

APLIKASI PENGONTROLAN ROBOT PEMANTAU DARI UDARA BERBASIS ANDROID

Jacqueline M.S. Waworundeng*¹, Hendra Sayago²

^{1,2}Universitas Klabat; Jl. Arnold Mononutu Airmadidi, 0431-891035

³Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Klabat, Airmadidi
e-mail: *¹jacqueline.morlav@unklab.ac.id, ²hendra_zerohunter@hotmail.com

Abstrak

Penelitian ini bertujuan untuk membuat Aplikasi pengontrolan robot pemantau dari udara berbasis Android untuk memantau daerah yang berbahaya atau daerah dengan ketinggian tertentu yang sulit dijangkau oleh manusia. Hardware yang digunakan berupa Helicopter R/C yang dimodifikasi dengan menambahkan kamera IP serta remote control yang terhubung dengan Arduino Uno dan Arduino Ethernet Shield. Sistem ini dilengkapi dengan aplikasi pengontrolan robot dan kamera untuk melakukan pemantauan dan proses pengambilan gambar dengan menggunakan jaringan wireless. Jaringan wireless berfungsi sebagai media transmisi data berupa gambar dan dapat menampilkan hasil pemantauan object kepada user, melalui layar tablet-PC yang menggunakan sistem operasi mobile berbasis Android. Metode pengembangan yang digunakan dalam pembuatan aplikasi pengontrolan ini adalah metode prototype, dengan bahasa pemrograman untuk pengembangan aplikasi yaitu Java Android dan bahasa pemrograman C yang digunakan untuk pengembangan perangkat pengontrol. Hardware dan software dalam penelitian ini dapat berfungsi sesuai dengan rancangan. Dengan aplikasi pengontrolan robot pemantau dari udara berbasis android ini dapat membantu user untuk melakukan observasi di suatu daerah.

Kata kunci—Aplikasi Pengontrolan, Robot Pemantau, Android, Arduino, Wireless

Abstract

The aim of this research is to make application of aerial surveillance robot control based on Android to monitor hazardous area or areas with height which difficult to reach by human. The hardware used in this system are R/C Helicopter modified by adding an IP camera, a remote control that connected to Arduino Uno and Arduino Ethernet Shield. The system equipped with a robot and camera control application that can monitor and view object by using a wireless network. Wireless network serves as a medium for data transmission such as pictures and can display the monitoring results to the user, via tablet-PC screen that use Anroid as a mobile operating system. The development method used for this research is a prototyping model. The programming language use Java application development on Android and the C programming language to develop devices control. The hardware and software can perform well, according to the design. By controlling the application of air surveillance robot based Android can help user to make observations in particular area.

Keywords—Control Application, Surveilence Robot, Android, Arduino, Wireless.

1. PENDAHULUAN

Perkembangan teknologi sangat berpengaruh terhadap aktivitas sehari-hari dari manusia. Teknologi telah digunakan dalam berbagai bidang seperti pertanian, pendidikan, militer, dan keamanan. Contohnya teknologi robot yang terus dikembangkan untuk membantu manusia dalam melakukan aktivitas kerja. Pada saat ini jenis-jenis robot telah banyak diciptakan dengan fungsi-fungsi khusus. Seperti dalam penelitian ini peneliti merancang robot pemantau dari udara menggunakan *Helicopter Remote Control (R/C)* yang telah dimodifikasi dan dilengkapi dengan kamera serta aplikasi pengontrolan dari *tablet-PC*. Fungsi khusus dalam sistem ini yaitu dapat melakukan pemantauan dari udara pada ketinggian tertentu.

Rumusan masalah dalam penelitian ini yaitu bagaimana membuat suatu aplikasi pengontrolan robot pemantau dari udara untuk melakukan pengambilan gambar, dengan menggunakan jaringan *wireless* sebagai media perantara pengiriman datanya dan menggunakan operating sistem berbasis android.

Tujuan dari penelitian ini untuk merancang suatu aplikasi pengontrolan robot dan monitoring melalui *wireless camera*. Hasil *monitoring* dari kamera dapat dilihat secara langsung oleh user melalui *tablet-PC* yang menggunakan operating sistem berbasis android.

1.1 Operating System

Operating system adalah suatu program yang mengontrol eksekusi dari program aplikasi dan bertindak sebagai antarmuka antara aplikasi dengan *hardware* komputer. *Hardware* dan *software* digunakan untuk menyediakan aplikasi-aplikasi kepada user [1].

Android adalah *operating system* untuk *mobile device*, dimana Android adalah sistem operasi yang tidak membedakan aplikasi inti dengan aplikasi pihak ketiga. Application Programming Interface (API) yang disediakan juga menawarkan akses ke *hardware*, maupun data ponsel, atau data sistem, bahkan pengguna dapat menghapus aplikasi inti dan mengganti aplikasi dari pihak ketiga [2].

Fitur dan Arsitektur Android mempunyai 8 fitur yaitu:

- 1) *Framework* aplikasi : memungkinkan penggunaan dan pemindahan dari komponen yang telah tersedia.
- 2) *Dalvik virtual machine* : pengoptimalan *virtual machine* untuk perangkat *mobile*.
- 3) Grafik : grafik 2D dan grafik 3D yang didasarkan pada *library OpenGL*.
- 4) *SQLite*: untuk proses penyimpanan data.
- 5) Mendukung *media* : *audio, video*, dan berbagai *format* gambar.
- 6) *GSM, Bluetooth, EDGE, 3G*, dan *WiFi*
- 7) *Camera, Global Positioning System (GPS), compass, dan accelerometer*
- 8) Lingkungan pengembangan yang kaya, termasuk *emulator*, peralatan *debugging*, dan *plugin* untuk eclipse *IDE*.

Operating system yang digunakan untuk menjalankan aplikasi pengontrolan robot pemantau dari udara adalah Android versi 3.2

1.2 Arduino

Mikrokontroler pada dasarnya adalah sebuah komputer chip tunggal (*single chip*). *Single chip* berarti seluruh sistem komputer berada dalam sebuah silikon yang dienkapsulasi dalam sebuah *board* circuit yang saling terhubung dan dapat menyimpan dan menjalankan program [3].

Arduino merupakan komputer mini yang dapat diprogram untuk memproses input maupun output yang keluar dan masuk dalam chip. *Arduino* dikenal sebagai *platform* komputasi fisik yang berinteraksi lingkungan *hardware* maupun *software*. *Arduino* dapat dihubungkan ke komputer untuk mengambil atau mengirim data [4].

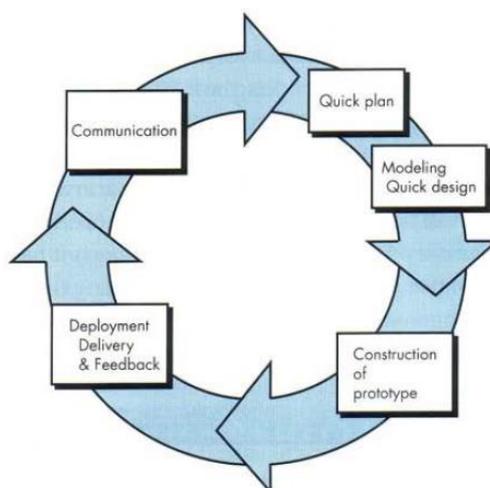
Dalam penelitian ini, tipe *Arduino* yang digunakan adalah *Arduino Uno* dan *Arduino Ethernet Shield* yang di program dengan software *Arduino Integrated Development Environment*. *Arduino Uno* terintegrasi dengan *Arduino Ethernet Shield* sehingga dapat terhubung dengan jaringan *wireless*. *Arduino Ethernet Shield* mempunyai *Ethernet* port dan *micro-SD* yang menjadi media penyimpanan data berupa gambar.

2. METODE PENELITIAN

Metode penelitian mencakup desain penelitian, kerangka konseptual sistem, perancangan *hardware*, dan perancangan *software* aplikasi pengontrolan berbasis Android.

2.1 Desain Penelitian

Desain penelitian menggunakan model *Prototyping* seperti pada Gambar 1.



Gambar 1. Model Prototyping [5]

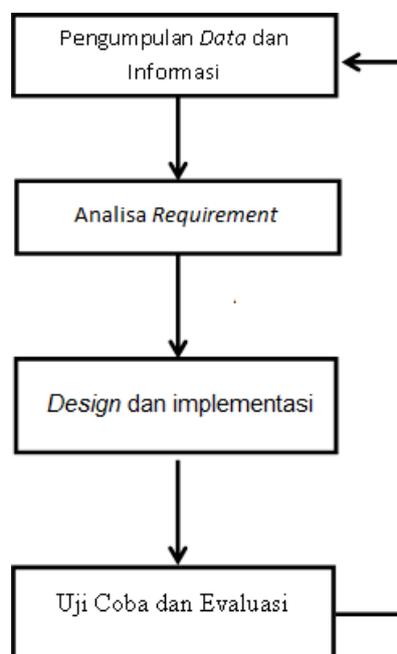
Proses pada model *Prototyping* dapat dijelaskan sebagai berikut:

1. *Communication*, tahap ini dilakukan dengan tujuan untuk mengumpulkan data yang dibutuhkan dalam membangun aplikasi. Komunikasi terjadi melalui hubungan langsung antara pengguna dan peneliti untuk membangun aplikasi. Kebutuhan diidentifikasi untuk mencari informasi dan *tools* yang dibutuhkan untuk membantu pembuatan aplikasi.
2. *Quick Plan*, tahap ini merupakan tahap perencanaan yang cepat karena semua data dan tools yang akan digunakan sudah diidentifikasi pada tahap sebelumnya.
3. *Modeling Quick Design*, Tahap ini merupakan tahap desain dan analisis aplikasi dengan cepat karena telah dilakukan perencanaan dan identifikasi kebutuhan pengguna pada tahap-tahap sebelumnya.
4. *Construction of Prototype*, tahap ini adalah tahap pembuatan prototipe dari aplikasi. Prototipe yang dimaksud merupakan konsep pembangunan aplikasi yang sederhana. Tahap ini juga disebut tahap implementasi aplikasi, dimana hasil analisis dan desain akan ditranslasikan ke dalam bahasa pemrograman.
5. *Deployment Delivery and Feedback*, tahap ini adalah tahap dimana aplikasi disebarkan ke tangan pengguna kemudian pengguna mengevaluasi dan memberikan *feedback* terhadap aplikasi yang telah dibuat. *Feedback* diberikan secara langsung dan akan digunakan pada pengembangan aplikasi atau iterasi selanjutnya.

Model *Prototyping* menjadi acuan dalam menyusun konsep penelitian aplikasi pengontrolan robot pemantau dari udara berbasis Android. Gambar 2 memperlihatkan tahapan kerangka penelitian yang dijelaskan sebagai berikut :

1. Pengumpulan Data dan Informasi
Dalam tahap ini peneliti mengumpulkan data dan informasi berupa referensi mengenai pembangunan aplikasi dan *hardware* yang dapat digunakan.
2. Analisa *Requirements*
Dalam tahap ini peneliti akan melakukan analisa *requirement* yang telah di dapat pada tahap pertama yaitu pengumpulan data dan informasi.
3. *Design* dan Implementasi
Tahap peneliti setelah menganalisa *requirement* yang di dapat, maka peneliti akan merancang aplikasi yang akan di gunakan, dari hasil desain tersebut, peneliti akan mengimplementasikannya, dengan membuat *prototype*. Dalam proses pembuatan *prototype* terdapat beberapa proses pendukung yaitu proses penghubungan antara komponen-komponen, dan proses pembuatan aplikasi yang nantinya akan mengontrol robot tersebut. Setelah itu proses koneksi antara *hardware* dan aplikasi agar *hardware* tersebut bisa dikendalikan
4. Uji coba dan Evaluasi
Tahap selanjutnya sesudah pembuatan *prototype* adalah tahap pengujian dan evaluasi. Dalam tahap ini peneliti menguji *prototype* yang telah di buat. Dari hasil pengujian itulah peneliti dapat mengetahui jika ada perangkat yang tidak sesuai atau terdapat *error* pada aplikasi yang dibuat. Jika terdapat *error* peneliti mencari penyebab *error* serta solusinya. Jika hasil *prototype* belum sesuai dengan *requirement* maka peneliti melakukan perbaikan sampai hasilnya sesuai dengan *requirement* yang ada.

Ketika proses pengujian dan evaluasi selesai dilakukan, maka dihasilkan suatu *prototype* robot pemantau dari udara dan aplikasi pengontrolan robot berbasis Android.

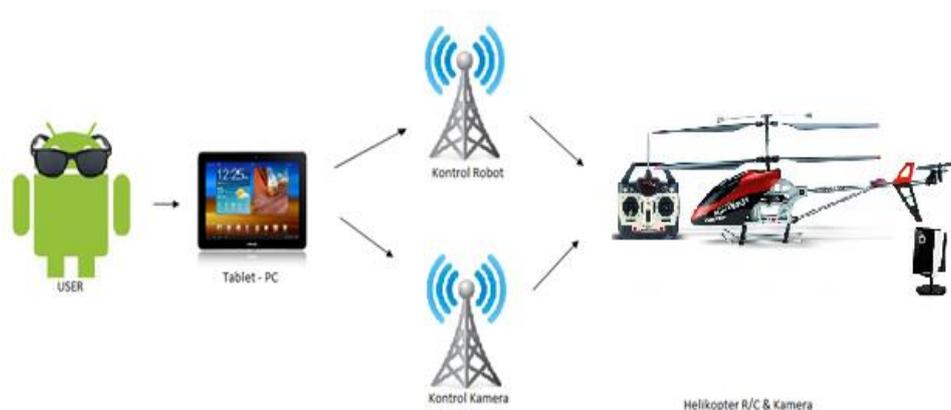


Gambar 2. Kerangka penelitian

2.2 Kerangka Konseptual Sistem

Kerangka konseptual sistem yang dibangun ditunjukkan pada Gambar 3. Sistem ini dibuat untuk pengontrolan robot dan pengontrolan kamera melalui aplikasi berbasis Android. *Hardware* robot pemantau udara dibuat dengan memodifikasi *Helicopter* (R/C) 3 channel dengan menambahkan kamera IP. *Helicopter* yang digunakan adalah jenis *helicopter* R/C 3 channel yang dapat bergerak naik atau turun, berputar ke kiri atau ke kanan serta dapat bergerak maju atau mundur. *Remote Control* dari *Helicopter* R/C 3 channel juga dimodifikasi dengan menghubungkannya dengan *Arduino board* yaitu *Arduino Uno* dan *Arduino Ethernet Shield*.

Aplikasi pengontrolan robot dan kamera terinstal pada *tablet-PC*, dan user dapat mengakses aplikasi dan melakukan pengontrolan robot *helicopter* R/C dan kamera dalam jaringan *wireless*. Gambar 3 menunjukkan kerangka konseptual sistem yang dirancang.



Gambar 3. Kerangka Konseptual Sistem

Cara kerja sistem ini berawal dari *user* yang menjalankan aplikasi pengontrolan robot berbasis Android yang terinstal di *tablet-PC*. Aplikasi tersebut terhubung dengan *Arduino* dan kamera IP dengan menggunakan *wireless router*. Setelah semuanya terhubung maka user dapat mengendalikan remote kontrol helikopter dan kamera IP melalui aplikasi di *tablet-PC*. Aplikasi android mengontrol robot dan mengontrol kamera dalam jaringan *wireless*. Setelah proses penghubungan *tablet-PC*, robot dan kamera melalui jaringan *wireless* terkoneksi maka robot dan kamera telah dapat di kontrol melalui *tablet-PC*.

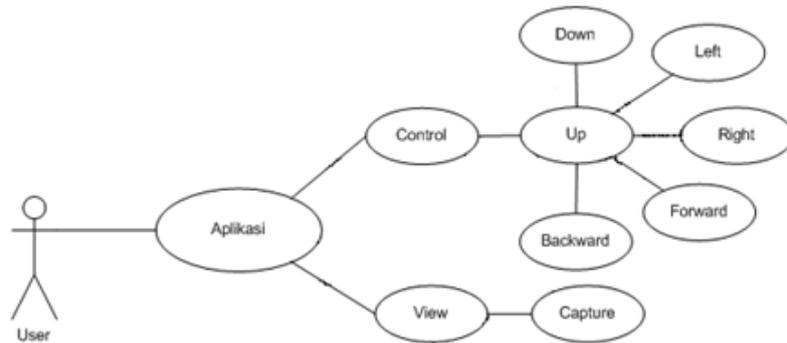
2.3 Perancangan Software Sistem

Analisis aplikasi yang digunakan peneliti pada penelitian ini yaitu menggunakan *Unified Modelling Language* (UML) yang merupakan pemodelan standar dalam industri *software* untuk visualisasi, perancangan, dan pedokumentasian suatu *software*.

Gambar 4 menunjukkan *Use Case Diagram* aplikasi pengontrolan. Use case diagram terdiri atas *actor* yaitu *user*. *Precondition* adalah kondisi dimana *user* belum bisa mengontrol robot helikopter dan kamera dengan menggunakan aplikasi. Sedangkan *Postcondition*, adalah kondisi dimana *user* telah berhasil mengontrol helikopter atau kamera. Tahapan yang dilakukan dijelaskan sebagai berikut :

1. *User* mengaktifkan *power* dari robot, *wireless router*, dan *IP camera*.
2. *User* melakukan pengaturan awal untuk *wireless router* dan *IP camera*.
3. *User* membuka aplikasi dari *tablet-PC*

4. Pada saat *user* membuka aplikasi maka akan tampil *menu* agar user dapat menentukan pilihan untuk mengakses aplikasi atau keluar dari aplikasi.
5. *Menu* akan ditampilkan pada layar untuk memasukkan alamat *URL* dari *controller* dan *IP camera*.
6. Jika salah satu atau kedua koneksi tidak berhasil maka fitur untuk melakukan kontrol dan *view camera* tidak dapat dijalankan.
7. Jika tidak tampil pesan error maka helikopter telah dapat di kontrol, dimana helikopter harus dalam kondisi *up* terlebih dahulu lalu diikuti dengan kontrol yang lain seperti *down*, *left*, *right*, *forward*, dan *backward*. Selain dapat mengendalikan helikopter, user dapat melakukan *capture* dan *record* dari *view* kamera yang di dapat.



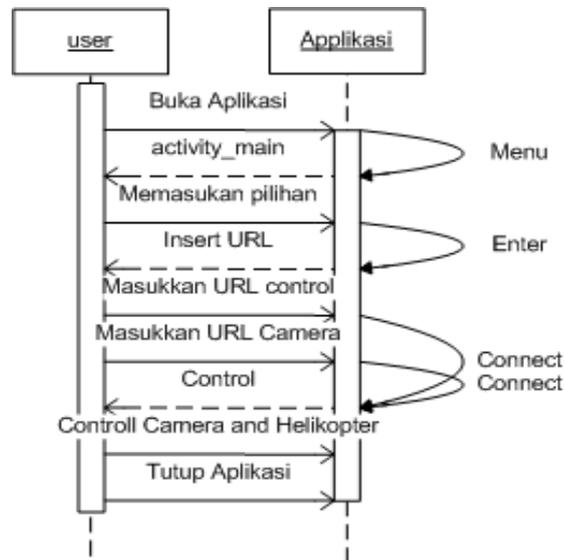
Gambar 4 Use Case Diagram

Gambar 5 memperlihatkan Class Diagram *MainActivity*. *Class Diagram* digunakan di dalam pembuatan aplikasi ini berhubungan dengan cara pemrograman berbasis *Object Oriented Progaming (OOP)*, dimana *class* diagram adalah diagram yang menggambarkan *class*, metode serta atribut yang digunakan dalam pengembangan aplikasi.



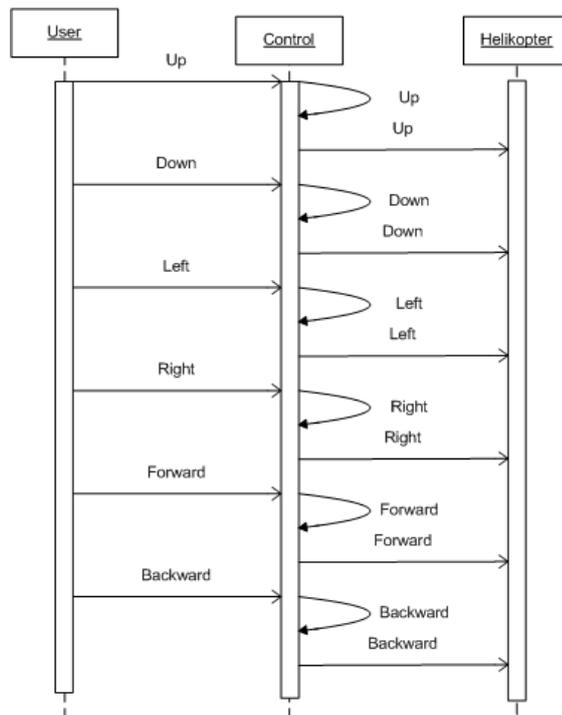
Gambar 5 Class Diagram MainActivity

Sequence Diagram adalah diagram yang akan menggambarkan proses interaksi antar *object* yang dimana, *object* tersebut akan melakukan proses eksekusi instruksi yang diberikan oleh *user*.



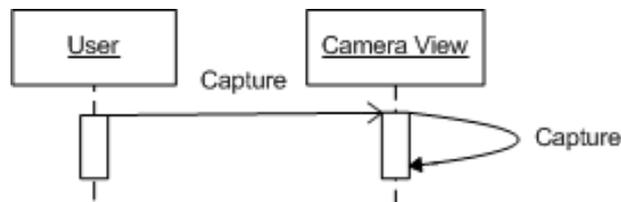
Gambar 6. *Sequence Diagram* User ke Aplikasi

Gambar 6 menunjukkan *sequence diagram* user ke aplikasi. Langkah awal adalah *user* membuka aplikasi. Setelah aplikasi terbuka maka aplikasi akan menampilkan *menu* dan menampilkan proses pengimputan URL untuk control dan camera. Setelah pengimputan URL selesai maka proses selanjutnya yaitu tampil view untuk control dan view dari kamera. Setelah selesai menggunakan aplikasi user dapat keluar dari aplikasi tersebut.



Gambar 7. *Sequence Diagram User, Remote Control, Robot Helicopter*

Gambar 7 menunjukkan *sequence diagram user, remote control* dan pengontrolan gerakan robot *helicopter* dengan variasi gerakan *up, down, left, right, forward* dan *backward*.

Gambar 8. *Sequence Diagram User ke Camera*

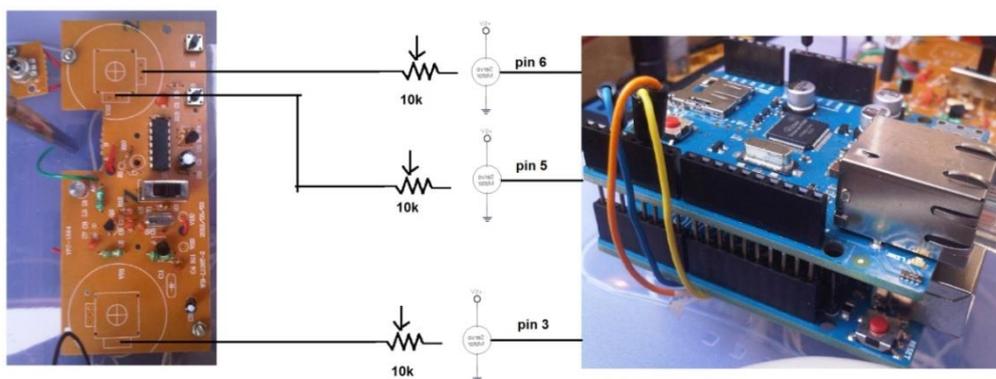
Gambar 8 menjelaskan proses interaksi *user* dengan kamera yang melalui aplikasi. Dimana menjelaskan bahwa kamera harus dinyalakan terlebih dahulu oleh *user*, setelah itu *user* akan mendapatkan *view* dari kamera, setelah mendapatkan *view* dari kamera *user* dapat melakukan proses *capture*.

3. HASIL DAN PEMBAHASAN

Hasil dan pembahasan meliputi implementasi *hardware* dan *software* serta hasil pengujian.

3.1 Implementasi Hardware Remote Control

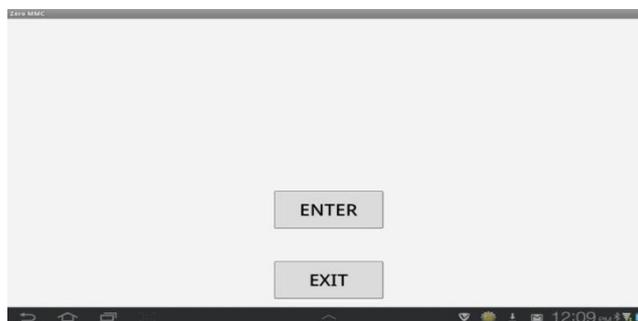
Implementasi *hardware* dilakukan dengan memodifikasi *remote control helicopter* dengan menambahkan *Arduino Uno* dan *Arduino Ethernet Shield*. Perangkat pengontrolan dibuat dengan menghubungkan *remote control, potentiometer* dan *Arduino board*. Perangkat ini akan terkoneksi dalam jaringan *wireless* dengan aplikasi pada *tablet-PC*. Gambar 9 menunjukkan modifikasi *remote control robot helicopter* yang terhubung dengan pin 3, 4, 6 pada *Arduino*.

Gambar 9. Modifikasi *Remote Control Helicopter* dan *Arduino*

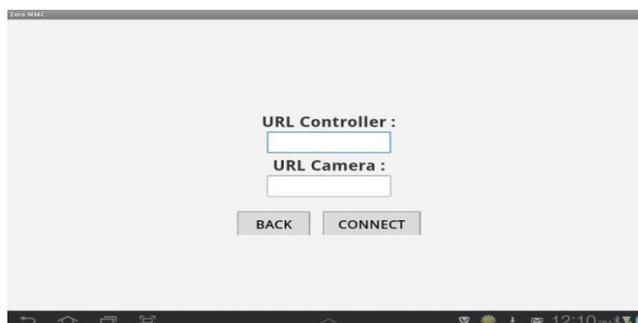
3.2 Implementasi Software

Implementasi *software* berupa tampilan antarmuka aplikasi dengan *user*. Aplikasi yang dibuat berbasis *Android* dan dioperasikan dengan menggunakan *tablet-PC Samsung Galaxy Tab*

10. Gambar 10 merupakan *form* awal untuk melakukan pemilihan apakah *user* ingin mengakses aplikasi atau keluar dari aplikasi pengontrolan berbasis Android.

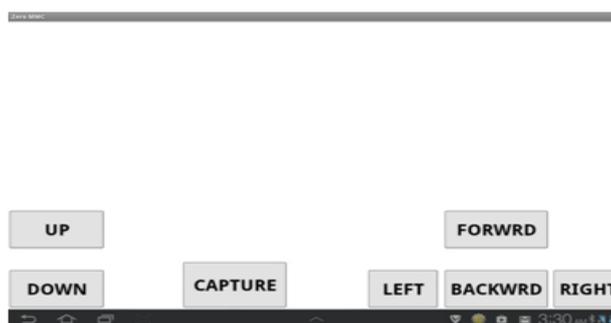


Gambar 10. Implementasi Menu Utama



Gambar 11. Implementasi Input *URL*

Gambar 11 merupakan tampilan *form* yang di tampilkan ketika *user* menekan *ENTER* di tampilan menu utama.



Gambar 12 Implementasi Pengontrolan

Gambar 12 merupakan tampilan yang muncul ketika *user* menekan tombol *connect* ada tampilan mengisi *URL*. Fungsi pengontrolan dapat menggerakkan robot pemantau dari udara untuk bergerak *up*, *down*, *forward*, *backward*, *left* dan *right*. Sedangkan pengontrolan kamera dapat melakukan *capture* gambar.

3.3 Pengujian

Pada penelitian ini peneliti menggunakan metode pengujian *blackbox testing*. Metode ini memungkinkan user memberikan input dan menghasilkan output. Output tersebut yang diamati jika sesuai dengan yang diharapkan atau tidak.

Tabel 5.1 Tabel pengujian *menu awal aplikasi*

| Input | Action | Output yang di inginkan | Hasil test |
|---------------------|-----------------------------|------------------------------------|------------|
| <i>Button Enter</i> | 1. Klik tombol <i>ENTER</i> | 1. Masuk ke <i>form insert URL</i> | OK |
| <i>Button Exit</i> | 2. Klik tombol <i>EXIT</i> | 1. Keluar dari aplikasi | OK |

Tabel 5.2 Tabel pengujian *insert URL*

| Input | Action | Output yang di inginkan | Hasil test |
|-----------------------|--|---|------------|
| <i>Button Connect</i> | 1. <i>Insert URL</i> kamera (192.168.0.100) 2. <i>Insert URL control</i> (192.168.0.104) 3. Klik tombol <i>CONNECT</i> | 1. Masuk ke <i>form control</i> 2. <i>View</i> kamera tampil 3. <i>Control</i> tidak <i>error</i> | OK |
| <i>Button Back</i> | 1. Klik <i>BACK</i> | 1. Kembali ke <i>form</i> utama | OK |

Tabel 5.3 Tabel pengujian *control*

| Input | Action | Output yang di inginkan | Hasil Test |
|------------------------|-----------------------------|-------------------------|------------|
| <i>Button Up</i> | Klik tombol <i>UP</i> | Bergerak ke atas | OK |
| <i>Button Down</i> | Klik tombol <i>DOWN</i> | Bergerak ke bawah | OK |
| <i>Button Left</i> | Klik tombol <i>LEFT</i> | Bergerak ke kiri | OK |
| <i>Button Right</i> | Klik tombol <i>RIGHT</i> | Bergerak ke kanan | OK |
| <i>Button Forward</i> | Klik tombol <i>FORWARD</i> | Bergerak ke depan | OK |
| <i>Button Backward</i> | Klik tombol <i>BACKWARD</i> | Bergerak ke belakang | OK |
| <i>Button Capture</i> | Klik tombol <i>CAPTURE</i> | <i>Capture</i> gambar | OK |

Hasil pengujian menunjukkan bahwa semua fungsi yang dirancang dapat terpenuhi untuk melakukan pengontrolan robot dan kamera melalui aplikasi berbasis Android.

4. KESIMPULAN

Dari penelitian mengenai aplikasi pengontrolan robot pemantau dari udara berbasis android maka diperoleh kesimpulan :

1. Pemantauan dapat dilakukan dengan memodifikasi *Helicopter* R/C sebagai robot pemantau dari udara.

2. Robot pemantau dari udara dapat dibuat dengan memodifikasi *Helicopter* R/C secara *wireless* dengan menggunakan *WiFi* yang dapat melakukan pemantauan serta pengambilan gambar.
3. Robot pemantau dari udara dapat di kontrol melalui aplikasi berbasis Android dengan menggunakan *Arduino Ethernet Shield* sebagai *module* untuk *Arduino Uno* yang menghubungkan *Arduino Uno* dengan *wireless router*. Jika menggunakan *Arduino Ethernet Shield* dan di kendalikan secara *wireless* menggunakan *WiFi* maka terdapat kendala dalam proses pengiriman data dari android ke *Arduino*. Dalam hal ini *Arduino* memproses instruksi tersebut satu detik atau lebih setelah *action* dilakukan, dan selain itu jika hal tersebut di implementasikan pada helikopter maka akan sangat berbahaya untuk helikopter tersebut.
4. Hasil pengujian menunjukkan bahwa fungsi pengontrolan robot maupun kamera berhasil sesuai dengan tujuan.

5. SARAN

Pengembangan penelitian dapat dilakukan dengan menambahkan fitur recording video dari IP Camera. Dalam penggunaan power supply dari kamera sebaiknya terpisah dengan robot *helicopter* agar dapat dioperasikan dalam waktu yang optimal.

DAFTAR PUSTAKA

- [1] Stallings, W., 2012, *Oprating System Internals and Design Principles*, Ed. 7, Prentice Hall, New Jersey.
- [2] Stephanus, H. S., 2011. *Mudah Membuat Aplikasi Android*. Yogyakarta: ANDI.
- [3] Iovine J., 2004, *PIC Robotics - A Beginner's Guide to Robotics Projects Using the PIC Micro*, McGraw Hill, New York.
- [4] Roberts, M., 2009, *A Complete Beginner Guide to Arduino*, Electronic Earthshine.
- [5] Pressman, R. S., 2009, *Software Engineering: A Practitioner's Approach*, Ed.7, McGraw-Hill, New York.