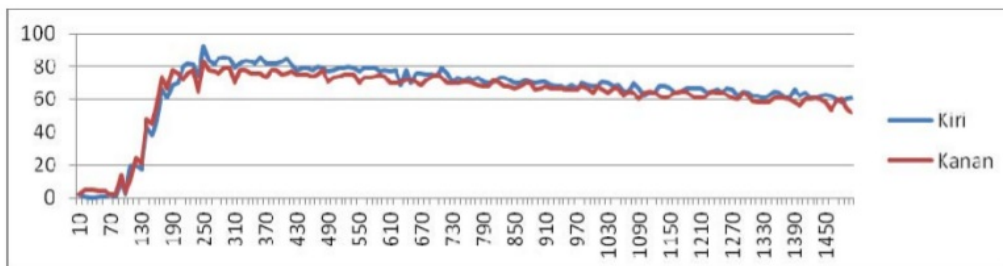
Gambar 19 grafik pengujian dengan set point 50 konstanta *proportional* 0.5



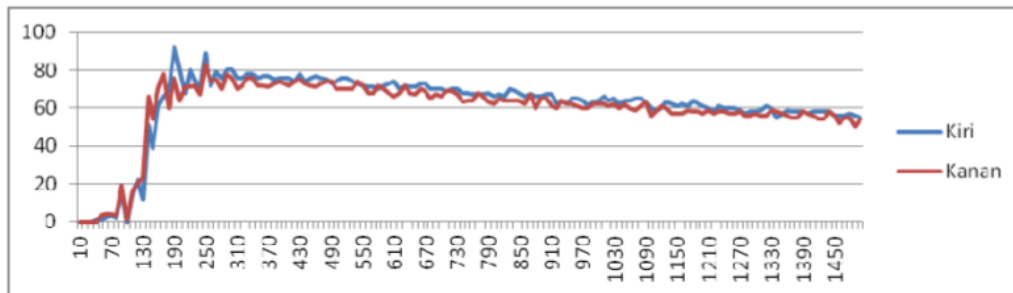
Gambar 20 grafik pengujian dengan set point 50 konstanta *proportional* 0.6



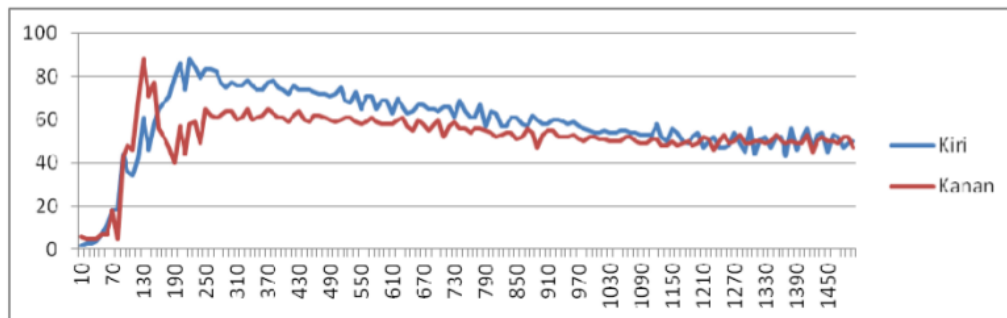
Gambar 21 grafik pengujian dengan set point 50 konstanta *proportional* 0.7



Gambar 22 grafik pengujian dengan set point 50 konstanta *proportional* 0.8



Gambar 23 grafik pengujian dengan set point 50 konstanta proportional 0.9



Gambar 24 grafik pengujian dengan set point 50 konstanta proportional 1.0

Hasil grafik diatas didapatkan dengan menjalankan sistem kendali yang dibuat berdasarkan flowchart pada bagian metodologi. Sistem kendali membaca kecepatan yang dikirim oleh komputer melalui RF. Kemudian sistem mengirim perintah tersebut ke motor DC dan mikrokontroler mengendalikannya agar kecepatan yang dihasilkan sesuai dengan set point atau kecepatan yang dikirimkan.

5. KESIMPULAN

Setelah dilakukan pembangunan sistem kendali pada mikrokontroler atmega8 terhadap motor DC robot mirosot dan dilakukan pengujian dapat ditarik kesimpulan bahwa sistem kendali yang dibangun mampu mengendalikan kecepatan motor DC untuk berputar sesuai set point. Dari grafik hasil pengujian dapat dilihat bahwa respon motor DC pada awalnya melebihi nilai set point dan tidak stabil, namun mampu dikendalikan oleh sistem kendali hingga kecepatan motor kembali berada pada di sekitar angka set point. Respon putaran motor DC untuk mencapai keadaan stabil sesuai set point berada di sekitar 800 millisecond untuk motor DC tanpa beban dan hampir 1,5 detik respon motor DC untuk stabil apabila terdapat beban robot.

DAFTAR PUSTAKA

- Decy Nataliana, M.T., Hari Wahyudi, M.T., Perdi Rusdiansyah, S.T., "Perancangan dan Implementasi Pengendalian Kecepatan Motor DC Penguatan Terpisah Menggunakan Pwm Berbasis Mikrokontroler Atmega 8535", Jurusan Teknik Elektro Institut Teknologi Nasional (ITENAS) Bandung, 2013.
- Djiwo Harsono, Joko Sunardi, Desi Biantara, "Pemantauan Suhu Dengan Mikrokontroler Atmega8 Pada Jaringan Lokal", Sekolah Tinggi Teknologi Nuklir – BATAN Yogyakarta, 2009.
- Dr. Norbert Jesse, "Robot Soccer: A Test-bed for Affordable Automation Solutions ?", Universität Dortmund, Germany, 2007.
- Frederico Carvalho Vieira, Pablo Javier Alsina, Adelardo Adelino Dantas de Medeiros, "Micro-Robot Soccer Team - Mechanical And Hardware Implementation", DCA – CT – Universidade Federal do Rio Grande do Norte, Campus Universitário – Lagoa Nova, 2011.

- G.S. Hyalij, A.U. Deshpande, P. D. Shendge, B.M. Patre, "Real Time Implementation of Time Delay Controller for DC Motor Speed Control", International Journal of Recent Trends in Engineering, 2009.
- Jiang H, Peng Q, Lee HA, C Teoh EL, Sng HL, "Colour Vision and Robot/Ball Identification for a Large Field Soccer Robot System", Centre for Advanced Robotics & Intelligent Control School of Electrical and Electronic Engineering Singapore Polytechnic, Singapore, 2004.
- Jong-Hwan Kim, Hyun-Sik Shim, Heung-Soo Kim, Myung-Jin Jung dan Prahlad Vadakkepat, "MICRO-ROBOT SOCCER SYSTEM: Action Selection Mechanism and Strategies", Dept. of Electrical Engineering, KAIST, Korea, 1997.
- Muhammad Rizki Setiawan, M. Aziz Muslim dan Goegoes Dwi Nusantara, "Kontrol Kecepatan Motor DC Dengan Metode PID Menggunakan Visual Basic 6.0 dan Mikrokontroler ATmega 16", Teknik Elektro Fakultas Teknik Universitas Brawijaya Malang, 2013.
- Peter Kopacek, "ROBOTSOCCER", Intelligent Handling and Robotics (IHRT), Vienna University of Technology Favoritenstr. 9-11, A-1040 Vienna, Austria, 2008.
- Rahul Malhotra, Tejbeer Kaur, Gurpreet Singh Deol, "Dc Motor Control Using Fuzzy Logic Controller", IJAEST, 2011.
- Ruzita Sumiati, "Analisis Pengendalian Motor DC Menggunakan Logika PID dengan Mikrokontroler atmega8535", Teknik Mesin Politeknik Padang, 2009.
- Seung-Hwan Choi, Seungbeom Han, and Jong-Hwan Kim, "Soty-Segment: Robust Color Patch Design to Lighting Condition Variation", Department of Electrical Engineering and Computer Science, KAIST, Daejeon, Republic of Korea, 2009.
- Tang Huabin, Wang Lei, Sun Zengqi, "Accurate and Stable Vision in Robot Soccer", Department of Computer Science and Technology, Tsinghua University, Beijing, China, 2004.
- www.fira.net, 2014.
- Yangmin Li, Wai Ip Lei, and Xiaoshan Li, "Multi-Agent Control Structure for a Vision Based Robot Soccer System", Department of Electromechanical Engineering Faculty of Science and Technology University of Macau, 2004.

PENGENDALIAN MOTOR DC ROBOT MIROSOT UPN VETERAN YOGYAKARTA MENGGUNAKAN MIKROKONTROLER ATMEGA 8

ORIGINALITY REPORT

8%

SIMILARITY INDEX

7%

INTERNET SOURCES

3%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

- | | | |
|--|---|--|
| <div style="background-color: red; color: white; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 0 auto;">1</div> | <p>Damar Nurcahyono, Willy Permana Putra, Abdul Najib, Tien Rahayu Tulili. "Analysis sentiment in social media against election using the method naive Bayes", Journal of Physics: Conference Series, 2020</p> <p>Publication</p> | <div style="font-size: 2em; font-weight: bold;">3%</div> |
| <hr/> | | |
| <div style="background-color: purple; color: white; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 0 auto;">2</div> | <p>eprints.upnyk.ac.id</p> <p>Internet Source</p> | <div style="font-size: 2em; font-weight: bold;">3%</div> |
| <hr/> | | |
| <div style="background-color: purple; color: white; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 0 auto;">3</div> | <p>pt.scribd.com</p> <p>Internet Source</p> | <div style="font-size: 2em; font-weight: bold;">2%</div> |

Exclude quotes

On

Exclude matches

< 2%

Exclude bibliography

Off