Pemrosesan SRF05, CMPS03, TPA81, Sistem Motor Secara MultiProsesor pada Robot KRPAI

Hendi Wicaksono (1)
Industrial Robotic Design
Teknik Elektro Universitas Surabaya
Surabaya, Indonesia
hendi@ubaya.ac.id

Ari Bengnarly Tanjung (2)
Industrial Robotic Design
Teknik Elektro Universitas Surabaya
Surabaya, Indonesia

***Abstrak*—Permasalahan pada respon robot KRPAI (Kontes Robot Pemadam Api) di tahun 2012 melatarbelakangi penelitian ini. Masalah keterlambatan respon robot tersebut diidentifikasi karena faktor pengaktifan dan pengambilan data *sensor* yang dilakukan saat diperlukan saja. Terutama pada sistem *sensor* jarak yang menggunakan SRF05, sistem *sensor* navigasi yang menggunakan *sensor* kompas digital CMPS03, dan *sensor* pendeteksi api TPA81. Hal ini dilakukan agar tidak membebani kerja mikrokontroler saat mengerjakan proses utama. Yang menjadi masalah adalah bagaimana semua *sensor* bisa selalu bekerja tanpa membebani mikrokontroler. Pada makalah ini membahas penerapan multiprosesor (banyak prosesor) sehingga semua *sensor* bisa selalu bekerja tanpa membebani mikrokontroler utama yang mengerjakan proses utama. Hasil yang diperoleh adalah penggunaan multiprosesor membantu meningkatkan respon jelajah robot yang dibuat tim *Industrial Robotic Design* Ubaya pada lapangan KRPAI 2013.**

Kata Kunci—KRPAI, SRF05, CMPS03, TPA81, MultiProsesor

I. PENDAHULUAN

Pada tahun 2012, Tim *Industrial Robotic Design* Ubaya membuat robot KRPAI (Kontes Robot Pemadam Api) yang dilengkapi dengan lebih dari 1 jenis *sensor*, bahkan pada *sensor* jarak digunakan 8 unit *sensor*. Pemrosesan *sensor* mulai dari pengaktifan *sensor*, pengambilan data dilakukan bila diperlukan saja. Semisal ketika ingin mendeteksi ada tidaknya dinding, maka saat itu baru dilakukan pengaktifan *sensor*, dan pengambilan data. Hal ini menyebabkan adanya waktu tunda pada respon kedua motor robot KRPAI 2012. Apabila dilakukan rutin pengaktifan semua *sensor* secara bergantian menyebabkan beban kerja mikrokontroler sangat tinggi. Belum lagi jika dilakukan hal tersebut, dapat mengakibatkan gerak robot yang terpatah-patah karena pengendalian gerak kedua motor tidak dapat terus-menerus. Terkait navigasi, masalah respon robot ini teridentifikasi 100% menjadi satu-satunya masalah pada KRPAI 2012 dikarenakan robot KRPAI saat itu sudah menggunakan motor Vexta lengkap dengan *driver motor*, dan *sensor* jarak SRF05 yang teruji stabil penggunaannya.

Penggunaan SRF05 sebagai *sensor* jarak sudah lazim digunakan seperti pada *obstacle avoidance vehicle robot* yang dikembangkan oleh Dheeman Paul [1]. Pada robot tersebut mengembangkan penggunaan 1 *sensor* jarak untuk dapat menghindari halangan dengan teknik *Potential Field Algorithm*. Dengan hanya 1 *sensor* saja, tidak memungkinkan robot dapat bergerak cepat sambil bernavigasi. Oleh karena itu, robot KRPAI menggunakan 8 buah *sensor* SRF05.

Muncul pemikiran untuk membuat semua *sensor* selalu bekerja mulai dari pengaktifan *sensor* dan mendapatkan data *sensor*. Begitu pula dengan aktuator berupa 2 unit motor yang selalu bekerja hingga terjadi perubahan kecepatan yang diinformasikan padanya. Namun, dengan catatan semua proses tersebut tidak boleh mengganggu atau membebani kerja mikrokontroler saat mengerjakan proses rutin. Manimaran di tahun 1998 juga menyatakan bahwa proses paralel memungkinkan dilakukan pada sistem *realtime* multiprosesor [2]. Pada sistem multiprosesor yang Manimaran kembangkan mengutamakan *scheduled time* atau waktu yang sudah direncanakan untuk pengambilan data dan pemberian data. Sistem tersebut tidak sesuai 100% dengan kebutuhan robot KRPAI 2013 ini, sehingga robot ini menggunakan sistem *request*. Sistem *request* dijalankan oleh *master* kontrol yang memerintahkan *slave* mengirimkan datanya.

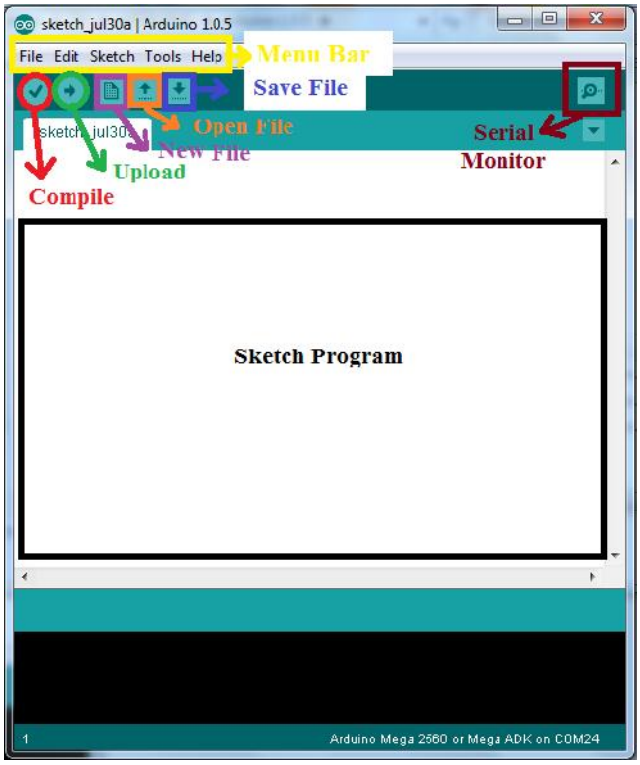
Konsep desentralisasi seperti yang dikemukakan di atas didapat dari model sistem kontrol *mobile* robot Bebot [3]. Penggunaan konsep tersebut pada Bebot *mobile* robot adalah setiap robot memiliki 1 kontroler dari multi-robot tersebut. Sedangkan pada robot KRPAI ini konsep tersebut diimplementasikan dengan membuat setiap proses kerja memiliki 1 kontroler dalam 1 robot.

Pada makalah ini membahas 5 bagian penting, dimulai dari bagian pertama tentang teori terkait multiprosesor, bagian kedua metode penelitian yang digunakan, bagian ketiga desain robot KRPAI khususnya pada multiprosesor, bagian keempat hasil dan analisis, bagian terakhir simpulan dan penelitian mendatang.

II. DASAR TEORI

Arduino merupakan *platform* berbasis *open source* pemrograman IC ATMEL AVR [4]. Dengan sistem *library* Arduino ini pemrograman AVR sangat mudah terutama pada proses pengaturan sistem *timer* dan masih banyak lainnya. Tampilan *board* Arduino dapat dilihat pada Gambar 1 dan tampilan *software* antar muka Arduino dapat dilihat pada Gambar 2.

Gambar 1. *Board* Arduino Uno [4]



Gambar 2. Antarmuka *Software* Arduino [4]

I^2C (*Inter Integrated Circuit*) merupakan komunikasi antar mikrokontroler yang menggunakan 2 jalur, yaitu *pin* SDA (*Serial Data*) dan SCL (*Serial Clock*) [5]. SDA digunakan sebagai jalur komunikasi data dua arah. SCL merupakan *clock* guna mengatur sinkronisasi pengiriman dan penerimaan data. Terdapat minimal 2 buah mikrokontroler untuk mengimplementasi komunikasi I^2C. Satu mikrokontroler sebagai mikrokontroler *master*, dan sisanya sebagai mikrokontroler *slave*. Hanya ada 1 buah mikrokontroler sebagai mikrokontroler *master*, namun bisa terdapat lebih dari 1 mikrokontroler *slave*. *Library* yang digunakan untuk mengaktifkan maupun menjalankan komunikasi I^2C ini pada arduino adalah `wire.h`. Seperti yang dikemukakan di atas bahwa dengan *platform* arduino ini, penggunaan fitur-fitur AVR dipermudah dengan adanya dukungan *library* yang beraneka ragam.

Laboratorium *Industrial Robotic Design* Universitas Surabaya

Pada *library* `wire.h` terdapat fungsi-fungsi seperti `begin()`, `beginTransmission()`, `endTransmission()`, `write()`, `read()` dan beberapa lainnya yang kesemua itu dapat langsung digunakan [4]. Jalur SDA pada Arduino terletak pada *pin* A4, dan jalur SCL pada Arduino terletak pada *pin* A5.

SPI (*Serial Peripheral Interface*) merupakan komunikasi *serial* yang menggunakan 4 buah jalur data [4]. Empat jalur data tersebut adalah MOSI (*Master Output Serial Input*), MISO (*Master Input Serial Output*), SCLK (*Serial Clock*), dan SS (*Slave Select*). Sama seperti komunikasi I^2C, *library* SPI juga sudah ada pada *library* Arduino. *Library* untuk komunikasi SPI adalah `SPI.h`. Pada *library* ini terdapat fungsi-fungsi seperti `begin()`, `setDataMode()`, `transfer()` dan beberapa lainnya [4].

III. METODE PENELITIAN

Metode penelitian desain sistem multiprosesor guna memperbaiki respon robot KRPAI 2013 sebagai berikut.

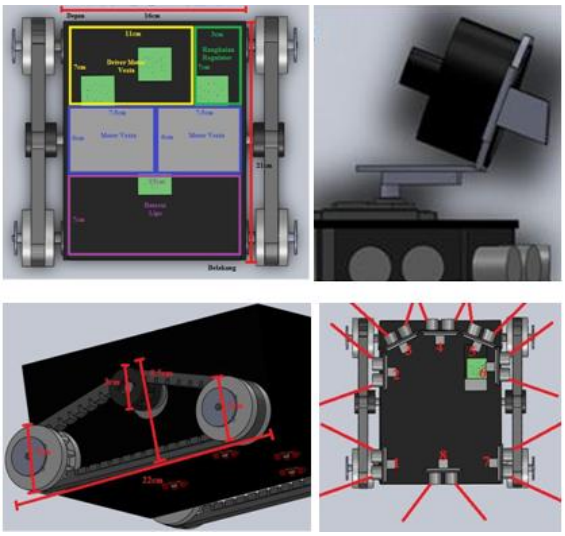
1. Desain mekanik dan realisasi robot KRPAI 2013.
2. Pendataan bagan-bagan *sensor* dan aktuator yang dibutuhkan, kemudian 1 prosesor mewakili 1 bagan.
3. Perancangan interkoneksi perangkat *sensor* dan aktuator tersebut.
4. Pembuatan program untuk masing-masing kerja bagan *sensor* atau aktuator.
5. Penggabungan setiap multiprosesor dengan komunikasi I^2C dan SPI.
6. Pengujian sistem navigasi dan pergerakan robot KRPAI 2013 pada lapangan yang digunakan terutama pada respon penyusuran setiap ruangan.

IV. DESAIN ROBOT

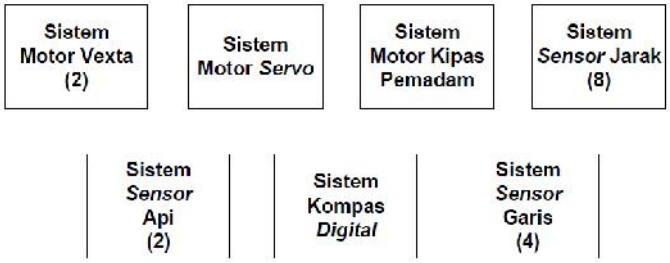
Desain robot Tim *Industrial Robotic Design* untuk KRPAI 2013 menggunakan *software* 3D populer dapat dilihat pada Gambar 3. Dari desain tersebut direalisasikan dengan bahan utama berupa akrilik. Motor yang digunakan adalah 2 unit motor Vexta, 8 unit SRF05 sebagai *sensor* jarak, TPA81 sebagai *sensor* pendeteksi panas, dan CMPS03 sebagai *sensor* deteksi arah untuk navigasi.

Teridentifikasi dari tahun-tahun sebelumnya tentang *sensor* dan aktuator yang dibutuhkan, maka semua bagan perangkat elektronika yang ada pada robot ini dapat dilihat pada Gambar 4. Terdapat 7 bagan perangkat elektronika, yang idealnya terdistribusi pada 7 prosesor berupa mikrokontroler Arduino. Namun, dikarenakan CMPS03 sebagai *sensor* kompas digital dan TPA81 sebagai *sensor* pendeteksi api sudah menggunakan sistem komunikasi I^2C, maka kedua *sensor* tersebut langsung terhubung pada mikrokontroler utama. Sedangkan 2 macam aktuator, yakni sistem motor *servo* dan sistem motor kipas pemadam, juga langsung terhubung pada prosesor utama.

ISBN: 978-602-7776-72-2



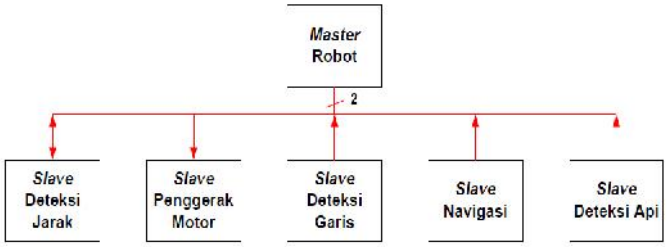
Gambar 3. Desain 3D Robot KRPAI 2013



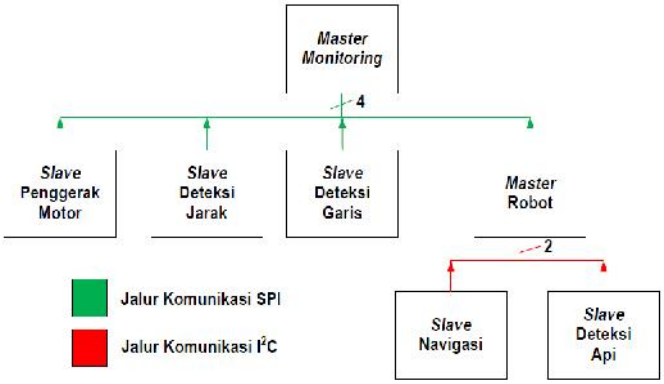
Gambar 4. Blok Perangkat Elektronika

Pada sistem komunikasi I^2C, semua bagan elektronika di atas menjadi prosesor *slave* dengan 1 unit *master*. Jalur komunikasi prosesor *master* dan *slave* dapat dilihat pada Gambar 5. Sedangkan sistem komunikasi SPI digunakan sebagai sistem *monitoring*. Dengan kata lain, masing-masing *slave* mengirimkan data ke *master* I^2C dan ke *master* SPI. Hubungan jalur data sistem *monitoring* dapat dilihat pada Gambar 6. Dengan begitu semua data *sensor* jarak, data *sensor* deteksi api, data *sensor* navigasi dapat terpantau dengan mudah.

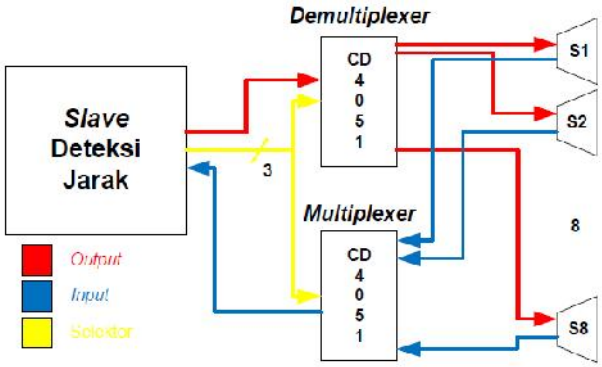
Ada desain khusus pada bagan sistem *sensor* jarak yang terdiri dari 8 unit SRF05. Setiap SRF05 ini membutuhkan 2 *pin*, 1 *pin* untuk *trigger*, *pin* sisanya untuk *echo*. Menurut perhitungan dari jumlah *pin* tersebut, maka terdapat 16 *pin* mikrokontroler yang digunakan untuk dapat mengakses 8 unit SRF05 itu. Hal bisa dioptimalkan dengan penggunaan CD4051, sebuah multiplekser demultiplekser. Bagan penggunaan CD4051 dapat dilihat pada Gambar 7.



Gambar 5. Jalur Komunikasi I^2C Mikrokontroler *Master* dan *Slave*



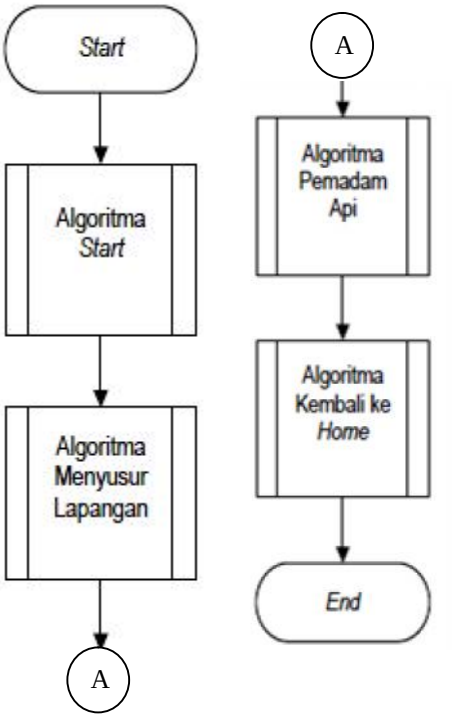
Gambar 6. Jalur Komunikasi SPI untuk *Monitoring*



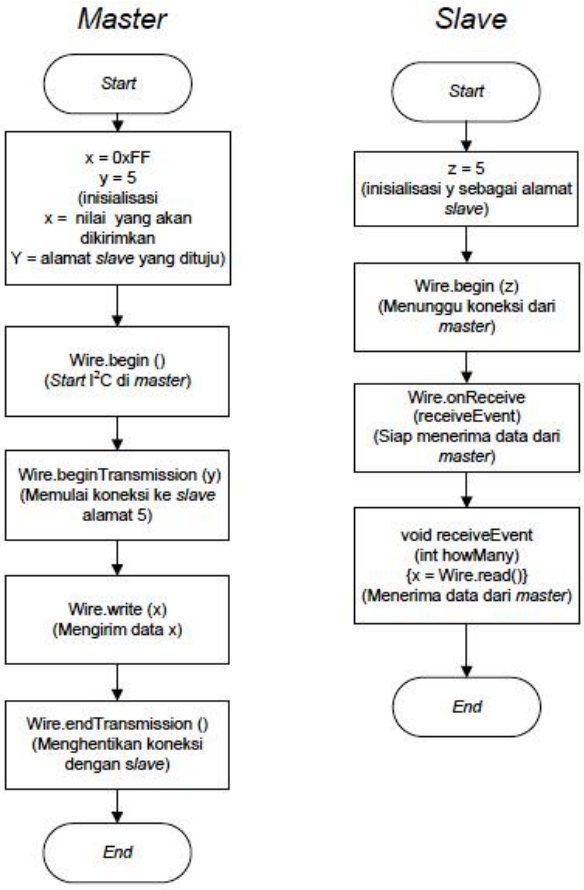
Gambar 7. Akses 8 Unit SRF05 Menggunakan CD4051

Setelah mendesain keseluruhan perangkat elektronik yang terfokus pada pengembangan model sistem multiprosesor, selanjutnya adalah pengembangan perangkat lunak untuk menjalankan robot tersebut. *Flowchart* perangkat lunak secara keseluruhan dapat dilihat pada Gambar 8.

Terdapat 4 buah algoritma yang dikembangkan pada robot ini. Mulai dari algoritma *start*, algoritma menyusur lapangan, algoritma pemadam api, dan algoritma kembali ke *home*. Sedangkan untuk algoritma komunikasi I^2C sendiri dapat dilihat pada Gambar 9. Dapat dilihat pada algoritma tersebut, pemilihan *master* ingin berkomunikasi dengan *slave* yang mana berdasarkan *address* yang dikirim dulu. Lain halnya dengan SPI, pemilihan *slave* mana yang ingin diajak berkomunikasi dengan cara mengaktifkan *pin* SS pada masing-masing *slave*.



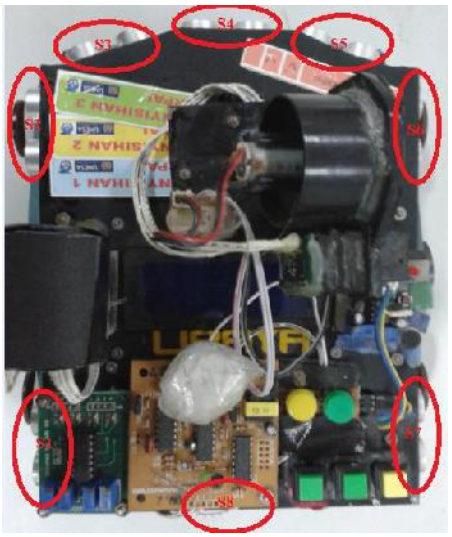
Gambar 8. *Flowchart* Perangkat Lunak secara Keseluruhan



Gambar 9. *Flowchart* Komunikasi I^{2}C *Master* dan *Slave*

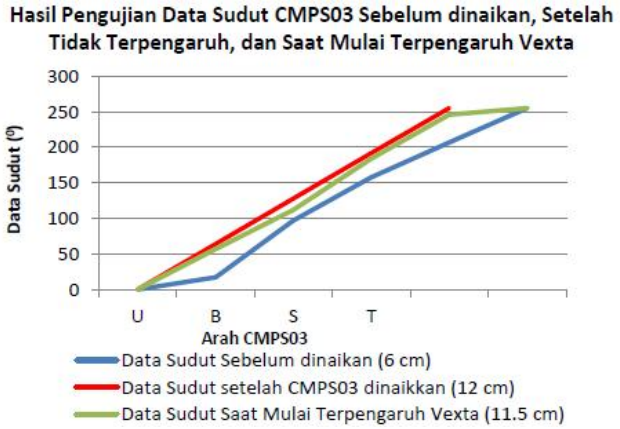
V. HASIL DAN DISKUSI

Setelah melalui serangkaian desain diperoleh bentuk mekanik dari robot KRPAI 2013 ini seperti pada Gambar 10.



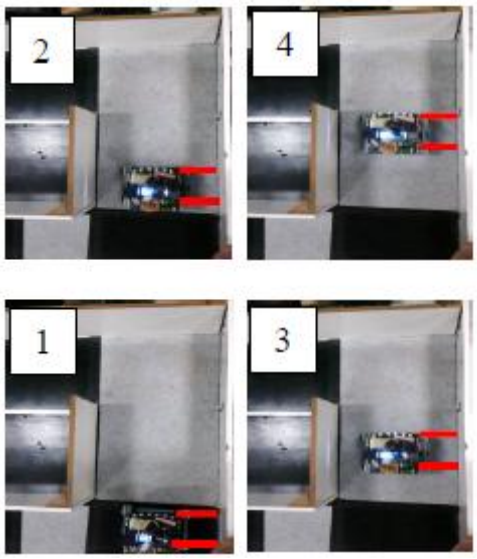
Gambar 10. Foto Robot KRPAI 2013

Yang menarik untuk didiskusikan adalah pembacaan *sensor* CMPS03 sebagai *sensor* navigasi. Adanya gangguan dari medan magnet motor yang membuat pembacaan CMPS03 ini sedikit tidak sama dibandingkan saat tidak terpasang pada robot. Diujikan pada jarak 6 cm, 11.5 cm, dan 12 cm dengan motor Vexta, didapatkan bahwa pada jarak 12 cm data CMPS03 benar-benar sama dengan pembacaan saat tidak terpasang pada robot. Namun, dengan peletakkan 12 cm tersebut, mengakibatkan tinggi robot melampaui batas tinggi maksimum yang diijinkan. Sehingga diperoleh jarak 11.5 cm terhadap motor yang paling mendekati dengan data saat jarak 12 cm itu. Grafik perbandingan data CMPS03 pada tiga buah jarak tersebut dapat dilihat Gambar 11.



Gambar 11. Perbandingan Data CMPS03 pada Tiga Buah Jarak antara CMPS03 dengan Motor Vexta

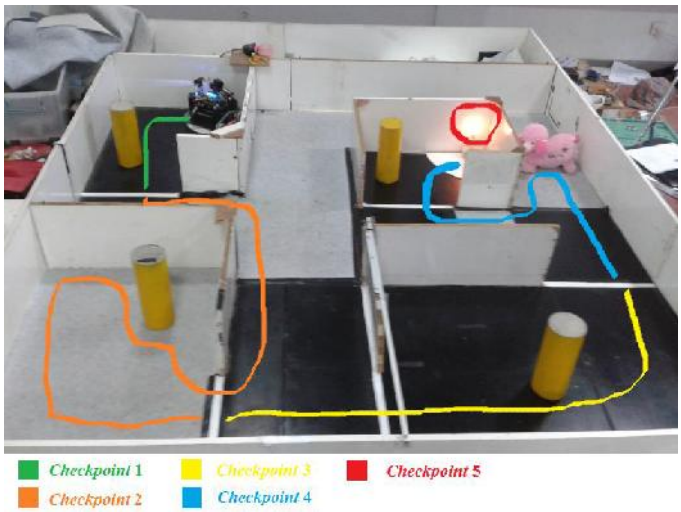
ISBN: 978-602-7776-72-2



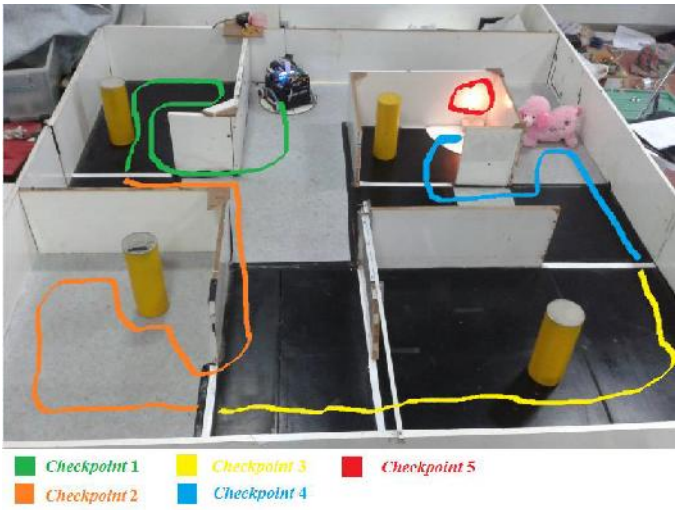
Gambar 12. Rekaman Gerak Robot Menyusuri Dinding Tiap *Frame*

Proses *sensing* dari kedelapan unit SRF05 mempengaruhi respon gerakan robot. Paling mudah dengan melihat tingkat osilasi dari gerakan robot menyusuri lapangan. Hal ini hanya bisa diukur melalui rekaman video dan dilihat posisi robot tiap *frame* per satuan waktu. Apabila robot tidak bergoyang kiri kanan, dengan kata lain osilasi sangat minim, ini artinya respon *sensing* cukup cepat sehingga mendukung perancangan algoritmanya. *Frame* dari gerakan robot tersebut dapat dilihat pada Gambar 12. Berikutnya pengujian bagaimana robot dapat bernavigasi di seluruh ruangan lapangan KRPAI untuk melihat respon dari robot setelah melakukan *sensing* terhadap dinding maupun halangan lainnya. Pengujian robot menyusuri ke semua ruangan dapat dilihat pada Gambar 13 dan Gambar 14.

Adapun pencatatan waktu yang diperoleh untuk setiap *checkpoint* dapat dilihat pada Tabel 1, Tabel 2.



Gambar 13. Pengujian Robot Menyusuri Semua Ruangan dengan Posisi *Start* dalam Ruangan



Gambar 14. Pengujian Robot Menyusuri Semua Ruangan dengan Posisi *Start* pada Lorong

Tabel 1. Waktu pada *CheckPoint* saat Robot *Start* pada Ruangan

Waktu pada *CheckPoint* (detik)					
CP1	CP2	CP3	CP4	CP5	Total
5.0	21.5	16.4	12.7	11.2	66

Tabel 2. Waktu pada *CheckPoint* saat Robot *Start* pada Lorong

Waktu pada *CheckPoint* (detik)					
CP1	CP2	CP3	CP4	CP5	Total
17.4	21.9	16.4	13.0	12.2	81

Dapat dilihat pada Tabel 1, Tabel 2, robot mampu menjelajah seluruh ruangan untuk *start* pada ruangan dengan waktu selama 66 detik, sedangkan saat *start* pada lorong membutuhkan waktu selama 81 detik. Waktu jelajah ini lebih cepat dibandingkan dengan robot tahun lalu yang membutuhkan waktu selama hampir 2 menit atau 120 detik.

Selain itu, pengujian proses pemadaman api juga penting. Mampunya robot bernavigasi menyusuri semua ruangan harus diakhiri dengan kemampuan memadamkan api. Berikut adalah pengujian berbagai variasi posisi robot menemukan titik api pada Gambar 15.

Robot ini saat diturunkan pada KRPAI 2013 waktu lalu belum bisa juara dikarenakan terdapat masalah pada *sensor* api UVTron (*sensor* api selain TPA81) yang sebelumnya tidak diidentifikasi sebagai masalah pada pengembangan robot ini.

Gambar 15. Variasi Penemuan Posisi Titik Api

VI. SIMPULAN

Pengembangan multiprosesor pada robot KRPAI 2013 mampu memperbaiki respon pergerakan robot. Dengan menggunakan metode multiprosesor, lima mikrokontroler *slave* dan satu mikrokontroler *master* dapat membuat semua *sensor* selalu bekerja sehingga respon robot cepat saat *sensing* dinding lapangan. Waktu yang dibutuhkan untuk menjelajah seluruh ruangan adalah 40 hingga 60 detik lebih cepat dibandingkan waktu tempuh robot tahun lalu. Penggunaan *platform* Arduino sangat memudahkan dalam implementasi komunikasi multiprosesor menggunakan I^2C dan SPI dikarenakan di dalamnya sudah terdapat *library* untuk 2 jenis komunikasi ini.

Hasil pembacaan CMPS03 6 cm di atas Vexta tidak linier sempurna karena masih terpengaruh motor Vexta. Hasil pembacaan CMPS03 linier mulai jarak 11.5 cm. Namun meskipun tidak linier sempurna hasil pembacaan CMPS03 saat tinggi 6 cm masih valid digunakan untuk mendeteksi ruangan acak karena data yang digunakan berupa *range* yang cukup luas.

Dari pengamatan lapangan dengan masih gagalnya robot yang dibangun oleh tim *Industrial Robotic Design* Ubaya ini memadamkan api, maka penelitian mendatang mengubah konstruksi robot lebih kecil sehingga manuver lebih lincah karena dukungan responsivitas *sensor-sensor* sudah bagus.

REFERENSI

[1] Paul, D., Muhit, M. S., & Khatun, M. (2013), "Implementation & evolution of obstacle avoidance vehicle robot," American Academic & Scholarly Research Journal, Vol. 5, Pages 72-79.

[2] Manimaran, G. and Murthy, C.SivaRam and Ramamritham, Krithi, "A New Approach for Scheduling of Parallelizable Tasks in Real-Time Multiprocessor Systems", Real-Time Systems Journal, Vol. 15, Pages 39-60, ISSN 0922-6443.

[3] Chia Choon Loh, Ansgar Traechtler, "Cooperative Transportation of Aload Using Nonholonomic Mobile Robots," Procedia Engineering, Volume 41, 2012, Pages 860-866, ISSN 1877-7058.

[4] A. Bengnarly, "Pengaplikasian TPA81 dan CMPS03 Pada Rancang Bangun Robot Beroda KRPAI 2013", Tugas Akhir, Teknik Elektro Universitas Surabaya, Agustus 2013.

[5] R. Silay, "Kontrol Gerak Robot Empat Kaki dengan Metode Trajektori Dua Derajat Kebebasan," Tugas Akhir, Teknik Elektro Universitas Surabaya, 2010.

ISBN: 978-602-7776-72-2