

REMOTE MONITORING BERBASIS SMARTPHONE ANDROID

REMOTE MONITORING BASED ANDROID SMARTPHONE

Denish Novenda¹, Agung Nugroho Jati², Erwin Susanto, Ph.D³

^{1,2}Prodi S1 Sistem Komputer, Fakultas Teknik Elektro, Universitas Telkom Bandung

³Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom Bandung

¹denishnovenda20@gmail.com, ²agungni@telkomuniversity.ac.id, ³erwinelektro@telkomuniversity.ac.id

Abstrak

Perkembangan *teleoperation* pada *mobile robot* semakin berkembang pesat. Hal ini yang membuat penulis mengimplementasikan *teleoperation* sebagai *remote monitoring* berbasis android pada bidang robotik. Klasifikasi robot yang diimplementasikan berupa robot dengan karakter tank sehingga hampir semua karakter bidang darat dapat dilalui. *Remote monitoring* ini dilengkapi *interface webcam* untuk *streaming video* dan tombol yang dapat mengatur arah gerak melalui koneksi *wifi*. Sistem *remote monitoring* ini telah berhasil diimplementasikan pada Raspberry Pi dan *smartphone* android. Selain itu sistem *remote monitoring* ini juga telah berhasil terhubung antara Raspberry Pi dan *smartphone* android menggunakan jaringan komunikasi *wifi*.

Kata Kunci : *remote, monitoring, android, robotika, wifi*

Abstract

The development of mobile robot teleoperation on growing. This makes the authors implement teleoperation as Android-based remote monitoring. Classification robot is implemented in the form of a robot with a tank character that nearly all land areas of the characters can be passed. The main feature of remote monitoring such as camera interface for streaming video and buttons that can set the direction of motion through a wifi connection. This remote monitoring system has been successfully implemented on a Raspberry Pi and android smartphones. In addition, the remote monitoring system has also been successfully connected between the Raspberry Pi and android smartphones using wifi communication network.

Keywords: *remote, monitoring, android, robotic, wifi*

1. Pendahuluan

Jarak radius *wifi* yang lebar dan dapat mengirim data yang tidak terbatas seperti pada penggunaan *remote* dengan koneksi *infrared* merupakan salah satu kelebihan dari koneksi *wifi*. Selain kelebihan pada jarak radius yang lebar, penggunaan *wifi* untuk *remote monitoring* tidak hanya berlaku pada suatu *device* tertentu saja melainkan dapat digunakan untuk *device* yang memiliki fitur *wifi*. Terlebih koneksi *wifi* ini banyak didukung oleh *smartphone* berbasis android. Android merupakan salah satu sistem operasi yang digunakan pada telepon seluler yang berbasis Linux. Salah satu keunggulan sistem operasi android adalah memakai basis kode komputer yang bisa didistribusikan secara terbuka (*open source*) sehingga pengguna dapat mengembangkan dan membuat aplikasi baru di dalamnya. [2]

Sistem yang dibuat dapat digunakan dalam perusahaan, rumah dan bahkan di militer. Selain itu sistem dapat dengan mudah dioperasikan oleh banyak orang dengan pengetahuan dasar komputer. Sistem robot sendiri sangat dijamin keamanannya, agar tidak sembarang orang dapat mengaksesnya untuk mengontrol robot dan *monitoring* pada seluruh perangkat android. Chirag B telah mengembangkan sistem dengan kontrol dan *monitoring* menggunakan perangkat Raspberry Pi dan Android [1]. *Teleoperation* merupakan suatu metode dimana ada *operator* dan robot. *Operator* mempunyai *interface* untuk menjalankan robot secara jarak jauh dan robot mempunyai sensor, *actuator* dan catu daya untuk dapat berkomunikasi dengan *operator* [11]. Yong-Ho Seo telah mengembangkan sistem robot berbasis *touch control* untuk memperoleh informasi dari robot dan mengontrol robot dengan *smartphone* [9]. Sung Wook Moon memperkenalkan *smartphone* yang dapat dioperasikan untuk mengontrol dan sistem *monitoring* pada robot laba-laba dengan *delay* 200ms antara kamera dan *smartphone* [10]. Letian Liu telah memperkenalkan *mobile robot* bernama "Robot-XD" dirancang dengan *control pad* pada *smartphone* android. *Mobile robot* ini menggunakan koneksi *wifi* untuk mengontrol robot dengan *smartphone* [7].

Kontribusi dari paper ini meliputi penerapan model sistem pada *mobile robot* dengan kontrol dan *monitoring* secara *real-time* menggunakan *smartphone* android. Sistem *monitoring* ini tidak terbatas pada

penggunaan untuk kontrol jarak jauh saja melainkan berguna sebagai *monitoring* jarak jauh. Selain kelebihanannya, penggunaan koneksi *wifi* untuk *remote monitoring* memiliki kelemahan seperti *delay* jika jarak antara *device* jauh dan performansi *remote monitoring* jika digunakan pada perangkat *smartphone* android. Dalam implementasinya, penulis menggunakan metode *trial and error* pada sisi *interface* sistem untuk menentukan apakah sistem *remote monitoring* ini berjalan baik pada perangkat *smartphone* android dengan sistem operasi minimal *Gingerbread* (2.3).

2. Metodologi

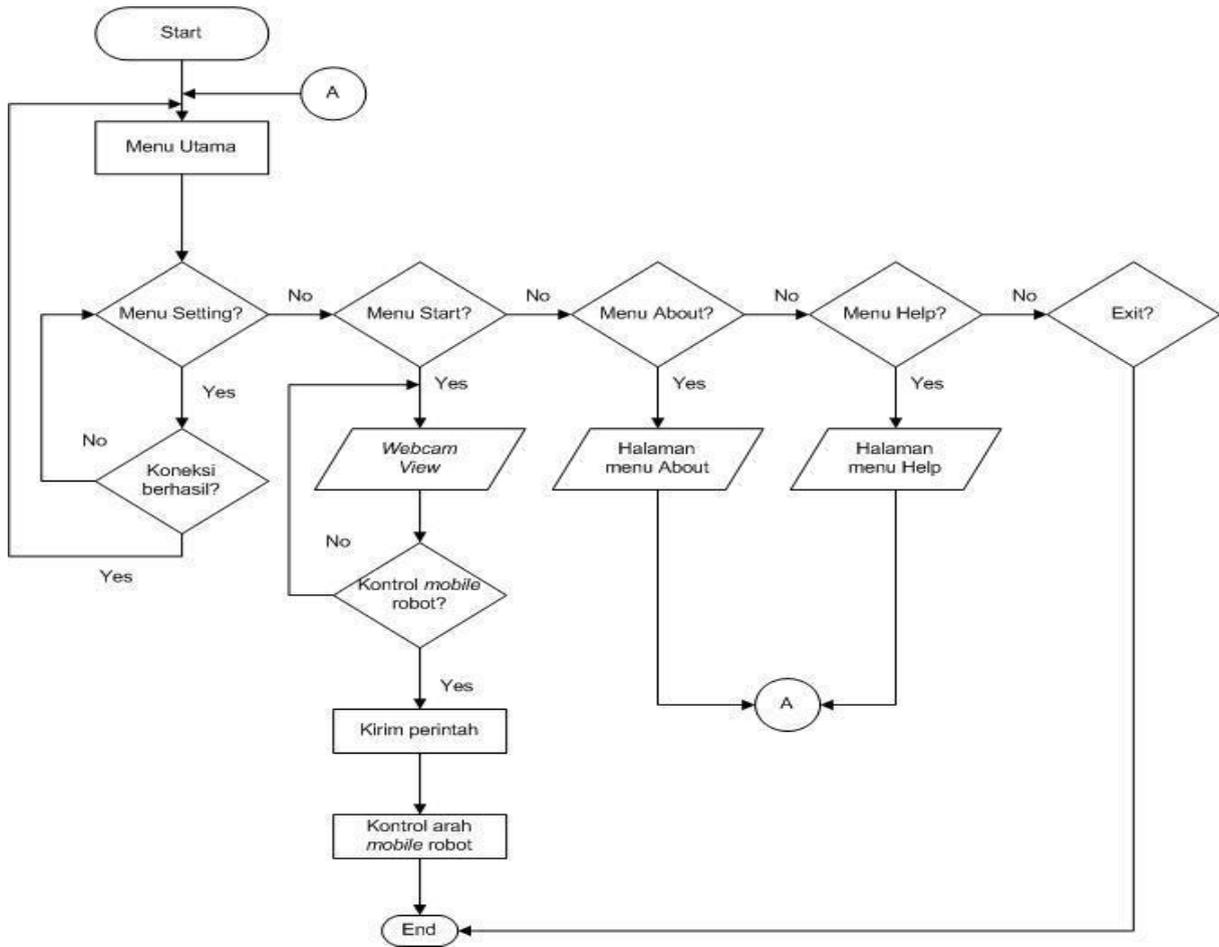
2.1 Desain Sistem

Sistem pada *remote monitoring* ini berbasis android dan terdiri dari kontrol gerak tank, kontrol lampu, *monitoring* pada webcam, *capture* gambar dan kontrol sistem. Setiap bagian memiliki fungsi tersendiri meliputi kontrol gerak tank yang berfungsi untuk mengatur arah tank sesuai dengan masukan *user*. Kontrol lampu berfungsi untuk mengaktifkan fungsi lampu agar kamera dapat menangkap objek ketika berada dilingkungan yang gelap. *capture* gambar berfungsi untuk menangkap objek gambar pada *streaming webcam*. Kontrol sistem berfungsi untuk menonaktifkan sistem. Sedangkan *monitoring* pada webcam berfungsi untuk mengetahui keadaan sekitar melalui kamera yang terletak di tank. Cara kerja sistem *remote monitoring* ini yaitu *user* menggunakan *smartphone* android untuk mengirim perintah melalui koneksi *wifi*. Kemudian *streaming* kamera dan tombol arah muncul dalam interface. *User* dapat melakukan gerak kontrol arah gerak robot.



Gambar 2.1. Gambaran umum sistem

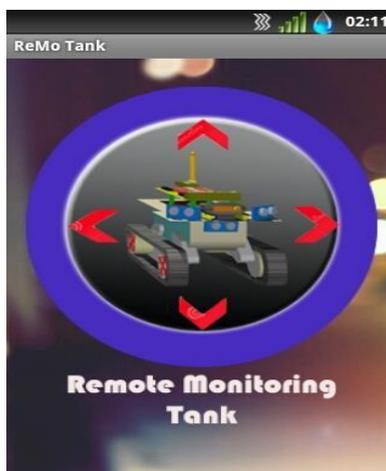
Aplikasi terdiri dari beberapa menu utama seperti *Start*, *Setting*, *About*, *Help* dan *Exit*. Untuk dapat mengaktifkan *sistem monitoring* ini, *user* harus mengaktifkan koneksi *wifi* pada menu *Setting*. Setelah terhubung *user* dapat langsung memilih menu *Start* untuk memulai *remote* dan *sistem monitoring*. Untuk lebih jelasnya dapat dilihat diagram alir dari gambaran umum sistem pada gambar 2.2. berikut.



Gambar 2.2. Diagram alir dari gambaran umum sistem

2.2 Desain Antarmuka

Ketika *user* mengaktifkan sistem *remote monitoring* ini maka akan muncul pertama kali layar pembuka yang berisi informasi nama sistem *monitoring* ini. Seperti pada gambar 2.3 berikut



Gambar 2.3. Desain antarmuka layar pembuka

Desain antarmuka yang telah dirancang terdiri dari tombol menu *start*, *setting*, *about*, dan *help* yang memiliki masing-masing fungsi. Tombol menu *start* merupakan antarmuka ketika *user* ingin melakukan *remote monitoring* dengan terlebih dahulu terhubung dengan koneksi *wifi*. Tombol menu *setting* berfungsi untuk mengaktifkan koneksi *wifi*. Sedangkan tombol menu *about* dan *help* berisi informasi singkat mengenai sistem *remote monitoring* ini beserta tutorial singkat penggunaan aplikasi *remote monitoring* ini.

Berikut desain antarmuka menu utama terdapat pada gambar 2.4 berikut



Gambar 2.4. Desain antarmuka menu utama

Pada bagian menu *remote monitoring* dapat diakses ketika *user* menekan tombol menu *start*. Terdapat bagian *streaming webcam* dan tombol kontrol arah gerak, *capture* gambar dan kontrol lampu LED. Bagian *streaming webcam* berfungsi untuk menampilkan hasil *monitoring* lingkungan sekitar yang dapat ditangkap oleh *webcam*. Sedangkan tombol kontrol arah gerak berfungsi untuk mengarahkan tank pada posisi yang diinginkan *user*. Tombol *capture* berfungsi untuk menangkap objek gambar pada *streaming webcam*. Tombol kontrol lampu atau LED berfungsi untuk memberi penerangan lebih ketika berada dilingkungan yang gelap dan tombol *shutdown* berfungsi untuk menonaktifkan sistem.

Pada gambar 2.5 berikut merupakan desain antarmuka *remote monitoring*



Gambar 2.5. Desain antarmuka *remote monitoring*

2.3 Desain Sistem

Sistem terdiri dari *single board computer* yaitu Raspberry Pi yang memiliki spesifikasi prosesor ARM 11 700 MHz, RAM 512MB, 2 USB port dan 26 pin GPIO (*General Purpose Input Output*) [5]. Raspberry Pi yang digunakan menggunakan sistem operasi raspbian yang berbasis Linux. Perangkat USB kamera *webcam* dan *wifi* adapter dihubungkan pada dua *port* USB. Sedangkan 6 pin GPIO dihubungkan dengan *motor driver* dan motor DC pada Rover 5 robot *chassis*.

2.4 Metode Kontrol

Metode kontrol yang digunakan untuk komunikasi menggunakan koneksi *wifi* yang terdapat pada *wifi* adapter. Konfigurasi pada *wifi* adapter meliputi keamanan WPA pada jaringan dan menjadikan akses poin agar dapat diakses oleh *user* yang menggunakan android. Keamanan jaringan diperlukan agar tidak sembarang orang dapat mengakses kontrol maupun *streaming* pada Raspberry Pi. Selain itu konfigurasi dengan alamat IP *static* dibutuhkan agar *user* dapat berkomunikasi dengan Raspberry Pi. Setelah keduanya dapat berkomunikasi maka perintah *streaming* dan kontrol dapat dilakukan secara *real-time*.

Pada gambar 2.6 berikut merupakan *screenshot* hasil *ping* dari *user* dengan IP 192.168.43.201 dengan *server* yang terdapat pada *mobile* robot dengan IP 192.168.43.200. Hasil ini diperoleh dari aplikasi *terminal emulator* yang terpasang pada *smartphone* android. Dengan perintah *ping -c 5 192.168.43.200* (IP *server*) pada *terminal emulator*. *Ping* jaringan dilakukan sebanyak 6 kali percobaan dengan jumlah *ping* berbeda yaitu 5 kali *ping* , 10 kali *ping* , 15 kali *ping* , 20 kali *ping* , 25 kali *ping* , 30 kali *ping* . Diperoleh hasil *ping* 0% *packet loss* selama 6 kali percobaan.



```

64 bytes from 192.168.43.200: icmp_seq=20 ttl=64 time
=5.56 ms
64 bytes from 192.168.43.200: icmp_seq=21 ttl=64 time
=9.11 ms
64 bytes from 192.168.43.200: icmp_seq=22 ttl=64 time
=5.63 ms
64 bytes from 192.168.43.200: icmp_seq=23 ttl=64 time
=6.35 ms
64 bytes from 192.168.43.200: icmp_seq=24 ttl=64 time
=6.50 ms
64 bytes from 192.168.43.200: icmp_seq=25 ttl=64 time
=5.52 ms
--- 192.168.43.200 ping statistics ---
25 packets transmitted, 25 received, 0% packet loss,
time 24030ms
rtt: min/avg/max/mdev = 5.151/6.165/9.110/0.954 ms
#

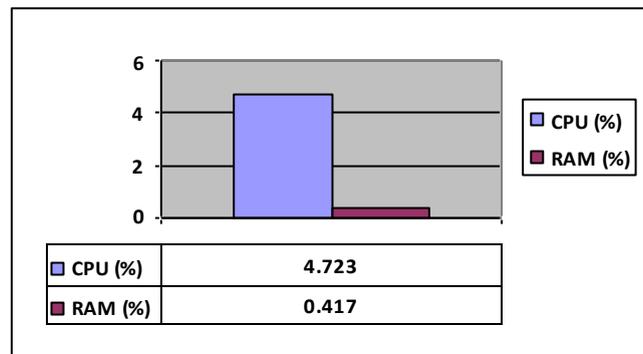
```

Gambar 2.6. *Screenshot* hasil *ping* dari perangkat android ke *server*

3. Hasil Pengujian

Dalam pengujian ini terbagi menjadi tiga bagian. Pengujian yang pertama meliputi ketahanan perangkat pada Raspberry Pi. Sedangkan pengujian kedua meliputi ketahanan perangkat pada android. Pengujian terakhir adalah pengujian koneksi *wifi* dari perangkat android ke *server* pada Raspberry Pi. Pengujian ketahanan perangkat berfungsi untuk mengetahui seberapa besar penggunaan RAM dan CPU ketika program dijalankan. Sedangkan pengujian koneksi *wifi* berfungsi untuk mengetahui apakah android dan *server* pada Raspberry Pi terhubung ketika *wifi* pada perangkat android telah aktif dan terkoneksi dengan SSID *server*.

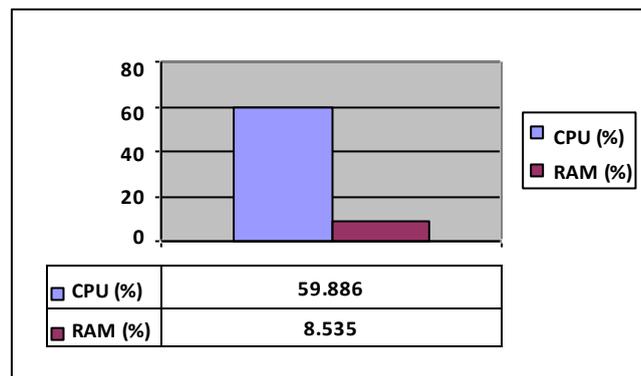
3.1 Pengujian ketahanan perangkat Raspberry Pi.



Gambar 3.1. Grafik rata-rata penggunaan CPU (%) dan RAM (%) Raspberry Pi

Pengujian ini berfungsi untuk mengetahui penggunaan CPU dan RAM ketika mengeksekusi program *streaming webcam*. Hasil pengujian ketahanan Raspberry Pi diperoleh hasil penggunaan CPU sebesar 4.723% dan penggunaan RAM sebesar 0.417% dari percobaan sebanyak 35 kali. Hasil pengujian ini diperoleh dari program *terminal* pada Raspberry Pi. Dengan menggunakan perintah *top* untuk melihat penggunaan CPU dan RAM ketika program *streaming* dijalankan. Penggunaan CPU dan RAM ini dipengaruhi dari konfigurasi program *streaming*, *driver webcam* yang digunakan dan *background process* ketika program *streaming* dijalankan.

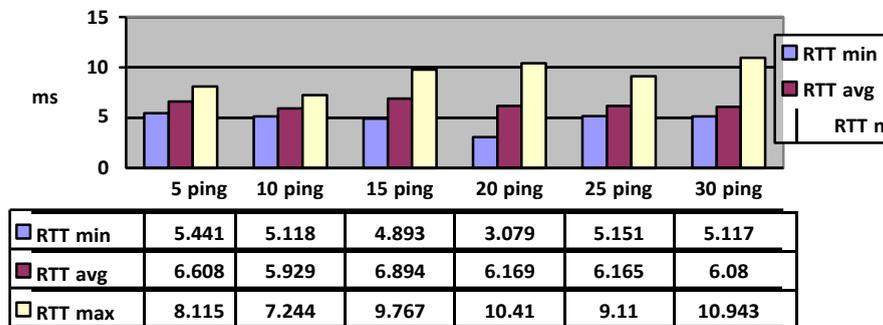
3.2 Pengujian perangkat Smartphone Android.



Gambar 3.2. Grafik rata-rata penggunaan CPU (%) dan RAM (%) pada Android

Pengujian ini berfungsi untuk mengetahui penggunaan CPU dan RAM ketika program Remo tank di jalankan pada *smartphone* android. Pengujian ini menggunakan *smartphone* android dengan sistem operasi *Gingerbread (2.3)*. Hasil pengujian ketahanan android diperoleh hasil penggunaan CPU sebesar 59.886% dan penggunaan RAM sebesar 8.535% dari percobaan sebanyak 35 kali. Hasil penggunaan CPU dan RAM ini diperoleh dengan menggunakan aplikasi *terminal emulator* pada *smartphone* android. Dengan menggunakan perintah *top* untuk melihat penggunaan CPU dan RAM selama aplikasi Remo tank dijalankan. Hasil penggunaan CPU dan RAM ini dipengaruhi dari spesifikasi perangkat *smartphone* android, konfigurasi program pada *server* atau Raspberry Pi dan *background process* selama pengujian berlangsung.

3.3 Pengujian koneksi *wifi* .



Gambar 3.3. Grafik hasil *ping* dari perangkat android ke *server*

Pengujian ini berfungsi untuk mengetahui koneksi *wifi* dari perangkat android ke *server* pada Raspberry Pi. Pada pengujian dilakukan dengan memberi perintah *ping* (IP *server*) pada *terminal emulator* android. Sebanyak enam kali pengujian dengan *ping* paling sedikit 5 kali dan paling banyak 30 kali diperoleh hasil 0% *packet loss*. Hasil pengujian ini diperoleh dari aplikasi *terminal emulator* yang terdapat pada *smartphone* android. Hasil pengujian koneksi *wifi* ini dipengaruhi oleh jarak antara *user* pada *smartphone* android dengan *server* pada Raspberry Pi, keadaan trafik jaringan ketika dilakukan pengujian dan spesifikasi dari perangkat *wifi* adapter yang terdapat pada *server* atau Raspberry Pi.

4. Kesimpulan

Kesimpulan dari hasil pengujian diatas dapat disimpulkan bahwa penggunaan CPU dan RAM pada perangkat Raspberry Pi ketika mengeksekusi program *streaming* sebesar 4.723% untuk CPU dan penggunaan RAM sebesar 0.417%. Hasil pengujian ketahanan perangkat Raspberry Pi sangat dipengaruhi oleh jenis perangkat *webcam* dan *driver* yang digunakan. Selain itu dipengaruhi oleh pengaturan *FPS* (*Frame Per Second*) dan resolusi kamera *webcam* pada software *streaming* yang digunakan.

Pengujian selanjutnya adalah pengujian penggunaan CPU dan RAM pada *smartphone* android ketika mengeksekusi program Remo tank sebesar 59.886% untuk CPU dan penggunaan RAM sebesar 8.535%. Hasil pengujian ini dipengaruhi oleh spesifikasi *smartphone* android.

Setelah hasil pengujian koneksi *wifi* dilakukan, diperoleh hasil bahwa koneksi dari perangkat android ke *server* yang terdapat pada Raspberry Pi berhasil. Hasil ini dapat disimpulkan dari test *ping* dengan jumlah *ping* berbeda dan tiap test *ping* diperoleh 0% *packet loss*. Pengujian ini dipengaruhi salah satunya dari jarak antara perangkat android dengan *server* yang berbeda-beda.

Diharapkan dengan adanya sistem *monitoring* ini dapat dikembangkan baik dalam bidang militer, penanganan bencana maupun penggunaan pribadi. Mengingat sistem ini tidak membutuhkan biaya yang besar dibandingkan dengan sistem lain. Selain itu sistem ini mudah digunakan oleh banyak orang.

Daftar Pustaka:

- [1] C. B. Manjunath and Badrinath, "Design and Implementation of Cost Effective Surveillance Robot," *IEEE*, pp. 141-145, 2014.
- [2] W. F. Ableson, R. Sen, C. King and C. E. Ortiz, *Android in Action Third Edition*, New York: Manning Publications Co., 2012.
- [3] W. W. Gay, *Raspberry Pi Hardware Reference*, New York: Apress, 2014.
- [4] M. Schmidt, *Raspberry Pi A Quick-Start Guide*, USA: The Pragmatic Programmers, 2012.
- [5] Broadcom, "Broadcom Corporation," 6 Februari 2012. [Online]. Available: <http://www.raspberrypi.org/wp-content/uploads/2012/02/BCM2835-ARM-Peripherals.pdf>. [Accessed 29 Desember 2014].
- [6] X. Z. Y. T. Letian Liu, "Indoor Surveillance Robot Controlled by a Smartphone," *IEEE*, pp. 1875-1880, 2013.
- [7] J. L. G. T. J. K. SeokJu Lee, "Autonomous Tour Guide Robot by using Ultrasonic Range Sensors and QR code Recognition in Indoor Environment," *IEEE*, pp. 410-415, 2014.
- [8] S.-S. K. T.-K. Y. Yong-Ho Seo, "Mobile Robot Control Using Smart Phone and Its Performance Evaluation," *Springer*, vol. 199, pp. 362-369, 2011.
- [9] Y. J. K. H. J. M. C. S. K. Sung Wook Moon, "Implementation of smartphone environment remote control and monitoring system for Android operating system-based robot platform," *IEEE*, pp. 211-214, 2011.
- [10] R. R. Murphy, *Introduction to AI Robotics*, London : Massachusetts Institute of Technology, 2000.