

Purwarupa Pengendalian Jarak Jauh Pada *Mobile Robot* Berbasis *Web* Melalui Jaringan *Wireless TCP/IP*

Fajar Rinto Hadi Putra^{*1}, Tri Kuntoro Priyambodo²

¹Program Studi Elektronika dan Instrumentasi, JIKE, FMIPA, UGM, Yogyakarta

²Departemen Ilmu Komputer dan Elektronika, Universitas Gadjah Mada, Yogyakarta

e-mail: ^{*}fajarrinto hp@gmail.com, ²mastri@ugm.ac.id

Abstrak

Telah berhasil dirancang sebuah purwarupa mobile robot yang dapat dikendalikan dari jarak jauh, dengan berbasis web melalui jaringan TCP/IP. Purwarupa mobile robot menggunakan sistem gerak menyerupai tank, yang dilengkapi sebuah kamera untuk keperluan surveillance untuk mendapatkan data pantauan lingkungan secara real time di sekitar mobile robot ini, pengguna menggunakan webcam. Karena purwarupa mobile robot ini dapat dikendalikan dari jarak jauh menggunakan aplikasi berbasis web dari komputer host melalui media jaringan wireless TCP/IP, ditujukan untuk keperluan penginderaan jarak jauh di lingkungan yang berbahaya bagi keselamatan manusia. Komunikasi yang dilakukan menggunakan sebuah perangkat wireless untuk dua arah mengirim dan menerima data. Pada tahap debugging, komunikasi wireless dilakukan tanpa penghalang (line of sight), antara komputer host dengan purwarupa mobile robot tersebut.

Kata kunci— mobile robot, komunikasi wireless, TCP/IP

Abstract

It has been successfully developed a prototype mobile robot, controlled remotely, based on web via wireless TCP/IP network. This prototype mobile robot adopts tank's movement, equipped with a surveillance camera to acquire real time data on surrounding environment, a webcam is used by user. Since it is controlled remotely from a web based application on host computer via wireless TCP/IP network, it could do some telemetry in such environment, which harmful for human. It uses a wireless device for birectional communication, to transmit and receive data. While on debugging, wireless communication are took place directly without any obstacle (line of sight), between host computer and prototype mobile robot.

Keywords— mobile robot, wireless communication, TCP/IP

1. PENDAHULUAN

Dalam menjalankan tugasnya sehari-hari, beberapa profesi menuntut kehadiran manusia pada lingkungan kerja yang berbahaya, seperti lingkungan yang tercemar unsur radioaktif, daerah yang mengandung gas beracun, goa pertambangan, ataupun tempat yang belum pernah dan sulit dijamah oleh manusia.

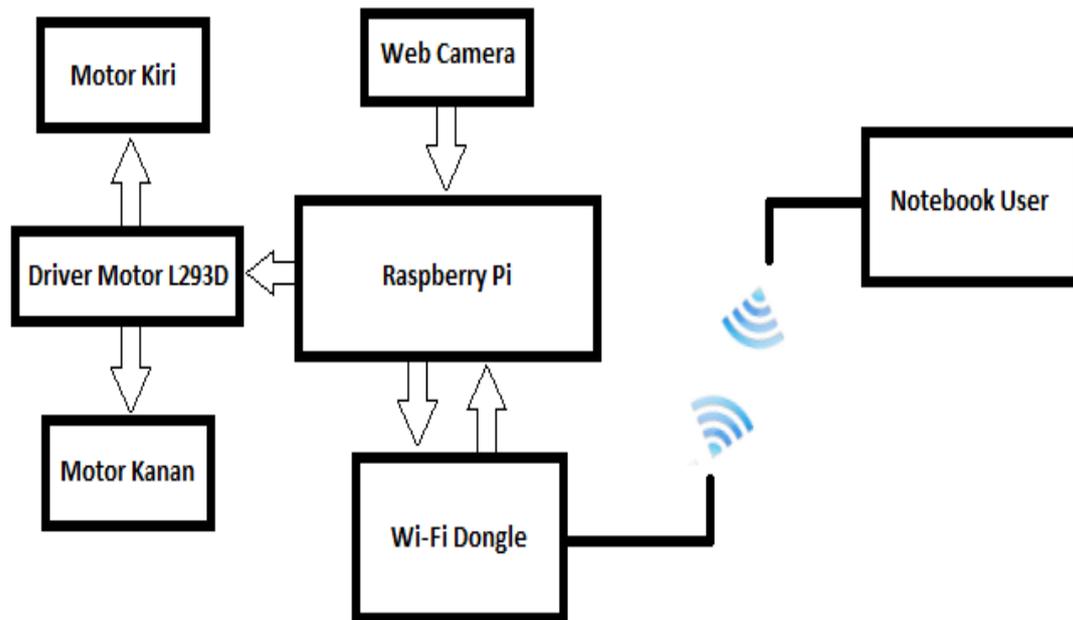
Dalam rangka mengurangi resiko kecelakaan dan bahaya secara langsung, maka diperlukan suatu alat yang dapat menggantikan kehadiran manusia pada lokasi-lokasi tersebut. Peralatan tersebut diharuskan dapat mewakili peran manusia pada lokasi yang dimaksud yaitu dapat memberikan informasi dari suatu lokasi ke stasiun pemantau dengan jarak yang jauh. Dengan demikian akan lebih menjamin keselamatan manusia, informasi yang diperlukan dari suatu lokasi tetap bisa didapatkan dan dipantau secara berkelanjutan. Salah satu solusi yang dapat menjawab permasalahan di atas adalah dengan mengembangkan *unmanned vehicle* untuk keperluan *surveillance* yang salah satunya diwujudkan dalam bentuk *mobile robot*. Robot adalah peralatan elektro-mekanik atau bio-mekanik, atau gabungan peralatan yang menghasilkan gerakan otonomi maupun berdasarkan gerakan yang diperintahkan [1]. Robot yang dilengkapi dengan peralatan komunikasi dimungkinkan untuk dikendalikan dari jarak jauh [2] serta memungkinkan untuk melakukan pengiriman informasi [3] dari suatu lokasi ke stasiun pemantau.

Salah satu peralatan yang dikendalikan berupa sebuah robot yang dilengkapi dengan kamera untuk keperluan pengendalian dan pemantauan sehingga operator tetap dapat mengendalikan robot tanpa harus melihat posisi robot. Sistem yang diadopsi berupa sistem penglihatan (mata), sistem pendengaran (telinga) ataupun sistem gerak [4], sehingga robot bergerak dengan kecerdasan yang telah diprogramkan [5]. Selain itu, peralatan jaringan komputer saat ini relatif mudah di dapat dan harganya cukup terjangkau serta mudah untuk dilakukan pengembangan jaringan. Dalam penelitian ini akan dikembangkan sebuah purwarupa *mobile robot* yang dilengkapi dengan kamera untuk keperluan *surveillance* yang dapat dikontrol secara jarak jauh menggunakan aplikasi berbasis *web* dari komputer *user/client* melalui media jaringan *wireless TCP/IP*.

2. METODE PENELITIAN

2.1 Perancangan Perangkat Keras

Robot yang dikembangkan merupakan *mobile robot* dengan sistem penggerak *tank (track)*. Robot dirancang untuk dapat dikendalikan dari jarak jauh melalui jaringan *wireless TCP/IP* dengan antarmuka aplikasi berbasis *web* yang benamkan pada Raspberry Pi [6] yang langsung terintegrasi dengan robot yang sekaligus berfungsi sebagai *web server*. Robot dilengkapi dengan kamera *webcam* sebagai kamera *navigasi* bagi operator untuk mengendalikan robot. Hasil visualisasi dari kamera akan ditampilkan secara *real time* pada halaman *web*. Komputer *client/user* (operator) bersifat fleksibel yang tidak terbatas pada komputer tertentu. Operator cukup memanggil alamat IP robot untuk dapat mengakses halaman *web* untuk mengendalikan robot. Gambar 1 menunjukkan blok diagram pengendalian *mobile robot* jarak jauh berbasis *web* dengan media jaringan *wireless TCP/IP*.

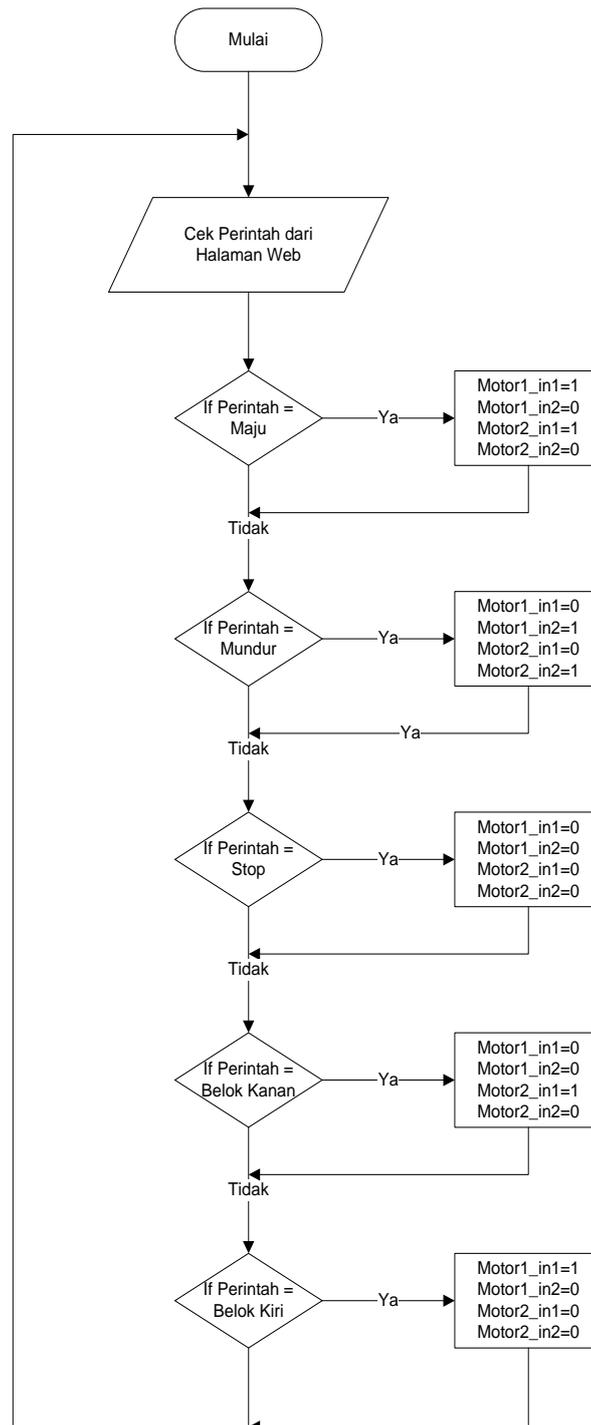


Gambar 1 Diagram blok sistem *mobile robot*

Raspberry Pi sebagai pusat kendali robot menerima *input* dari *web camera* dan perintah dari *user*. Berdasarkan perintah dari *user*, Raspberry Pi akan memberikan sinyal *output* untuk *driver L293D* untuk mengendalikan motor kiri dan kanan. Untuk berkomunikasi dengan *user*, Raspberry Pi menggunakan perangkat tambahan USB Wi-Fi *dongle*.

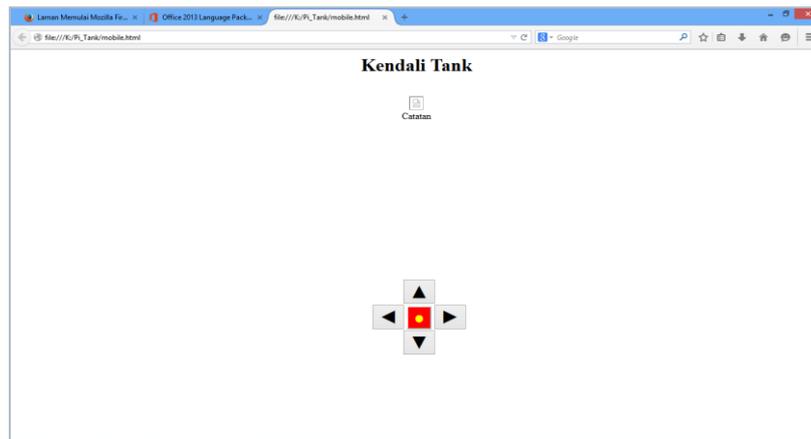
2.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada penelitian ini terdiri atas tiga bagian, yaitu instalasi perangkat lunak yang digunakan pada Raspberry Pi, konfigurasi pengendalian gerakan robot, dan konfigurasi halaman *web*. Perangkat lunak pendukung yang digunakan antara lain, *mjpegstreamer*, *lighttpd*, konfigurasi jaringan. Program pengendali gerakan robot digunakan dengan menggunakan perangkat lunak Python. Diagram alir program pengendalian gerakan robot ditunjukkan oleh Gambar 2.



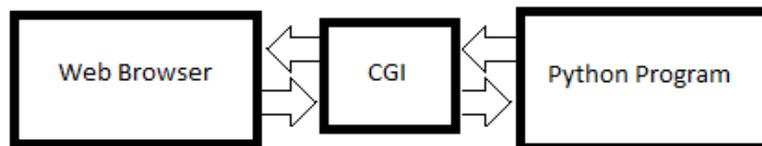
Gambar 2 Diagram alir program kendali Python

Halaman *web* yang menjadi antarmuka untuk mengendalikan robot terdiri atas *frame* video yang berfungsi untuk menampilkan video *streaming* dari robot yang digunakan untuk navigasi dalam mengendalikan robot, tombol panah maju untuk memerintahkan robot bergerak maju, tombol panah ke kanan untuk memerintahkan robot berbelok ke kanan, tombol panah kiri, untuk memerintahkan robot berbelok ke kiri, tombol panah mundur untuk memerintahkan robot bergerak mundur, serta tombol *stop* untuk memerintahkan robot berhenti. Tampilan antarmuka *web* untuk mengendalikan robot dapat dilihat seperti pada Gambar 3.



Gambar 3 Antarmuka kendali *web mobile* robot

Halaman *web* yang menjadi antarmuka untuk mengendalikan robot terdiri atas *frame* video yang berfungsi untuk menampilkan video *streaming* dari robot yang digunakan untuk navigasi dalam mengendalikan robot, tombol panah maju untuk memerintahkan robot bergerak maju, tombol panah ke kanan untuk memerintahkan robot berbelok ke kanan, tombol panah kiri, untuk memerintahkan robot berbelok ke kiri, tombol panah mundur untuk memerintahkan robot bergerak mundur, serta tombol *stop* untuk memerintahkan robot berhenti. Adapun Gambar 4 menunjukkan arsitektur sederhana *web* untuk pengendalian *mobile* robot.



Gambar 4 Arsitektur sederhana *web*

Untuk *web browser* bersifat *user-end*, menampilkan data *streaming* video dan *button* kendali yang diperlukan *user*. Sedangkan pada sisi robot terdapat program Python, yang akan menjalankan *input request* dari *user* melalui *web browser*. Untuk menjembatani antara *web browser* dengan program Python, diperlukan CGI (*Common Gateway Interface*) yakni dengan FCGI. Fungsi dari CGI ialah menjembatani antara *web browser* dengan program Python, menerima *request* dari *web browser* untuk disampaikan ke program Python, dan bila *request* tersebut sesuai, maka akan dijalankan oleh program Python tersebut. Selanjutnya program Python juga akan memberikan tanggapan balik atas *request* tersebut kepada *user* (*web browser*), juga melalui CGI terlebih dahulu.

2.3 Perancangan Pengujian

Perancangan pengujian untuk mendapatkan data untuk analisis pencapaian hasil penelitian. Pengujian yang dilakukan meliputi jarak optimal komunikasi *wireless*, kemampuan mobilitas robot, dan kemampuan respon robot. Tabel 1 menunjukkan data pengujian dan capaian yang diharapkan.

Tabel 1 Pengujian dan capaian yang diharapkan

No	Pengujian	Capaian
1	Jarak Komunikasi Optimal <i>Wireless</i> TCP/IP Robot	Diketahui jarak optimal pengendalian robot
2	Kemampuan Mobilitas Robot	Diketahui kemampuan mobilitas robot dalam gerakan maju, mundur, belok kanan, belok kiri, serta kecepatan robot
3	Respon Robot Terhadap Kendali Manual Melalui <i>Web</i>	Didapat respon terbaik robot dalam menanggapi setiap masukan kendali manual dari <i>web</i> .

3. HASIL DAN PEMBAHASAN

3.1 Pengujian Variasi Jarak Pengendalian Robot terhadap Kualitas Wi-Fi

Pelemahan sinyal *wireless* dipengaruhi oleh banyak faktor, seperti penyerapan, pemantulan, pemecahan, pembelokan sinyal, dan *Free Space Path Loss* atau FSPL. Pengujian *mobile* robot dilakukan di lingkungan tanpa penghalang tembok. Meski banyak faktor yang mempengaruhi pelemahan sinyal *wireless*, namun dalam analisis yang dilakukan hanya mencakup faktor FSPL. Hal tersebut dikarenakan faktor pelemahan sinyal lain sangat terpengaruh pada faktor lingkungan tempat pengujian *mobile* robot tersebut.

Faktor *Free Space Path Loss* (FSPL), yakni banyaknya kekuatan sinyal yang hilang (melemah) dalam ruang bebas tanpa hambatan yang berbanding lurus dengan jarak sumber sinyal. Besarnya pelemahan sinyal dari faktor FSPL akan dibandingkan dengan hasil pengukuran dengan aplikasi inSSIDer 2.0. Persamaan (1) menunjukkan analisis perhitungan nilai pelemahan sinyal FSPL.

$$FSPL = 20 \log d + 20 \log f + K \quad (1)$$

Keterangan d ialah jarak, f ialah frekuensi, dan K ialah konstanta yang nilainya tergantung dari satuan dari d dan f. Bila d dalam km dan f dalam MHz maka nilai K ialah 33,44, sedangkan jika f dalam GHz, maka nilai K ialah 92,45. Berbeda lagi bila d dalam meter dan f dalam MHz, maka nilai K ialah -27,55. Berdasarkan Persamaan 1 pelemahan FSPL dengan frekuensi *wireless* yang digunakan ialah 2,4 GHz, jarak pengujian 1,2 meter, maka dapat dihitung besarnya pelemahan sinyal seperti pada perhitungan di bawah ini :

$$FSPL = 20 \log d + 20 \log f + K$$

$$FSPL = (20 \log (1,2) + 20 \log (2400) - 27,55) \text{ dB}$$

$$FSPL = (20 \times 0,07918124604 + 20 \times 3,38021124171 - 27,55) \text{ dB}$$

$$FSPL = (1,583624921 + 67,60422483 - 27,55) \text{ dB}$$

$$FSPL = 41,63784976 \text{ dB}$$

Diperoleh nilai pelemahan FSPL sebesar 41,63784976 dB, yang dibulatkan menjadi 41,64 dB. Karena merupakan suatu pelemahan sinyal maka nilai tersebut menjadi negatif, sehingga menjadi -41,64 dB. Dengan persamaan yang sama kemudian dilakukan perhitungan kembali untuk variasi jarak yang berbeda-beda, dengan interval 1,2 meter. Data perhitungan jarak terhadap kekuatan sinyal, tampak pada Tabel 2.

Tabel 2 Data Perhitungan kekuatan sinyal berbanding terhadap jarak

No.	Jarak (m)	Kekuatan Sinyal (dB)
1	1,20	-41,64
2	2,40	-47,66
3	3,60	-51,18
4	4,80	-53,68
5	6,00	-55,62
6	7,20	-57,20
7	8,40	-58,54
8	9,60	-59,70
9	10,80	-60,72

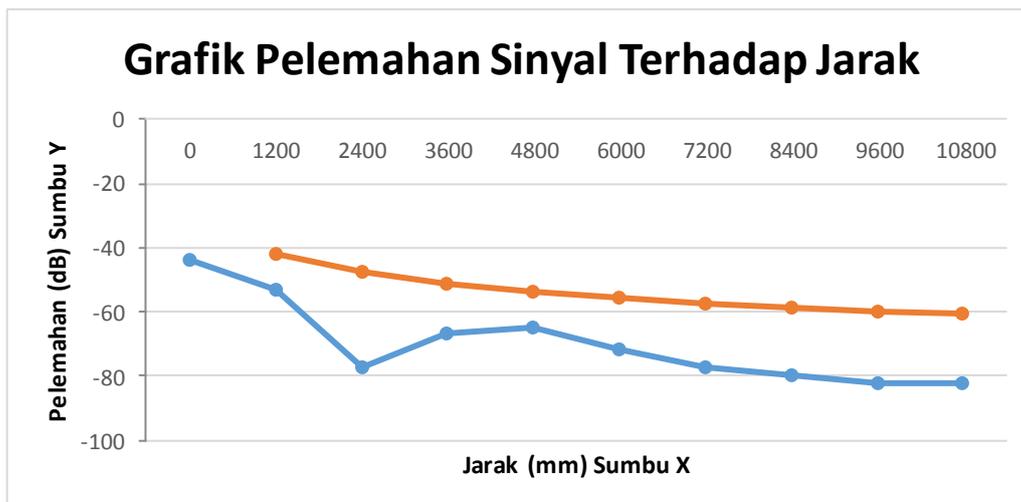
Sementara itu kekuatan sinyal yang terukur dengan aplikasi ialah pada Tabel 3 data pengukuran kekuatan sinyal berbanding terhadap jarak.

Tabel 3 Data pengukuran kekuatan sinyal berbanding terhadap jarak

No.	Jarak (m)	Kekuatan Sinyal (dB)	<i>Time Reply</i> (ms)
1	0	-44	771
2	1.20	-53	519
3	2.40	-77	616
4	3.60	-67	627
5	4.80	-65	636
6	6.00	-72	704
7	7.20	-77	798
8	8.40	-80	0
9	9.60	-82	0
10	1080	-82	0

Jarak pengujian diambil dengan interval setiap 1,2 meter. Kekuatan sinyal dalam dB dengan nilai negatif menandakan pelemahan sinyal. Mulai pada data jarak ke-8, yakni 8,4 meter sinyal sudah sangat melemah hingga tidak lagi memberikan tanggapan balik, *time out service*. Dari data pengujian tersebut dapat direpresentasikan dalam sebuah grafik antara jarak (mm) sumbu X, dengan kekuatan sinyal (dB) sumbu Y, tampak pada Gambar 5.

Berdasarkan pada data Tabel 2 dan Tabel 3, dapat disusun Gambar 5, grafik perbandingan antara hasil perhitungan dengan pengukuran pelemahan sinyal dengan aplikasi menggunakan inSSIDer 2.0. Garis berwarna jingga merepresentasikan hasil perhitungan pelemahan sinyal, sedangkan garis dengan warna biru untuk pengukuran pelemahan sinyal dengan aplikasi inSSIDer 2.0.

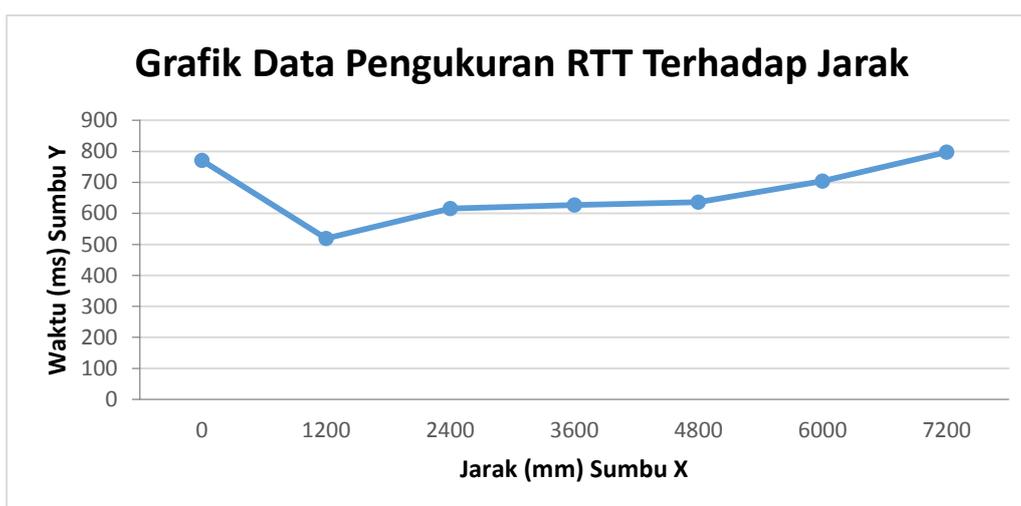


Gambar 5 Grafik jarak dengan kekuatan sinyal

Terdapat selisih antara hasil perhitungan dengan pengukuran dengan inSSIDer 2.0. Dimana pelemahan yang terukur dengan inSSIDer 2.0 lebih besar. Hal tersebut dapat ditolerir, mengingat banyak faktor yang menyebabkan pelemahan sinyal dan semua itu diukur dengan inSSIDer 2.0, sedangkan untuk hasil perhitungan hanya berdasarkan faktor FSPL saja.

3.2 Pengujian Waktu Tanggap Mobile Robot

Round trip delay time (RTD) atau *round trip time (RTT)*, terkadang juga disebut *ping time*, merupakan lamanya waktu yang diperlukan oleh sebuah sinyal untuk dikirimkan ditambah dengan lamanya waktu untuk sinyal *acknowledge* diterima. Dengan kata lain RTT ialah lamanya waktu yang diperlukan suatu sinyal dari *source* (sumber) yang dikirimkan ke suatu *destination* (tujuan), dan kembali ke *source* dengan memberikan *acknowledge* paket data terkirim atau tidak. Untuk analisis kaitan antara jarak dengan RTT hanya akan ada pada data yang diperoleh dengan menggunakan aplikasi. Gambar 6 grafik data waktu *reply* atau RTT terhadap jarak berdasarkan pada Tabel 3.



Gambar 6 Grafik jarak dengan waktu *reply*

Pada Gambar 6, sumbu X untuk jarak dalam mm dan sumbu Y untuk waktu *reply* (RTT) dalam ms. Pada grafik Gambar 6 tersebut hanya menampilkan data hingga jarak 7200

mm (7,2 meter), karena berdasarkan data pada Tabel 2 setelah jarak lebih dari itu *reply time out*, tidak ada balasan. Berdasarkan grafik Gambar 6 dan Tabel 2, terdapat kecenderungan semakin jauh dari sumber, maka waktu *reply* semakin lama. Selain itu diperoleh pula jarak efektif yang disarankan yakni tidak lebih dari 7,2 meter.

3.3 Pengujian Kemampuan Streaming Video

Pengujian kualitas kemampuan *streaming* video dilakukan dengan merubah nilai *framerate* dan resolusi video kemudian melihat penggunaan *resource* CPU pada Raspberry Pi sehingga diharapkan dapat diketahui nilai resolusi dan *framerate* yang paling baik yang dapat digunakan oleh *mobile* robot. Untuk data hasil pengujian resolusi video dan *resource* CPU yang digunakan ditunjukkan pada Tabel 4.

Tabel 4 Data pengujian *streaming* video

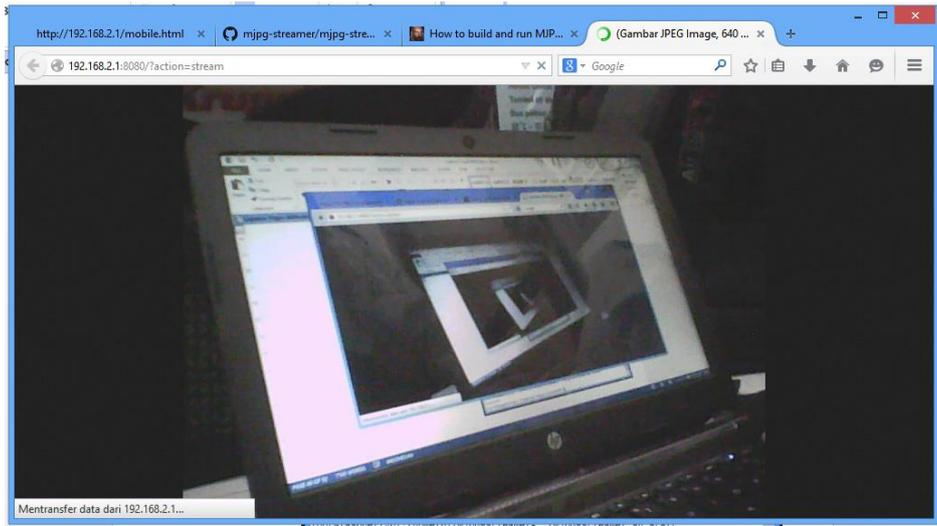
Resolusi	Framerate	CPU Usage
320x240	15 fps	0,3 %
640x480	15 fps	1,6 %
640x480	25 fps	1,6 %
1024x720	25 fps	1,6 %
1024x720	30 fps	1,6 %
720p (HD) (1280x720)	<i>Not supported</i>	<i>Not supported</i>
1080p (Full HD) (1920x1080)	<i>Not supported</i>	<i>Not supported</i>

Pada pengujian *streaming* video ini, divariasikan beberapa resolusi dan *framerate* video untuk aplikasi *mjpg_streamer*, yaitu 320x240 pada 15 fps, 640x480 pada 15 fps, 640x480 pada 25 fps, 1024x720 pada 25 fps, 1024x720 pada 30fps, serta video HD 720p (1280x720), dan video *full* HD 1080p (1920x1080). Pada pengujian video dengan resolusi mulai 320x240 sampai 1024x720 dan dengan *framerate* antara 15 fps sampai 30 fps, diperoleh nilai rata-rata penggunaan CPU sebesar 1,34%.

Sedangkan untuk video HD 720p dengan resolusi 1280x1080 serta video *full* HD 1080p dengan resolusi 1920x1080, kemampuan *webcam* belum mendukung. Berdasarkan data pada Tabel 4, pada pengujian ini diketahui bahwa untuk melakukan *streaming* video dengan resolusi 320x240 pada *framerate* 15 fps *resource* CPU yang digunakan hanyalah sebesar 0,3 %, sedangkan untuk video dengan resolusi lebih tinggi antara 640x480 sampai 1024x720 dengan *framerate* 15 fps sampai 30 fps, *resource* CPU yang digunakan relatif sama yaitu sebesar 1,6 %.

Dengan demikian dapat disimpulkan bahwa kualitas video optimal yang dapat diaplikasikan pada *mobile* robot yaitu video dengan tingkat resolusi 1024x720 dengan *framerate* 25 atau 30 *frame per second*. Hasil *streaming* video ditampilkan pada sebuah *frame* di halaman antarmuka *web* pengendali *mobile* robot.

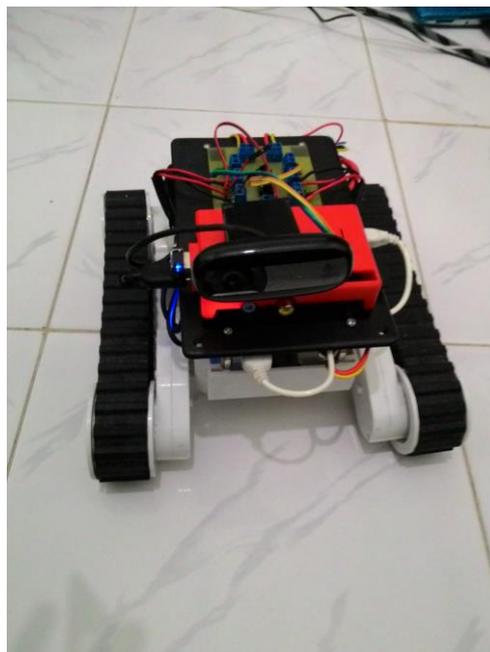
Hanya saja sudut pandang yang ditangkap oleh *web camera* yang terintegrasi dengan *mobile* robot masih bersifat tetap, yakni hanya satu arah lurus ke depan. Hal ini tentu saja membatasi kemampuan dari *mobile* robot untuk *surveillance* dan mengirimkan informasi lingkungan di sekitarnya melalui *streaming* video. Adapun Gambar 7 menampilkan hasil tampilan *streaming* video dari *web camera* *mobile* robot.



Gambar 7 Hasil *streaming* video kamera *mobile* robot

3.4 Pengujian Kendali *Mobile* Robot

Pengujian kendali *mobile* robot dilakukan dengan cara melakukan analisis terhadap kemampuan dari *mobile* robot tersebut untuk menjalankan perintah dari *user* melalui *web browser*. Parameter yang digunakan untuk menilai berhasil atau tidaknya pengendalian gerak *mobile* robot ini ialah kesesuaian antara perintah yang diberikan oleh *user* melalui *web browser* dengan pergerakan dari *mobile* robot tersebut. Bila terdapat hasil yang tidak sesuai antara perintah yang diberikan oleh *user* melalui *web browser* dengan hasil pergerakan dari *mobile* robot tersebut, maka hasil pengujian gerak *mobile* robot tersebut gagal, sehingga diperlukan perbaikan dalam sistem, baik dari perangkat keras maupun perangkat lunak. Lain halnya bila, pergerakan dari *mobile* robot tersebut sesuai dengan perintah dari *user*, meski terlambat menjalankannya, maka hal tersebut akan dianggap sebagai sebuah keberhasilan. Adapun Gambar 8 menunjukkan tampilan *mobile* robot dalam proses pengujian pergerakan.



Gambar 8 Tampilan *mobile* robot

Hasil pengujian kendali *mobile* robot disajikan sebagaimana pada Tabel 5 yang berisikan perintah dan pencapaian dari *mobile* robot tersebut.

Tabel 5 Pengujian kendali *mobile* robot

Nomor	Perintah	Hasil
1	Maju	Berhasil
2	Mundur	Berhasil
3	Belok Kanan	Berhasil
4	Belok Kiri	Berhasil

Berdasarkan pada Tabel 5 data hasil pengujian kendali *mobile* robot tersebut mampu melakukan gerak yang sesuai dengan perintah yang diberikan oleh *user* melalui *web browser*. Untuk mengatur kerja motor sehingga diperoleh gerak *mobile* robot maju, mundur, juga berbelok, Raspberry Pi memberikan sinyal kendali *output* pada *port* GPIO ke *driver* motor. Berikut ini ialah Tabel 6 kendali motor.

Tabel 6 Kendali motor

Nomor	GPIO Raspberry Pi				Gerak Robot	Motor Kiri	Motor Kanan
	22	24	25	27			
1	<i>High</i>	<i>High</i>	<i>Low</i>	<i>Low</i>	Maju	<i>On</i>	<i>On</i>
2	<i>Low</i>	<i>Low</i>	<i>High</i>	<i>High</i>	Mundur	<i>Reverse</i>	<i>Reverse</i>
3	<i>Low</i>	<i>High</i>	<i>Low</i>	<i>Low</i>	Belok Kanan	<i>On</i>	<i>Off</i>
4	<i>High</i>	<i>Low</i>	<i>Low</i>	<i>Low</i>	Belok Kiri	<i>Off</i>	<i>On</i>

Berdasarkan pada Tabel 6, kerja dari kedua motor dikendalikan dengan empat buah *port* GPIO Raspberry Pi. Kendali motor yang dilakukan bersifat *on-off* dan dengan pengaturan polaritas dari motor tersebut. Berdasarkan pada Tabel 6, kondisi motor *on* berarti motor berputar mengakibatkan robot bergerak maju, *reverse* putaran motor mengakibatkan robot bergerak mundur, dan *off* artinya motor tidak aktif.

Untuk dapat mengendalikan motor di sisi kanan agar menghasilkan suatu gerakan maju maka *port* GPIO 22 harus bernilai *high* dan *port* GPIO 25 bernilai *low*. Demikian pula sebaliknya, untuk menghasilkan gerakan mundur maka *port* GPIO 25 harus bernilai *high* dan *port* GPIO 22 bernilai *low*. Kedua *port* GPIO 22 dan 25 tidak boleh bernilai *high* pada saat yang bersamaan, karena hal tersebut dapat merusak *driver* motor. Akan tetapi diperbolehkan kedua *port* GPIO tersebut untuk bernilai *low* pada saat yang bersamaan.

Sementara untuk mengendalikan motor di sisi kiri agar menghasilkan gerakan maju maka *port* GPIO 24 harus bernilai *high* dan *port* GPIO 27 bernilai *low*. Demikian pula sebaliknya, untuk menghasilkan gerakan mundur maka *port* GPIO 27 harus bernilai *high* dan *port* GPIO 24 bernilai *low*. Sama seperti pada motor kanan, kedua *port* GPIO 24 dan 27 tidak boleh bernilai *high*, tetapi diperbolehkan bernilai *low* pada saat yang bersamaan.

Untuk mendapat gerakan *mobile* robot yang sesuai dengan yang diharapkan, diperlukan kombinasi kerja dari kedua motor tersebut. Agar *mobile* robot dapat bergerak maju secara lurus maka kedua motor di sisi kiri dan kanan harus sama-sama berputar maju (*on*), sedangkan untuk bergerak mundur secara lurus maka kedua motor di sisi kanan dan kiri harus berputar dengan arah sebaliknya, *reverse*. Sementara itu agar *mobile* robot tersebut berbelok salah satu motor harus berputar (arah maju) sedangkan motor satunya lagi dalam keadaan *off*. Pada saat motor kanan *on* dan motor kiri *off*, maka *mobile* robot akan berputar ke arah kiri, sebaliknya pada saat motor kanan *off* dan motor kiri *on*, maka *mobile* robot akan berputar ke arah kanan.

Seperti telah disebutkan sebelumnya, bahwa pengendalian pergerakan *mobile* robot dilakukan dengan cara *switch on-off*, hal tersebut mengakibatkan kecepatan pergerakan dari *mobile* robot tersebut relatif konstan. Kecepatan gerakan *mobile* robot dipengaruhi oleh

kemampuan motor yang digunakan, tanpa dapat dilakukan pengendalian kecepatan dengan memberikan percepatan atau perlambatan.

4. KESIMPULAN

Telah berhasil dilakukan penelitian Purwarupa Pengendalian Jarak Jauh Pada *Mobile Robot* Berbasis *Web* Melalui Jaringan *Wireless TCP/IP*. Berdasarkan pada hasil pengujian dan analisis percobaan *mobile* robot yang telah dilakukan, jarak optimal pengendalian dari *mobile* robot tidak lebih dari 7,2 meter, dengan tanpa adanya penghalang (*line of sight*) antara pengendali dengan *mobile* robot. Kemampuan kualitas *streaming* video untuk *surveillance*, *mobile* robot optimal ialah pada tingkat resolusi 1024x720 dengan *framerate* 30 fps, yang menggunakan *resource* CPU 1,6%, dengan arah sudut pandang *web camera* yang bersifat tetap. Kemampuan arah pergerakan *mobile* robot ini telah sesuai dengan perintah yang diberikan oleh *user* melalui *web browser*, meski belum dapat melakukan perubahan kecepatan pergerakan baik dengan percepatan maupun perlambatan.

Beberapa hal dapat dilakukan untuk penelitian dan pengembangan selanjutnya. Diantaranya adalah Pengembangan *mobile* robot yang memungkinkan untuk dikendalikan meski terdapat berbagai penghalang seperti dinding atau dari ruangan yang berbeda, seperti dengan memperkuat sinyal WiFi. Peningkatan jarak optimal kendali *mobile* robot, yang dapat dilakukan dengan cara memasang *repeater* atau memperluas jaringan WLAN *mobile* robot atau memberikan IP *Public* pada *mobile* sehingga terhubung dengan jaringan internet dan dapat diakses dari mana saja tanpa ada batasan. Peningkatan kemampuan yang memungkinkan pengendalian arah sudut pandang dari *web camera* pada *mobile* robot, sehingga cakupan area yang dapat dipantau menjadi lebih luas. Peningkatan kemampuan *mobile* robot untuk dapat bergerak secara lebih dinamis dengan melakukan percepatan dan perlambatan.

DAFTAR PUSTAKA

- [1] Halim, S., 2007, *Merancang Mobile Robot Pembawa Objek Menggunakan OOPic-R*, Elex Media Komputindo, Jakarta.
- [2] Wong, C. dan Rydgren A., 2003, *Controlling Bluetooth Mini Race Car from the P900*, http://developer.sonyericsson.com/site/global/techsupport/tipstrickscode/java/p_tips_java_1202.jsp
- [3] Rahmani, B., Putra, A.E., Harjoko, A., dan Priyambodo, T.K., 2015, Review of Vision Based Robot Navigation Method. *IAES International Journal of Robotics and Automation (IJRA)*, 4(4), 31-38. <http://doi.org/10.11591/ijra.v4i4.8514>
- [4] Anadhar, 2007, Al Qur'an dan Teknologi Sensor Robot, <http://www.anadhar.multiply.com/journal/item/11>, diakses pada tanggal 28 Agustus 2012.
- [5] Rahmani, B., Harjoko, A., Priyambodo, T.K., dan Aprilianto, H., 2016, Research of smart real-time robot navigation system. In *AIP Conference Proceedings*. <http://doi.org/10.1063/1.4940853>
- [6] Raspberry Pi, 2013, Raspberry Pi, <http://www.raspberrypi.org>, diakses pada tanggal 15 Februari 2013.