PENGAPLIKAISAN TPA81 DAN CMPS03 PADA RANCANG BANGUN ROBOT BERODA KRPAI 2013

Ari Bengnarly [(1)], Hendi Wicaksono [(2)]
Teknik Elektro Universitas Surabaya [(1), (2)]
Bengnarly@gmail.com [(1)], Hendi@ubaya.ac.id [(2)]

Abstrak

Pada KRCI 2011 dan 2012 robot memiliki masalah dalam hal tidak dapat memadamkan api. Kegagalan disebabkan karena *sensor* UV-TRON dan *photodioda* bermasalah saat mendeteksi api. Robot juga memiliki masalah dalam sistem navigasi karena robot tidak dapat menentukan posisi ruangan yang telah diacak. Banyaknya sistem yang bekerja pada robot menyebabkan sistem-sistem tersebut tidak dapat dijalankan secara *serial*. Pada Tugas Akhir ini dibuat rancang bangun robot beroda yang mengaplikasikan *sensor* api TPA81 dan kompas *digital* CMPS03 yang seluruh sistem pada robot bekerja secara *paralel*. Pembuatan robot beroda dibagi menjadi tiga tahapan besar, yaitu perancangan mekanik, perancangan *hardware*, dan perancangan *software*. Perancangan mekanik terdiri atas perancangan bentuk robot bagian bawah, konstruksi roda, peletakkan *sensor* garis, *sensor* ultrasonik dan pemadam api. Sedangkan pada perancangan *hardware* terdapat perancangan sistem multiprosesor yang menggunakan komunikasi I^2C dan SPI, sistem penggerak motor, sistem deteksi garis, sistem navigasi, dan sistem deteksi api yang digunakan pada robot, serta sistem *display* LCD. Untuk perancangan *software* pada Tugas Akhir ini terdiri atas algoritma *start*, algoritma menyusur lapangan, algoritma memadamkan api biasa serta cadangan, algoritma kembali ke posisi *home* dan algoritma untuk melakukan komunikasi I^2C serta SPI yang menggunakan Arduino. Dari hasil pengujian keseluruhan pada Tugas Akhir didapatkan bahwa waktu robot untuk menyusuri lapangan hingga dapat memadamkan api tercepat sebesar 01:04.58 dan waktu terlama dalam menyusuri dan memadamkan api sebesar 01:37.54. Konsistensi keberhasilan robot hingga dapat memadamkan api sebesar 50%.

Kata Kunci: robot pemadam api, Arduino, multiprosesor, I^2C, TPA81, CMPS03

Abstract

On the KRCI 2011 and 2012 robot have problems could not extinguish the fire. Failure is caused the sensor UV-TRON and photodiode troubled when to detect flames. Robot also have problems in navigation system because robots can not determine position room had been scrambled. Many system worked on robot cause the systems cannot be executed in serial. In this final project was build a wheeled robot applied flame sensor TPA81 and digital compass CMPS03 which system on robot worked parallel. Manufacture of wheeled robot is divided into

three major phases, namely the mechanical design, hardware design, and software design. Mechanical design consists of design robot form the bottom, wheel construction, laying a line sensor, ultrasonic sensor and fire extinguisher. While the hardware design are multiprocessor system design uses I^2C and SPI communication, motor drive system, line detection system, navigation system, and fire detection system are used on the robot, as well as the LCD display system. For design of the software on this final project consists of algorithm start, algorithm run along the field, fire extinguisher algorithm regular and reserves, algorithm back to the home position and algorithms to perform I2C and SPI communication using the Arduino. Overall result of the final project test it was found that while the robot to run along the field to be able to extinguish the fire quickest at 1:04:58 and longest time in the down and extinguish the fire at 1:37:54. Consistency success robot to be able to extinguish the fire by 50%.

Keywords: fire extinguisher robot, Arduino, multiprocessor, I^2C, TPA81, CMPS03

1. PENDAHULUAN

Pada perlombaan KRCI divisi beroda (Kontes Robot Cerdas Indonesia) atau yang tahun ini berganti nama menjadi KRPAI (Kontes Robot Pemadam Api Indonesia) pada tahun 2011 dan tahun 2012. Robot beroda Teknik Elektro Ubaya (Universitas Surabaya) selalu gagal dalam usaha untuk memadamkan api. Hal tersebut disebabkan karena *sensor* api UV-TRON dan *photodioda* mengalami masalah pendeteksian api saat perlombaan yang disebabkan suhu udara serta lampu sorot. Gangguan tersebut menyebabkan UV-TRON dan *photodioda* tidak berfungsi secara maksimal untuk mendeteksi api. Selain *sensor* api, robot juga bermasalah dengan sistem navigasi karena hanya menggunakan *sensor* ultrasonik. Penggunaan sistem navigasi yang berupa *sensor* ultrasonik saja menyebabkan robot susah untuk menentukan posisi ruangan yang telah diacak. Selain itu, karena terdapat banyak sistem bekerja yang pada robot, maka tidak mungkin seluruh sistem pada robot dijalankan secara *serial* (berurutan). Seperti permasalahan yang telah dijelaskan sebelumnya maka untuk mengatasi masalah tersebut pada Tugas Akhir ini dibuat rancang bangun robot beroda yang mengaplikasikan *sensor* api TPA81untuk membantu mengatasi permasalahan pendeteksian api dan kompas *digital* CMPS03 yang digunakan sebagai sistem navigasi untuk menentukan ruangan acak. Agar seluruh sistem pada robot menggunakan sistem yang dapat bekerja secara *paralel* (bersamaan) maka pada robot dibuat sistem multiprosesor yang menggunakan komunikasi I^2C.

2. DASAR TEORI

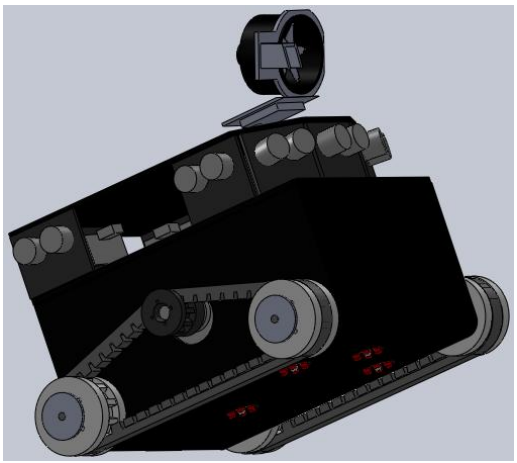
Arduino merupakan *platform prototyping* elektronik berbasis *open source* yang mudah digunakan (*flexibel*) yang terdiri atas perangkat keras (*hardware*) dan perangkat lunak (*software*) [1]. Pada Tugas Akhir ini *hardware* (*board Arduino*) tidak menggunakan *board* jadi namun membuat *board* sendiri dengan konfigurasi yang sama dengan *board* asli. *Board* yang dibuat sendiri merupakan *board* Arduino NG dengan mikrokontroler ATMega8 yang sudah banyak dibahas dan diuraikan pada dasar teori Tugas Akhir [2], [3], dan [4]. Serta *board* Arduino Uno yang menggunakan mikrokontroler ATMega328 [5].

Penggunaan kedua *board* tersebut disebabkan karena pada Tugas Akhir ini robot menggunakan sistem multiprosesor yang menggunakan komunikasi I^2C dan SPI. *Board* Arduino NG digunakan sebagai *slave* dan *board* Arduino Uno digunakan sebagai *master*. Pada program Arduino untuk melakukan komunikasi I^2C antar mikrokontroler hanya perlu menggunakan *library* yang telah disediakan Arduino yaitu `wire.h` [6]. Dalam komunikasi I^2C pada Arduino terdapat dua macam pengiriman data yaitu pengiriman data dari *master* menuju *slave* dan pengiriman data dari *slave* menuju *master* [7]. Untuk menggunakan komunikasi I^2C *pin* SDA dan SCL harus saling dihubungkan pada masing-masing Arduino. Sedangkan untuk komunikasi SPI (*Serial Peripheral Interface*) pada Arduino menggunakan *library* `SPI.h` [8], dan menggunakan mode *independent slave* [9]. Komunikasi SPI menggunakan *pin* MOSI, MISO, SCLK, dan SS.

Komponen-komponen utama yang digunakan untuk membangun robot beroda pada Tugas Akhir ini terdiri atas aktuator dan *sensor*. Aktuator yang digunakan yaitu motor Vexta dan motor *servo* yang dapat menghasilkan besaran sudut putar [10], untuk penggerak kipas pemadam. Untuk *sensor* yang digunakan terdiri atas SRF05 yang merupakan *sensor* ultrasonik dengan jarak maksimum deteksi 4 m [11], *Sensor* kompas digital CMPS03 yang memiliki ketelitian hingga 0.1^0 [12], UV-TRON untuk mendeteksi api hingga jarak 5 m [13], dan *sensor* api TPA81 yang dapat mendeteksi infra merah dengan panjang gelombang 2 um hingga 22 um [14].

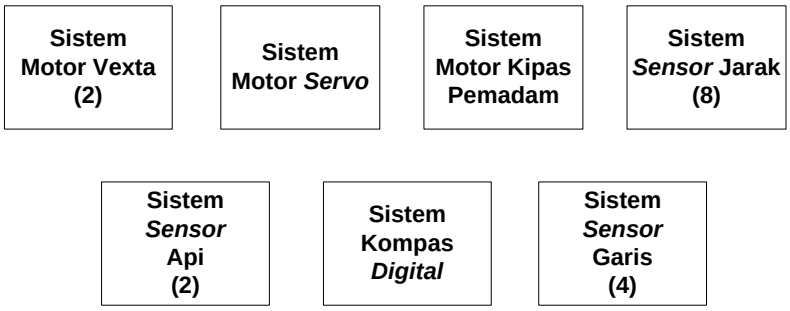
3. METODE PENELITIAN

Perancangan pada Tugas Akhir ini terdiri atas perancangan mekanik, perancangan *hardware*, dan perancangan *software*. Perancangan mekanik pada robot beroda dilakukan dengan merancang bagian-bagian robot menggunakan *software* desain 3D. Bagian-bagian robot yang dirancang antara lain konstruksi robot bawah, konstruksi roda, peletakan *sensor* garis, peletakan *sensor* ultrasonik, dan peletakan pemadam api. Gambar 1 menunjukan hasil perancangan robot menggunakan *software* 3D.



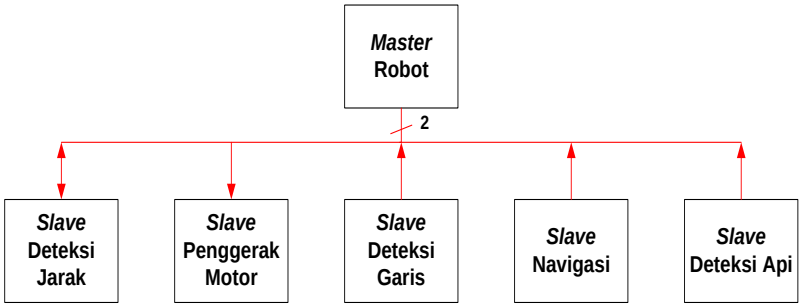
Gambar 1. Hasil Perancangan Robot Beroda Menggunakan *Software* 3D

Perancangan *hardware* mengacu pada banyaknya I/O yang terdapat pada robot. I/O utama pada robot ini berupa motor Vexta, motor *servo*, motor kipas pemadam, *sensor* jarak, *sensor* api, kompas *digital*, dan *sensor* garis. I/O yang terdapat pada robot beroda dapat dilihat pada Gambar 2. Beberapa I/O pada robot beroda yang harus bekerja bersamaan (*paralel* proses) seperti sistem motor Vexta dan sistem *sensor* jarak menyebabkan kinerja robot menjadi berat jika hanya menggunakan 1 mikrokontroler. Oleh karena itu pada robot beroda ini dirancang sistem multiprosesor (banyak mikrokontroler).



Gambar 2. I/O yang Terdapat pada Robot Beroda

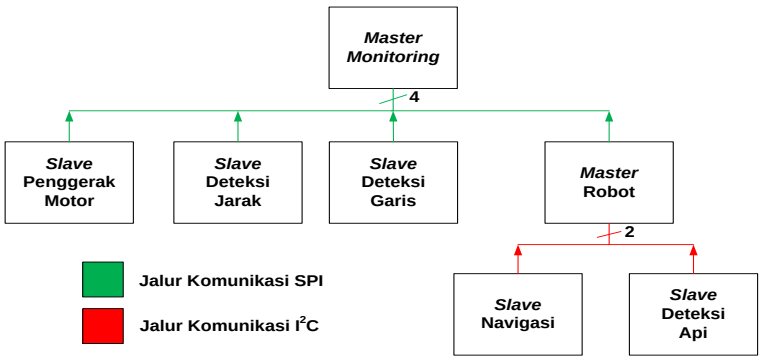
Sistem multiprosesor pada robot beroda ini dihubungkan menggunakan komunikasi *serial* I^2C (*Inter Integrated Circuit*) yang menggunakan sistem *master* dan *slave*. Jadi pada satu atau beberapa I/O tersebut, ditangani oleh satu mikrokontroler yang bertindak sebagai *slave.* Sehingga nantinya terdapat banyak *slave* yang ada pada robot. *Slave-slave* tersebut terdiri atas *slave* deteksi jarak, *slave* penggerak motor, *slave* deteksi garis, *slave* navigasi, dan *slave* deteksi api. *Slave-slave* tersebut nantinya dikontrol menggunakan sebuah *master* yang terhubung secara *serial* dengan 2 jalur data yaitu SDA (*Serial Data*) dan SCL (*Serial Clock)* dan pada masing-masing jalur tersebut diberi *pull up* 5 V dengan resistor 1K8Ω. Blok jalur komunikasi I^2C antara *master* dan *slave* dapat dilihat pada Gambar 3.



Gambar 3. Blok Jalur Komunikasi I^2C antara *Master* dengan *Slave*

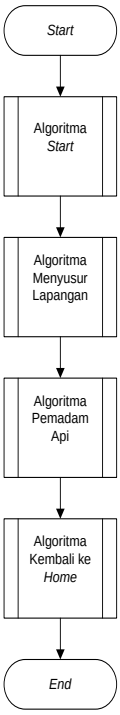
Display berupa LCD diperlukan agar dapat melihat sampai dimana program tersebut berjalan dan data I/O yang dihasilkan oleh *sensor*. Karena *display* LCD mengambil data dari *slave* maka *display* LCD harus dijadikan *master* pada komunikasi I^2C, namun karena tidak boleh terdapat dua *master* dalam komunikasi I^2C maka untuk mengambil data dari *slave-slave* komunikasi I^2C digunakan

komunikasi *serial* lainnya yaitu SPI (*Serial Peripheral Interface*). Jalur komunikasi SPI pada robot ini menghubungkan antara mikrokontroler LCD yang bertindak sebagai *master* dengan *slave-slave* pada komunikasi I^2C. Namun karena *slave* navigasi dan *slave* deteksi api berupa modul jadi yang hanya dapat melakukan komunikasi I^2C sehingga kedua *slave* tersebut tidak dapat dihubungkan langsung dengan *master monitoring*. Oleh karena itu jalur komunikasi SPI pada *master monitoring* dihubungkan ke *master* robot untuk mengambil data dari *slave* navigasi dan *slave* deteksi api. Jalur komunikasi SPI antara *master monitoring* dengan *slave* dapat dilihat pada Gambar 4.



Gambar 4. Sistem Komunikasi SPI antara *Master Monitoring* dengan *Slave*

Konfigurasi lapangan dan posisi *start* yang tidak dapat ditebak menyebabkan dibutuhkannya perancangan *software* untuk menyusun algoritma agar robot dapat menyusuri lapangan KRPAI. Oleh karena itu algoritma pada robot ini dirancang secara *general* dengan menggunakan sistem *wall follower* dengan dinding sebelah kanan sebagai acuannya (rata kanan). Jadi robot selalu berjalan mengikuti dinding sebelah kanan yang terdapat pada lapangan KRPAI. Namun sistem *wall follower* tersebut hanya digunakan sebagai dasar cara berjalan robot, sehingga agar robot dapat berjalan dari posisi *start*, kemudian memadamkan api, dan terakhir kembali ke posisi *home*, maka disusunlah algoritma *general* yang terdiri atas algoritma *start*, algoritma menyusur lapangan, algoritma pemadam api dan algoritma kembali ke posisi *home* seperti yang dapat dilihat pada *flowchart* Gambar 5.



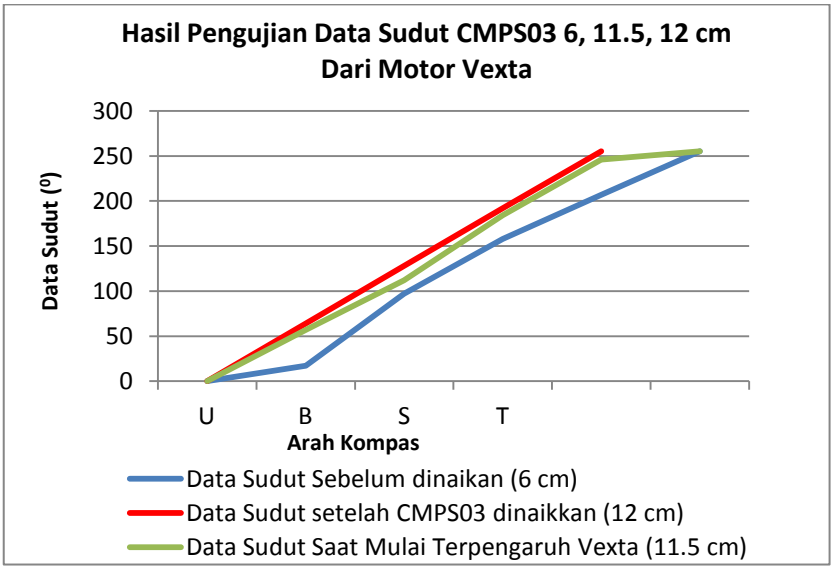
Gambar 5. *Flowchart* Algoritma Keseluruhan Robot Beroda

Pengacakan posisi arah *start*, menyebabkan posisi *sensor* kanan yang digunakan untuk menyusur lapangan terkadang tidak dapat mendeteksi dinding kanan. Oleh karena itu diperlukan algoritma tambahan (algoritma *start*) sebelum algoritma menyusur lapangan yang digunakan untuk membantu robot saat *start*. Setelah melakukan algoritma *start* robot menjalankan algoritma menyusur lapangan yang merupakan algoritma robot untuk berjalan menyusur lapangan, algoritma ini merupakan algoritma rata kanan. Algoritma rata kanan pada robot ini tidak dibuat menggunakan sistem cerdas seperti *fuzzy* atau PID, namun algoritma rata kanan dibuat melalui pengecekan *sensor* ultrasonik yang menggunakan *statement if*, *else if*, *else*. Algoritma menyusur lapangan terdiri atas algoritma osilasi, algoritma belok kiri, dan algoritma belok kanan. Saat robot menyusur lapangan dan mendeteksi adanya garis putih maka algoritma robot berpindah ke algoritma pemadam api. Pada algoritma pemadam api terdapat dua algoritma yaitu algoritma pemadam api biasa dan algoritma pemadam api cadangan. Algoritma pemadam api biasa merupakan algoritma pemadam api yang menggunakan pengecekan *sensor* UV-TRON dan TPA81. Sedangkan algoritma

pemadam api cadangan merupakan algoritma yang hanya menggunakan pengecekan *sensor* TPA81 saja.

4. HASIL DAN PEMBAHASAN

Dari pengujian kelinieran data CMPS03, CMPS03 tidak terpengaruh motor Vexta pada jarak 12 cm di atas motor. Saat CMPS03 mulai diturunkan sedikit yaitu sekitar 11.5 cm, data CMPS03 menjadi tidak linier karena mulai terpengaruh oleh motor Vexta. CMPS03 tidak mungkin diletakkan 12 cm diatas motor vexta sebab dapat melebihi dimensi tinggi maksimal pada peraturan KRPAI. Sehingga CMPS03 diletakkan 6 cm di atas motor Vexta karena *range* data yang dihasilkannya valid dan tidak mempengaruhi dimensi tinggi maksimal robot. Gambar 6 menunjukan grafik hasil pengujian kelinieran CMPS03 pada jarak 6 cm, 11.5 cm dan 12 cm.

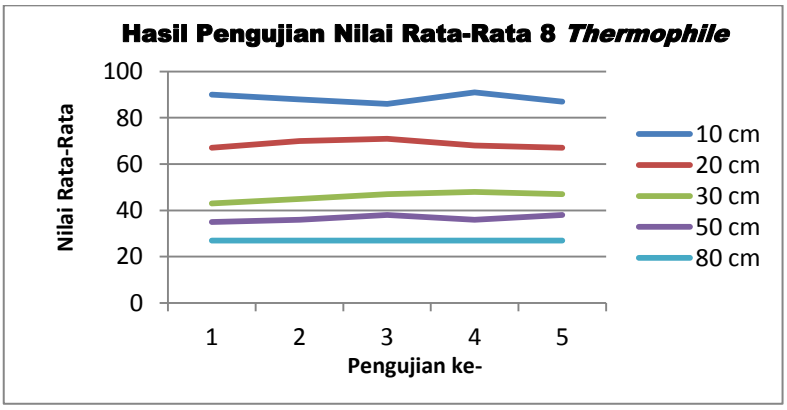


Gambar 6. Grafik Kelinieran CMPS03 pada Jarak 6, 11.5, dan 12 cm

Tabel 1. Hasil Pengujian Nilai Rata-Rata 8 *Thermophile* pada TPA81

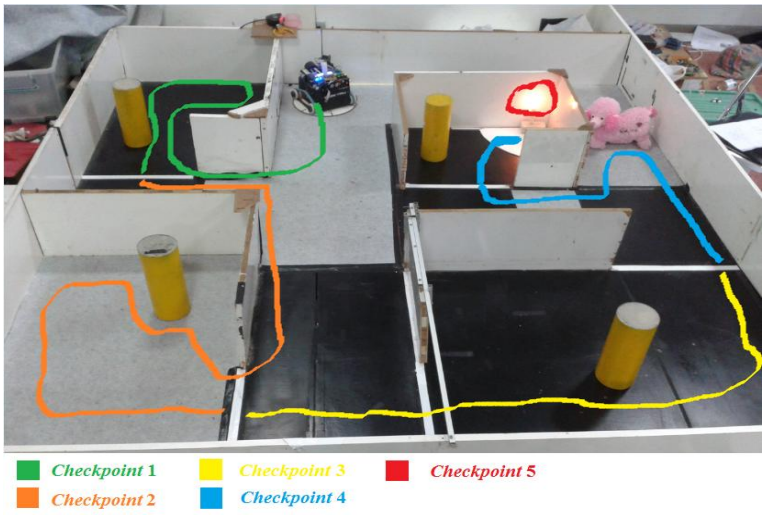
Pengujian ke-	Nilai Rata-Rata pada *Display* LCD				
	10 cm	20 cm	30 cm	50 cm	80 cm
1	90	67	43	35	27
2	88	70	45	36	27
3	86	71	47	38	27
4	91	68	48	36	27
5	87	67	47	38	27

Dari pengujian jarak jangkauan TPA81 yang terpasan di kipas pemadam pada jarak 10 cm, 20 cm, 30 cm, 50 cm, dan 80 cm dalam Tabel 1 didapatkan bahwa nilai rata-rata 8 *thermophile* pada TPA81 pada jarak 10 cm sampai 50 cm dapat digunakan sebagai acuan untuk mendeteksi adanya api karena nilai rata-rata pada jarak tersebut lebih besar dari nilai rata-rata pada saat TPA81 tidak mendeteksi adanya api. Sedangkan pada jarak 80 cm nilai rata-rata saat mendeteksi api sama dengan nilai rata-rata saat TPA81 tidak mendeteksi adanya api yang sebesar 27 (nilai normal). Grafik hasil pengujian dapat dilihat pada Gambar 7.



Gambar 7. Grafik Hasil Pengujian Nilai Rata-Rata 8 *Thermophile* pada TPA81

Dari hasil perancangan, pengujian robot dilakukan dengan menguji waktu robot untuk mematikan api dari posisi *start* yang berada di lorong hingga memadamkan api yang terdapat dalam ruangan sebanyak 5 kali pemadaman pertama menggunakan algoritma pemadam api biasa dan algoritma pemadam api cadangan. Perhitungan waktu pada pengujian dibagi menjadi 5 *checkpoint*. *Checkpoint* 1 dari posisi *start* hingga pintu keluar dari ruangan 3, *checkpoint* 2 dari pintu keluar ruangan 3 hingga pintu keluar ruangan 2, *checkpoint* 3 dari pintu keluar ruangan 2 hingga pintu keluar ruangan 1, *checkpoint* 4 dari pintu keluar ruangan 1 hingga juring pada ruangan 4, dan *checkpoint* 5 dari robot *scanning* api hingga dapat mematikan lilin yang terdapat pada ruangan 4. Gambar 8 menunjukan *checkpoint* pengujian.



Gambar 8. *Checkpoint* Pengujian

Tabel 2. Hasil Pengujian Waktu *Checkpoint* pada Algoritma Pemadam Api Biasa

Pengujian	**Waktu pada *Checkpoint* (detik)**					
	1	**2**	**3**	**4**	**5**	**Total**
1	17.56	22.10	16.34	13.57	10.93	01:20.50
2	17.44	21.86	16.21	13.04	11.32	01:19.87
3	17.01	21.78	16.45	12.43	12.81	01:20.48
4	17.84	22.20	16.63	12.98	12.96	01:22.61
5	17.11	21.54	16.27	13.11	13.01	01:21.04
Rata-rata	17.392	21.896	16.38	13.03	12.206	01:20.90

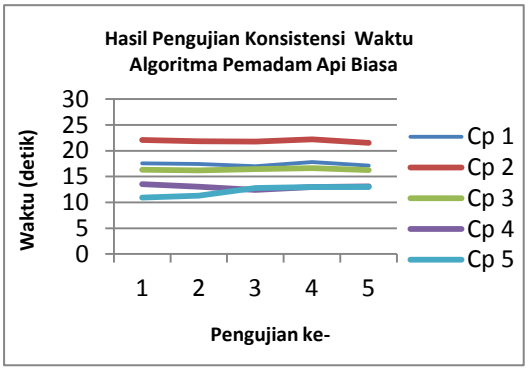
Tabel 3. Hasil Pengujian Waktu *Checkpoint* pada Algoritma Pemadam Api Cadangan

Pengujian	**Waktu pada *Checkpoint* (detik)**					
	1	**2**	**3**	**4**	**5**	**Total**
1	20.84	25.41	19.03	16.09	16.03	01:37.40
2	21.12	24.97	19.35	17.20	16.27	01:38.91
3	19.85	25.72	19.57	16.31	16.16	01:37.61
4	20.59	25.01	19.21	16.56	15.06	01:36.43
5	20.87	25.30	19.12	16.76	15.31	01:37.36
Rata-rata	20.654	25.282	19.26	16.58	15.766	01:37.54

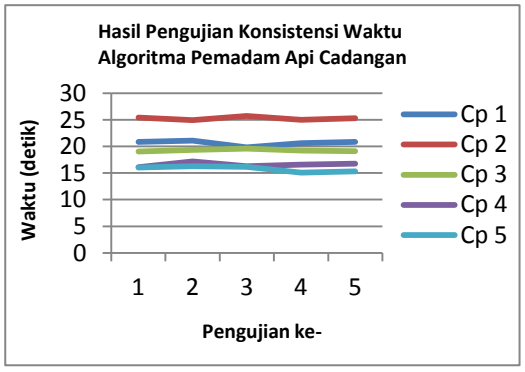
Tabel 2 dan Tabel 3 menunjukan hasil pengujian dengan menggunakan algoritma pemadam api biasa maupun algoritma pemadam api cadangan saat robot melakukan *start* di lorong. Hasil dari pengujian memadamkan api dari keempat tabel tersebut didapatkan setelah melakukan sekitar 9-10 kali pengujian, dari 9-10 kali pengujian tersebut didapatkan 5 pengujian pertama yang berhasil memadamkan api dan sisanya gagal dalam memadamkan api. Kegagalan dalam memadamkan api terdapat saat robot tidak dapat masuk ke dalam ruangan dengan urutan *checkpoint* yang telah ditentukan. Kegagalan robot biasanya disebabkan

terlalu melebarnya robot saat berbelok memasuki ruangan sehingga robot tidak dapat masuk ke ruangan yang dituju namun melewati ruangan yang dituju tersebut. Ruangan yang gagal dilewati robot meliputi ruangan 2, 3, dan 4. Terutama ruangan 4 yang sering gagal dilewati robot karena bentuk ruangan yang lebih kecil dari ruangan lainnya.

Namun kegagalan memasuki ruangan-ruangan tersebut tidak sering terjadi pada setiap pengujian, tetapi kegagalan tersebut hanya terjadi kadang-kadang saja. Jadi misal pada pengujian algoritma pemadam api biasa, robot sekali gagal memasuki ruangan 3 dan 3 kali gagal memasuki ruangan 4. Dari pengujian memadamkan api saat *start* di lorong, pada 9-10 kali pengujian didapatkan 5 keberhasilan robot untuk memadamkan api. Sehingga dapat dilihat bahwa robot memiliki persentase memadamkan api sebesar 50% pada ruangan 4.

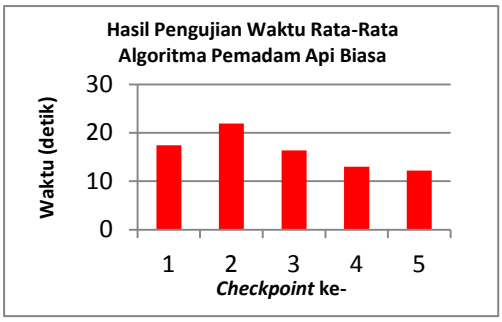


Gambar 9. Grafik Konsistensi Waktu Algoritma Pemadam Api Biasa

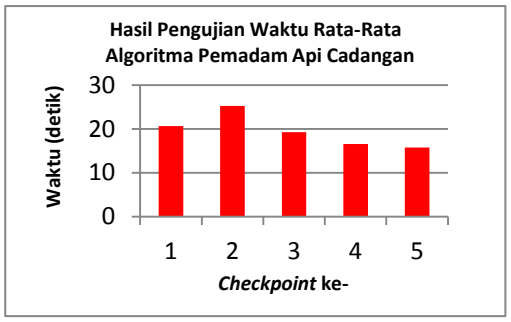


Gambar 10. Grafik Konsistensi Waktu Algoritma Pemadam Api Cadangan

Dari kedua tabel tersebut didapatkan grafik hasil konsistensi waktu pada setiap *checkpoint* yang dapat dilihat pada Gambar 9 dan Gambar 10. Dari kedua grafik tersebut dapat dilihat bahwa konsistensi waktu yang tidak stabil terdapat pada *checkpoint* ke 4 dan ke 5. Pada *checkpoint* ke 4, tidak stabilnya konsistensi waktu disebabkan karena adanya boneka anjing terdapat di lorong. Sedangkan pada *checkpoint* ke 5, tidak stabilnya konsistensi waktu disebabkan karena saat robot masuk dan berhenti di juring, posisi robot tidak menentu arahnya ke titik api sehingga meyebabkan terdapat perbedaan waktu karena robot harus menentukan terlebih dulu daerah dimana titik api berada. Selain mendapatkan grafik konsistensi waktu, dari kedua tabel tersebut juga didapatkan grafik waktu rata-rata pada setiap *checkpoint* yang dapat dilihat pada gambar dibawah ini.



Gambar 11. Hasil Pengujian Waktu Rata-Rata Algoritma Pemadam Api Biasa



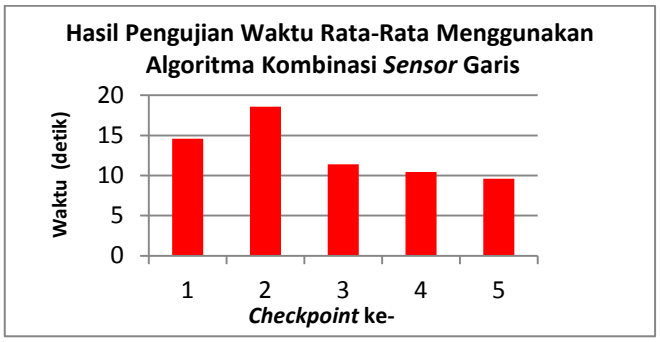
Gambar 12. Hasil Pengujian Waktu Rata-Rata Algoritma Pemadam Api Biasa

Gambar 11 dan Gambar 12 menunjukan grafik waktu rata-rata pada setiap *checkpoint* yang didapatkan dari Tabel 1 dan Tabel 2. Dari kedua grafik tersebut didapatkan bahwa total waktu rata-rata saat menggunakan algoritma pemadam api biasa sebesar 01:20.09 dan saat menggunakan algoritma pemadam api cadangan sebesar 01:37.54. Selain itu dari grafik tersebut dapat dilihat bahwa waktu tercepat saat robot menyusur lapangan terdapat pada *checkpoint* ke 4 sedangkan waktu terlama terdapat pada *checkpoint* ke 2. Dari kedua total waktu grafik tersebut didapatkan selisih sebesar 17.54 detik. Perbedaan waktu tersebut disebabkan oleh perbedaan lama waktu pengecekan UV-TRON dan TPA81. Lama pengecekan UV-TRON sebesar 2 detik dan lama waktu pengecekan TPA81 sebesar 4.25 detik sehingga dari kedua waktu tersebut terdapat selisih waktu pengecekan 2.25 detik. Karena dari pengujian tersebut terdapat 8 kali pengecekan, maka 2.25 detik dikalikan 8 sehingga didapatkan selisih sebesar 18 detik. Dari kedua algoritma pemadam api yaitu algoritma pemadam api biasa dan algoritma pemadam api cadangan, dapat dilihat bahwa banyak waktu yang terbuang untuk melakukan *scanning* api. Oleh sebab itu untuk perlombaan tahun depan, jika aturan perlombaan tidak berubah maka algoritma pemadaman api yang digunakan seharusnya hanya menggunakan kombinasi *sensor* garis untuk melakukan pengecekan terhadap juring. Untuk membuktikan bahwa pengecekan kombinasi *sensor* garis memiliki waktu yang lebih cepat maka dibuat program algoritma yang menggunakan kombinasi *sensor* garis depan dan kanan yang terdapat pada robot untuk mendeteksi juring. Sama seperti pengujian sebelumnya, pengujian algoritma *sensor* garis dilakukan dari posisi *start* di lorong hingga dapat

memadamkan api pada ruangan 4. Pengujian juga dilakukan hingga robot dapat memadamkan api hingga sebanyak 5 kali pertama.

Tabel 4. Hasil Pengujian Algoritma Kombinasi *Sensor* Garis Saat *Start* di Lorong

Pengujian	**Waktu pada *Checkpoint* (detik)**					
	1	**2**	**3**	**4**	**5**	**Total**
1	14.56	18.10	11.34	10.57	09.93	01:04.50
2	14.44	18.86	11.21	10.04	09.32	01:03.87
3	15.01	19.18	11.45	10.43	09.81	01:05.88
4	14.84	18.20	11.63	10.98	09.96	01:05.61
5	14.11	18.54	11.27	10.11	09.01	01:03.04
Rata-rata	14.59	18.57	11.38	10.43	09.60	01:04.58



Gambar 13. Hasil Pengujian Waktu Rata-Rata Menggunakan Algoritma Kombinasi *Sensor* Garis

Tabel 4 menunjukan hasil pengujian waktu rata-rata dan waktu total pada setiap *checkpoint* saat robot *start* di lorong menggunakan algoritma kombinasi *sensor* garis. Dari tabel tersebut didapatkan grafik waktu rata-rata pada setiap *checkpoint* yang dapat dilihat pada Gambar 13. Dari hasil pengujian tersebut dapat dilihat bahwa waktu rata-rata yang dihasilkan lebih kecil pada setiap *checkpoint* jika dibandingkan dengan algoritma pemadam api biasa dan algoritma pemadam api cadangan. Total waktu rata-rata yang didapatkan sebesar 01:04.58. Dari total waktu rata-rata tersebut didapatkan selisih sebesar 16.32 detik. Sehingga terbukti bahwa algoritma kombinasi *sensor* garis lebih cepat dibandingkan dengan algoritma pemadam api biasa dan algoritma pemadam api cadangan.

5. SIMPULAN DAN SARAN

Dari hasil pengujian perancangan robot beroda KRPAI 2013, baik pengujian mekanik, pengujian *sensor-sensor* maupun pengujian algoritma, terdapat simpulan dan saran sebagai berikut.

5.1 SIMPULAN

1. TPA81 dengan penutup pada kipas pemadam tidak dapat mendeteksi api lebih dari jarak 50 cm.
2. Hasil Pembacaan CMPS03 6 cm di atas Vexta tidak linier sempurna karena masih terpengaruh motor Vexta. Hasil Pembacaan CMPS03 linier pada jarak 12 cm. Namun meskipun tidak linier sempurna hasil pembacaan CMPS03 saat tinggi 6 cm masih valid digunakan untuk mendeteksi ruangan acak karena untuk mendeteksi ruangan acak tersebut digunakan *range* dari data sudut CMPS03.
3. Waktu tempuh robot menyusuri dan memadamkan api tidak konsisten pada *checkpoint* ke 4 dan ke 5.
4. Pengecekan api pada garis putih pintu ruangan menggunakan UV-TRON memiliki waktu yang lebih cepat 2.25 detik dari pengecekan TPA81.
5. Metode tercepat untuk menyusur lapangan hingga memadamkan api saat *start* di lorong adalah metode kombinasi *sensor* garis karena tanpa menggunakan cek api di ruangan dengan lama waktu sebesar 01:04.58. Berikutnya metode pengecekan UV-TRON dengan lama waktu sebesar 01:20.09. Dan yang terakhir metode pengecekan menggunakan TPA81 dengan lama waktu 01:37.54.
6. Waktu robot untuk menyusuri lapangan hingga dapat memadamkan api tercepat sebesar 01:04.58 dan waktu terlama dalam menyusuri dan memadamkan api sebesar 01:37.54. Konsistensi keberhasilan robot hingga dapat memadamkan api sebesar 50%.

5.2 SARAN

1. Pada perlombaan selanjutnya sebaiknya menggunakan algoritma pemadam api kombinasi *sensor* garis karena memiliki waktu yang lebih cepat dari

algoritma pemadam api biasa dan algoritma pemadam api cadangan. Hal tersebut disebabkan karena pada algoritma pemadam api kombinasi *sensor* garis, tidak ada pengecekan menggunakan UV-TRON dan TPA81 pada garis pintu masuk.

2. Sebaiknya nanti jika masih menggunakan motor Vexta, penggunaan CMPS03 dinaikkan 12 cm agar data sudut yang dihasilkan linier namun perlu diperhatikan juga tinggi maksimal pada peraturan KRPAI sebesar 27 cm.

DAFTAR PUSTAKA

[1] http://www.arduino.cc/

[2] B. Abraham, *Aplikasi Mobile Automatic Number Plate Recognition (mobile Anpr) Pada Robot Mobil Menggunakan Embedded Prosesor Nios II Pada Altera Fpga Board*. Tugas Akhir, Teknik Elektro, Universitas Surabaya. Surabaya, 2013.

[3] P. Hutomo, *Penggunaan Mikrokontroler AVR untuk Pengontrolan Jarak Jauh Robosoccer melalui Stick Playstation 2*. Tugas Akhir, Teknik Elektro, Universitas Surabaya. Surabaya, 2010.

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

[5] www.atmel.com/images/atmel-8271-8-bit-avr-microcontroller-atmega48a-48pa-88a-88pa-168a-168pa-328-328p_datasheet_summary.pdf

[6] http://arduino.cc/en/reference/wire

[7] http://arduino.cc/en/Tutorial/MasterWriter

[8] http://arduino.cc/en/Reference/SPI

[9] http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

[10] Y. L. Wijaya, *Rancangan Bangun Robot Penyelamat Untuk Medan Reruntuhan Bangunan Di Darat*. Tugas Akhir, Teknik Elektro, Universitas Surabaya. Surabaya, 2012.

[11] http://www.robot-electronics.co.uk/htm/srf05tech.htm

[12] http://www.robot-electronics.co.uk/htm/cmps3tech.htm

[13] http://www.robotstorehk.com/R2868.pdf

[14] http://www.robot-electronics.co.uk/htm/tpa81tech.htm