

## Implementasi Pengendalian *Quadcopter* Dengan Prinsip *Virtual Reality* Menggunakan *Google Cardboard*

Dimas Angger Pribadi<sup>1</sup>, Eriq Muh. Adams Jonemaro<sup>2</sup>, Gembong Edhi Setyawan<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>dimasanggerpribadi@gmail.com, <sup>2</sup>eriq.adams@ub.ac.id, <sup>3</sup>gembong@ub.ac.id

### Abstrak

*Quadcopter* merupakan *RC craft* yang sangat diminati oleh manusia baik dalam penggunaan secara pribadi, maupun sebagai objek penelitian untuk pengembangan teknologi *quadcopter*. Salah satu *quadcopter* yang sering menjadi objek penelitian adalah Parrot AR.Drone. Parrot AR.Drone merupakan perangkat yang bisa dikendalikan menggunakan *android*, di mana *drone* lain masih menggunakan *remote control*. Selain menggunakan *android*, Parrot AR.Drone juga bisa dikendalikan dengan menggunakan perangkat lain seperti *kinect*, *joystick*, *oculus rift* dan masih banyak lagi. Hal ini dikarenakan Parrot AR.Drone bersifat sumber terbuka yang dapat dikembangkan oleh semua orang. Minat penulis pada penelitian ini adalah mengendalikan Parrot AR.Drone dengan prinsip *virtual reality* menggunakan gerakan kepala. Penelitian ini bertujuan untuk membuat program kendali *quadcopter* menggunakan *input* gerakan kepala dengan perangkat *google cardboard*, yang akan memberikan pengalaman *first person view (fpv)* pada pengguna seakan pengguna berada di dalam *quadcopter* tersebut. Pada penelitian ini *google cardboard* digunakan sebagai perangkat pendukung *virtual reality* untuk mengendalikan pergerakan Parrot AR.Drone. Hasil penelitian ini adalah program kendali *quadcopter drone* dengan menggunakan gerakan kepala. Pengujian dilakukan dengan menguji setiap gerakan kepala untuk mengetahui respon dari pergerakan Parrot AR.Drone. Berdasarkan hasil pengujian yang dilakukan, didapatkan hasil tingkat akurasi kendali sebesar 97% dan waktu yang berbeda dari respon pergerakan *drone* ketika dikendalikan.

**Kata Kunci** : *Google Cardboard*, *Parrot AR.Drone*, pengendalian, *Quadcopter*, *Virtual Reality*

### Abstract

*Quadcopter* is an *RC Craft* that is in great demand by humans both in personal use, or used as a research object for the development of *quadcopter* technology. One type that is often used as the object of research is Parrot AR.Drone. Parrot AR.Drone is a device that can be controlled by using *android* smartphone, where other drones still use the *remote control*. In addition to using *android*, Parrot AR.Drone can also be controlled by using other devices such as *kinect*, *joystick*, *oculus rift* and much more. This is because Parrot AR.Drone is an open source that can be developed by everyone. The author's interest in this research is to control Parrot AR.Drone with the principle of *virtual reality* using head movement. The purpose of this research was to create a *quadcopter* control program using head movement input with *google cardboard* device, which will provide *first person view (fpv)* experience to the user as if the user is in the *quadcopter*. In this research, *google cardboard* is used as a *virtual reality* support device to control the movement of Parrot AR.Drone. The result of this research is the *quadcopter drone* control program using head movement. Testing phase is done by testing each head movement to find out the response of Parrot AR.Drone movement. Based on the results of the tests conducted, obtained the result of control accuracy level is 97%, and different response times of *drone* movement when controlled.

**Keyword** : *Google Cardboard*, *Parrot AR.Drone*, controlling, *Quadcopter*, *Virtual Reality*

## 1. PENDAHULUAN

Perkembangan pada dunia teknologi

kedirgantaraan sangatlah pesat. Hal ini ditandai dengan perkembangan pesawat tanpa awak atau *drone*. *Drone* sangat diminati salah satunya Parrot AR.Drone. Parrot AR.Drone merupakan

perangkat yang bisa dikendalikan menggunakan *smartphone android*, di mana *drone* lain dikendalikan menggunakan *remote control*. *Drone* ini disertai dengan kamera yang digunakan untuk pengambilan gambar dan video. Di dalam komponen *drone* sudah tertanam PC sendiri yang didalamnya terdapat banyak sensor seperti *accelerometer*, *gyroscope*, dan masih banyak sensor lain yang ada di dalamnya. Untuk menghubungkan antara *drone* dan aplikasi *smartphone* yaitu dengan menggunakan sinyal *wi-fi* yang dipancarkan langsung dari *drone* itu sendiri.

Pada penelitian sebelumnya yang dilakukan oleh Andrea et al. (2013), *drone* dikendalikan dengan menggunakan *kinect*. Penelitian ini menggunakan gerakan tangan sebagai pengendali gerakan *drone* (Andrea et al., 2013). Penelitian mengenai pengendalian *drone* juga dilakukan oleh Byung Hyun Kim et al. (2014) dengan menggunakan *eye tracker* atau pembaca gerakan mata dan juga gerakan kepala sebagai pengendali *drone* (Kim et al., 2014). Berdasarkan dua penelitian tersebut bisa diambil kesimpulan bahwa, *drone* bisa dikendalikan menggunakan perangkat lain yaitu *kinect* dan *eye tracker*.

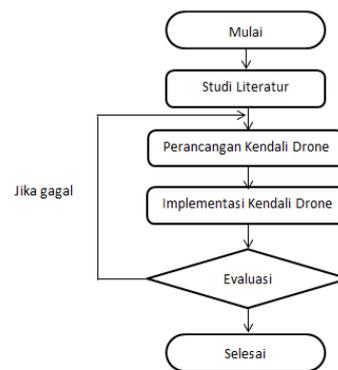
Banyak sekali jenis aplikasi *android* yang memiliki fungsi sebagai pengendali Parrot AR.Drone, namun semuanya masih menggunakan prinsip pengendalian konvensional. Sehingga penulis merasa perlu adanya pemanfaatan teknologi virtual reality pada *drone* dengan memanfaatkan perangkat pendukung yaitu *google cardboard*. Berdasarkan penelitian sebelumnya, penulis beranggapan bahwa pengembangan aplikasi pengendali *drone* ini karena penggunaan *google cardboard* sangatlah ekonomis dan ergonomis.

**2. METODOLOGI PENELITIAN**

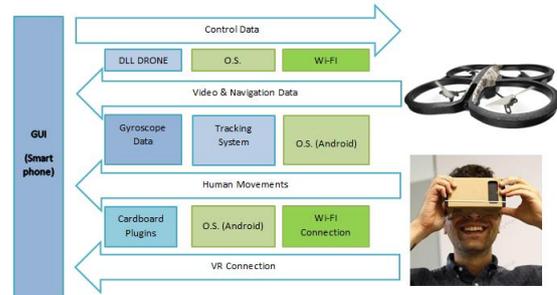
Metode yang digunakan pada pengembangan aplikasi kendali ini merupakan metode pengembangan iteratif. Seperti pada Gambar 1, metode ini dimulai dengan studi literatur dari setiap sumber pustaka kode program dan juga *software* pendukung yang digunakan untuk pengembangan kendali *drone*. Proses selanjutnya adalah perancangan kendali *drone*, yang mana pada bagian ini dijelaskan mengenai perancangan *stereoscopic display*, pembacaan data *gyroscope* dan perancangan pemetaan data dari data *gyroscope* ke kontrol atau kendali *drone*. Pada bagian implementasi

berisi penjelasan tentang hasil penerapan dari perancangan yang sudah dijelaskan pada bab perancangan. Terakhir, proses evaluasi yang dimaksud adalah pengujian serta analisis dari hasil penelitian implementasi kendali *drone* yang meliputi akurasi pembacaan data *gyro*, akurasi ketinggian *drone* dan juga analisis dari kendali *drone* tersebut.

Penelitian ini dititikberatkan pada implementasi kendali *drone*, yang menggunakan jenis Parrot AR.Drone 2.0 . Pemanfaatan teknologi *virtual reality* dan juga perangkat *google cardboard* dimaksudkan untuk memberi efek *first person view* dan perangkat *virtual reality* yang mudah dijangkau atau ekonomis.



**Gambar 1.** Gambar Diagram Alir Program



**Gambar 2.** Diagram Alir Pertukaran Data Pada Perangkat

Diagram alir pertukaran data pada Gambar 2 di atas juga dijelaskan mengenai data yang digunakan untuk menggerakkan *drone*. *Drone* dan *smartphone* terintegrasi dengan menggunakan sinyal *wi-fi*, sedangkan untuk pengendalian *drone* memanfaatkan data *gyroscope* dari *smartphone* android. Data yang digunakan berupa sudut x, y dan z dari perangkat yang digunakan.

Rumus persamaan matematika yang digunakan untuk mendapatkan data *gyroscope* adalah rumus pembacaan sudut *quaternion*. Penggunaan rumus ini sangatlah tepat mengingat

*quaternion* bersifat tidak mudah diinterpolarsi atau jika dilakukan rotasi tidak mempengaruhi sumbu pergerakan lain. Rumus yang digunakan adalah rumus Hamilton *Quaternion* yang ditemukan oleh Rowan Hamilton pada tahun 1843. Penerapan rumus Hamilton pada rotasi objek adalah sebagai berikut:

$$H = w + xi + yj + zk \tag{1}$$

Dengan keterangan:

$w, x, y, z$  = bilangan real sebagai koefisien  
 $i, j, k$  = fundamental dari *quaternion*

Pada dasarnya, sebuah *quaternion* adalah bilangan kompleks dengan  $w$  sebagai bagian sebenarnya dan  $x, y, z$  sebagai bagian imajiner. Jika sebuah *quaternion* mewakili sumbu dalam rotasi sebuah objek, maka penjabarannya :

$$w = \cos \frac{\theta}{2} \tag{2}$$

$$x = v1, v1 = \sin \frac{\theta}{2} \tag{3}$$

$$y = v2, v2 = \sin \frac{\theta}{2} \tag{4}$$

$$z = v3, v3 = \sin \frac{\theta}{2} \tag{5}$$

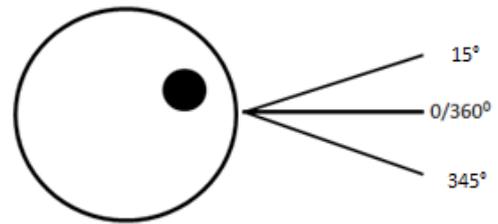
Manfaat dari penggunaan *quaternion* ini adalah ketika objek 3 dimensi dilakukan rotasi tidak akan mengalami pertumbukan nilai sudut dan rotasi objek lebih stabil.

### 3. PERANCANGAN

Pada bagian ini perancangan yang dilakukan adalah pemetaan dari data gyro ke kontrol *drone* menyesuaikan dari gerakan kepala pengguna. Pembacaan data gyro dalam bentuk sudut derajat diteruskan ke *syntax* gerakan *drone*. Beberapa perancangan gerakan *drone* adalah sebagai berikut :

#### 1. Gerakan kepala ke atas dan ke bawah

Gerakan kepala ke atas dan ke bawah mengikuti pergerakan sumbu x atau yang pada pergerakan *drone* mengikuti perubahan nilai *altitude*. Gerakan kepala ke atas dibaca ketika sudut sumbu x mengalami perubahan nilai dari  $0/360^0$  menjadi  $15^0$ . Sedangkan gerakan kepala ke bawah dibaca ketika sudut sumbu x mengalami perubahan nilai dari sudut  $0/360^0$  menjadi  $345^0$ . Gerakan kepala ke atas diterjemahkan pergerakan *up* pada *drone* sedangkan gerakan kepala ke bawah diterjemahkan pergerakan *down* pada *drone*. Penampang gerakan tersebut bisa dilihat pada Gambar 3 di bawah ini:



**Gambar 3.** Ilustrasi Gerakan Kepala ke Atas dan ke Bawah

#### 2. Gerakan kepala menoleh ke kanan dan kiri

Gerakan kepala menoleh ke kanan dan ke kiri mengikuti pergerakan sumbu y pada unity atau yang pada pergerakan *drone* mengikuti perubahan nilai yaw. Gerakan kepala menoleh ke kiri dibaca ketika sudut sumbu y mengalami perubahan nilai dari  $0/360^0$  menjadi  $345^0$ . Sedangkan gerakan kepala menoleh ke kanan dibaca ketika sudut sumbu y mengalami perubahan nilai dari sudut  $0/360^0$  menjadi  $15^0$ . Gerakan kepala menoleh ke kiri diterjemahkan pergerakan memutar ke arah kiri pada *drone* sedangkan gerakan kepala menoleh ke kanan diterjemahkan pergerakan memutar ke arah kanan pada *drone*. Gerakan yang sudah dijelaskan bisa dilihat pada Gambar 4 di bawah ini.



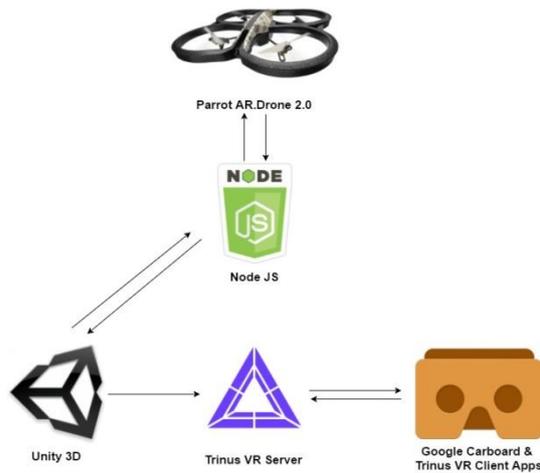
**Gambar 4.** Ilustrasi Gerakan Kepala Menoleh Ke Kanan Dan Ke Kiri

Berdasarkan Gambar 3 dan Gambar 4 dapat diambil kesimpulan, untuk mengetahui sudut pada gerakan kepala yang dilakukan oleh pengguna menggunakan rumus *quaternion*. Sesuai dengan Persamaan (1), penggunaan rumus *quaternion* untuk menghindari adanya sudut yang terkunci karena sumbu rotasi yang bertumbukan.

### 4. IMPLEMENTASI

Implementasi dari pengendalian *drone* dengan prinsip *virtual reality* dilakukan berdasarkan perancangan yang sudah dijelaskan

pada metodologi penelitian. Penerapan implementasi program kendali *drone* menggunakan bahasa pemrograman *C#* untuk antarmuka dan *node.js* untuk mengakses koneksi jaringan *drone*. Penggunaan bahasa pemrograman tersebut menyesuaikan dengan bahasa pemrograman yang bisa digunakan pada *unity* dan akses jaringan *drone*. Alur dari implementasi program kendali *drone* bisa dilihat pada Gambar 5 di bawah ini:



Gambar 5. Diagram Implementasi Program

Berdasarkan perancangan program yang sudah dibuat, maka tahapan proses program kendali adalah sebagai berikut:

1. Menyambungkan program kendali dengan jaringan drone.
2. Membagi jaringan dengan menggunakan prinsip *Ad-Hoc* untuk diakses melalui trinus vr dan diteruskan pada *smartphone* dengan tampilan *stereoscopic*.
3. Drone diterbangkan.
4. Membaca sudut pergerakan kepala.
5. Mengirim sinyal kontrol untuk pengendalian *drone*.
6. Melakukan perulangan tahap 4.

## 5. HASIL PENGUJIAN

Untuk menguji kinerja program kendali *drone* yang telah dibangun, maka pengujian melakukan pengujian fungsionalitas dan pengujian berdasarkan navigasi data *drone*. Navigasi data *drone* yang digunakan adalah data *altitude* dan data *yaw*. Data *altitude* digunakan sebagai dasar pengujian yang mengacu pada ketinggian, sedangkan data *yaw* sebagai dasar pengujian rotasi atau perputaran *drone*.

Pengujian fungsionalitas dilakukan dengan menguji fungsi tiap *input* yang diberikan dengan

pergerakan *drone*. Sedangkan untuk pengujian navigasi data untuk melihat waktu tempuh dari respon pergerakan *drone* setelah diberikan *input*.

### 5.1 Pengujian Fungsionalitas

Pengujian ini dimaksudkan untuk mengetahui bagaimana fungsi program pengendalian *drone* yang sudah dijelaskan pada bab implementasi. Pengujian fungsionalitas yang dimaksud adalah proses menguji kesesuaian antara kondisi *input* yang sudah dibuat pada program kendali dengan respon pergerakan *drone*.

Proses Pengujian ini meliputi pengujian secara manual pada tiap kondisi yang sudah di inisialisasi pada bab implementasi. Kesesuaian antara *input* dengan pergerakan *drone* sangatlah diperlukan untuk melihat tingkat akurasi program kendali *drone*. Dari 10 kali pengujian yang dilakukan dapat dijabarkan sebagai berikut:

Tabel 1 Tabel Tingkat Keberhasilan Pergerakan

Jenis Pergerakan <i>Drone</i>	Tingkat Keberhasilan
F1 ( <i>take off</i> )	100%
F2 ( <i>landing</i> )	100%
F3 ( <i>hover</i> )	100%
F4 ( <i>naik / up</i> )	100%
F5 ( <i>turun / down</i> )	100%
F6 ( <i>rotate right</i> )	100%
F7 ( <i>rotate left</i> )	80%

Berdasarkan pengujian yang sudah dilakukan, penulis dapat mengasumsikan keberhasilan pengujian akurasi kendali sebesar 97% dari 10 kali pengujian.

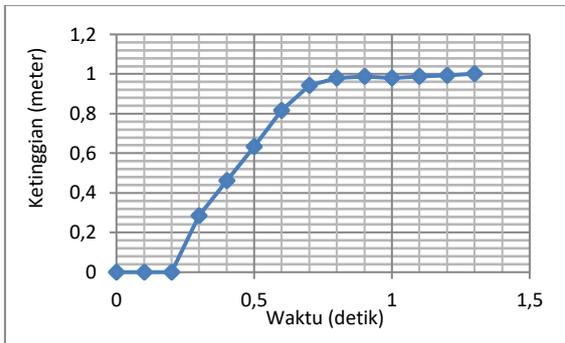
### 5.2 Pengujian Berdasarkan Navigasi Data

Pengujian pada tahap ini untuk mengambil navigasi data ketika program kendali mulai terhubung dengan *drone* sampai ketika program kendali dinonaktifkan. Dari navigasi data tersebut dihasilkan grafik ketinggian, grafik *roll*, grafik *pitch* dan grafik *yaw* untuk dijadikan sebagai hasil analisa. Analisa navigasi data merupakan perbandingan antara data ketinggian, data *roll*, data *pitch* dan data *yaw*. Dari setiap pergerakan *drone* didapat navigasi data yang berbeda, jadi setiap pergerakan *drone* navigasi data yang didapat tidak selalu sama dan menghasilkan grafik yang berbeda-beda.

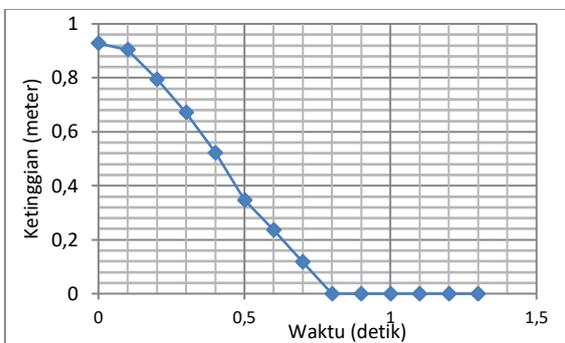
#### 1. Navigasi Data *take off* dan *landing*

Navigasi data yang didapatkan pada pengujian *take off* dan *landing* yang di dapatkan digunakan untuk menguji waktu yang dibutuhkan *drone* untuk *terbang dan mendarat*.

Grafik navigasi data dipisah untuk mengetahui perbedaan waktu respon tiap *input*.



**Gambar 6.** Hasil Grafik Navigasi Data Pengujian *Take Off*

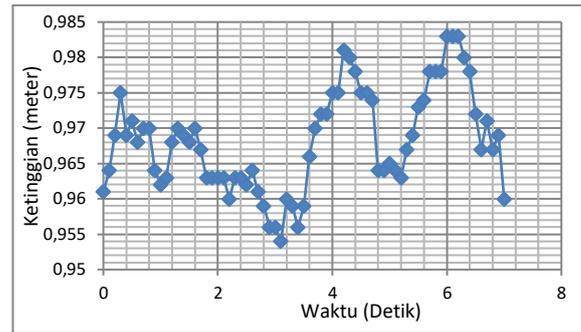


**Gambar 7.** Hasil Grafik Navigasi Data Pengujian *Landing*

Pengujian *take off* dan *landing* lebih dititikberatkan pada waktu yang dibutuhkan untuk *drone* terbang dan mendarat. Kesimpulan dari pengujian berdasarkan pada Gambar 6 dan Gambar 7 adalah waktu yang dibutuhkan *drone* untuk *take off* sebesar 0,6 detik dan untuk *landing* sebesar 0,8 detik.

2. Navigasi Data *Hover*

Pengujian *hover* pada *drone* dilakukan untuk mengetahui ketinggian *drone* ketika berada pada posisi *hover*. Selain itu pengujian ini digunakan untuk melihat kestabilan ketinggian *drone* ketika berada pada posisi *hover*.



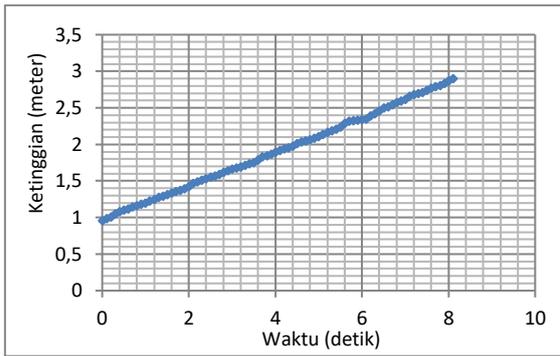
**Gambar 8.** Hasil Grafik Navigasi Data Pengujian *Hover*

Berdasarkan grafik yang terdapat pada Gambar 8, dapat diambil kesimpulan ketinggian *drone* ketika berada pada posisi *hover* tidak menunjukkan nilai yang stabil, melainkan masih terdapat perubahan nilai walaupun kita melihat ketika *drone* berada pada posisi *hover* stabil. Ketinggian *drone* ketika berada pada posisi *hover* berada pada kisaran 0,96 meter.

3. Navigasi Data *Drone Naik (up)*

Pengujian *drone* naik dilakukan untuk melihat pertambahan ketinggian *drone* setelah berada pada posisi *hover*. Selain itu, pengujian ini juga digunakan untuk melihat waktu tempuh untuk setiap pertambahan ketinggian 1 meter. Setelah *drone* diterbangkan dan berada pada posisi stabil (*hover*) ditambahkan inputan untuk *drone* naik dengan gerakan kepala mendongak ke atas.

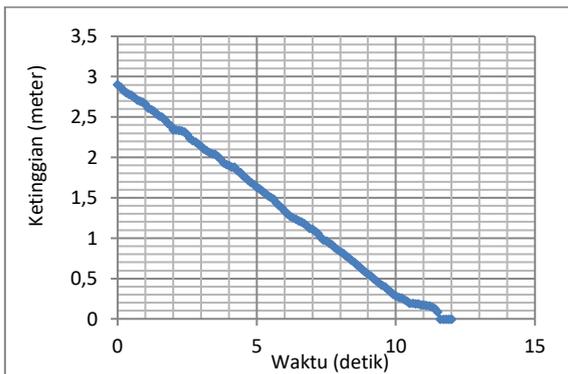
Gambar 9 di bawah ini merupakan hasil grafik dari pengujian *drone* naik. Setelah pengguna memberikan *input* gerakan kepala mendongak ke atas, di dapatkan ketinggian *drone* semakin naik dari ketinggian posisi *hover* (0,98 meter) semakin naik sampai pada ketinggian 2,9 meter atau ketinggian maksimal dari tempat pengujian dengan waktu tempuh 8 detik. Berdasarkan pengujian ini dapat diambil kesimpulan waktu yang dibutuhkan *drone* untuk setiap penambahan ketinggian 1 meter sebesar 4,2 detik.



Gambar 9. Hasil Grafik Navigasi Data Pengujian Drone Naik

4. Navigasi Data Drone Turun (down)

Pengujian drone turun dilakukan melihat perubahan ketinggian drone setelah berada pada posisi hover. Setelah drone diterbangkan dan berada pada posisi stabil (hover) ditambahkan input untuk drone turun dengan gerakan kepala menunduk.



Gambar 10. Hasil Grafik Navigasi Data Pengujian Drone Turun

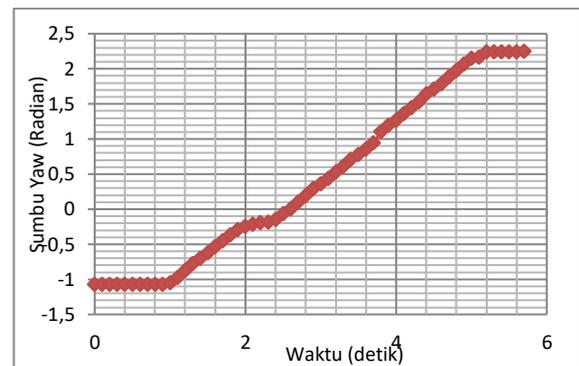
Gambar 10 di atas merupakan hasil grafik dari pengujian drone turun. Setelah pengguna memberikan input gerakan kepala menunduk, di dapatkan ketinggian drone semakin turun dari ketinggian 2,9 meter semakin turun sampai dengan ketinggian 0,1 meter dengan waktu 12 detik. Berdasarkan pengujian ini dapat diambil kesimpulan bahwa ketika drone diberikan input untuk pergerakan turun, ketinggian drone akan berangsur-angsur turun namun tidak sampai pada posisi landing. Waktu yang dibutuhkan untuk setiap penurunan ketinggian sebanyak 1 meter sebesar 3,8 detik.

5. Navigasi Data Drone Berputar ke Kanan (Rotate Right)

Pengujian drone berputar ke kanan dilakukan untuk melihat respon rotasi drone ketika diberikan input pergerakan berputar ke

kanan. Yang menjadi acuan untuk pengujian drone turun adalah perubahan data sumbu yaw, karena gerakan drone berotasi berdasar kepada perputaran sumbu yaw.

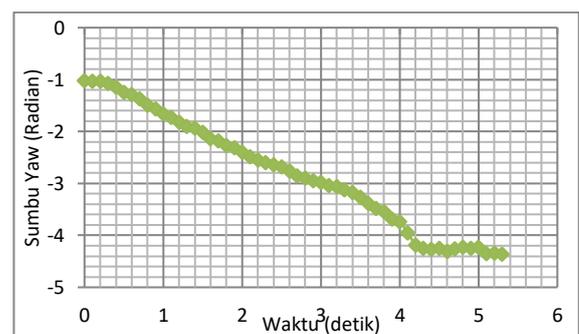
Gambar 11 merupakan hasil pengujian drone berputar ke kanan. Perubahan nilai terjadi ketika drone berputar ke kanan, yang awalnya bernilai -1 radian menjadi bernilai 2,2 radian. Dari nilai tersebut di dapatkan selisih sebesar 3,2 radian yang jika dikonversikan ke dalam satuan derajat ( $^{\circ}$ ), drone berputar ke arah kanan  $180^{\circ}$ . Berdasarkan hasil pengujian drone berputar ke kanan dapat diambil kesimpulan bahwa jika drone berputar ke kanan nilai radian berubah semakin ke arah nilai positif. Waktu yang dibutuhkan untuk drone berputar ke kanan  $180^{\circ}$  adalah dengan gerakan kepala menoleh ke kanan selama 4 detik.



Gambar 11. Hasil Grafik Navigasi Data Pengujian Drone Berputar ke Kanan

6. Navigasi Data Drone Berputar ke Kiri (Rotate Left)

Pengujian drone berputar ke kiri dilakukan untuk melihat respon rotasi drone ketika diberikan input pergerakan berputar ke kiri. Yang menjadi acuan untuk pengujian drone berputar ke kiri adalah perubahan data sumbu yaw, karena gerakan drone berotasi berdasar kepada perputaran sumbu yaw.



Gambar 12. Hasil Grafik Navigasi Data Pengujian Drone Berputar ke Kiri

Dari pengujian *drone* berputar ke kiri, didapatkan hasil grafik seperti pada Gambar 12 di atas. Perubahan nilai terjadi ketika *drone* berputar ke kiri, yang awalnya bernilai -1 radian menjadi bernilai -4,3 radian dengan waktu tempuh 4 detik. Dari nilai tersebut di dapatkan selisih sebesar 3,3 radian radian yang jika dikonversikan ke dalam satuan derajat ( $^{\circ}$ ), *drone* berputar ke arah kiri  $188^{\circ}$ . Berdasarkan hasil pengujian *drone* berputar ke kiri dapat diambil kesimpulan bahwa jika *drone* berputar ke kiri nilai radian berubah semakin ke arah nilai negatif. Waktu yang dibutuhkan untuk drone berputar ke kiri  $180^{\circ}$  adalah dengan gerakan kepala menoleh ke kiri selama 4 detik.

**Tabel 2** Hasil Pengujian Navigasi Data

Jenis Pergerakan Drone	Waktu Tempuh
F1 ( <i>take off</i> )	0,6 detik
F2 ( <i>landing</i> )	0,8 detik
F3 ( <i>hover</i> )	Menyesuaikan waktu pengujian
F4 (naik / <i>up</i> )	4,2 detik (pertambahan 1 meter)
F5 (turun / <i>down</i> )	3,8 detik (pengurangan 1 meter)
F6 ( <i>rotate right</i> )	4 detik (rotasi $180^{\circ}$ searah jarum jam)
F7 ( <i>rotate left</i> )	4 detik (rotasi $180^{\circ}$ berlawanan arah jarum jam)

Tabel 2 merupakan ringkasan hasil dari pengujian berdasarkan navigasi data. Dari tabel di atas bisa dilihat waktu tempuh pada setiap pengujian untuk merespon setiap input berbeda-beda. Untuk waktu tempuh *hover* menyesuaikan dengan pengujian yang dilakukan, karena yang diuji adalah ketinggian rata-rata saat berada pada posisi *hover*.

## 6. KESIMPULAN

Berdasarkan pada perancangan, implementasi dan pengujian yang telah dilakukan, dapat diambil kesimpulan sebagai berikut :

1. Perangkat *Google Cardboard* atau *Google VR* sangatlah membantu kinerja program kendali *drone* dengan menggunakan prinsip *virtual reality*. Perangkat *Google Cardboard* memberikan kesan *first person view (FPV)* dan juga menambah kesan *immersive* atau seakan pengguna berada di dalam *drone* dan mengendalikannya secara langsung. Implementasi program kendali *drone* menggunakan prinsip *virtual reality* memakai *software* tambahan yaitu *Trinus VR*.

2. Implementasi program kendali *drone* dengan prinsip *virtual reality* terbatas pada gerakan kepala. Penulis hanya bisa mengimplementasikan 4 dari 6 gerakan kepala. Hal ini dikarenakan keterbatasan implementasi pada *trinus vr* tidak bisa membaca gerakan kepala miring ke kanan dan ke kiri. Mengacu pada Tabel 1, didapatkan nilai akurasi program kendali sebesar 97%. Hal ini dikarenakan ketika dilakukan pengujian fungsionalitas gerakan *drone* berputar ke kiri terjadi *error* 2 kali dari 10 kali pengujian.
3. Berdasarkan hasil pengujian navigasi data yang terdapat pada Tabel 2, bisa dilihat hasil kinerja pergerakan *drone* ketika *take off*, *landing*, *hover*, rotasi ke kanan dan rotasi ke kiri menghasilkan respon yang berbeda beda. Respon tersebut merupakan waktu tempuh dari *drone* ketika diberikan *input* sampai dengan *drone* melakukan gerakan sesuai *input* yang diberikan.

## DAFTAR PUSTAKA

- Andrea, S., Lamberti, F., Paravati, G. & Manuri, F., 2013. A Kinect-based natural interface for quadrotor control. *A Kinect-based natural interface for quadrotor control*, pp.179-82.  
Available at: [https://id.wikipedia.org/wiki/Realitas\\_maya](https://id.wikipedia.org/wiki/Realitas_maya)  
[Accessed 5 September 2016].
- Brandon, A., 2010. *Control your own augmented reality aerial drone? There's an app for that*-. [Online]  
Available at: <http://newatlas.com/parrot-ardrone-iphone-controlled-remote-helicopter/13741/>  
[Accessed 21 Agustus 2016].
- Discover Magazine, 2009. *20 Things You Didn't Know About. Tunnels*. [Online]  
Available at: <http://discovermagazine.com/2009/may/20-things-you-didnt-know-about-tunnels>  
[Accessed 4 May 2017].
- Google Cardboard, 2014. *Buy cardboard - Google VR*. [Online]  
Available at: [https://vr.google.com/intl/id\\_id/cardboard/get-cardboard/](https://vr.google.com/intl/id_id/cardboard/get-cardboard/)  
[Accessed 1 September 2016].

Kim, B.H., Minho, K. & Jo, S., 2014. Quadcopter flight control using a low - cost hybrid interface. *Elsevier*, pp.82-92.

Utama, B., 2015. *FAQ : Pertanyaan Umum Tentang Google Cardboard - Google Cardboard Indonesia*. [Online] Available at: <http://cardboard-id.com/faq-pertanyaan-umum-tentang-google-cardboard/> [Accessed 20 Augustus 2016].

Wikipedia, 2016. *Parrot AR. Drone - Wikipedia, the free encyclopedia*. [Online] Available at: [https://en.wikipedia.org/wiki/Parrot\\_AR.Drone](https://en.wikipedia.org/wiki/Parrot_AR.Drone) [Accessed 4 September 2016].