

## RANCANG BANGUN DAN IMPLEMENTASI PATH BUILDER PADA QUADCOPTER

### Design and implementation mapping of certain waypoint on quadcopter

Azmi Rasyid<sup>1</sup>, Ir. Porman Pangaribuan<sup>2</sup>, Ramdhan Nugraha Spd.,MT.<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Teknik Elektro, Fakultas Teknik, Universitas Telkom

<sup>1</sup>[azmirasyid@students.telkomuniversity.ac.id](mailto:azmirasyid@students.telkomuniversity.ac.id), <sup>2</sup>[porman@telkomuniversity.ac.id](mailto:porman@telkomuniversity.ac.id), <sup>3</sup>[ramdhan@telkomuniversity.ac.id](mailto:ramdhan@telkomuniversity.ac.id)

#### Abstrak

Quadcopter (*drone*) adalah sebuah *multicopter* yang memiliki empat rotor (baling-baling) dimana sistem kendalinya menggunakan *Flight Controller* yang telah memiliki algoritma dan sensor – sensor didalamnya. Sistem autonomous pada zaman sekarang banyak dikembangkan pada *quadcopter*, khususnya dalam hal pengamatan objek – objek fisik di alam semesta menggunakan *quadcopter*. Dari keunggulan tersebut terdapat kekurangan, yaitu pada umumnya *quadcopter* masih dikendalikan oleh seseorang dengan sebuah *remote control*.

Pada tugas akhir ini, dirancang *autonomous quadcopter* dengan kemampuan bergerak sesuai *pathbuild* yang ditentukan pada layar Android. *Quadcopter* dapat dikendalikan melalui aplikasi Android yang dikomunikasikan menggunakan modul *Wireless 3DR* dengan Frekuensi 433MHz, dimana *quadcopter* akan bergerak tanpa menggunakan *remote control* dan bergerak sesuai dengan *path* yang telah di tujukan melalui Android.

Hasil tugas akhir ini berupa sistem kontrol gerak *autonomous* yang memudahkan pengguna untuk mengontrol *quadcopter* dalam melakukan mobilisasi melalui sebuah perangkat *smartphone* berbasis Android. Dengan persentase performansi hasil yang dicapai dalam percobaan yang dilakukan sebesar 83%

**Kata kunci:** Java, *autonomous*, UAV, *quadcopter*, *drone*, *waypoint*, Android, *gesture*, GPS

#### Abstract

Quadcopter (*drone*) is a *multicopter* that has four rotor (blades) which the control system using the *Flight Controller* who already have the algorithms and sensors inside it. An autonomous system today developed on *quadcopter*, particularly in terms of observing physical objects in the universe using *quadcopter*. From the advantage there's shortage, which is generally *quadcopter* is still controlled by a person with a *remote control*.

In this thesis, the *autonomous quadcopter* was designed with the ability to move in accordance *pathbuild* specified on the Android screen. *Quadcopter* can be controlled via an Android application that communicated using *3DR Wireless* modules with 433MHz frequency, which *quadcopter* will move without using the *remote control* and move in accordance with the *path* that has been addressed via Android.

The results of this thesis is an *autonomous motion control* system that allows users to control *quadcopter* in mobilizing via a *smartphone* device based on Android. With performance percentage result that accomplished on trial is 83%

**Keywords:** Java, *autonomous*, UAV, *quadcopter*, *drone*, *waypoint*, Android, *gesture*, GPS

### 1. Pendahuluan

UAV (*Unmanned Aerial Vehicle*) secara umum dapat diartikan sebagai sebuah wahana udara. Dengan kemajuan teknologi, UAV saat ini sudah dilengkapi perangkat pendukung yang memungkinkan dapat sepenuhnya dikendalikan secara *autonomous*. Pada saat ini perkembangan sistem *autonomous* digunakan pada sistem wahana bertipe *rotary-wing*. Salah satu jenis *rotary-wing* yang berkembang saat ini adalah *quadrotor* atau *quadcopter*. [1]

*Quadcopter* merupakan pesawat tanpa awak yang memiliki empat buah motor dan *propeller* di tiap sudut kerangka utama. Bagian kerangka utama digunakan untuk peletakkan sumber daya (baterai), sistem kontrol dan sensor

dari *quadcopter*. Sistem kontrol tersebut digunakan untuk mengatur kecepatan dari setiap motor sesuai dengan gerakan yang diinginkan. [2]

*Quadcopter* memiliki GPS (*Global Positioning System*) yang berfungsi melacak koordinat dan lokasi, sehingga dapat mempertahankan posisinya saat terbang, maka *quadcopter* dapat diimplementasikan untuk *aerial photography*, *aerial mapping*, dan pantauan udara. Selain keunggulan dan masalah dalam *aerial photography* dan *mapping*, pada umumnya *quadcopter* masih dikendalikan oleh seorang pilot dengan *radio control* untuk *mapping*. Berdasarkan masalah tersebut, penulis berkeinginan untuk merancang bangun sebuah *quadcopter* dengan kemampuan dapat bergerak berdasarkan *pathbuild* yang telah di tentukan pada program yang telah terintegrasi.

Jadi, *quadcopter* dapat bergerak berdasarkan *pathbuild* yang ditentukan pada layar Android dan menganalisa hasil mobilisasi alat ketika berada di tiap titik *pathbuild* dengan dibantu *software* Mission Planner setelah mengendalikan *quadcopter* melalui GPS yang dikontrol posisinya melalui Android.

## 2. Dasar Teori

### 2.1. Quadcopter

Salah satu jenis UAV (*Unmanned Aerial Vehicle*) yang banyak diteliti saat ini adalah *quadcopter*. UAV jenis *helicopter* yang menggunakan empat motor untuk menggerakkan baling-balingnya. Kelebihan *quadcopter* adalah kemampuannya dalam melakukan *take-off* dan *landing* secara vertikal. [1]

Pada umumnya, *quadcopter* dikendalikan melalui *radio transmitter* dari jarak jauh oleh manusia sehingga sering sekali terjadi kesalahan pengendalian terutama dalam mekanisme *landing*.

### 2.2 Gerakan Dasar Quadcopter

*Quadcopter* memiliki empat baling-baling penggerak, masing-masing baling-baling dan motor penggeraknya menghasilkan daya angkat dan memiliki jarak yang sama terhadap pusat massa pesawat. Untuk menghindari terjadinya momen putar pada *body*, arah putaran baling-baling pada setiap rotornya berbeda. Terdapat 2 rotor yang bergerak searah jarum jam (CW) dan 2 rotor yang bergerak berlawanan arah jarum jam (CCW). [2]

Berikut gerakan dasar pada *quadcopter* [3]:

- *Throttle* atau *Height*  
Gerak ini dapat dilakukan dengan menaikkan atau menurunkan kecepatan semua *propeller* dalam jumlah yang sama.
- *Roll*  
Gerak ini dapat dilakukan dengan menambah atau mengurangi kecepatan salah satu *propeller* yang kiri atau yang kanan.
- *Pitch*  
Gerak ini dapat dilakukan dengan menambah atau mengurangi kecepatan salah satu *propeller*, yang depan atau yang belakang.
- *Yaw*  
Gerak ini dapat dilakukan dengan menambah atau mengurangi *propeller* depan belakang atau kanan-kiri secara bersamaan.

### 2.3 Flight Controller

Fixhawk memungkinkan APM dan PX4 operator yang ada untuk transisi ke sistem ini dan menurunkan hambatan masuk bagi pengguna baru untuk berpartisipasi dalam dunia otonom.

Keunggulan Fixhawk modul disertai dengan opsi perifer baru, termasuk sensor digital kecepatan udara, dukungan untuk indikator multi-warna LED eksternal dan magnetometer eksternal. Semua periferal secara otomatis terdeteksi dan dikonfigurasi

### 2.5 Radio Telemetry 3DR 433 MHz

Radio telemetry adalah cara ideal untuk *setup* koneksi antara *multicopter* dengan *ground control station* yang berupa PC (*Personal Computer*) atau Laptop, *smartphone* Android dan juga berupa firmware open source dengan menggunakan protokol MAVlink. Radio telemetry memiliki modul *transmitter* yang dipasang di *multicopter* dan modul *receiver* yang dipasang pada PC / laptop atau *smartphone*. Telemetry akan memudahkan pengguna dalam *interface* dengan *multicopter* secara *real time*.

## 2.5 Sistem Kontrol PID (Proportion Integral Derivative)

Pada *flight controller* Fixhawk memiliki sistem kontrol PID (*Proportional Integral Derivative*), yang menggunakan *negative feedback control*. *Input* kontroler PID adalah sinyal *error*  $e(t)$  dan *output*nya adalah sinyal kontrol  $u(t)$ . hubungan Antara *input* dan *output* controller PID dapat dituliskan

$$u(t) = K_p e(t) + T_d \frac{de(t)}{dt} + \frac{1}{T_i} \int e(t) dt$$

## 2.6 Mode Pathbuild

Mode PathBuild adalah fitur masa depan dengan sistem yang mudah dioperasikan untuk aktifitas outdoor, seperti *aerial photography*, pantauan udara, olahraga, *lifestyle* dan lain – lain. *PathBuild* atau sebuah *waypoint* adalah posisi GPS terdiri dari *Latitude* dan *Longitude* ditransmisikan ke robot oleh pengguna [mission planing]. Mode *PathBuild* mengikuti hasil gambar dari pengguna pada layar Android yang berupa titik – titik koordinat, lalu dapat dipantau catatan tempuhnya pada *software Mission Planner*. Adapun kekurangan dari ini yaitu, GPS (*Global Positioning System*) pada *smartphone* yang mengendalikan *quadcopter* tidak terlalu akurat ketika berada pada titik koordinat yang telah ditentukan oleh pengguna. Namun hal ini tidak berpengaruh terhadap terbang alat dalam mode *PathBuild*.

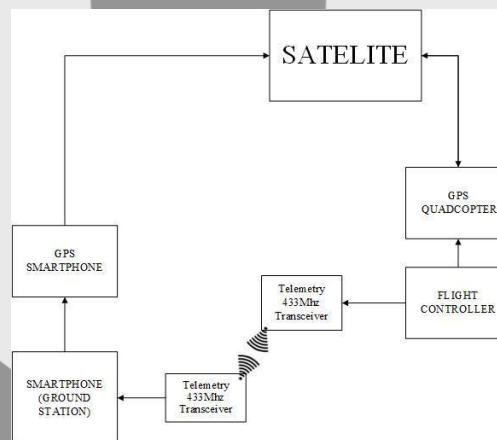
## 2.7 Android Studio[7]

Android Studio merupakan sebuah *Integrated Development Environment (IDE)* untuk *platform* Android. Android Studio mempunyai banyak fitur-fitur baru dibandingkan dengan Eclipse IDE, Android Studio menggunakan *Gradle* sebagai *build environment*. Fitur – fitur yang meningkatkan ketika membangun Android Apps, sebagai berikut:

- Sistem bangun *Gradle-based* yang fleksibel
- Fitur emulator yang cepat dan banyak
- Bisa mem-*build multiple APK (Android PacKage)*
- Template support* untuk Google Services dan berbagai macam tipe perangkat
- Layout editor* yang lebih bagus
- Built-in support* untuk *Google Cloud Platform*, sehingga mudah untuk integrase dengan *Google Cloud Messagin* dan *App Engine*
- Import library* langsung dari *Maven repository*
- Extensive Testing Tools and Frameworks*

## 3. Perancangan Sistem

### 3.1 Diagram Blok Sistem



Gambar 1 Blok Diagram Sistem

Perancangan dari pembentukan *pathbuild* dengan sistem *waypoint* secara garis besar dapat dilihat pada Gambar 1 *Flight Controller* yang terdapat pada *quadcopter* mengontrol pergerakan motor secara garis besar sehingga bisa berjalan dengan *autonomous* dan mengikuti jalur yang telah dibuat oleh penulis. Komponen- komponen yang terhubung dengan

*Flight Controller* yaitu GPS (Global Positioning System) dan Telemetry 433 MHz Transmitter.

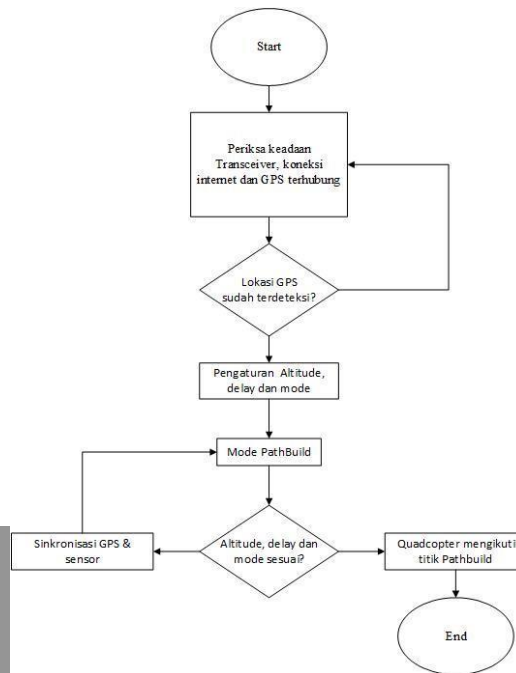
Cara kerja dari sistem tersebut adalah GPS pada *quadcopter* mengirimkan data lokasi berupa titik koordinat ke satelit. Begitu juga satelit memberikan balasan ke GPS pada *quadcopter*. Data tersebut kemudian diolah di dalam *Flight Controller*, lalu untuk mengontrol pergerakan motor digunakan sistem pengolahan PID secara *auto tuning* menggunakan program *Mission Planner*, lalu *3DR Telemetry 433 MHz Transmitter* mengirimkan informasi data yang terkoneksi pada *Flight Controller* ke Receivernya yang sudah terhubung ke *smartphone*. Untuk mengendalikan *quadcopter* dengan *smartphone* digunakan aplikasi Android. Aplikasi ini dapat menampilkan posisi *quadcopter* dan si pengguna. Hal ini dapat terjadi karena peran GPS dan satelit pada sistem pembentukan *PathBuild* dengan dengan sistem waypoint ini sangat besar, karena apabila koneksi terputus menyebabkan *quadcopter* kehilangan arah yang akan dituju. Jika semua komponen dari sistem telah terkoneksi maka *quadcopter* siap untuk dikendalikan menggunakan Android.

### 3.2 Kebutuhan perangkat

Kebutuhan perangkat lunak (*software*) dan perangkat keras (*Hardware*), yaitu:

- A) Perangkat Lunak (*Software*)
  - OS Android
  - Aplikasi Android Studio
  - Mission Planner
- B) Perangkat Keras (*Hardware*)
  - Laptop
  - Autodesk Inventor
  - *Flight Controller* Fixhawk
  - *Electric Speed Control* (ESC)
  - Motor Brushless
  - Baterai Li-po
  - Low Voltage Buzzer Alarm
  - *Global Positionin System* (GPS)
  - *Transmitter Remote Control* 2.4 GHz
  - *Receiver* FR-Sky 8CH
  - Modul 3DR 433 MHz
  - Smartphone Android
  - Kabel *On The Go* (OTG) Smartphone

### 3.3 Perancangan Perangkat Lunak (*Software*)

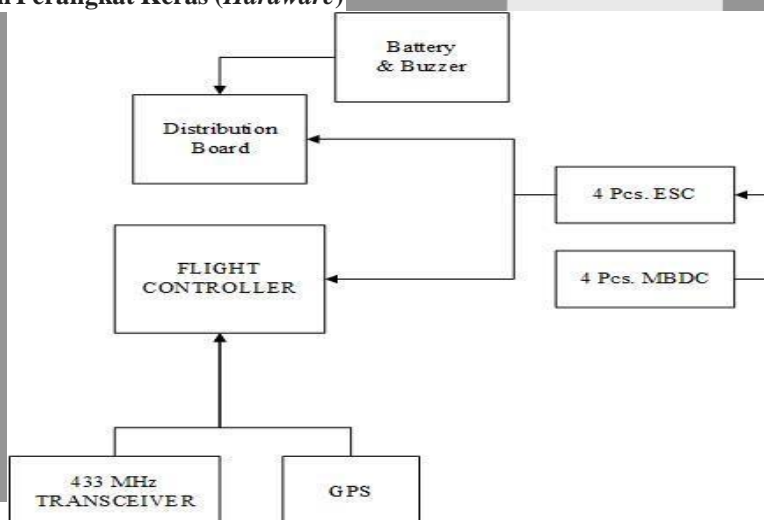


**Gambar 2** Flowchart Sistem dan Software

Gambar 2 menjelaskan tentang bagaimana cara kerja dari sistem dan software yang digunakan untuk pathbuild. Aplikasi Android sangat dibutuhkan dalam sistem ini. Pada tahap pertama koneksikan quadcopter dengan aplikasi Android yang sudah terpasang pada Android. Komunikasi quadcopter dengan Android menggunakan komponen Modul 3dr 433 MHz. Setelah quadcopter terkoneksi dengan aplikasi, lalu periksa sambungan GPS (Global Positioning System) pada quadcopter dan Android. Koneksi internet dibutuhkan saat proses ini. Lalu masuk kasus tentang pendeteksian lokasi GPS. Jika tidak terdeteksi, maka ulangi proses pemeriksaan koneksi internet dan GPS. Jika terdeteksi, maka pada aplikasi Android pengguna membuat Path untuk arah tujuan quadcopter, setelah itu set altitude, delay dan mode ketika berada pada setiap titik, kemudian upload ke quadcopter.

Quadcopter masuk mode Armed kemudian tekan tombol pathbuild, maka quadcopter akan terbang menuju path yang telah ditentukan oleh penulis. Hasil dari analisis yang diamati dapat berupa posisi titik koordinat Longitude, Latitude, Altitude, keadaan battery, jumlah titik ketika bergerak dari titik koordinat yang satu ke titik koordinat lainnya.

**3.4 Perancangan Perangkat Keras (Hardware)**



### Gambar 3 Diagram Blok Hardware

Perancangan perangkat keras dari *quadcopter* bisa dilihat pada Gambar 3. Penjelasan dari gambar di atas adalah *Flight Controller* pada diagram adalah sebagai pusat kendali utama dan baterai sebagai catuan utama dari *quadcopter*. Sedangkan *distribution board* adalah base atau badan dasar yang biasanya terbuat dari PCB untuk menggabungkan semua sambungan listrik dari baterai ke ESC (*Electronic Speed Control*). Seluruh komponen ESC tersambung ke *Flight Controller*. Bagian yang tersambung adalah kabel – kabel Vcc, Ground, dan Sinyal. Masing – masing ESC terkoneksi dengan motor *brushless*. *Receiver RC (Remote Control)*, GPS (*Global Positioning System*), dan Modul 3dr 433 MHz tersambung ke *Flight Controller*. Setelah semua komponen tersambung sesuai dengan gambar di atas, maka *quadcopter* telah siap digunakan.

#### 3.5 Pengujian dan Pengeluaran Yang Diharapkan

Dalam pengujiannya, *quadcopter* akan diterbangkan dengan kondisi yang telah ditentukan secara berulang kali dan melihat akurasi GPS pada *quadcopter*. Untuk mengujinya, cara yang dilakukan adalah mengarahkan *quadcopter* dengan mode *PathBuild*, lalu diamati respon dari *quadcopter*. Pengguna melakukan pengujian dengan dua cara, manual dan analisa dengan program *Mission Planner*. Pengujian secara manual dilakukan dengan mengikuti *quadcopter* ketika terbang mengikuti perintah yang dilakukan oleh pengguna, lalu meletakkan sebuah tanda (batu) di setiap titik koordinat yang ditentukan, lalu mengukur mulai dari titik tujuan, titik akhir dengan *landing*, keadaan battery ketika mulai penerbangan, dan waktu penerbangan. Sedangkan pengujian secara analisa program dilakukan dengan mengambil data log dari Android yang nantinya dapat di analisa mulai dari Latitude, Longitude, Altitude, Grafik Pitch and Roll.

Keluaran yang diharapkan setelah melakukan beberapa pengujian di atas yaitu, *quadcopter* bisa terbang dengan stabil dan dapat secara akurat mengikuti *PathBuild* yang telah ditentukan oleh penulis sesuai dengan *waypoint* yang dimasukkan, serta akurasi proporsional di setiap perubahan titik koordinat yang ditentukan oleh si penulis.

## 4. Pembahasan

### 4.1 Pengujian Akurasi GPS

Pengujian ini dilakukan dengan menyambungkan Android menggunakan kabel USB untuk pengambilan data log yang tersimpan ketika waktu pengujian. Lalu load data log ke *software Mission Planner* untuk menampilkan hasil dari *quadcopter* dari Latitude dan Longitude, setiap kali hasil uji coba akan dibandingkan dengan hasil titik koordinat awal dengan latitude -6.974201500 dan longitude 107.63234600, titik koordinat terbang dengan latitude -6.974342300 dan longitude 107.632317600 dan titik koordinat mendarat dengan latitude -6.974302300 dan longitude 107.632318200 untuk pengujian pertama. Pada pengujian kedua dengan hasil titik koordinat awal dengan latitude -6.974227200 dan longitude 107.632322700, titik koordinat terbang dengan latitude -6.974292600 dan longitude 107.632320700, dan titik koordinat mendarat dengan latitude -6.974271500 dan longitude 107.632318900. Adapun data titik koordinat wahana sebagai berikut:

Tabel 4 Uji pengujian 2 Waypoint Terbang

n	Koordinat terbang GPS Quadcopter		n	Koordinat percobaan GPS Quadcopter		Error (m)
	Latitude	Longitude		Latitude	Longitude	
1	-6.97429 2600	107.63232 0700	1	-6.97429 5100	107.63232 0600	0.278 m
2	-6.97429 2600	107.63232 0700	2	-6.97429 5200	107.63232 0100	0.296 m
3	-6.97429 2600	107.63232 0700	3	-6.97429 5400	107.63232 0100	0.318 m
4	-6.97429	107.63232	4	-6.97429	107.63232	0.181 m

	2600	0700		4200	0400	
5	-6.97429	107.63232	5	-6.97428	107.63232	0.835 m
	2600	0700		5100	0300	
6	-6.97429	107.63232	6	-6.97429	107.63232	0.167 m
	2600	0700		4100	0600	
7	-6.97429	107.63232	7	-6.97429	107.63232	0.159 m
	2600	0700		4	1	
8	-6.97429	107.63232	8	-6.97429	107.63231	0.199
	2600	0700		2	9	
9	-6.97429	107.63232	9	-6.97429	107.63232	0.142
	2600	0700		3600	1500	

Tabel 5 Uji pengujian 2 Waypoint Mendarat

n	Koordinat mendarat		n	Koordinat percobaan		Error (m)
	GPS Quadcopter			GPS Quadcopter		
	Latitude	Longitude		Latitude	Longitude	
1	-6.97427	107.63231	1	-6.97427	107.63232	0.407 m
	1500	8900		4700	0700	
2	-6.97427	107.63231	2	-6.97427	107.63232	0.269 m
	1500	8900		0700	1200	
3	-6.97427	107.63231	3	-6.97427	107.63232	0.217 m
	1500	8900		0700	0700	
4	-6.97427	107.63231	4	-6.97427	107.63231	0.091 m
	1500	8900		1700	9700	
5	-6.97427	107.63231	5	-6.97427	107.63231	0.135 m
	1500	8900		2200	9900	
6	-6.97427	107.63231	6	-6.97427	107.63231	0.129 m
	1500	8900		1400	7800	
7	-6.97427	107.63231	7	-	107.63231	0.510 m
	1500	8900		6.974267	8	
8	-6.97427	107.63231	8	-6.97426	107.63231	0.602 m

	1500	8900		6100	8500	
9	-6.97427	107.63231	9	-6.97427	107.63232	0.346 m
	1500	8900		1000	2000	

## 5. Kesimpulan

Berdasarkan hasil penelitian yang dilakukan pada tugas akhir, dapat disimpulkan bahwa:

1. Menggunakan Raspberry Pi lebih rumit daripada menggunakan PC. Selain rumit, memori pada Raspberry Pi sangatlah terbatas.
2. Menggunakan Arduino sebelum RF adalah cara yang lebih baik daripada langsung ke Raspberry Pi, karena pemrogramannya lebih mudah pada Arduino daripada Java di Raspberry Pi.
3. Data yang dikirim dari Arduino ke Raspberry Pi akan selalu looping, cara menghentikan looping tersebut dengan *reset*. Dan faktor yang memberikan data error itu bisa dari looping Arduino itu sendiri.
4. Penggunaan memori untuk program berbasis Java sangatlah besar, karena banyaknya fitur-fitur yang disediakan oleh Java.

## DAFTAR PUSTAKA

1. Utama Wiguna, Rizky, M. Komarudin dan Agus Trisanto, 2013. Sistem Kendali Holding Position Pada Quadcopter Berbasis Mikrokontroler ATmega 328P. Bandar Lampung. Universitas Lampung
2. Tomasso, Bresciani, 2008. Modelling, Identification and Control of a Quadcopter Helicopter. Department of Automatic Control Lund University
3. Domingue, Jorge M.B., 2009. Quadrotor Prototype. Universidade Tecnica de Lisboa. Dissertacio
4. Aziz, Mohamad Yusuf Abdul, 2015. Perancangan dan Implementasi Self-tuning Proportional Integral and Derivatif untuk kestabilan sistem manuver pada Autonomous. Bandung. Telkom University
5. Ma, Li, 2014. Research and Development of Mobile Application for Android Platform. University of Information Science & Technology. Nanjing
6. Singhal Rinku, Padhee Subhransu, & Kaur Gagandeep, 2012. Design of fractional order PID controller for Speed Control of DC Motor. Punjab: Patiala.