



Perancangan Sistem Pengendalian *Drone Autonomous Quadcopter* Dengan *Smartphone* Berbasis *Android*

Muhammad Adi Saputra^{#1}, Nur Alif Mardiyah², Muhammad Nasar³

Info Artikel dan Penulis

1,2,3 - Program Studi Teknik Elektro,
Universitas Muhammadiyah Malang,
Jl. Tlogomas No. 246, Malang,
Indonesia

Email: adi_413029@webmail.umm.ac.id

Kata Kunci:

Unmanned Aerial Vehicle, Autonomous, Quadcopter, Smartphone, Tracking.

Proses Artikel:

Dikirim 23 Februari 2021

Direvisi 24 Februari 2021

Diterima 22 April 2021

Diterbitkan online 22 April 2021

Abstrak

Unmanned Aerial Vehicle (UAV) adalah kendaraan udara tak berawak yang dapat terbang secara *autonomous*. Sistem *autonomous* pada *drone quadcopter* memiliki kemampuan untuk memperkecil pengendalian langsung dari user. *Autonomous* adalah sebuah misi pada *drone* yang dapat terbang dan mendarat secara otomatis sesuai dengan misi yang dibuat sebelumnya, walaupun *remote* dalam keadaan mati. Salah satu jenis nya yang berkembang pesat yaitu *UAV* jenis *drone*. Diantara tipe-tipe *drone* yang digunakan pada penelitian ini adalah *drone quadcopter* yang memiliki 4 *propeller* dan disinkronisasi dengan konfigurasi *frame* yang berbentuk (X). Pada penelitian sebelumnya *drone* menggunakan *google maps* dan diterbangkan secara *autonomous* sesuai dengan *waypoint* yang ditentukan oleh pengendali pada aplikasi android namun perintah terbang *drone* dilakukan secara *manual* dengan menggunakan aplikasi *putty* pada laptop. Maka dari itu akan diteliti perbaikan dari versi yang ada sebelumnya. Dan pada penelitian ini *drone* dapat dikendalikan via *smartphone* berbasis android dan dengan fitur *tracking*. Karena, android memiliki berbagai keunggulan sebagai software yang memakai basis kode computer yang bisa di distribusikan secara terbuka (*open source*). Dan dengan menggunakan aplikasi yang mendukung dengan *flight controller* dan aplikasi *tracking* yang dirancang sendiri menggunakan komunikasi *Internet of Things (IoT)*. dimana *IoT* tersebut berupa jaringan internet yang bisa mengkomunikasikan benda-benda elektronik dan virtual yang ada disekitar, contohnya *user* dan *UAV* jenis *drone* tersebut. Dimana tujuan *drone* ini dibuat agar dapat mempermudah pengendali dalam menggunakannya dan mudah untuk monitoring *drone* tersebut.

Abstract

An *Unmanned Aerial Vehicle (UAV)* is an unmanned aircraft that can fly autonomously. *Autonomous* system in a *drone quadcopter* can minimize user control. An *autonomous* mode enables the *drone* to operate automatically based on the previously created setting, although the *remote* is off. One type of *UAV* that is still developed is the *drone*. This research used a *drone quadcopter* type with 4 *propellers* synchronized with a configured *frame* in (X) shape. The previous *drone* research used *google maps* and flew autonomously according to the *waypoint* directed by an android controller application, next, the *drone* flight command was manual by using a *putty* application on a laptop. Since, the previous research should be improved, therefore, in this research, the *drone* used android-based control complete with a *tracking* feature. Because, android enables the use of computer basic code that can be distributed in *open source* with an application that supports *flight controllers* along with a *tracking* application that was a self-designed application using *Internet of Things* communication (*IoT*). *IoT* used in this research is an internet network that can connect electronics and virtual objects around, such as users and types of *UAV* like a *drone*. The purpose of making the *drone* in this research was to ease the user in operating and monitoring.

Saputra, M. A., Mardiyah, N. A., & Nasar, M. (2021). Perancangan Sistem Pengendalian Drone Autonomous Quadcopter Dengan Smartphone Berbasis Android . Journal of Mechatronic and Electrical Engineering, Vol. 1(1), pp: 33-41, April 2021, doi: <https://doi.org/10.22219/jmee.xxxx.xxxx>

1. PENDAHULUAN

Pada masa sekarang ini banyak macam dari teknologi sengaja dibuat agar lebih mempermudah manusia dalam hal pekerjaan. Diantara nya *Unmanned Aerial Vehicle (UAV)* atau biasa disebut dengan kendaraan udara tanpa awak. Salah satu jenis nya yang berkembang pesat yaitu *UAV* jenis *drone*. Pengendalian navigasi adalah hal yang penting untuk dikuasai karena untuk mengatur arah pergerakan pada *drone* tersebut terlebih *drone* ini menggunakan *remote control*.

Drone memiliki banyak tipe yang beredar pada saat ini, di antara nya ada *singlecopter*, *tricopter*, *quadcopter*, *heksacopter*, *oktacopter* dan sebagainya. Tetapi yang paling sering digunakan yaitu *singlecopter* dan *quadcopter* [1]. *Singlecopter* sering digunakan karena memiliki *propeller single* jadi cukup bisa bertahan lama karena tenaga yang digunakan juga sedikit. Adapun gerakan *singlecopter Yaw*, gerak berputar dari kiri atau dari kanan. *Pitch*, gerak maju dan mundur. *Roll*, gerak kekanan atau kekiri. Pengujian pertama *singlecopter* pada versi pertama saat pengujian di *outdoor* mengalami masalah yaitu kehilangan keseimbangan ketika angin kencang dan *singlecopter* menjadi terlalu *responsive* dan bertambah sulit dikendalikan [1].

Quadcopter adalah salah satu tipe *drone* yang memiliki 4 *propeller* yang mana keempat *propeller* tersebut menggunakan motor *brushless* yang disinkronisasikan dengan konfigurasi *frame* yang berbentuk tanda tambah (+) [2]. Motor pada bagian depan kanan dan belakang kiri berputar searah dengan jarum jam, sedangkan bagian kiri depan dan kanan belakang berputar berlawanan arah dari jarum jam. *Drone* yang dikendalikan secara manual, diarahkan sesuai yang *user* inginkan. Berikut gerakan dasarnya *Throttle height*, gerak untuk mengatur semua kecepatan *propeller* menaikkan ataupun menurunkan pada jumlah yang sama. *Roll*, gerak untuk mengatur kecepatan *propeller* menambah ataupun mengurangi dari salah satu *propeller* yang berlawanan arah jarum jam. *Pitch*, gerak yang mengatur kecepatan *propeller* menambah ataupun mengurangi dari salah satu *propeller* yang searah dengan jarum jam saja. Sedangkan *Yaw*, gerak yang mengatur kecepatan *propeller* menambah ataupun mengurangi yang berlawanan arah jarum jam dan yang searah jarum jam secara bersama [2]. Dari gerakan dasar tersebut terdapat perbedaan antara *singlecopter* dan *quadcopter*, yaitu *singlecopter* tidak memiliki gerakan *throttle height*. Sehingga *drone quadcopter* bisa lebih menjaga keseimbangan ketika *takeoff* ataupun *landing* dan lebih kecil kemungkinan untuk sulit dikendalikan dibanding *singlecopter*.

Sistem *autonomous quadcopter* memiliki kemampuan untuk memperkecil pengendalian langsung dari *user*. Sistem ini di desain terutama untuk bekerja di area *outdoor* [3]. Sehingga mudah untuk menerbangkannya dan mendaratkannya. Mengatur program perintah dari terbangnya *drone* hingga mendarat dengan aplikasi yang telah dirancang pada *smartphone* berbasis android. Karena, *smartphone* yang berbasis android lebih banyak memiliki keunggulan sebagai software, yaitu menggunakan kode komputer yang open source sehingga dapat kustom aplikasi buatan sendiri didalamnya dan selain itu, aplikasi native google sudah terdapat dalam android tersebut sehingga bisa secara langsung terintegrasi seperti Google Maps, Google Calendar dan Gmail [4]. Bahasa pemrograman utama yang digunakan yaitu Java Android yang tidak secara langsung memproses ke paket UDP dan memiliki delay yang kurang dari 140ms[5].

Pada penelitian sebelumnya *drone* menggunakan *google maps* dan diterbangkan secara *autonomous* sesuai dengan *waypoint* yang ditentukan oleh *user* pada aplikasi android namun perintah terbang *drone* dilakukan secara *manual* dengan menggunakan aplikasi *putty* pada laptop dan menambahkan *raspberry* pada *drone* yang berkomunikasi langsung dengan mikrokontroler pada *drone*, agar dapat lebih mudah komunikasi pada *smartphone* [6]. *Drone* terbang *autonomous* mulai dari *takeoff* sampai dengan *landing*. Pengujian yang lain *drone quadcopter* dapat terbang sesuai tujuan menggunakan metode *fuzzy logic control* dan sensor *ultrasonic* yang berfungsi untuk membaca jarak *drone* dengan objek agar dapat mudah menghindari *accident/crash*, tetapi GPS mikrokontroler kurang presisi dan akurat [7]. Maka dari itu akan diteliti perbaikan dari versi yang ada sebelumnya. Penelitian ini akan menggunakan *smartphone* berbasis android sebagai *monitoring tracking* dan *IoT (Internet of Things)* dimana *IoT* tersebut berupa jaringan internet yang bisa mengkomunikasikan benda-benda elektronik dan virtual yang ada disekitar, contohnya *user* dan *UAV* jenis *drone* tersebut. Sehingga *remote control* mempunyai tambahan fitur *tracking* dengan *smartphone* sebagai monitornya. Dan menggunakan mikrokontroler yang memiliki GPS lebih presisi dan dapat menghasilkan fitur *tracking* yang akurat.

Fitur *tracking* berfungsi untuk mengetahui keberadaan *drone* pada saat terbang hingga mendarat, fitur tersebut dapat dilihat di aplikasi yang akan dirancang dan dapat mempermudah *user* dalam memantau *drone* dari jarak jauh.

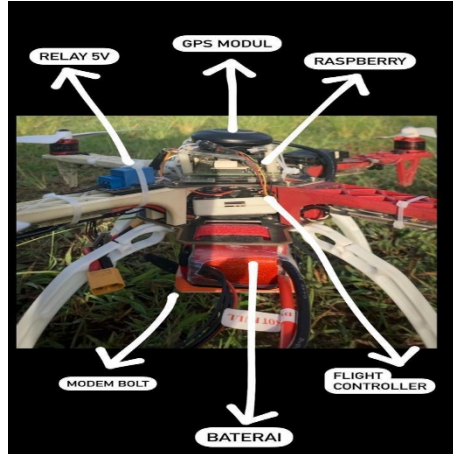
Berdasarkan latar belakang tersebut, dilakukan penelitian dengan membuat aplikasi pada *smartphone* berbasis android agar dapat berfungsi sebagai pengganti *remote* pada *drone* yang secara *online* terhubung dengan *IoT* dan

menambah fitur *tracking*. Dengan android dan *IoT* diharapkan dapat mengurangi dan mempermudah *user* menggunakan *drone*.

2. METODE

2.1. Perakitan Hardware dan Mekanik

Bentuk drone sebagaimana umumnya tipe quadcopter, berikut gambar 1 adalah gambar drone dan menjelaskan beberapa tata letak komponen.

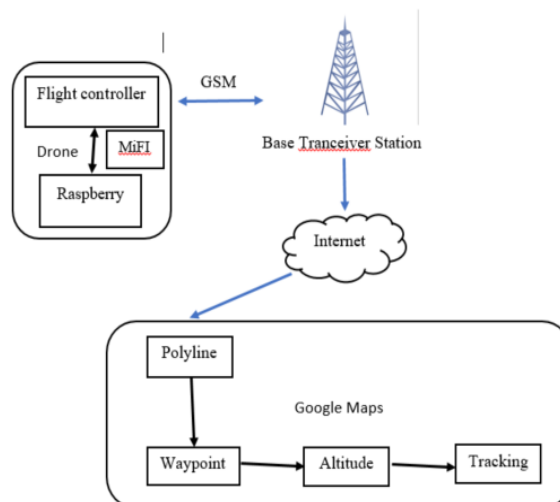


Gambar 1. Tampak drone dari permukaan depan

Gambar 1 merupakan gambar Drone yang telah dirakit yang terdiri dari :

1. *Propeller* / Baling-baling,
2. *Frame*/ Rangka yang berbentuk huruf "X",
3. *Relay 5V*, sebagai saklar *drone*,
4. *Gps Module*, untuk aplikasi *tracking*,
5. *Flight controller* / Pengendali penerbangan,
6. *Modem bolt*, sumber internet untuk *drone*,
7. *Baterai*, sumber daya pada *drone*,
8. *Raspberry*, *controller* kedua pada *drone* berfungsi untuk menerima sinyal internet dan mengirim data *gps* ke *user*.

Pada Gambar 2 dapat dijelaskan , *Drone* yang dipasang modem *mifi GSM* yang terkoneksi dengan internet melalui *Base Transceiver Station (BTS)* mengirim data ke agar *user* dapat memonitoring posisi *drone* secara *realtime*, adapun data yang dikirim berbasis *GPS* berupa *Polyline*, objek yang tersusun atas garis sebagai satu objek tunggal, sedangkan *Waypoint*, titik koordinat untuk sebuah navigasi dan *Altitude* / Ketinggian.



Gambar 2. Blok diagram logic sistem berbasis *IoT*

Dan beberapa rincian komponen pada tabel 1 berikut.

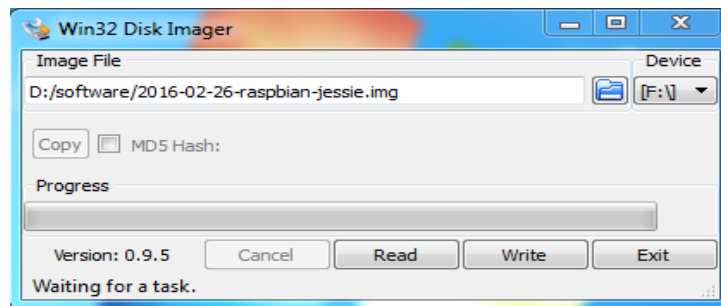
Tabel 1. Perincian Komponen Quadcopter

Komponen	Tipe
Flight Controller	Pixhawk
Motor Brushless	2212/920KV
Propeller	9450
Electronic Speed Controller	30A Opto
Frame	F450
Battery	3S 6200MAh 50C
Mikrokontroler	Raspberry Pi 3B
Relay	5V
Modem	Bolt
Telemetry	V8FR-II
GPS Module	Ublox 7M
Remote Control	Turnigy 9XR Pro

2.2. Penkonfigurasi Raspberry

Raspberry Pi adalah mikrokontroler sebagai fungsi sama seperti komputer pada umumnya dengan beberapa tambahan fitur GPIO dan modul wifi [8]. *Raspi* berfungsi sebagai mikrokontroler setelah di install dengan langkah-langkah berikut ini:

1. Menginstall SD card setelah itu menjalankan aplikasi *SD formatter*. Jika *raspi* sudah digunakan sebelumnya, klik *format*, karena *SD formatter* memiliki fungsi membersihkan dari file sebelumnya..
2. Menjalankan *Win32 Disk Imager*, kemudian meneliti ketika memilih *device* yang sesuai dengan kartu memori SD karena jikalau *device* yang dipilih salah maka dapat menghapus data pada kartu memori SD tersebut. Langkah selanjutnya klik “write” untuk menginstall *raspbian* dan menunggu beberapa menit sampai proses install selesai. Seperti pada gambar 3.

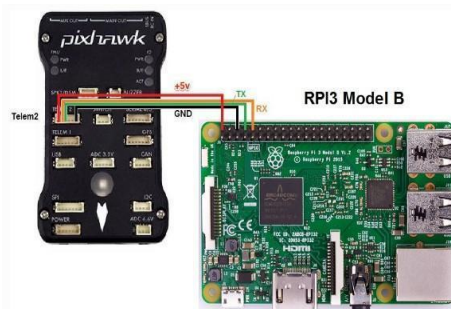


Gambar 3. Proses Penginstallan Raspbian Image Pada Aplikasi Win32 Disk Imager.

3. Menginstall dengan klik *install raspbian*, selanjutnya memasukkan kartu memori SD pada *raspi* dan *raspi* siap untuk digunakan.

2.3. Pengaturan Sistem Komunikasi Raspberry Pi Via Mavlink

Komunikasi antara *pixhawk* dan *raspi*, dapat terjadi dengan cara menyambungkan port telem 2 pada *pixhawk* ke *raspi*. seperti pada gambar 4 dibawah ini.

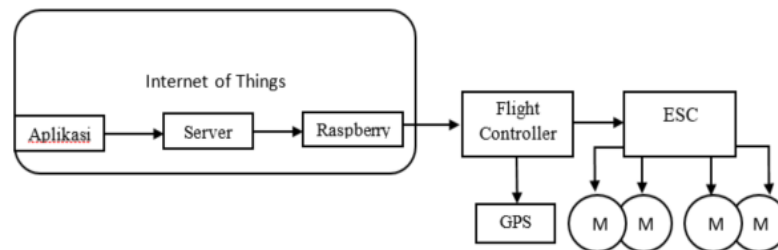


Gambar 4. Wiring Raspi Dengan Pixhawk

Gambar 4 diatas TELEM 2 pada *pixhawk* dihubungkan ke pin TX, RX dan *Ground raspi*. Dan untuk power 5V dihubungkan melalui *USB port* dengan adaptor dikarenakan *power* yang didapat dari *pixhawk* itu mencukupi untuk kebutuhan *raspberry*.

2.4. Perancangan Komunikasi IoT

Perancangan komunikasi *internet of things (IoT)*. *IoT* adalah sebuah konsep untuk memperluas manfaat internet, dimana benda-benda elektronik dan *virtual* yang ada disekitar dapat berkomunikasi satu sama lain melalui jaringan internet [9]. Gambar 5 dibawah menjelaskan blok diagram kerja *drone* dengan menggunakan aplikasi yang ada pada *smartphone*.



Gambar 5. Blok Diagram IoT

Drone dapat diterbangkan dengan perintah aplikasi *smartphone*, karena *smartphone* disini berfungsi sebagai *planner* atau yang membuat misi. Penghubung komunikasi antara *smartphone* dengan *drone* yaitu internet yang menggunakan jaringan data atau *provider*. *Mifi* pada *drone* memancarkan sinyal *wifi* yang diterima *raspi* karena *Raspi* tersebut sebelumnya sudah di konfigurasi agar dapat menerima sinyal *wifi* yang dipancarkan oleh *mifi*. IP yang dimiliki *mifi* berupa *dynamic* atau berubah-ubah, maka dari itu dibutuhkan *server* yang memiliki IP *static*. *Server* berfungsi sebagai pihak ketiga yang menghubungkan aplikasi dan *drone*. Dalam penelitian ini menggunakan *server hosting* dan *domain* berbayar dari Niagahoster. jaringan yg sama.

2.5. Perancangan Aplikasi Android

Aplikasi mampu dijalankan pada sebagian besar versi jenis android sehingga para responden akan dengan mudah untuk memasang dan menjalankan aplikasi [10]. Tahap pertama dalam pembuatan aplikasi berbasis *android* adalah dengan pembuatan *design* terlebih dahulu. Aplikasi ini yang dibuat menggunakan *software ionic* yang dapat mendukung penggunaan fitur *GPS*, Senter, Kamera dan lainnya yang terdapat di *smartphone*. Dan pada perancangan ini membutuhkan fitur *GPS*, dan beberapa fungsi *on/off drone*, *start/stop tracking*, riwayat, dan menutup aplikasi.

2.5.1. Fitur Aplikasi

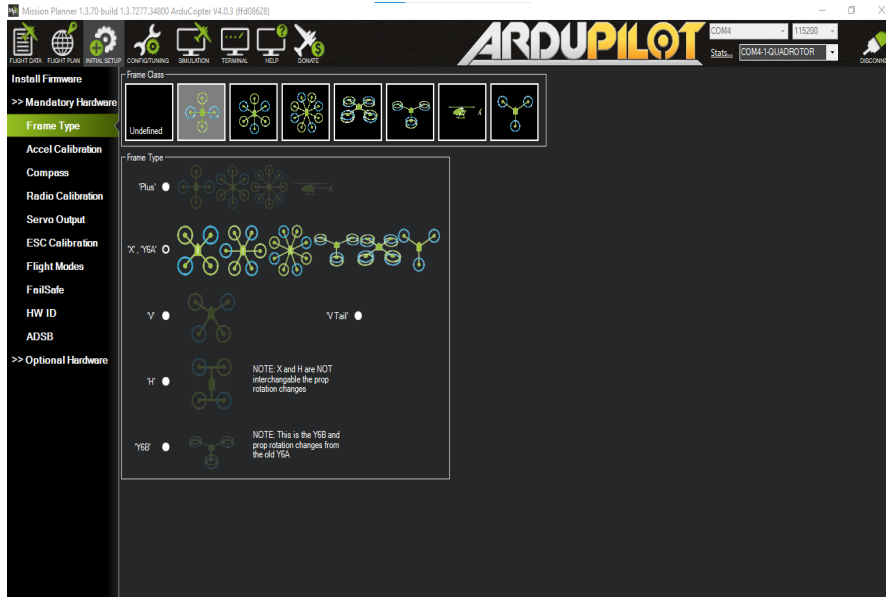
Tabel 2 ini dibuat untuk menjelaskan fungsi dari tombol yang terdapat di aplikasi yang telah dirancang.

Tabel 2. Tabel Antarmuka

<i>Fitur</i>	<i>Fungsi</i>
On/Off	Menghidupkan dan mematikan drone
Start/Stop Tracking	Memulai dan berhenti dari fitur tracking.
Riwayat	Menampilkan riwayat pada tracking.
Keluar	Menutup aplikasi

2.5.2. Mekanisme Autonomous

Sistem *autonomous* ini sudah terdapat pada *flight controller* tersebut. Setelah *drone* dirakit *step* selanjutnya dapat di sambungkan ke *mission planner* lalu ke *mandatory hardware* untuk mengkalibrasi beberapa komponen *drone* tersebut. Seperti gambar 6 berikut.



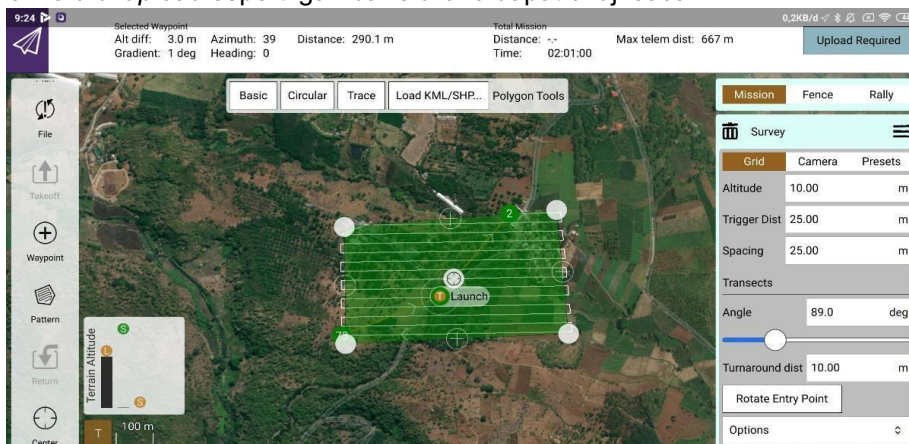
Gambar 6. Mandatory Hardware

Cara kalibrasi beberapa komponen tersebut hanya dengan mengikuti perintah pada aplikasi mission planner dan setelah kalibrasi beberapa komponen dari tipe *frame* sampai dengan *esc calibration*, selanjutnya bisa menentukan *flight mode* yang ingin di input ke channel *remote control*. Seperti pada gambar 7 di bawah ini.



Gambar 7. Setting Mode Autonomous.

Selanjutnya *drone* bisa di sambungkan ke *smartphone* melalui kabel *usb* dan membuat misi *autonomous* di aplikasi *QGroundControl* pada *smartphone*. Setelah semua *altitude* 10 meter dan kecepatan 1 meter/second sudah diatur sesuai keinginan lalu di *upload* seperti gambar 8 *drone* dapat di uji coba.



Gambar 8. Membuat Misi Autonomous.

Setelah misi diupload, *drone* sudah dapat terbang dengan misi *autonomous* dari *take off* serta *landing* tanpa bantuan *remote control*.

3. HASIL DAN PEMBAHASAN

Pengujian ini bertujuan agar dapat terlihat respon dari apa yang telah dirancang dari *drone* maupun aplikasi *tracking* pada *smartphone* bahwa dapat hasil yang sesuai dengan perancangan..

3.1. Pengujian Aplikasi Android

Pada aplikasi yang telah dirancang tentu saja memiliki beberapa fitur yang tujuannya untuk mempermudah *user*. Berikut tabel 3 adalah beberapa fitur yang terdapat dalam aplikasi tersebut.

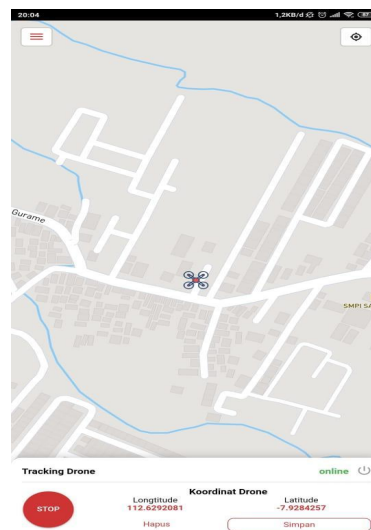
Tabel 3, Tabel Antarmuka

Jenis Fitur	Status
On/Off	Berfungsi
Start/Stop Tracking	Berfungsi
Simpan/Hapus Tracking	Berfungsi
Riwayat Tracking	Berfungsi

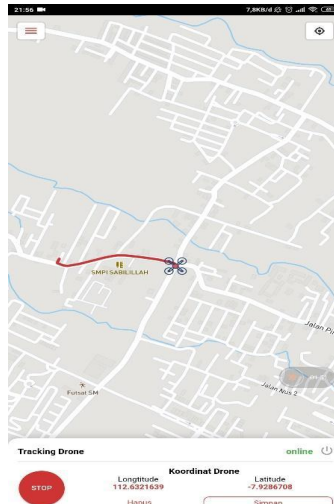
Dapat dilihat dari hasil percobaan pada aplikasi yang telah dirancang mungkin bisa dikatakan dapat berjalan sesuai dengan tujuan yang telah ditulis di BAB 1 dan juga menu yang dibutuhkan semua berjalan dengan sesuai, seperti *on*, *off*, *start stop tracking*, *simpan* dan *hapus riwayat tracking*.

3.2. Pengujian Fitur Tracking Quadcopter Pada Aplikasi Smartphone

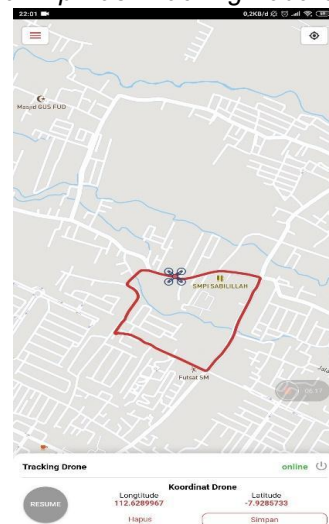
Pada aplikasi ini sangat bergantung penuh dengan jaringan dari kedua pihak, yaitu antara *drone* dan *smartphone*. Apabila dari perangkat *drone* memiliki jaringan yang memiliki potensi maka *tracking* dapat berjalan baik dan aplikasi pada *smartphone* pun juga begitu. Tetapi, dalam tugas akhir ini sudah diperkirakan dan menggunakan jaringan *provider* yang memiliki potensi baik. Berikut adalah gambar 9, 10, 11 adalah hasil dari *tracking* tersebut.



Gambar 9. Tampilan Aplikasi Tracking Pada Drone Pada Saat Start.



Gambar 10. Tampilan Aplikasi Tracking Pada Saat Drone Terbang.



Gambar 11. Tampilan Aplikasi Tracking Saat Kembali Ke Titik Awal.

Dari ketiga gambar diatas sudah terlihat jelas bahwa aplikasi *tracking* pada *drone* ini dapat berjalan sesuai dan dapat secara *realtime*.

3.3. Pengujian Daya Jangkau Quadcopter Dengan User

Pada uji daya jangkau *drone quadcopter* tersebut bergantung dari *remote control* dan *telemetry* yang digunakan. Pada pengujian ini, jarak jangkau yang diperoleh sekitar 100 sampai dengan 200 meter jika di *control manual* dikarenakan *remote control* dan sistem *telemetry* yang digunakan bekerja pada *frekuensi* yang cukup tinggi namun daya pancar yang kecil sehingga daya jangkaunya tidak dapat jauh. Daya jangkau tersebut tidak termasuk jika menggunakan misi *autonomous*, karena dalam misi *autonomous* sendiri *drone quadcopter* sudah dapat berjalan mengerjakan misi walaupun *remote* keadaan *off*. Tabel 4 berikut menjelaskan jarak dari daya jangkau *drone* tersebut.

Tabel 4. Tabel Uji Daya Jangkau

<i>Daya Jangkau</i>	<i>Status</i>
100 meter	Berhasil
200 meter	Berhasil
250 meter	Tidak

3.4. Pengujian terbang Drone Secara Autonomous

Tahapan ini untuk mengetahui *drone* dapat menjalankan misi *Autonomous* tersebut. Namun, pada pengujian tersebut terdapat satu kondisi yang tidak sesuai dengan *rule* yang ditentukan yaitu ketika *drone* menuju titik *waypoint* pertama terjadi *crash landing* secara tiba-tiba. Berdasarkan analisa kegagalan penerbangan *drone* secara *Autonomous* disebabkan karena, berat *drone* yang melebihi dari kemampuan bila menggunakan misi *autonomous* adapun kapasitas

berat pada *drone* ini hanya 1863 *gram* dan berat *drone* 2500 *gram*. Batas kapasitas *drone* didapat dari dorongan motor karena setiap motor punya daya dorong (*trust*), misalkan

$$\begin{aligned} \text{Trust motor } 621\text{gr} \times 4 &= 2484\text{gr} \\ \text{Maka berat maksimal } \textit{drone} &\frac{3}{4} \times 2484\text{gr} = 1863\text{gr}. \end{aligned}$$

Selain itu juga alat yang digunakan sudah tidak seutuhnya berfungsi *normal* dan yang jadi penyebab terakhir pengemudi masih dapat dikatakan pemula dan masih belum paham akan banyak fungsi dari *drone quadcopter* tersebut. Sehingga dapat dikatakan gagal pada misi *autonomous* tersebut. Berikut tabel 5 adalah keterangan dari percobaan.

Tabel 5. Tabel Keterangan Status

Misi	Berat	Status
Manual	2500 gr	Berhasil
Autonomous	2500 gr	Gagal

4. KESIMPULAN

Dari pengujian maka didapatkan kesimpulan pada aplikasi, semua menu di aplikasi berjalan dengan sesuai perintah dan tracking dapat berjalan secara realtime. Dan *drone*, dalam misi *autonomous* seharusnya *drone* terbang sesuai waypoint akan tetapi ketika menuju waypoint pertama sering kali terjadi crash landing atau kecelakaan. Sehingga dapat disimpulkan dalam uji coba *autonomous* tersebut belum sepenuhnya sempurna.

Daftar Pustaka

- [1] Pradana, M. G. A., Prasakti, R., Worsito, S. B., dan Fajaryati, N., "Single Propeller Drone (Singrone): Inovasi Rancang Bangun Drone Single Propeller Sebagai Wahana Pemetaan Lahan Berbasis Unmanned Aerial Vehicle (UAV)," *J. Electron. Informatics, Vocat. Educ.*, vol. 1, no. November, pp. 1–6, 2016.
- [2] Hamdani, C. N., Rusdhianto, E. A. K., dan Iskandar, E., "Perancangan Autonomous Landing pada Quadcopter Menggunakan Behavior-Based Intelligent Fuzzy Control," *J. Tek. POMITS*, vol. 2, no. 2, pp. 63–68, 2013.
- [3] Latif, M., dan Hairil, B., "Perancangan Sistem Autonomous Quadcopter," *Progr. Stud. Mekatronika*, no. November, pp. 1–5, 2014.
- [4] Zulkarnain, T., Susanto, E., dan Wibowo, A. S., "Perancangan Dan Implementasi Autonomous Quadcopter Dengan Kemampuan Follow Me Yang Terintegrasi Pada Android," *e-Proceeding Eng.*, vol. 3, no. 2, pp. 1336–1343, 2016.
- [5] Nathanael, R., Widodo, S., Santoso, P., dan Khoswanto, H., "Aplikasi Android Untuk Remote Control Quadcopter," *J. Dimens. Tek. Elektro*, vol. 1, no. 1, pp. 7–11, 2013.
- [6] Pranata, A., Zhulfatman, dan Alif, N., "Perancangan Sistem Pengendalian Drone Quadcopter Secara Autonomous Berbasis Aplikasi Android," *Artik. Ilm. Tek. Elektro*, vol. xx, no. xx, pp. 103–113, 2018.
- [7] Turoha, D. H., Zhulfatman, dan Alif, N., "Perancangan Dan Implementasi Obstacles Collision Avoidance Pada Drone Quadcopter Menggunakan Fuzzy Logic Control," *AITEL Artik. Ilm. Tek. Elektro*, 2018.
- [8] Purnomo, I. D., Murti, M. A., dan Susanto, E., "Rancang Bangun Aplikasi Networked Control System (Ncs) Pada Unmanned Aerial Vehicle (Uav) Quadcopter Untuk Kontrol Kestabilan Posisi," *e-Proceeding Eng.*, vol. 2, no. 3, pp. 6919–6926, 2015.
- [9] Kusumaningrum, A., Pujiastuti, A., dan Zeny, M., "Pemanfaatan Internet Of Things Pada Kendali Lampu," vol. 6, no. 1, pp. 53–59, 2015.
- [10] Deta, G. U. H., Kumalasari, E., dan Hamzah, A., "Aplikasi Mobile E-Voting Studi Kasus Di-Institute For Research And Empowerment (IRE) Yogyakarta Berbasis Android," *J. Scr.*, vol. 3, no. 2, pp. 79–92, 2016.