

## ABSTRAK

**Jamil Febrian : RANCANG BANGUN SISTEM KONTROL  
PENYEMPROTAN CAIRAN PESTISIDA OTOMATIS  
MENGUNAKAN *DRONE* UAV *HEXACOPTER***

Perkembangan teknologi di zaman revolusi industri 4.0 telah berkembang dengan pesat, yang dulunya dilakukan dengan manual hingga saat ini dapat dilakukan secara otomatis. Teknologi *drone* sudah menjadi teknologi yang bisa digunakan dalam bidang apapun, salah satunya pada bidang pertanian dalam melakukan penyemprotan cairan pestisida. Penelitian ini bertujuan untuk membuat dan mengembangkan sistem penyemprotan cairan pestisida secara otomatis menggunakan *drone* UAV *Hexacopter*. *Drone* ini menggunakan 6 motor brushless 1400kV dan *propeller* 1045 sebagai unit penggerak, *Drone* ini menggunakan Pixhawk sebagai *flight controller* dan dilengkapi dengan sistem navigasi GPS untuk penerbangan secara otomatis. Pada penelitian ini dilakukan serangkaian pengujian dan analisis terhadap kemampuan gaya angkut *drone* dan efektivitas sistem penyemprotan. Metode pengujian meliputi simulasi beban dan pengamatan langsung terhadap proses penyemprotan di lapangan. Hasil pengujian yang dilakukan drone mampu mengangkat maksimal beban sebesar 1,5kg beban cairan dan melakukan penyemprotan selama 60 detik di udara sejauh 6 meter sepanjang jalur penyemprotan. Hasil pengujian yang diperoleh memberikan pemahaman yang mendalam terkait performa dan potensi penggunaan *drone hexacopter* dalam aplikasi penyemprotan pestisida secara otomatis. Penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam pengembangan teknologi pertanian yang lebih efisien dan berkelanjutan di masa mendatang.

**Kata kunci :** *Drone, Hexacopter, Flight Controller, Motor Brushless, Propeller, Cairan Pestisida.*

## KATA PENGANTAR

Alhamdulillahirabbil 'alamin, puji syukur kita ucapkan kepada Allah SWT yang telah memberikan rahmat dan karunia-Nya sehingga Penulis dapat menyelesaikan tugas akhir ini dengan judul **“RANCANG BANGUN SISTEM KONTROL PENYEMPROTAN CAIRAN PESTISIDA OTOMATIS MENGGUNAKAN *DRONE* UAV *HEXACOPTER*”**. Tugas Akhir ini disusun untuk memenuhi syarat menyelesaikan Program Studi Strata 1, Departemen Teknik Elektronika, Fakultas Teknik, Universitas Negeri Padang.

Penyelesaian proposal tugas akhir ini tidak terlepas dari bantuan dan bimbingan serta dorongan berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih kepada :

1. Allah Subhanahu Wata'ala yang telah memberikan rahmat dan hidayahnya dalam penulisan Tugas Akhir ini.
2. Kedua Orang Tua dan semua keluarga yang telah banyak berjasa baik moral ataupun materil serta doa sehingga Penulis dapat menyelesaikan tugas akhir ini
3. Bapak Ir. Krismadinata, S.T., M.T, Ph.D selaku Dekan Fakultas Teknik Universitas Negeri Padang.
4. Bapak Dr. Hendra Hidayat, S.Pd., M.Pd. selaku Ketua Departemen Teknik Elektronika Fakultas Teknik UNP, selaku Ketua Program Studi Pendidikan Teknik Elektronika
5. Ibu Sartika Anori, S.Pd., M.Pd.T selaku dosen Pembimbing Akademik
6. Bapak Dr. Yasdinul Huda, S.Pd., M.T. selaku pembimbing yang telah memberikan arahan dan bimbingan dalam pengerjaan Tugas akhir ini.

7. Bapak dan Ibu dosen serta seluruh staf Jurusan Teknik Elektronika Fakultas Teknik Universitas Negeri Padang yang telah membimbing dan membantu Penulis selama menuntut ilmu.
8. Bapak Purwantono Selaku Komandan TNI Landasan Udara Sutan Sjahrir Padang yang telah mengizinkan untuk melakukan penelitian di Lapangan Lanud Sutan Sjahrir Padang.
9. Rekan-rekan mahasiswa Jurusan Teknik Elektronika UNP, khususnya program studi Pendidikan Teknik Elektronika angkatan 2019 yang telah memberikan dukungan moral ataupun materil serta do'a sehingga Penulis dapat menyelesaikan Tugas Akhir ini.
10. Keluarga besar Unit Kegiatan Robotika Universitas Negeri Padang, khususnya rekan-rekan seperjuangan di tim Robot Terbang Kuau King UNP dan teman-teman serta adik-adik yang telah membantu dan memberika dukungan.
11. Semua pihak yang telah membantu Penulis untuk mewujudkan tugas akhir ini dan menyelesaikan studi yang tidak bisa penulis sebutkan satu per satu.

Semoga bantuan dan bimbingan serta arahan yang diberikan menjadi amal shaleh dan mendapatkan pahala dari Allah SWT, aamiin. Tugas Akhir ini tidak terlepas dari kesalahan dan kekeliruan, oleh sebab itu penulis mengharapkan saran dan kritik yang bersifat membangun. Semoga Tugas Akhir ini dapat bermanfaat bagi kita semua. Aamiin...

Padang, 29 Februari 2024  
Penulis,

Jamil Febrian

## DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iv
DAFTAR TABEL.....	vi
BAB I.....	1
A. Latar Belakang .....	1
B. Identifikasi Masalah .....	4
C. Pembatasan Masalah.....	5
D. Perumusan Masalah .....	5
E. Tujuan Penelitian .....	6
F. Manfaat Penelitian .....	6
BAB II .....	8
A. Sistem Kontrol .....	8
B. Drone Pertanian.....	9
C. Perangkat Kontrol Penyemprotan Cairan Pestisida.....	10
D. Bahasa Pemrograman <i>Python</i> dan <i>Flask Framework</i> .....	19
E. Metode Pengembangan Sistem .....	19
F. Penelitian Relevan .....	22
BAB III.....	24
A. Analisis Kebutuhan ( <i>Requirement</i> ) .....	24
B. Desain Rancangan Sistem ( <i>Design</i> ).....	30
C. Alur Pembuatan Sistem .....	37
BAB IV .....	40
A. Perancangan Sistem Kontrol Penyemprotan Cairan Pestisida Menggunakan <i>Drone UAV Hexacopter</i> .....	40
B. Integrasi Sistem Kontrol Penyemprotan Pestisida Menggunakan Drone UAV Hexacopter .....	46
C. Implementasi Sistem Kontrol Penyemprotan Pada <i>Drone UAV         Hexacopter</i> .....	54
BAB V.....	67
A. Kesimpulan.....	67
B. Saran .....	67
DAFTAR PUSTAKA .....	69

## DAFTAR GAMBAR

Gambar 1 Blok Diagram Sistem Kontrol.....	8
Gambar 2 Aplikasi Drone Pada Pertanian .....	9
Gambar 3 Pixhawk.....	11
Gambar 4 Bagian dan Port Konektor Pixhawk .....	12
Gambar 5 Pixhawk Digital Input dan Output .....	14
Gambar 6 Electronics Speed Controller (ESC).....	14
Gambar 7 Brushless DC Motor.....	15
Gambar 8 Jetson Nano .....	16
Gambar 9 Relay 2 Channel .....	17
Gambar 10 Waterpump DC Motor .....	18
Gambar 11 Nozzle Sprayer .....	18
Gambar 12 Siklus Metode Waterfall.....	21
Gambar 13 Activity Diagram Sistem.....	29
Gambar 14 Diagram Blok Sistem .....	31
Gambar 15 Desain Rancangan Alat. ....	32
Gambar 16 Web Control Interface .....	34
Gambar 17 Flowchart Perencanaan Misi Drone .....	35
Gambar 18 Alur Pembuatan Sistem.....	37
Gambar 19 Tampilan Fisik Alat.....	41
Gambar 20 Posisi Komponen Bagian Bawah Drone .....	41
Gambar 21 Posisi Komponen Bagian Atas Drone .....	42
Gambar 22 Letak Komponen Bagian Lengan Drone.....	42
Gambar 23 Tampilan Web Kontrol Penyemprotan .....	43
Gambar 24 Bagian Kontrol Penyemprotan pada drone .....	44
Gambar 25 Bagian Monitoring Data Sensor Pada Drone .....	46
Gambar 26 Rangkaian Sistem Penyemprotan.....	47
Gambar 27 Pengukuran Tegangan Baterai .....	54
Gambar 28 Pengujian Koordinat GPS .....	60
Gambar 29 Pengujian Tegangan Input pada waterpump DC.....	61
Gambar 30 Aliran Penyemprotan Cairan .....	62
Gambar 31 Pengujian Daya Angkat Drone hexacopter .....	63
Gambar 32 Pengujian Mekanisme Penyemprotan .....	64

## DAFTAR TABEL

Tabel 1 Analisis Kebutuhan Sistem .....	25
Tabel 2 Analisis Kebutuhan Alat dan Bahan .....	26
Tabel 3 Pengukuran Tegangan Port Pin GPS Pixhawk .....	55
Tabel 4 Pengukuran Tegangan Port Pin I2C Pixhawk.....	57
Tabel 5 Pengukuran Tegangan pin GPIO Jetson Nano.....	58
Tabel 6 Hasil Pengujian GPS .....	60
Tabel 7 Tegangan Output Pin AUX5 Pixhawk.....	61
Tabel 8 Hasil Pengukuran Debit Air .....	62
Tabel 9 Hasil Pengujian Daya Angkat Drone Hexacopter.....	63

# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang**

Indonesia merupakan negara yang dikenal sebagai negara yang kaya akan sumber daya alam, salah satunya di sektor pertanian yang menjadi mata pencarian bagi masyarakat Indonesia (Kusumaningrum, 2019). Petani merupakan pelaku utama dalam sektor pertanian yang berperan penting dalam mewujudkan sumber ketersediaan pangan. Pertanian modern menghadapi berbagai tantangan, salah satunya adalah efektivitas dan efisiensi dalam pengendalian hama. Hama tanaman merupakan organisme atau makhluk hidup yang bersifat merusak dan mengganggu pertumbuhan tanaman, sehingga berdampak pada kualitas atau kuantitas hasil tanaman (Feriadi, 2017). Serangan hama tersebut mengakibatkan terjadinya gagal panen sehingga menyebabkan kerugian yang sangat besar bagi para petani (Astuti & Widyastuti, 2016).

Kehadiran hama sangat tidak diinginkan dalam kegiatan sehari-hari manusia sehingga hama tanaman perlu dibasmi dengan menggunakan pestisida sebagai salah satu cara yang umum dilakukan untuk mengatasi masalah hama tanaman agar tanaman dapat tumbuh dengan baik. Namun, metode konvensional penyemprotan pestisida seringkali kurang efektif dan memerlukan banyak tenaga kerja, waktu, dan biaya yang tidak sedikit. Hingga saat ini petani menghabiskan sumber daya tenaga manusia yang cukup besar untuk melakukan proses penyemprotan cairan pestisida pada area yang cukup luas.

Penggunaan cairan pestisida dilakukan untuk menjaga dan meningkatkan hasil panen dari serangan hama tanaman (Hidayat et al., 2019). Namun proses penyemprotan cairan pestisida yang dilakukan secara manual memiliki beberapa dampak negatif seperti cairan pestisida yang terkena kulit dapat menyebabkan gatal-gatal atau iritasi, karena cairan pestisida mengandung zat kimia yang berbahaya (Widiasari & Este Dulan Agustinus S., 2020). Kecelakaan akibat pestisida yang sering dialami seperti pusing-pusing ketika sedang menyemprot maupun setelah penyemprotan, muntah-muntah, mulas, iritasi mata, kulit terasa gatal, kejang-kejang, pingsan, dan tidak sedikit kasus akan berpotensi pada kematian. Penyemprotan secara manual juga berpotensi merusak tanaman karena dalam proses penyemprotan banyak tanaman yang terinjak. Kondisi tersebut menjadikan perlu adanya inovasi untuk mengurangi risiko akibat kontak fisik petani dengan penggunaan cairan pestisida dan juga untuk mencegah tanaman yang terinjak saat melakukan penyemprotan cairan pestisida yang lebih efisien, aman, dan ramah lingkungan. (Widiasari & Este Dulan Agustinus S., 2020)(Hidayat et al., 2019).

Hingga saat ini perkembangan teknologi pada zaman revolusi industri 4.0 telah berkembang pesat, seperti yang dulunya dilakukan secara manual dan membutuhkan efisiensi waktu yang cukup lama, hingga saat ini dapat dilakukan dengan lebih cepat secara otomatis dengan penggunaan robot terbang tanpa awak yang disebut dengan *drone*. *Drone* merupakan kendaraan udara tanpa awak yang dapat terbang di udara dikendalikan dari jarak jauh melalui kendali komputer. Teknologi drone menawarkan solusi inovatif untuk mengatasi masalah hama



tanaman, terutama *drone* tipe hexacopter yang memiliki kemampuan manuver yang tinggi dan stabilitas yang baik sehingga ideal untuk aplikasi di bidang pertanian (Febri et al., 2021). Penggunaan drone untuk penyemprotan pestisida tidak hanya dapat mengurangi kebutuhan tenaga kerja tetapi juga dapat meningkatkan presisi dan efisiensi dalam aplikasi pestisida. *Drone hexacopter* dilengkapi dengan berbagai sensor dan sistem kontrol yang memungkinkan penyemprotan dilakukan secara otomatis dan terprogram sesuai dengan kebutuhan di lapangan.

Pengembangan sistem kontrol penyemprotan cairan pestisida otomatis menggunakan *drone hexacopter* memerlukan perencanaan dan rancang bangun yang matang. Sistem ini harus mampu mengintegrasikan berbagai komponen teknologi seperti GPS untuk navigasi, sensor untuk mendeteksi kondisi lingkungan, dan aktuator untuk mengendalikan penyemprotan cairan pestisida. Selain itu, sistem kontrol harus dapat diprogram untuk mengikuti pola penyemprotan yang efisien dan sesuai dengan kondisi tanaman serta tingkat infestasi hama. Sehubungan dengan itu, penelitian ini bertujuan untuk merancang dan membangun sebuah sistem kontrol penyemprotan cairan pestisida otomatis yang menggunakan drone hexacopter sebagai platform utama. Sistem ini diharapkan dapat memberikan solusi praktis dan inovatif bagi petani dalam mengelola hama dengan cara yang lebih efektif dan efisien. Dengan penerapan teknologi ini, diharapkan dapat tercipta praktik pertanian yang lebih berkelanjutan dan produktif.

Secara keseluruhan, inovasi ini tidak hanya memberikan manfaat dalam hal peningkatan hasil pertanian tetapi juga dapat mengurangi dampak negatif penggunaan pestisida terhadap lingkungan dan kesehatan manusia. Oleh karena itu,

penulis menawarkan alternatif solusi dengan judul “**Rancang Bangun Sistem Kontrol Penyemprotan Cairan Pestisida Otomatis Menggunakan Drone AUV *Hexacopter***“ yang merupakan langkah penting menuju modernisasi pertanian yang lebih cerdas dan berkelanjutan.

## **B. Identifikasi Masalah**

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, dapat diuraikan identifikasi masalah yang terkait dengan penelitian ini :

1. Penyemprotan pestisida secara manual kurang efektif dan membutuhkan banyak tenaga kerja, waktu, dan biaya. Metode manual mengharuskan petani memerlukan sumber daya manusia yang besar untuk menyemprot area yang luas, sehingga berdampak pada produktivitas dan efisiensi kerja petani.
2. Penyemprotan pestisida secara manual memiliki risiko terhadap kesehatan petani. Kontak langsung dengan pestisida dapat menyebabkan berbagai masalah kesehatan seperti iritasi kulit, pusing, muntah, hingga potensi kematian. Kondisi ini menunjukkan perlunya inovasi dalam proses penyemprotan pestisida untuk mengurangi kontak langsung antara petani dengan pestisida.
3. Proses penyemprotan manual berpotensi merusak tanaman karena banyak tanaman yang terinjak selama proses penyemprotan. Hal ini dapat mengurangi kualitas dan kuantitas hasil panen, sehingga diperlukan metode penyemprotan yang lebih efisien untuk mencegah kerusakan tanaman.

### C. Pembatasan Masalah

Berdasarkan latar belakang dan identifikasi masalah di atas, maka dalam perancangan tugas akhir ini penulis membatasi perancangan alat ini yaitu :

1. Tugas akhir ini berfokus pada merancang dan membangun sistem kontrol penyemprotan cairan pestisida secara otomatis dengan menggunakan *drone hexacopter* sebagai media kontrol.
2. Sistem kontrol penyemprotan dibuat menggunakan aplikasi berbasis *website* yang dirancang menggunakan bahasa pemrograman python.
3. Sistem kontrol penyemprotan dirancang mampu dikontrol secara otomatis dan manual oleh user.
4. Sistem kontrol *drone* dirancang untuk terbang di luar ruangan (*outdoor*) pada lahan yang luas tanpa adanya halangan saat penerbangan.

### D. Perumusan Masalah

Berdasarkan latar belakang, identifikasi masalah, dan batasan masalah diatas maka diperoleh rumusan masalah pada tugas akhir ini adalah :

1. Bagaimana merancang sistem kontrol penyemprotan cairan pestisida menggunakan *Drone UAV Hexacopter* ?
2. Bagaimana mengintegrasikan berbagai teknologi seperti GPS, sensor, dan aktuator ke dalam *Drone UAV Hexacopter* ?
3. Bagaimana implementasi sistem kontrol penyemprotan yang dilakukan dengan menggunakan *Drone Hexacopter*?

## **E. Tujuan Penelitian**

Tujuan dari pembuatan tugas akhir ini adalah :

1. Mengetahui proses perancangan sistem kontrol penyemprotan cairan pestisida otomatis dengan menggunakan *Drone Hexacopter*.
2. Mengetahui proses pengintegrasian berbagai teknologi seperti GPS, sensor, dan aktuator ke dalam *Drone Hexacopter*.
3. Mengetahui bagaimana mekanisme kerja sistem penyemprotan cairan pestisida yang dilakukan dengan menggunakan *Drone Hexacopter*.

## **F. Manfaat Penelitian**

Pembuatan tugas akhir ini memiliki manfaat praktis dan manfaat teoritis yang merupakan dua jenis manfaat yang diperoleh dari suatu konsep, teori, atau pengetahuan. Berikut manfaat dari pembuatan penelitian ini adalah :

1. Manfaat bagi petani:
  - a. Mengoptimalkan hasil panen dengan penggunaan pestisida yang lebih tepat dan efisien.
  - b. Mencegah dampak negatif dari resiko penyemprotan cairan pestisida seperti iritasi kulit, pusing, dan keracunan.
  - c. Mencegah kerusakan pada tanaman karena dalam proses penyemprotan secara manual banyak tanaman yang terinjak.
2. Manfaat bagi pengembang sistem :
  - a. Meningkatkan keahlian dalam merancang dan mengembangkan sistem kontrol otomatis, khususnya dalam teknologi drone dan aplikasi pertanian.

- b. Meningkatkan kemampuan dalam berpikir kreatif dan memecahkan masalah kompleks yang melibatkan berbagai aspek teknologi dan aplikasi praktis.
- c. Mengembangkan keterampilan dalam melakukan penelitian yang komprehensif, analisis data, dan evaluasi hasil untuk memecahkan masalah nyata di lapangan.

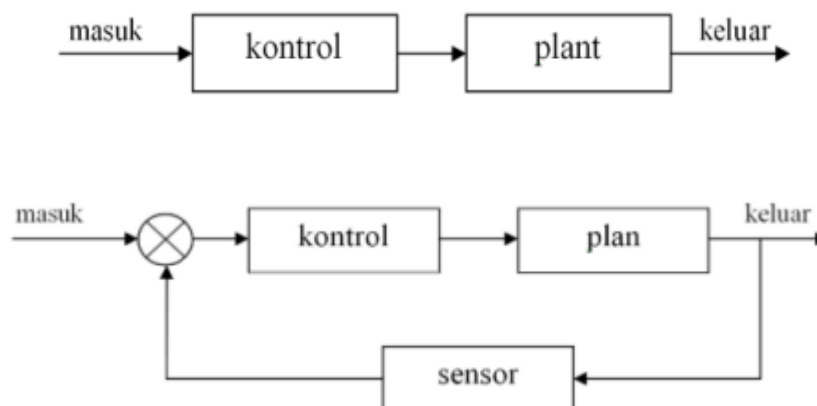
## BAB II

### KAJIAN PUSTAKA

#### A. Sistem Kontrol

Sistem kontrol adalah suatu sistem yang berfungsi untuk melakukan operasi pengendalian terhadap suatu *plant* melalui pengaturan masukannya. Dalam suatu sistem kontrol, kontroler memegang peranan penting dalam pengoperasian sistem.

Sistem kontrol dibagi menjadi 2, yaitu sistem kontrol *loop* terbuka dan sistem kontrol *loop* tertutup. Sistem kontrol *loop* terbuka merupakan sebuah sistem dimana variabel input dari suatu sistem mempengaruhi variabel output dan aksi kontrolnya tidak bergantung dari output sistem. Sistem kontrol *loop* tertutup adalah sistem kontrol dimana variabel output dari sistem diukur dan dibandingkan dengan variabel lain seperti *set point*, sehingga aksi kontrolnya selalu bergantung dari output sistem. Gambar 1 merupakan blok diagram dari Sistem kontrol (a) *loop* terbuka, (b) *loop* tertutup.



*Gambar 1 Blok Diagram Sistem Kontrol*

## B. Drone Pertanian

Pertanian telah menjadi tulang punggung perekonomian diberbagai negara di dunia. Seiring dengan perkembangan teknologi, sektor pertanian juga mengalami transformasi besar. Salah satu inovasi terbaru yang sedang banyak digunakan pada industri pertanian adalah penggunaan *drone* atau pesawat udara tanpa awak (UAV). *Drone* merupakan pesawat tanpa pilot yang dikendalikan secara otomatis melalui program komputer yang dirancang melalui kendali jarak jauh dari pilot yang berada di dataran atau di kendaraan lainnya (Indreswari Suroso, 2021)(Scott & Scott, 2017). *Drone* umumnya dikenal sebagai *Unmanned Aerial Vehicle* (UAV) yang merupakan perangkat terbang yang mampu menerbangi jalur yang ditentukan sebelumnya dengan bantuan autopilot dan koordinat GPS(Almetania & Huda, 2022)(Septian & Huda, 2022).

Drone bidang pertanian mulai diciptakan untuk membantu melakukan perawatan terhadap lahan pertanian yang cukup luas. Penerapan teknologi drone di sektor pertanian digunakan untuk mendukung berbagai aspek, mulai dari pemantauan lahan hingga pengelolaan tanaman.



*Gambar 2 Aplikasi Drone Pada Pertanian*  
(Sumber : faperta.umsu.ac.id)

Keunggulan penggunaan drone di pertanian tidak hanya terbatas pada pemantauan. Seperti pada Gambar 2 *Drone* juga dapat digunakan untuk aplikasi, seperti penyemprotan pupuk atau pestisida. Dengan kemampuan ini, petani dapat mengoptimalkan penggunaan bahan kimia dan mengurangi dampak lingkungan. Selain itu, drone dapat mengakses area yang sulit dijangkau secara manual, seperti lahan yang luas atau daerah dengan topografi yang rumit.

### C. Perangkat Kontrol Penyemprotan Cairan Pestisida

Perangkat penggerak dan elektronik pada drone merupakan komponen-komponen elektronik yang digunakan untuk mengontrol sistem drone agar dapat terbang dengan fungsinya. Perangkat penggerak *drone* terdiri dari baling-baling, motor, dan sistem propulsi yang menghasilkan daya angkat serta sistem pengendalian yang memungkinkan drone untuk bergerak pada pola tertentu, komponen elektronik yang digunakan seperti *flight controller*, baterai, sensor, dan pemrosesan data yang memberikan drone kemampuan untuk menjaga stabilitas, navigasi, serta berkomunikasi dengan pengguna melalui pengontrol jarak jauh atau menggunakan kendali otomatis seperti GPS. Perangkat penggerak dan elektronik yang digunakan diantaranya:

#### 1. *Flight Controller (Pixhawk)*

*Flight Controller* sering juga disebut dengan *FC* merupakan otak dari sebuah pesawat udara tanpa awak. *Flight controller* terdapat terdapat beberapa sensor yang digunakan dalam pemrosesan data untuk melakukan penerbangan, diantaranya sensor *gyroscope*, sensor *accelerometer*, sensor *barometer* dan



sensor *magnetometer* (kompas). Data dari sensor-sensor tersebut diproses oleh *mikrokontroler* di dalam *flight controller* dan kemudian diteruskan ke motor atau baling-baling untuk mengatur gerakan *drone*.

Pixhawk merupakan serangkaian autopilot berbasis *hardware* dan *software* yang digunakan pada aplikasi penerbangan otonom, terutama pada *drone* dan kendaraan udara tanpa awak (UAV). Gambar 3 berikut merupakan tampilan dari Pixhawk.



Gambar 3 Pixhawk  
(Sumber : Ardupilot.org)

Sisi bagian atas permukaan pixhawk terdapat konektor DF13 yang merupakan konektor kunci dalam sistem autopilot dan kontrol penerbangan yang digunakan pada *drone*. Konektor DF13 berfungsi sebagai antarmuka utama untuk menghubungkan *Pixhawk* dengan berbagai sensor, aktuator, dan perangkat tambahan lainnya. Desain konektor DF13 pada Pixhawk yang kokoh dan tahan lama, memastikan pengiriman sinyal yang baik antara autopilot dengan komponen lainnya seperti GPS, kompas, telemetri, baterai, dan motor. Konektor ini memungkinkan integrasi perangkat yang mulus dan meminimalkan risiko kesalahan pemasangan.



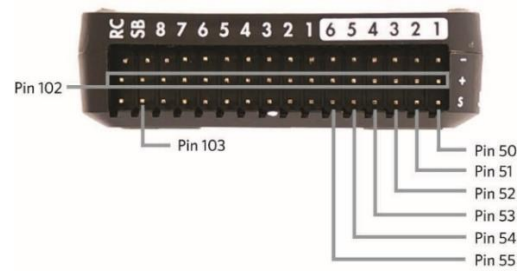
Gambar 4 Bagian dan Port Konektor Pixhawk  
(Sumber : ardupilot.org)

Gambar 4 merupakan tampilan dari port konektor DF13 Pixhawk dan fungsi dari masing-masing *port* konektor pada *Pixhawk* yaitu:

- a. *Power Port* : Port ini digunakan untuk menyediakan daya ke Pixhawk. Biasanya terdiri dari pin positif dan negatif untuk menghubungkan baterai atau sumber daya eksternal yang sesuai dengan kebutuhan daya Pixhawk.
- b. *Telemetry Port* : Port ini digunakan untuk komunikasi dua arah antara Pixhawk dan perangkat luar seperti pengendali jarak jauh (RC), *Ground Controll Station* (GCS), atau sistem monitor. Biasanya terdiri dari pin Tx (*transmit*) dan Rx (*receive*) yang mengirimkan dan menerima data telemetri.
- c. *GPS Port* : Port ini digunakan untuk menghubungkan modul GPS eksternal. Modul GPS menyediakan data lokasi dan navigasi yang penting dalam sistem kontrol penerbangan otonom.
- d. *I2C Port* : Port ini menggunakan protokol komunikasi I2C (Inter-Integrated Circuit) untuk menghubungkan sensor dan perangkat lain yang mendukung I2C. Contoh penggunaan I2C pada Pixhawk adalah untuk menghubungkan sensor tekanan, magnetometer, atau modul sensor lainnya.

- e. *SPI Port* : Port ini menggunakan protokol komunikasi SPI (*Serial Peripheral Interface*) untuk menghubungkan sensor atau perangkat lain yang mendukung SPI. Beberapa perangkat yang mungkin terhubung melalui port ini adalah sensor IMU (*Inertial Measurement Unit*) atau modul sensor lainnya.
- f. *RC Input Port* : Port ini digunakan untuk menghubungkan penerima sinyal radio pada *remote controll* (RC) dari pengendali jarak jauh ke Pixhawk. Ini memungkinkan pengendali RC untuk mengontrol penerbangan *drone* secara langsung.
- g. *Servo Output Ports* : Port ini digunakan untuk menghubungkan aktuator atau servo motor yang menggerakkan permukaan kendali (seperti kendali arah atau kendali *throttle*) pada pesawat atau *drone*.
- h. *Power Distribution Port* : Port ini digunakan untuk menghubungkan dan mendistribusikan daya ke aktuator atau perangkat lain seperti motor ESC (*Electronic Speed Controller*) atau gimbal kamera.
- i. *Serial Ports* : Pixhawk memiliki beberapa port serial tambahan yang dapat digunakan untuk menghubungkan perangkat eksternal, seperti modul radio, sensor tambahan, atau perangkat lain yang memerlukan komunikasi serial.

Pixhawk tidak memiliki pin output atau input digital khusus pada konektor DF13, tetapi terdapat hingga 6 pin konektor "AUX SERVO" sebagai output/input GPIO digital. Pin ini ditandai sebagai pin AUX 1-6 seperti pada Gambar 6.



*Gambar 5 Pixhawk Digital Input dan Output*  
(Sumber : ardupilot.org)

Fungsi AUX pin pada *Pixhawk* adalah untuk memperluas kemampuan pengendalian dan pemantauan sistem, seperti menghubungkan kamera, modul telemetri, sensor, atau perangkat lainnya. Dengan pin AUX, pengguna dapat mengintegrasikan komponen tambahan sesuai kebutuhan dan meningkatkan fungsionalitas dan fleksibilitas sistem.

## **2. *Electronics Speed Controller (ESC)***

*Electronic Speed Controller* (ESC) adalah sirkuit elektronik yang digunakan untuk mengubah kecepatan motor listrik, dan juga berfungsi sebagai rem dinamis, ESC bertindak seperti otak sistem dengan memberi tahu motor seberapa cepat untuk berputar berdasarkan sinyal data yang diterimanya dari pengontrol *throttle* yang bertujuan untuk mengubah kecepatan dan arah manuver *drone*. Bentuk dan tampilan *Electronics Speed Controller* (ESC) terdapat pada Gambar berikut.



*Gambar 6 Electronics Speed Controller (ESC)*  
(Sumber : tytorobotics.com)

ESC bertindak sebagai perantara pengatur antara baterai dan motor listrik. ESC menggunakan arus searah dari baterai yang digabungkan dengan sistem saklar untuk mencapai arus tiga fasa bolak-balik yang dikirim ke motor. ESC mengontrol gerakan atau kecepatan *motor brushless* dengan mengaktifkan MOSFET yang sesuai untuk menciptakan medan magnet yang berputar sehingga akan membuat motor berputar.

### 3. *Brushless DC Motor*

Motor DC adalah motor listrik yang mengubah energi listrik menjadi energi mekanik dengan menggunakan arus listrik DC. Motor DC berfungsi sesuai pada kemampuan untuk berputar pada kecepatan tinggi, dan torsi awal yang tinggi.

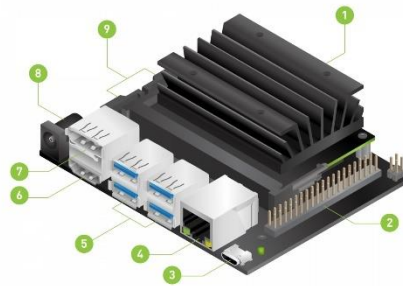


*Gambar 7 Brushless DC Motor*  
(Sumber : andalanelektro.id)

*Motor brushless* juga disebut dengan BLDC atau *BrushLess Direct Current* adalah motor DC yang tidak memiliki sikat atau *brush* untuk mengaliri arus listrik pada setiap kumparan. *Motor brushless* terdiri dari dua bagian utama, yaitu *stator* dan *rotor*. Motor brushless memiliki magnet permanen yang dipasang pada *rotor*, dan *stator* terdiri dari belitan untuk kutub tertentu.

#### 4. Mini Computer (Jetson Nano)

Mini komputer merupakan sebuah komputer yang berukuran kecil yang tersusun oleh beberapa jenis modul. *Jetson Nano* merupakan salah satu perangkat mini computer yang dirancang untuk aplikasi *Internet of Things* (IoT) kecerdasan buatan (*Artificial Intelligence*, AI) dan *computer sains*. Jetson Nano merupakan modul komputer yang dilengkapi dengan berbagai komponen yang memungkinkan pemrosesan data secara real-time, sehingga pengembang mampu melakukan kontrol dan monitorong perangkat secara *realtime*.

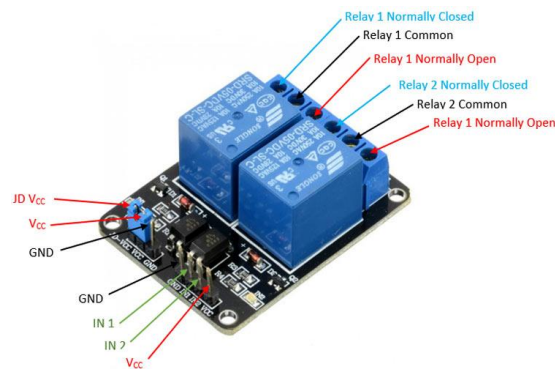


Gambar 8 Jetson Nano

#### 5. Relay

*Relay* adalah saklar (*switch*) yang dioperasikan dengan aliran listrik dan merupakan komponen *electromechanical* (elektromekanikal) yang terdiri dari 2 bagian utama yakni *elektromagnet (coil)* dan mekanikal (seperangkat kontak saklar/switch). Relay dimanfaatkan sebagai penggerak kontaktor untuk menyambungkan rangkaian secara tidak langsung. Relay memiliki bagian berupa coil yang biasanya memiliki tegangan kerja DC 5V, 9V, 12V atau sebagainya. Jika komponen tersebut diberi tegangan kerjanya maka arus akan mengalir pada coil sehingga menyebabkan kontak pada relay dapat bekerja.

Kegunaan relay pada penelitian ini yaitu sebagai menyambung dan memutus aliran arus listrik yang masuk pada *waterpump*, sehingga *waterpump* dapat dinyalakan dan dimatikan secara otomatis melalui kendali dari program *flight controller*.



*Gambar 9 Relay 2 Channel*  
(Sumber : components101.com)

Seperti yang terlihat pada Gambar 9, Relay memiliki 2 kondisi awal yaitu *Normally Open* (NO) dan *Normally Close* (NC). *Normally Open* merupakan kondisi relay terbuka atau tidak terhubung saat tidak dialiri arus listrik. Sedangkan *Normally Close* relay akan menutup saat tidak dialiri arus listrik.

## 6. Waterpump DC

*Waterpump DC* atau yang biasa disebut dengan *pompa air DC* adalah perangkat yang digunakan untuk mengalirkan atau mengangkat air menggunakan tenaga listrik yang berasal dari sumber arus searah. Pompa air DC mengandalkan motor DC yang berputar ketika diberi daya listrik, sehingga menciptakan aliran air. Pada sistem drone *waterpump* digunakan untuk mengalirkan cairan pestisida yang dikendalikan oleh *flight controller*.



*Gambar 10 Waterpump DC Motor*  
(Sumber : Shopee.co.id)

## 7. Nozzle Spray

*Nozzle spray* merupakan salah satu komponen penting dalam sistem dispersi cairan atau semprotan pada *drone*. *Nozzle* ini berfungsi sebagai saluran yang mengarahkan dan mengatur aliran cairan yang akan disemprotkan oleh *drone*. Keunggulan menggunakan nozzle ini adalah kemampuannya untuk mengatur pola semprotan dan volume cairan yang dihasilkan, memungkinkan drone untuk mengaplikasikan bahan kimia atau zat cair lainnya dengan presisi tinggi. Pola semprotan yang dapat diatur ini memungkinkan penyesuaian dengan kebutuhan aplikasi tertentu, seperti penyebaran pestisida atau pupuk secara merata di lahan pertanian.



*Gambar 11 Nozzle Sprayer*  
(Sumber : jakartanotebook.com)



#### **D. Python dan Flask**

Python adalah bahasa pemrograman tingkat tinggi yang diciptakan oleh *Guido van Rossum* dan pertama kali dirilis pada tahun 1991. Bahasa python mendukung berbagai paradigma pemrograman, seperti pemrograman orientasi objek (OOP), pemrograman fungsional, dan pemrograman imperatif. Python memiliki ekosistem yang kaya dengan banyak pustaka pihak ketiga yang tersedia, yang memungkinkan pengguna untuk melakukan berbagai tugas, seperti *data analysis* dan *machine learning* hingga pengembangan web dan otomatisasi sistem. Selain itu, Python adalah bahasa yang lintas platform, artinya program yang ditulis dalam Python dapat dijalankan di berbagai sistem operasi tanpa perubahan signifikan, menambah fleksibilitas dan daya tariknya di dunia pemrograman.

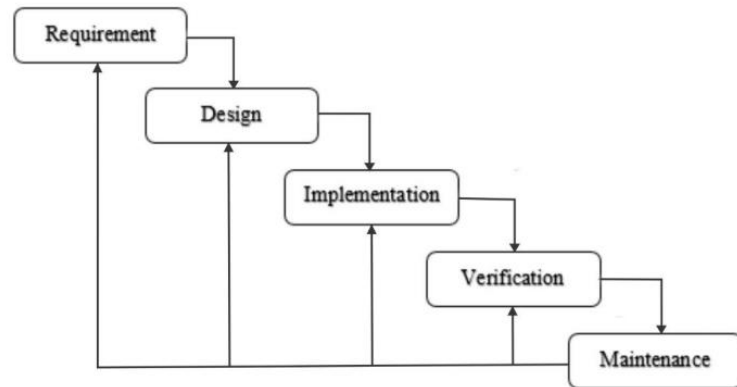
Bahasa pemrograman python memiliki berbagai macam framework untuk pengembangan website dan pengembangan sistem lainnya. Salah satu framework yang paling populer digunakan yaitu Flask. Flask merupakan salah satu framework Python yang ringan dan fleksibel untuk pengembangan aplikasi web. Framework ini memungkinkan pengembang untuk membangun aplikasi web dengan cepat, menggunakan Python sebagai bahasa dasar, dan menambahkan komponen atau ekstensi sesuai kebutuhan, seperti autentikasi, *database*, atau *managemen session*.

#### **E. Metode Pengembangan Sistem**

Metode pengembangan sistem adalah rencana atau pendekatan yang digunakan oleh tim pengembangan perangkat lunak atau sistem untuk merancang, mengembangkan, menguji, dan menerapkan sistem model perancangan yang akan

dilakukan. *System Development Life Cycle* atau yang dikenal dengan istilah SDLC merupakan metodologi umum yang digunakan untuk mengembangkan sistem perangkat lunak dengan menggunakan model dan metodologi yang digunakan untuk mengembangkan sistem perangkat lunak sebelumnya (Abdul Wahid, 2020; Hidayati, 2019). Pemilihan model SDLC yang digunakan untuk pengembangan sistem akan menentukan kualitas dari sistem yang akan dibuat atau dikembangkan dan juga menentukan kebutuhan lainnya dalam pengembangan sistem tersebut (Abdul Wahid, 2020).

Model *Waterfall* merupakan salah satu model SDLC yang sering digunakan dalam pengembangan sistem menggunakan pendekatan sistematis dan berurutan. Tahapan dalam model ini dimulai dari tahap perencanaan hingga tahap pengelolaan (*maintenance*) dan dilakukan secara bertahap. Disebut *waterfall* karena tahap demi tahap yang dilalui harus menunggu tahap sebelumnya selesai dan berjalan berurutan. Model pengembangan ini bersifat linear dari tahap awal pengembangan sistem yaitu tahap perencanaan sampai tahap akhir pengembangan sistem yaitu tahap pemeliharaan. Tahapan berikutnya tidak akan dilaksanakan sebelum tahapan sebelumnya selesai dilakukan dan tidak dapat kembali atau mengulang ke tahap sebelumnya. Tahapan-tahapan model waterfall terlihat pada Gambar 12 berikut :



*Gambar 12 Siklus Metode Waterfall*  
(Sumber : (Abdul Wahid, 2020))

### 1. Analisis Kebutuhan

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami sistem alat yang diharapkan oleh pengguna dan batasan alat tersebut. Informasi dapat diperoleh melalui mencari referensi, wawancara, diskusi atau survei secara langsung. Informasi yang diperoleh akan dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

### 2. Desain Sistem

Tahap desain sistem yaitu pengembang membuat desain sistem yang dapat membantu menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan. Desain sistem yang dilakukan meliputi pembuatan desain rancangan sistem secara keseluruhan, pengkabelan, manufaktur, dan pemrograman.

### 3. Implementasi

Tahap implementasi merupakan sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit testing.

#### 4. Verifikasi

Tahap ini dilakukan verifikasi dan pengujian apakah sistem sepenuhnya atau sebagian memenuhi persyaratan sistem, pengujian dapat dikategorikan ke dalam unit testing (dilakukan pada modul tertentu kode), sistem pengujian (untuk melihat bagaimana sistem bereaksi ketika semua modul yang terintegrasi) dan penerimaan pengujian (dilakukan dengan atau nama pelanggan untuk melihat apakah semua kebutuhan pelanggan puas)

#### 5. Pemeliharaan (*maintenance*)

Pemeliharaan adalah tahap akhir dari metode *waterfall*. alat yang telah berhasil dijalankan juga memungkinkan akan dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

### F. Penelitian Relevan

Penelitian yang akan dirancang mengacu pada penelitian terdahulu yang relevan dengan perancangan alat yang akan dilakukan. Berikut ini beberapa hasil penelitian relevan yang dijadikan bahan telaah bagi peneliti:

1. Penelitian (Hidayat et al., 2019) dengan judul “*Rancang bangun Prototype Drone Penyemprot Pestisida Untuk Pertanian Padi Secara Otomatis*” menyimpulkan bahwa berdasarkan hasil pengujian dan data yang diperoleh dengan menggunakan sumber baterai berkapasitas 2200mAh, *drone* frame quadcopter dapat terbang selama kurang dari 3 menit 46 detik ketika beban

*drone* sebesar 1,3kg dan ketika beban *drone* sebesar 2,5kg *drone* hanya dapat terbang selama kurang lebih 1 menit 45 detik

2. Penelitian (Taufik et al., 2022) dengan judul “*Drone Untuk Deteksi Hama dan Penyemprotan Pestisida Pada Tanaman Padi*” melakukan uji penyemprotan dengan menerbangkan *drone* di atas karton pada ketinggian  $\pm 1$  meter hingga  $\pm 1,5$  meter. Kemudian modul relay akan diaktifkan selama 20 detik dengan menjalankan program yang telah dibuat pada raspberry pi, sehingga pompa wiper akan menyemprotkan cairan.
3. Penelitian (Widiasari & Este Dulan Agustinus S., 2020) dengan judul “*Rancang Bangun Drone Quadcopter Tanpa Awak Penyiram Pupuk Tanaman*” Menyimpulkan bahwa daya angkat *drone* quadcopter mampu mengangkat beban dengan stabil dari berat 200-600 ml. Dengan berat maksimum yang dapat diangkut oleh *drone quadcopter* sebesar 600 ml pada lahan pertanian seluas 30m<sup>2</sup> dengan ketinggian 2-3 meter
4. Penelitian (Yallappa, D. et al., 2017) dengan judul “*Development and Evalution of Drone Mounted Sprayer For Pesticide Aplications to Crops*” Melakukan uji penyemprotan *drone* yang dioperasikan pada tanaman kacang tanah dan padi dengan luas lahan 1,15ha dan 1,08ha dengan ketinggian 1 m.

### **BAB III**

## **METODOLOGI PERANCANGAN**

BAB ini menjelaskan mengenai metode perancangan dan pengembangan sistem untuk merancang dan membuat sistem kontrol penyemprotan cairan pestisida menggunakan *drone hexacopter*. Perancangan sistem dilakukan dari segi perangkat keras (*hardware*) maupun dari segi perangkat lunak. Perancangan dan pembuatan sistem menggunakan metode pengembangan *System Development Life Cycle* (SDLC) model *Waterfall*, model *Waterfall* dapat menggambarkan pendekatan sistem yang sistematis dan proses yang berurutan, dimulai dari analisis kebutuhan, desain sistem, implementasi, pengujian dan *maintenance*. Model ini melibatkan penyelesaian satu tahap secara lengkap sebelum melangkah ke tahap berikutnya. Ketika satu tahap selesai langsung dilakukan evaluasi untuk memastikan alat berjalan sesuai rencana dan layak diteruskan ke tahap berikutnya.

#### **A. Analisis Kebutuhan (*Requirement*)**

Proses awal yang dilakukan adalah analisis kebutuhan yaitu mengidentifikasi kebutuhan-kebutuhan dalam pembuatan tugas akhir ini, kebutuhan yang diperlukan dalam perancangan dan pembuatan sistem penyemprotan cairan pestisida otomatis menggunakan *drone UAV hexacopter* yaitu sebagai berikut :

##### **1. Analisis Kebutuhan Sistem**

Analisis kebutuhan sistem merupakan kebutuhan yang diperlukan untuk mengidentifikasi bagaimana gambaran sistem alat yang digunakan agar

berfungsi dengan semestinya. Tabel 1 berikut merupakan daftar kebutuhan yang diperlukan pada penelitian ini.

*Tabel 1 Analisis Kebutuhan Sistem*

No	Uraian Sistem	Penjelasan
1.	Jenis Drone	Jenis <i>drone</i> yang digunakan adalah <i>drone hexacopter</i> F550 yaitu drone yang memiliki 6 motor dan baling-baling sebagai penggerak dengan jarak antara 2 motor yang sejajar sebesar 550mm. <i>Drone hexacopter</i> digunakan bertujuan untuk mampu membawa beban yang cukup besar seperti cairan pestisida.
2.	Perangkat Kontrol	Hardware yang diperlukan untuk sistem kontrol penyemprotan adalah Mini Computer yang mana berfungsi sebagai <i>server</i> atau pusat pengendalian sistem untuk memproses data sensor dan kontrol penyemprotan.
3	Aplikasi Kontrol	Sistem penyemprotan cairan pestisida dikontrol melalui sebuah aplikasi web server yang dirancang menggunakan bahasa pemrograman python framework Flask
3.	Mekanisme Sistem	Sistem Penyemprotan cairan pestisida dilakukan secara otomatis yang dikendalikan oleh modul mini komputer (Jetson nano), jetson nano

		bertindak sebagai sever yang menerima request dari client melalui interface website menggunakan bahasa pemrograman python.
4.	Pengujian Sistem	Pengujian penyemprotan dilakukan di area lapangan yang cukup luas tanpa halangan. pengujian dilakukan untuk menguji tingkat keberhasilan suatu sistem penyemprotan menggunakan drone hexacopter dengan lebih optimal

## 2. Analisis Kebutuhan Alat dan Bahan

Untuk memastikan keberhasilan pembuatan suatu sistem, diperlukan analisis terhadap kebutuhan alat dan bahan yang akan digunakan. Analisis ini bertujuan untuk mengidentifikasi secara detail fungsi dari setiap komponen yang diperlukan. Rincian kebutuhan alat dan bahan yang akan digunakan disajikan dalam Tabel berikut:

*Tabel 2 Analisis Kebutuhan Alat dan Bahan*

No	Alat dan Bahan	Fungsi
1.	Flight Controller (Pixhawk)	Mengatur, menstabilkan, dan mengoordinasikan semua pergerakan drone sesuai perintah pilot atau program penerbangan otomatis.
2.	Mini Computer (Jetson Nano)	Sebagai Server dan sistem antarmuka untuk mengendalikan sistem penyemprotan cairan pestisida
3.	GPS Module	Sebagai sensor pada drone yang berfungsi untuk mengetahui posisi drone secara real time



4	Wifi Dongle	Sebagai adaptor jaringan tanpa kabel untuk menghubungkan mini komputer ke jaringan WiFi
5.	Relay	Sebagai saklar otomatis untuk menghubungkan dan memutuskan arus yang mengalir pada waterpump DC
6.	Waterpump 12V	Sebagai pompa cairan pestisida ketika diberikan tegangan 12 V
7.	Nozzle Spray	Sebagai pendispersi cairan pestisida agar cairan semprotan menyebar dengan merata
8.	Tangki Pestisida	Sebagai tempat untuk menampung beban cairan pestisida

Berdasarkan Tabel 2, setiap komponen yang telah diidentifikasi memiliki fungsi spesifik yang mendukung proses pembuatan dan pengoperasian sistem secara keseluruhan. Alat dan bahan tersebut dirancang untuk bekerja secara terintegrasi guna memastikan sistem dapat berfungsi secara optimal sesuai dengan kebutuhan.

### 3. Analisis Kebutuhan Software

Analisis kebutuhan software dilakukan untuk mengidentifikasi berbagai perangkat lunak yang akan digunakan dalam proses perancangan dan pembuatan sistem. Software yang dibutuhkan dalam pembuatan tugas akhir ini adalah sebagai berikut:

#### a. *Software Solidworks*

*Software Solidworks* adalah perangkat lunak desain dan rekayasa 3D yang sangat banyak digunakan dalam industri manufaktur dan rekayasa. Pada hal ini *software solidworks* digunakan untuk pembuatan rancangan desain alat yang akan dikembangkan seperti tampilan alat secara keseluruhan, dan susunan struktur komponen.

b. *Software Mission Planner*

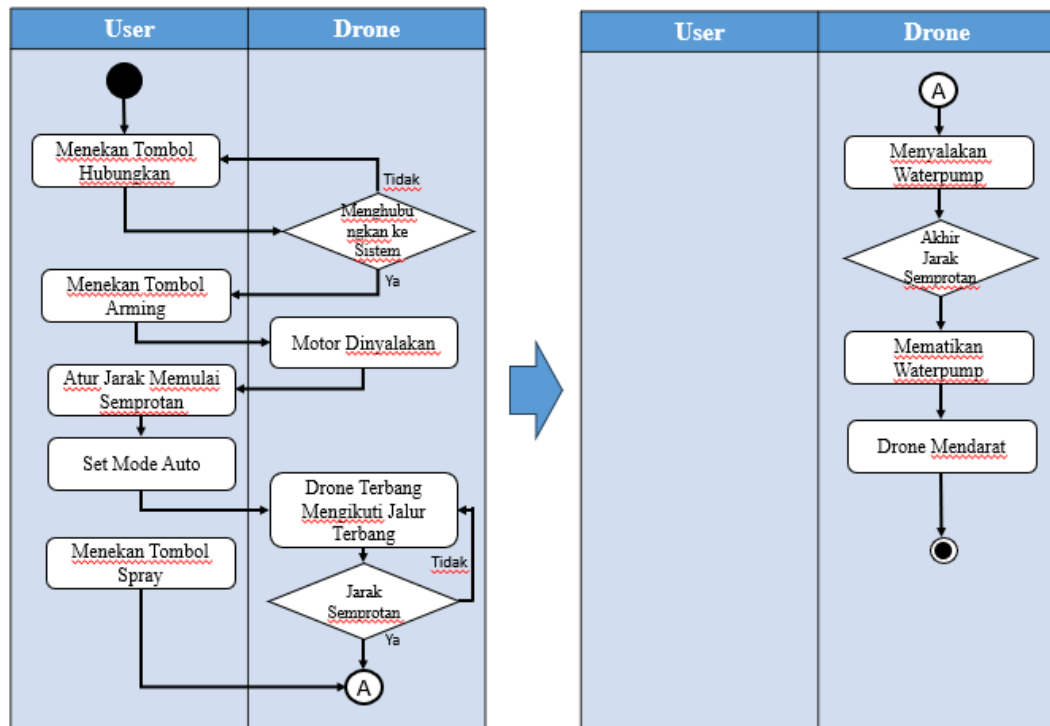
*Mission Planner* merupakan aplikasi perangkat lunak open-source yang digunakan untuk pengoperasian dan pemrograman pesawat udara tanpa awak (*drone*) yang menggunakan autopilot PX4 atau ArduPilot. pada hal ini software mission planner digunakan untuk misi penerbangan drone.

c. *Python dan Flask*

Python merupakan bahasa pemrograman tingkat tinggi yang digunakan untuk menulis perintah untuk mengontrol sistem navigasi drone. Sedangkan Flask adalah framework dengan bahasa python yang digunakan untuk pengembangan software dan website sistem kontrol penyemprotan.

#### **4. Analisis Kebutuhan Fungsional**

Analisis kebutuhan fungsional dilakukan bertujuan untuk merancang suatu sistem penyemprotan pestisida agar bekerja secara otomatis menggunakan *drone hexacopter* dengan adanya interaksi antara drone dengan user. prinsip kerja sistem penyemprotan pada penelitian ini dapat dilihat pada penjelasan gambar *activity diagram* berikut.



Gambar 13 Activity Diagram Sistem

Berdasarkan *activity diagram* diatas, sistem penyemprotan cairan pestisida ini melibatkan interaksi antara pengguna (*user*) dan *drone* yang dikendalikan. Langkah pertama yang dilakukan oleh pengguna adalah menghubungkan koneksi antara drone dan sistem kontrol dengan menekan tombol *connect* pada sistem kontrol. Setelah koneksi berhasil, pengguna perlu menyalakan mesin atau motor drone dengan menekan tombol *arming* pada sistem kontrol.

Pengaturan waktu penyemprotan cairan pestisida dilakukan dengan memasukkan nilai jarak dalam satuan meter ke sistem kontrol. Ketika drone mencapai jarak yang telah ditentukan, sistem secara otomatis akan mengaktifkan penyemprotan dengan menyalakan *waterpump*. Selain penyemprotan otomatis, pengguna juga dapat melakukan penyemprotan secara manual dengan menekan tombol *spray* pada sistem kontrol. Selama penyemprotan berlangsung, *waterpump* akan bekerja sesuai kondisi sistem—menyala saat penyemprotan

aktif dan mati saat penyemprotan dihentikan. Penyemprotan otomatis akan berhenti secara otomatis ketika drone mencapai akhir jarak yang telah ditentukan.

## **B. Desain Rancangan Sistem (*Design*)**

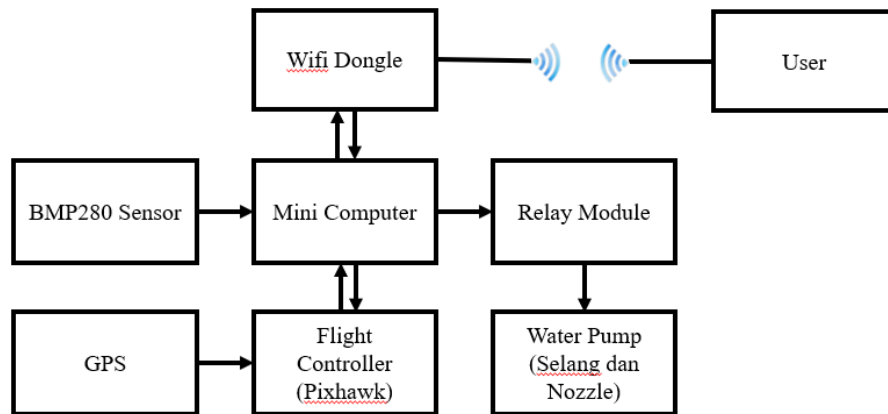
Tahap desain perancangan yaitu membuat desain rancangan sistem agar dapat membantu menentukan perangkat keras (*hardware*) dan sistem perangkat lunak (*software*) yang dibutuhkan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan seperti desain rancangan alat secara keseluruhan, pengkabelan sistem, blok diagram sistem dan sistem kontrol penyemprotan.

### **1. Perancangan *Hardware***

Perancangan *hardware* berfungsi untuk mengidentifikasi peralatan dan bahan yang akan digunakan. Pembuatan rancangan desain alat menggunakan aplikasi desain yaitu *software Solidworks*. Pada tahapan ini berfungsi untuk membuat desain bentuk jadi alat yang akan dibuat, mulai dari pembuatan bagian-bagian pada *drone* hingga desain perakitan *drone* yang akan digunakan.

#### **a. Diagram Blok Sistem**

Perencanaan alat dibuat dalam bentuk diagram blok. Tiap blok atau sub sistem mempunyai fungsi masing masing, yang kemudian akan digabungkan menjadi suatu alat yang utuh, dan dapat bekerja sesuai dengan tujuan yang diharapkan. Diagram blok sistem ditunjukkan pada Gambar 15



Gambar 14 Diagram Blok Sistem

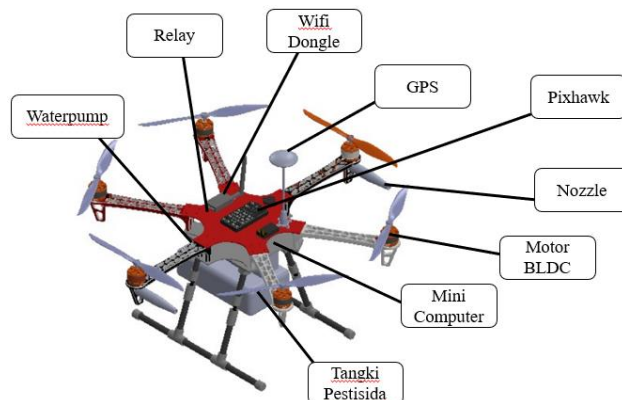
Berdasarkan pada Gambar 14 dari masing-masing blok diagram memiliki fungsi sebagai berikut :

- 1) **Flight controller (Pixhawk)** digunakan sebagai sistem navigasi dan pengatur arah terbang *drone*. pada *flight controller* terdapat beberapa sensor yang digunakan untuk memperoleh data penerbangan pada *drone* diantaranya : *accelerometer*, *barometer*, dan sensor kompas.
- 2) **Mini Computer** berfungsi sebagai media pemrosesan data untuk mengontrol output sprayer cairan pestisida.
- 3) **Relay** berfungsi sebagai saklar (*switch*) untuk menyalakan dan mematikan water pump yang digunakan sebagai penyemprotan pestisida cair.
- 4) **Water pump** berfungsi untuk memompa dan mengalirkan cairan pestisida dari tangki hingga cairan tersebut disemprotkan melalui *nozzle*.
- 5) **BMP280 Sensor** berfungsi untuk memproses data ketinggian, suhu dan tekanan udara ketika drone lepas landas, data tersebut akan ditampilkan ke monitor user.

- 6) **GPS** berfungsi sebagai perangkat penentu posisi yang menggunakan sinyal satelit untuk menentukan lokasi, kecepatan, dan waktu secara real-time
- 7) **Wifi Dongle** berfungsi sebagai adaptor jaringan nirkabel untuk menghubungkan perangkat mini komputer ke jaringan WiFi tanpa memerlukan kartu jaringan internal.

#### b. Desain Rancangan Alat

Desain rancangan alat merupakan gambaran alat yang akan dikembangkan dengan menggunakan aplikasi desain 3D yaitu *Solidworks*. Gambar 15 merupakan bentuk dari desain rancangan alat yang akan dikembangkan.



*Gambar 15 Desain Rancangan Alat.*

Gambar 15 merupakan bentuk dan susunan komponen yang telah dirakit hingga alat siap untuk dilakukan pengujian. *Drone* yang akan dikembangkan menggunakan *body frame hexacopter* F550 berbahan serat fiber yang kuat dan ringan. Bagian bawah *drone* juga terdapat *landing gear* yang terbuat dari bahan aluminium holo dan akrilik sebagai penopang *drone* tersebut untuk berdiri, serta tangki penyimpanan cairan pestisida yang dapat

membawa muatan berupa cairan pestisida agar saat lepas landas drone dapat terbang melakukan penyemprotan dengan seimbang dan stabil. Pixhawk yang berperan sebagai *flight controller* untuk sistem navigasi diposisikan di titik tengah *drone* yang bertujuan untuk menjaga stabilitas dan keseimbangan *drone*. Hal ini memungkinkan pusat gravitasi drone tetap seimbang, yang sangat penting untuk menghindari ketidakstabilan dalam penerbangan. Posisi baterai diletakkan di bagian bawah dengan memberi jarak dari pixhawk. Hal ini bertujuan agar sensor kompas pada *pixhawk* tidak terganggu akibat medan magnet yang dihasilkan oleh baterai. Kemudian tangki air diposisikan di bagian tengah kendaraan dan bawah baterai agar saat posisi *takeoff drone* dapat terbang dengan seimbang, serta untuk peletakan *nozzle* diletakkan di bagian bawah motor bagian kiri dan kanan *drone* untuk melakukan penyemprotan cairan agar menyebar hingga merata.

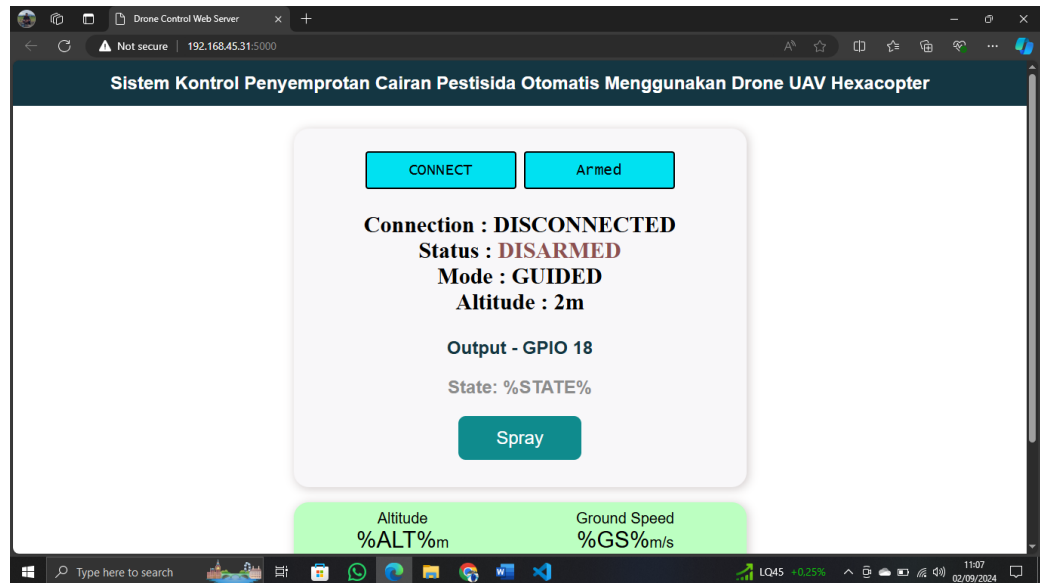
## **2. Perancangan Software.**

Setelah memastikan perangkat keras (*hardware*) telah dapat bekerja sesuai dengan fungsinya, maka tahap selanjutnya yaitu tahap perancangan *software*. Perancangan *software* sendiri meliputi rancangan antarmuka web aplikasi kontrol, dan pembuatan perencanaan misi terbang yang dilakukan dengan bantuan aplikasi *Mission Planner*

### **a. Antarmuka Kontrol (*Control Interface*)**

.Antarmuka kontrol (*control interface*) adalah tampilan dari pengaplikasian sistem kontrol cairan pestisida berbasis *website* yang dirancang

menggunakan bahasa pemrograman *python* dan *framework Flask*, antarmuka kontrol berfungsi untuk memonitoring dan kontrol sistem drone ketika melakukan penerbangan (*takeoff*). Berikut merupakan desain dari antarmuka sistem kontrol drone penyemprotan cairan pestisida :



Gambar 16 Web Control Interface

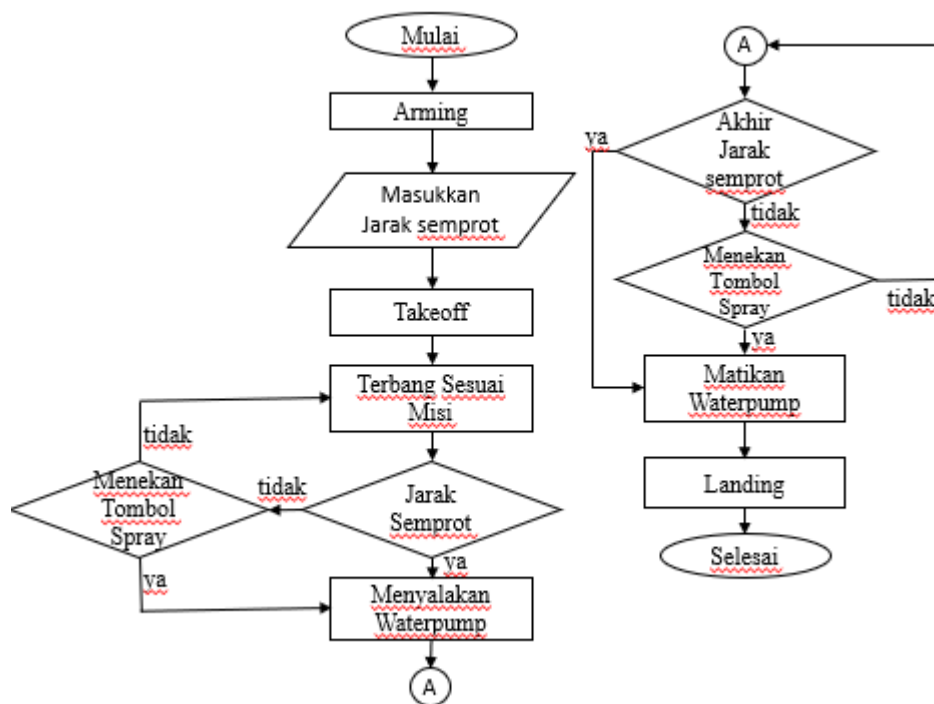
Berdasarkan Gambar 16 diatas aplikasi web kontrol dibuka melalui IP *address server*. Aplikasi ini terdapat tombol *connect* dan *armed*, tombol *connect* difungsikan untuk menghubungkan komunikasi antara perangkat *server (jetson nano)* dengan *flight controller (pixhawk)*, sedangkan tombol *armed* digunakan untuk menghidupkan mesin/motor pada drone. Tombol pada bagian bawah digunakan untuk menyalakan dan mematikan spray, serta status spray akan ditampilkan diatas tombol spray. Bagian bawah sistem merupakan bagian dari sistem monitoring perilaku drone, perilaku drone meliputi data-data drone yang diterima ketika melakukan penerbangan, diantaranya data ketinggian (*altitude*), data arah sudut drone



(yaw), data tekanan udara (*pressure*), data suhu udara (*temperature*), dan kecepatan manuver drone (*groundspeed*)

### 3. Diagram Alir (*Flowchart*)

Diagram alir biasa disebut dengan *flowchart* adalah representasi grafis dari urutan langkah-langkah atau proses dalam sebuah sistem, algoritma, atau pekerjaan. *Flowchart* digunakan untuk memvisualisasikan aliran informasi, keputusan, atau tugas dari awal hingga akhir. Gambar 18 berikut *flowchart* misi penerbangan drone yang akan dilakukan pada tugas akhir ini :



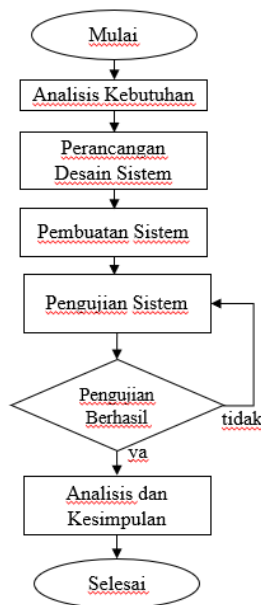
Gambar 17 Flowchart Perencanaan Misi Drone

Pada *flowchart* diatas menjelaskan algoritma dan urutan-urutan perencanaan misi pada penerbangan *drone*. berikut adalah penjelasan mengenai urutan-urutan *flowchart* :

- 1) Proses pertama yang dilakukan drone yaitu *Arming* motor yang merupakan proses menyalakan motor dan sistem kendali *drone* hingga *drone* siap untuk beroperasi.
- 2) Setelah drone berhasil dinyalakan sebelum melakukan penerbangan user terlebih dahulu perlu memasukkan data jarak semprot,
- 3) Drone akan melakukan proses penerbangan (*takeoff*) hingga ketinggian 1-2 meter dari permukaan.
- 4) Ketika drone sudah diterbangkan pada jarak terbang yang sudah ditentukan maka secara otomatis akan menyalakan waterpump untuk menyemburkan cairan pestisida.
- 5) Saat drone masih belum mencapai jarak tertentu, user juga bisa menyalakan penyemprotan secara manual dengan menekan tombol spray pada web kontrol.
- 6) Ketika penyemprotan berhasil dinyalakan drone akan terbang sesuai jalurnya hingga ujung jalur penyemprotan.
- 7) Ketika drone telah mencapai akhir jalur penyemprotan, penyemprotan secara otomatis akan mematikan semprotan.
- 8) Jika drone masih belum mencapai jarak akhir penyemprotan maka penyemprotan juga dapat dimatikan secara manual dengan menekan tombol spray pada web kontrol
- 9) Saat di lokasi akhir penyemprotan drone akan melakukan pendaratan

### C. Alur Pembuatan Sistem

Alur pembuatan sistem merupakan serangkaian alur dan langkah-langkah proses perancangan dan pembuatan sistem. Proses rancangan pembuatan dimulai dengan beberapa proses, diantaranya perancangan dan perakitan mekanik drone, penentuan komponen yang diperlukan, penyusunan dan pengkabelan sistem, dan yang terakhir pembuatan mekanisme sistem penyemprotan. Adapun flowchart dan langkah-langkah alur penelitian untuk proses perancangan dan pembuatan sistem penyemprotan menggunakan *drone* sebagai berikut :



Gambar 18 Alur Pembuatan Sistem

#### 1. Analisis kebutuhan

Tahapan ini merupakan tahap untuk mengidentifikasi apa saja yang dibutuhkan dalam perancangan dan pembuatan sistem semprot yang akan dilakukan pada drone. Kebutuhan tersebut seperti : study literatur, identifikasi komponen yang dibutuhkan, dan penyediaan alat dan bahan.

## 2. Perancangan Desain Sistem

Tahapan desain sistem merupakan proses awal yang dilakukan untuk mengetahui kebutuhan yang diperlukan pada proses pembuatan alat. Proses perancangan desain sistem ini juga merupakan suatu rencana terinci dan spesifik mengenai proses pengerjaan tugas akhir ini dengan membuat blok diagram sistem dan pembuatan desain dan struktur perancangan alat.

## 3. Pembuatan Sistem

Tahapan ini merupakan proses untuk melakukan pembuatan dan perakitan sistem penyemprotan cairan pestisida dengan *drone hexacopter* secara utuh dan fungsional. Perakitan ini dibuat seperti pada desain perancangan yang telah dibuat sebelumnya, mulai dari penyusunan frame, pemasangan motor, penyusunan letak komponen, pengkabelan sistem, dan melakukan pengecekan lainnya.

## 4. Pengujian Sistem

Tahapan ini dilakukan serangkain pengujian alat dalam melakukan proses penyemprotan cairan pestisida secara otomatis di atas udara dengan menggunakan *drone hexacopter*. Hal yang akan diuji yaitu pengujian pengeluaran daya baterai terhadap gaya dorong (*trusth*) motor BLDC terhadap spesifikasi *propeller*, pengujian rentang jarak pembacaan sinyal radio pada remot, pengujian rentang jarak pembacaan sinyal telemetri, pengujian error pembacaan titik koordinat GPS, dan melakukan perhitungan terhadap lama durasi terbang drone secara otomatis dari awal drone aktif hingga drone selesai melakukan misi dengan memberikan beban pada drone.

## **5. *Maintenance***

Tahapan *maintenance* merupakan tahapan tambahan dari metode pengembangan model *waterfall*, Tahapan *maintenance* merupakan tindakan yang dilakukan untuk menjaga, memperbaiki, atau memperbarui sistem *drone* agar tetap berfungsi dengan baik. Tujuan dari pemeliharaan ini adalah untuk mencegah atau memperbaiki kerusakan atau kegagalan yang terjadi pada sistem.

## **6. Analisis dan Kesimpulan**

Setelah melakukan serangkaian pengujian pada alat maka tahapan terakhir pada penelitian ini adalah melakukan analisis keberhasilan alat dan menarik kesimpulan terhadap data yang telah diperoleh dari hasil pengujian.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **A. Perancangan Sistem Kontrol Penyemprotan Cairan Pestisida Menggunakan *Drone UAV Hexacopter***

Suatu sistem dapat dikatakan berfungsi dengan baik apabila telah dilakukan pengujian kinerja dari peralatan tersebut. Pengujian kinerja alat merupakan salah satu langkah penting yang harus dilakukan untuk mengetahui apakah sistem yang dibuat telah sesuai dengan yang direncanakan. Perancangan sistem kontrol penyemprotan bertujuan untuk menciptakan mekanisme yang efektif dan efisien dalam mengatur proses penyemprotan secara otomatis. Sistem ini dirancang dengan mempertimbangkan kebutuhan pengguna, baik dari segi akurasi penyemprotan, kestabilan sistem, maupun kemudahan operasional. Dalam perancangan ini, digunakan perangkat keras dan perangkat lunak yang saling terintegrasi untuk mengontrol jalannya aktuator, mendeteksi kondisi lingkungan, dan mengatur parameter penyemprotan sesuai kebutuhan. Dengan pendekatan ini, sistem diharapkan dapat meningkatkan produktivitas sekaligus mengurangi kesalahan manual dalam proses penyemprotan.

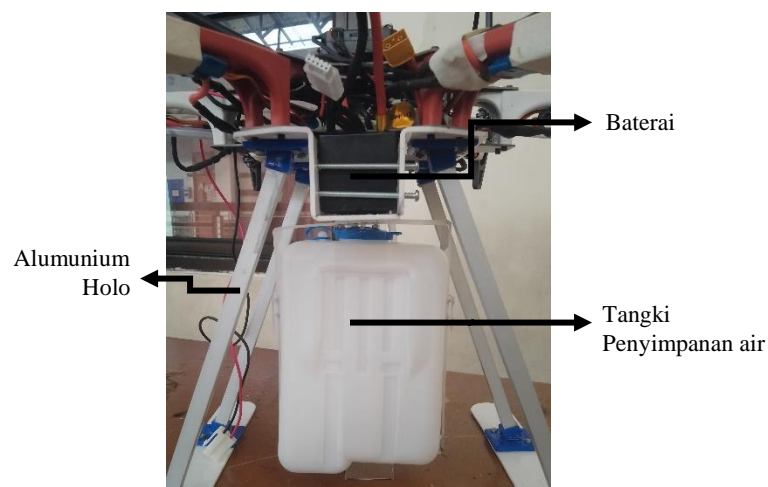
##### **1. Hasil Rancangan Hardware**

Hasil rancangan *hardware* merupakan bentuk dari hasil rancangan fisik yang telah disusun hingga drone siap untuk dioperasikan. tampilan penyemprotan cairan pestisida menggunakan *drone hexacopter* secara keseluruhan dapat dilihat pada Gambar 19.



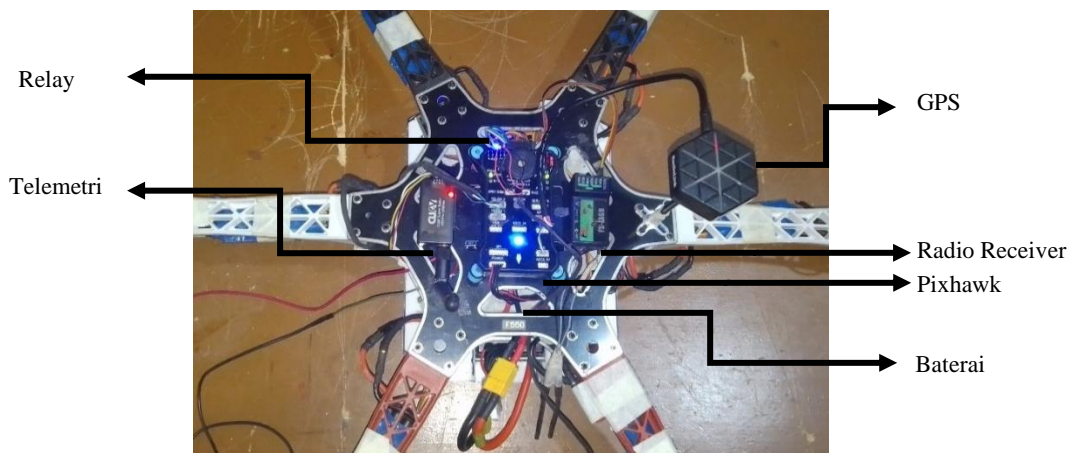
*Gambar 19 Tampilan Fisik Alat.*

Gambar 19 terlihat secara menyeluruh racangan alat yang telah dibangun dengan menggabungkan seluruh komponen. *Body frame drone* terbuat dari bahan dasar *fiber glass* dengan panjang diagonal 550 mm dengan baling-baling berbahan plastik yang telah terpasang. Posisi peletakan komponen dan rangkaian elektronik diposisikan di bagian tengah, diatas, di lengan dan dibawah drone. Pada bagian bawah drone terdapat beberapa komponen atau bahan berat seperti tangki penyimpanan pestisida, baterai dan dudukan untuk kaki drone, sedangkan pada bagian dalam atau tengah drone terdapat pusat dari sistem tersebut yaitu jetson nano / mini komputer



*Gambar 20 Posisi Komponen Bagian Bawah Drone*

Pada posisi atas terdapat komponen untuk pemrosesan data dan sistem navigasi seperti Pixhawk dan berbagai komponen yang berperan dalam menerima sinyal dari luar seperti sensor BMP280, *Telemetry*, *Radio Receiver*, dan *GPS*.



*Gambar 21 Posisi Komponen Bagian Atas Drone*

Pixhawk diposisikan pada posisi tengah drone dikarenakan pixhawk merupakan komponen inti untuk navigasi drone, tujuannya untuk mempermudah pada saat kalibrasi sensor pada drone. Perangkat lainnya seperti GPS, Telemetry dan radio receiver diposisikan diatas drone karena ketiga perangkat tersebut menerima data eksternal secara *wireless*, tujuannya agar komponen tersebut dapat menerima data sinyal dari luar dengan lebih baik.



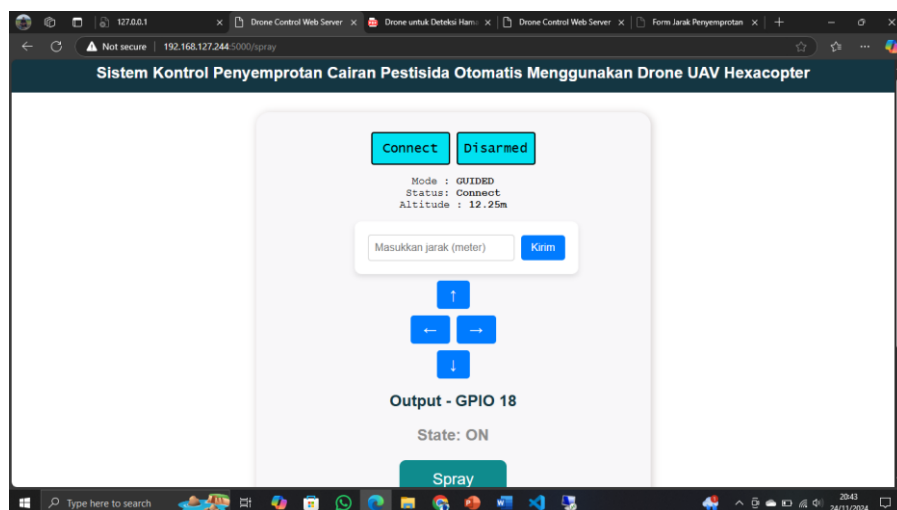
*Gambar 22 Letak Komponen Bagian Lengan Drone*



Gambar 22 merupakan susunan peletakan komponen pada bagian lengan kiri dan kanan drone. Komponen yang digunakan seperti ESC yang terpasang pada bawah lengan drone, Motor Brushless di ujung lengan drone, *propeller* yang terpasang pada motor, slang air dan *nozzle* untuk melakukan penyemprotan cairan pestisida dipasang pada bawah lengan drone.

## 2. Hasil Rancangan Software

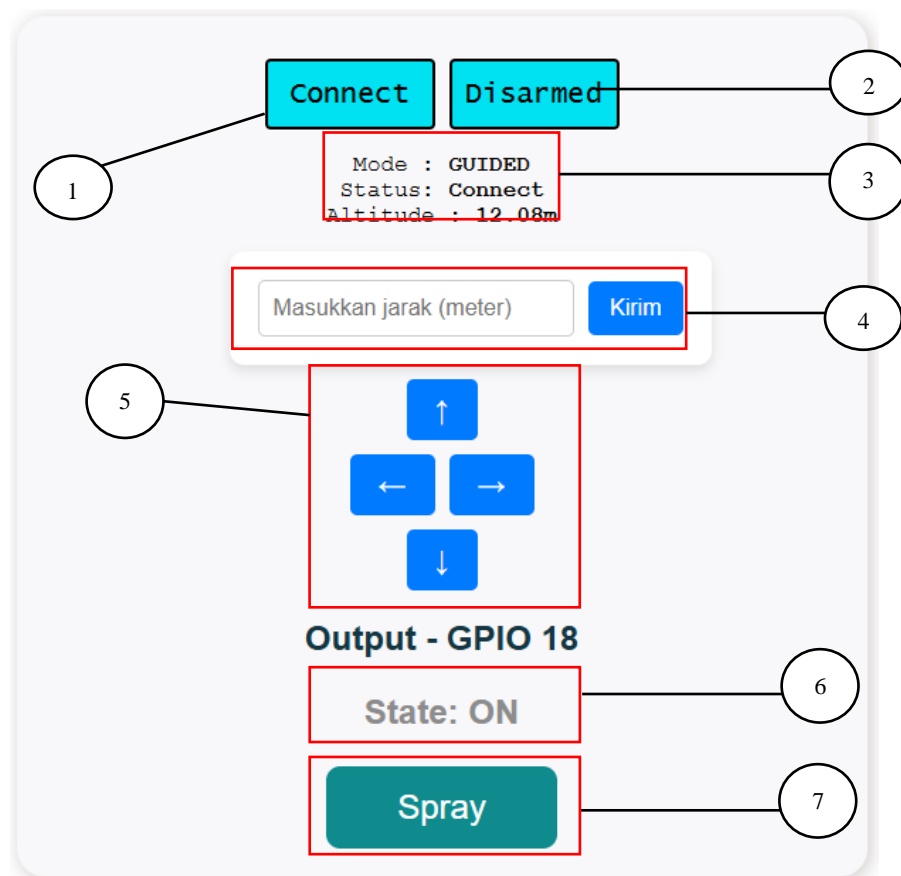
Hasil rancangan software merupakan implementasi dari desain sistem yang telah dirancang untuk mengendalikan dan memonitor perangkat sesuai kebutuhan. Pada tahap ini, perangkat lunak dirancang untuk mengintegrasikan berbagai komponen, seperti pengolahan data sensor, pengaturan aktuator, dan antarmuka pengguna.:



*Gambar 23 Tampilan Web Kontrol Penyemprotan*

Berdasarkan pada Gambar 23 diatas, aplikasi web kontrol dan monitoring drone beroperasi dengan menggunakan jaringan wifi lokal, jaringan wifi yang digunakan server drone dan perangkat kontrol harus berada di jaringan yang sama untuk bisa mengakses halaman tampilan webkontrol. Halaman web

kontrol diakses dengan mengirimkan request berupa IP address server seperti “192.168.84.1” dengan port :5000. Port 5000 menandakan server menggunakan framework flask dengan bahasa pemrograman python. Pada aplikasi kontrol terdapat 3 bagian dengan fungsi yang berbeda yaitu yang digunakan untuk mengontrol navigasi drone, kontrol penyemprotan dan monitoring data sensor drone yang terpasang pada drone. Gambar 25 berikut merupakan bagian status dan kontrol penyemprotan drone.



*Gambar 24 Bagian Kontrol Penyemprotan pada drone*

Berdasarkan Gambar 24 diatas pada bagian kontrol terdapat beberapa fitur yang telah dirancang dengan fungsi sebagai berikut :

- 1) Button Connect : Button koneksi digunakan untuk menghubungkan koneksi antara flight controller (pixhawk) dengan modul mini komputer (jetson nano)
- 2) Button Arm : Button ini digunakan untuk menyalakan mesin motor sebelum melakukan takeoff dan juga bisa digunakan untuk menghentikan mesin motor.
- 3) Drone Status : Drone Status digunakan untuk mengetahui keadaan drone saat ini. Berdasarkan gambar diatas mode penerbangan yang sedang berlangsung adalah mode GUIDED dengan ketinggian drone 2 meter dari permukaan tanah.
- 4) GPIO PIN : GPIO pin menunjukkan nomor pin yang digunakan untuk menjalankan motor pada penyemprotan drone.
- 5) Spray Status : Spray status merupakan keadaan sprayer saat ini. Berdasarkan gambar diatas status pada spray adalah off yaitu waterpump saat ini dalam keadaan mati. Sebaliknya akan bernilai ON ketika waterpump dinyalakan.
- 6) Button Spray : Button spray digunakan untuk menyalakan dan mematikan waterpump yang terhubung pada drone.

Altitude	Ground Speed
1.82m	1.23m/s
Sprayer	Yaw
OFF	180°deg
Temperature	Pressure
31.59°C	1013.15hPa

*Gambar 25 Bagian Monitoring Data Sensor Pada Drone*

Gambar 25 merupakan bagian monitoring data sensor yang terdapat pada drone. Nilai data yang dapat diambil dari drone yaitu altitude/ketinggian drone terbang, status sprayer, temperatur udara, kecepatan (*speed*) drone bermanuver, sudut arah drone menghadap, dan nilai tekanan udara.

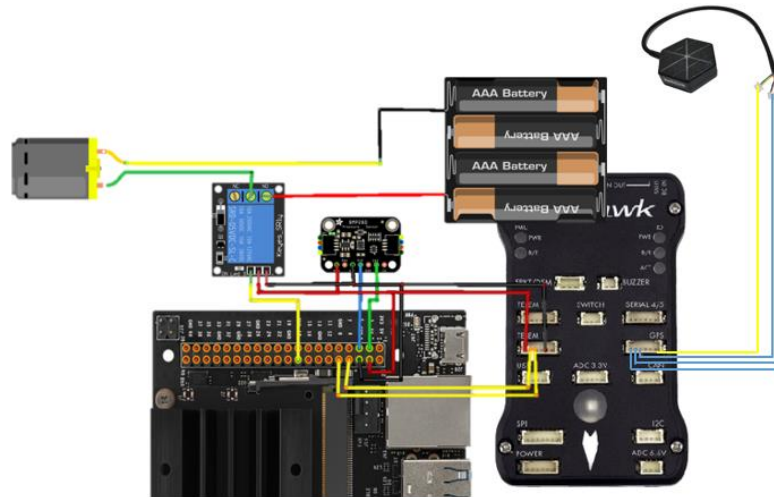
## **B. Integrasi Sistem Kontrol Penyemprotan Pestisida Menggunakan Drone UAV Hexacopter**

Sistem penyemprotan cairan pestisida menggunakan *drone* dirancang untuk beroperasi secara otomatis atau melalui perangkat kontrol. Proses ini memerlukan integrasi perangkat keras dan perangkat lunak dengan protokol komunikasi yang sesuai untuk memastikan kompatibilitas dan efisiensi antar komponen. Berikut ini adalah tahapan integrasi sistem penyemprotan otomatis pada drone hexacopter:

### **1. Rangkaian Sistem Penyemprotan :**

Rangkaian sistem penyemprotan merupakan komponen utama dalam mendukung proses distribusi cairan pestisida secara efisien dan tepat sasaran. Sistem ini dirancang dengan mengintegrasikan berbagai elemen seperti pompa, nozzle, saluran distribusi, dan aktuator yang bekerja secara terkoordinasi.

Rangkaian sistem kontrol penyemprotan cairan pestisida dapat dilihat pada Gambar berikut:



*Gambar 26 Rangkaian Sistem Penyemprotan*

Berdasarkan Gambar 26 komponen yang berperan sebagai perangkat kontrol adalah modul mini komputer (jetson nano). Sedangkan Pixhawk berperan sebagai sistem navigasi penerbangan drone. Kedua perangkat tersebut saling berkomunikasi menghubungkan kabel transmitter dan receiver yang terhubung satu sama lain dengan menggunakan protokol komunikasi yang disebut dengan MAVlink. MAVlink merupakan protokol komunikasi ringan yang dirancang untuk pertukaran data antara pixhawk dan mini computer menggunakan library python dronekit dan pymavlink. MAVLink bekerja dengan mengumpulkan data dalam bentuk pesan yang terdiri dari *header*, *payload*, dan *checksum* untuk memastikan integritas data. Data yang diambil dari Pixhawk seperti titik lokasi GPS yang tersambung ke pixhawk, ketinggian drone terbang, kecepatan dan arah terbang drone.

Sensor lainnya yaitu sensor BMP280 merupakan sensor yang digunakan untuk mengukur tekanan udara dan suhu dengan akurat. Sensor ini bekerja menggunakan teknologi piezoresistif, sehingga dapat menghasilkan data akurat dengan konsumsi daya rendah. BMP280 digunakan dalam sistem pengukuran ketinggian pada drone. Sensor ini menggunakan protokol komunikasi I2C. I<sup>2</sup>C (*Inter-Integrated Circuit*) adalah protokol komunikasi serial yang digunakan untuk menghubungkan perangkat-perangkat elektronik dalam satu sistem. I<sup>2</sup>C menggunakan dua jalur utama yaitu SDA untuk mengirim dan menerima data, serta SCL untuk sinkronisasi. Data dikirim dalam bentuk frame yang terdiri dari alamat, data, dan bit kontrol.

Perangkat output pada sistem penyemprotan menggunakan Waterpump DC 12V yang dikendalikan otomatis oleh relay yang terhubung pada pin GPIO18 jetson nano. Ketika pin GPIO bernilai rendah maka relay akan terhubung dengan tegangan 12 volt dari baterai, sebaliknya ketika pin GPIO bernilai HIGH maka relay akan memutus aliran listrik yang mengalir dari baterai ke waterpump DC.

## 2. Integrasi Hardware dan Software

Integrasi hardware dan software merupakan langkah krusial dalam memastikan sistem penyemprotan bekerja secara optimal. Proses ini melibatkan penyatuan berbagai perangkat keras, seperti sensor, aktuator, dan modul komunikasi, dengan perangkat lunak yang mengatur logika kontrol dan pengolahan data. Setelah pemasangan rangkaian sistem seperti pada Gambar

29 selanjutnya menghubungkan sistem dengan program yang dirancang sebagai berikut :

```
from flask import Flask, render_template, request
from flask_socketio import SocketIO, emit
from dronekit import connect, VehicleMode
from pymavlink import mavutil

from threading import Thread

import Jetson.GPIO as GPIO

import time

from smbus2 import SMBus

from bmp280 import BMP280
```

program diatas merupakan baris program python yang berfungsi untuk mengambil library yang telah terinstal seperti library flask,websocket, dronekit dan sebagainya.

```
connection_string = '/dev/ttyAMA0'

baud_rate = 921600

app = Flask(__name__)
app.config['SECRET_KEY'] = 'secret!'
socketio = SocketIO(app)

relay_pin = 18

GPIO.setmode(GPIO.BOARD)

GPIO.setup(relay_pin, GPIO.OUT)
```

```

koneksi = "Connect"
armStatus = "Disarmed"
status = "OFF"

```

```

logicConnect = False
logicArm = False
logicStatus = False
bus = SMBus(1)

```

program diatas merupakan variabel statis yang berfungsi untuk menyimpan beberapa nilai yang diperlukan nantinya seperti status koneksi, status spray, dll.

```

def relay_on():
    GPIO.output(relay_pin, GPIO.LOW)
def relay_off():
    GPIO.output(relay_pin, GPIO.HIGH)

```

Program diatas merupakan baris program function yang berfungsi untuk menyalakan dan mematikan relay ketika fungsi tersebut dipanggil.

```

def generate_temperature():
    bmp280 = BMP280(i2c_dev=bus)

    while True:
        temperature = bmp280.get_temperature()
        pressure = bmp280.get_pressure()
        altitude = 44330 * (1 - (pressure / 1013.25) ** 0.1903)
        yaw = "180°"
        groundSpeed = 1.8

```



```

socketio.emit('temperature_update',
              {'temperature': temperature,
               "pressure" : pressure,
               "yaw": yaw,
               "altitude": altitude,
               "groundSpeed" : groundSpeed
              })

time.sleep(1)

```

program diatas merupakan baris program block function yang berfungsi untuk membaca data dari berbagai sensor yang digunakan seperti sensor BMP280 dan sensor di dalam pixhawk. Data tersebut dikirimkan dan ditampilkan ke monitor web kontrol.

```

@app.route('/')
def index():
    return render_template('index.html', status=status,
                           koneksi=koneksi, armStatus=armStatus)

@app.route('/connect', methods=['POST'])
def command():
    global koneksi
    global logicConnect

    if request.form['connect'] == 'connect':
        logicConnect = not logicConnect
        if logicConnect == True :
            koneksi = "Connect"
        else :
            koneksi = "Disconnect"

```

```

        return render_template('index.html', status=status,
koneksi=koneksi, armStatus=armStatus)

```

program diatas berfungsi untuk menghubungkan koneksi antara pixhawk dengan jetson nano ketika user menekan tombol connect pada web kontrol.

```

@app.route('/armed', methods=['POST'])
def armed():
    global armStatus
    global logicArm

    if request.form['armed'] == 'armed':
        logicArm = not logicArm
        if logicArm == True :
            armStatus = "Armed"
        else :
            armStatus = "Disarmed"
        return render_template('index.html', status=status,
koneksi=koneksi, armStatus=armStatus)

```

program diatas berfungsi untuk menghubungkan mengaktifkan mesin/motor pada drone ketika user menekan tombol arming pada web kontrol

```

@app.route('/spray', methods=['POST'])
def control_relay():
    global logicStatus
    global status

    if request.form['sprayer'] == 'on':
        logicStatus = not logicStatus

```

```

        if logicStatus==True :
            status = "ON"
            relay_on()
        else :
            status = "OFF"
            relay_off()

    return render_template('index.html', status=status,
koneksi=koneksi, armStatus=armStatus)

```

program diatas berfungsi untuk tigger untuk menyalakan dan mematikan waterpump ketika tombol spraying ditekan

```

@socketio.on('connect')
def handle_connect():
    print('Client connected')

if __name__ == '__main__':
    thread = Thread(target=generate_temperature)
    thread.start()

    socketio.run(app, host='0.0.0.0', port=5000, debug=True)

```

Program diatas merupakan baris program utama untuk menjalankan aplikasi web kontrol yang menggunakan Flask, dan untuk mengirimkan data sensor melalui program websocket

Berdasarkan program diatas Program Python menggunakan aplikasi web berbasis Flask yang berinteraksi dengan Pixhawk, sensor BMP280, serta menggunakan WebSocket melalui Flask-SocketIO untuk menampilkan data

secara real-time di web. Serta relay mampu mengatur sprayer menggunakan trigger dari pin GPIO Jetson Nano.

### C. Implementasi Sistem Kontrol Penyemprotan Pada *Drone UAV Hexacopter*

Implementasi sistem kontrol penyemprotan merupakan tahap akhir dalam merealisasikan desain dan integrasi yang telah dirancang sebelumnya. Pada tahap ini, sistem diuji untuk memastikan semua komponen perangkat keras dan perangkat lunak bekerja sesuai dengan fungsinya. Implementasi ini meliputi pemasangan sistem pada drone, konfigurasi parameter kontrol, serta pengujian performa dalam berbagai kondisi operasional. Langkah ini bertujuan untuk memastikan bahwa sistem dapat beroperasi secara efisien, stabil, dan memenuhi kebutuhan penyemprotan secara otomatis. Berikut merupakan beberapa pengujian komponen kontrol penyemprotan pestisida menggunakan *Drone Hexacopter*

#### 1. Pengujian Daya Baterai

Pengujian daya baterai yang dilakukan bertujuan untuk mengukur besar tegangan output pada baterai. Jenis baterai yang digunakan adalah baterai lipo spesifikasi 4 sel baterai dengan tegangan 14,8V saat keadaan kosong dan 16,8V saat baterai dalam keadaan penuh.



Gambar 27 Pengukuran Tegangan Baterai

Gambar 31 merupakan hasil dari pengukuran tegangan baterai yang diukur pada konektor series XT60 dan konektor parallel JST pada baterai. Pengukuran pada konektor XT60 diukur menggunakan digital multimeter sedangkan pengukuran tegangan pada konektor JST menggunakan digital Li-Po Alarm. Nilai tegangan yang terukur pada multimeter dan lipo alarm adalah sama sebesar 16,7V ketika baterai dalam kondisi penuh..


## 2. Pengujian Pixhawk






Pengukuran tegangan output pada modul pixhawk dilakukan dengan mengukur besar keluaran tegangan pada pin port pixhawk yang digunakan. Pengukuran tegangan pada pin pixhawk diukur menggunakan multimeter digital. Pengukuran dilakukan untuk mengetahui nilai tegangan dan memastikan port konektor pada pin dapat bekerja pada tegangan kerjanya. Berikut merupakan nilai tegangan yang terukur pada port pin pixhawk :

### a. Pengukuran Tegangan Port GPS Pixhawk

Pin GPS berfungsi untuk menghubungkan modul GPS eksternal ke pixhawk untuk mendapatkan data lokasi drone. Modul GPS menyediakan data lokasi dan navigasi yang penting dalam sistem kontrol penerbangan otonom.

*Tabel 3 Pengukuran Tegangan Port Pin GPS Pixhawk*

Pin	Signal	Pengukuran	Dokumentasi
1 (red)	VCC	5,245V	

2 (blk)	TX (OUT)	3,298V	
3 (blk)	RX (IN)	3,072V	
4 (blk)	CAN2 TX	3,296V	
5 (blk)	CAN2 RX	0,166V	
6 (blk)	GND	0V	





Tabel 5 merupakan hasil dari pengukuran tegangan pin GPS pada *pixhawk*.

Berdasarkan hasil pengukuran pada Tabel 9 nilai tegangan yang terukur pada pin GPS normal dan mendekati dengan nilai tegangan yang terdapat pada datasheet.

b. Pengukuran Tegangan Port Pin I2C Pixhawk

Pin I2C berfungsi untuk menghubungkan sensor dan perangkat lain yang mendukung I2C dan menggunakan protokol komunikasi I2C (*Inter-Integrated Circuit*). Pada kasus ini pin I2C digunakan untuk menghubungkan *pixhawk* dengan sensor kompas yang tertanam di dalam modul GPS.

Tabel 4 Pengukuran Tegangan Port Pin I2C Pixhawk

Pin	Signal	Pengukuran	Dokumentasi
1 (red)	VCC	5,254V	
2 (blk)	SCL	3,293V	
3 (blk)	SDA	3,293V	
4 (blk)	GND	0V	











Tabel 6 merupakan hasil dari pengukuran tegangan pin I2C pada *pixhawk*.

Berdasarkan hasil pengukuran pada Tabel 10 nilai tegangan yang terukur pada pin I2C normal dan mendekati dengan nilai tegangan yang terdapat pada datasheet *pixhawk*.

### 3. Pengujian Jetson Nano

Pengukuran tegangan GPIO pada Jetson Nano dilakukan dengan cara mengukur besar tegangan output pada pin GPIO Jetson nano. Pengukuran tegangan dilakukan dengan menggunakan multimeter digital. Pengukuran bertujuan untuk mengetahui nilai tegangan dan memastikan output konektor pada pin GPIO dapat bekerja dengan baik. Berikut merupakan nilai tegangan yang terukur pada pin GPIO Jetson nano :

*Tabel 5 Pengukuran Tegangan pin GPIO Jetson Nano*

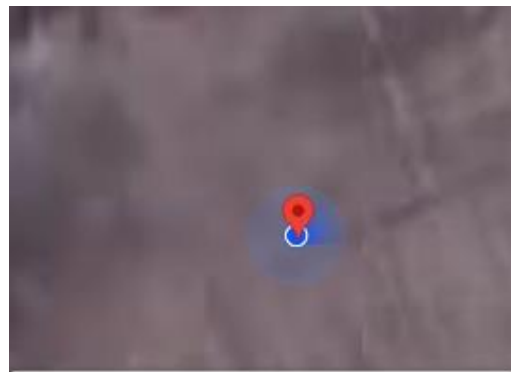
No	Pin	Pengukuran	Dokumentasi
1.	5V	5,012	
2.	GND	0	
3.	TXD/D8	3,342	
4.	RXD/D10	3,340	
5	D18	3,370	
6	3.3V	3,355	
7	SDA/D3	3,354	
8	SCL/D5	3,350	
9	D7	3,350	
10	GND	0	

Tabel 7 diatas merupakan hasil dari pengukuran besaran nilai tegangan yang terukur pada pin GPIO jetson nano. Berdasarkan tabel diatas tegangan kerja normal pada jetson nano yang terukur yaitu rentang nilai 0-3,3V

#### **4. Pengujian GPS**

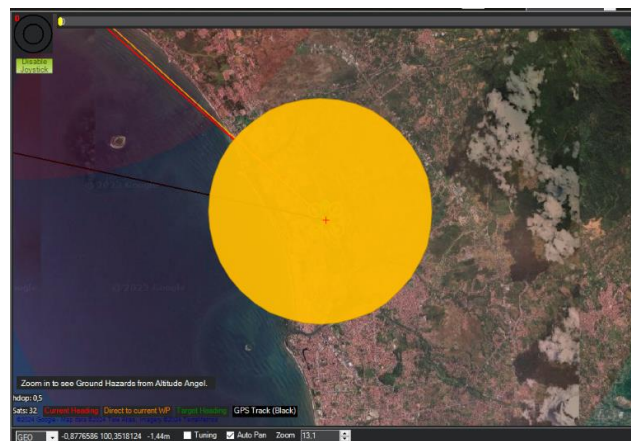


Pengujian GPS merupakan suatu pengujian dari keandalan sistem penerima GPS yang terpasang pada *drone*. Tujuan pengujian yaitu untuk memastikan bahwa drone dapat mengandalkan akurasi dan ketersediaan sinyal GPS yang memadai selama melakukan penerbangan. Saat pengujian, drone ditempatkan pada 6 titik koordinat yang berbeda di lokasi terbuka supaya penerimaan sinyal GPS dapat lebih optimal. Lokasi pengujian dilakukan di Landasan Udara TNI AU Sutan Sjahrir Padang. Proses ini dilakukan untuk melakukan pengukuran presisi posisi drone dan akan dibandingkan dengan tingkat akurasi posisi GPS pada handphone. Gambar 32 dan Tabel 8 berikut merupakan hasil dari pengujian GPS.



-0.877351,100.351832

(a) Pengujian Koordinat GPS pada HP



(b) Pengujian Koordinat GPS pada Drone  
Gambar 28 Pengujian Koordinat GPS

Tabel 6 Hasil Pengujian GPS

Titik posisi	Koordinat GPS Drone		Koordinat GPS Handphone		error	
	<i>Latitude</i>	<i>Longitude</i>	<i>Latitude</i>	<i>Longitude</i>	<i>Latitude</i>	<i>Longitude</i>
Posisi 1	-0,8776955	100,3517839	-0,877595	100,351735	1%	0%
Posisi 2	-0,8773827	100,3518020	-0,877658	100,351801	3%	0%
Posisi 3	-0,8773549	100,3519934	-0,877458	100,352088	1%	0%
Posisi 4	-0,8774738	100,3520869	-0,877331	100,351995	2%	0%
Posisi 5	-0,8776132	100,3519675	-0,877624	100,351965	0%	0%
Posisi 6	-0,8776586	100,3518124	-0,877351	100,351832	3%	0%

Berdasarkan hasil pengujian GPS pada Tabel 12 terdapat rata-rata error 3% pada *latitude* dan 0% error pada *longitude*. Berikut rumus perhitungan error pembacaan GPS :

$$\bullet \quad error = \left| \frac{\text{koordinat GPS drone}}{\text{koordinat GPS HP}} - 1 \right| \times 100$$

Pada hasil pengujian terdapat error pembacaan pada GPS dikarenakan pada saat pengujian kondisi cuaca kurang baik sehingga mengakibatkan perbedaan pembacaan GPS pada drone.

## 5. Pengujian Komponen Penyemprotan

Pengujian komponen penyemprotan dilakukan untuk memastikan setiap perangkat dalam sistem penyemprotan bekerja sesuai dengan fungsinya. Proses ini melibatkan pengujian elemen-elemen utama seperti pompa, nozzle, dan saluran distribusi untuk mengevaluasi performa, daya tahan, serta efisiensi penyemprotan. Pengujian ini bertujuan untuk mendeteksi potensi kendala teknis dan memastikan semua komponen dapat beroperasi secara optimal sebelum diintegrasikan ke dalam sistem kontrol utama. Pengujian Komponen

penyemprotan seperti waterpump bekerja efektif pada tegangan 12 volt dari sumber daya baterai seperti yang terlihat pada Gambar 29.



*Gambar 29 Pengujian Tegangan Input pada waterpump DC*

Gambar 29 merupakan hasil pengukuran tegangan yang masuk pada waterpump DC ketika relay dalam kondisi on. *Waterpump* DC dinyalakan secara otomatis dengan menggunakan modul relay sebagai menghubungkan dan memutus arus listrik yang dialirkan ke *waterpump* secara otomatis melalui kendali dari PIN GPIO18 pada jetson nano. Besar tegangan yang terukur pada pin relay dapat dilihat pada Tabel 7 berikut :

*Tabel 7 Tegangan Output Pin AUX5 Pixhawk*

Pin Relay	Tegangan Kondisi ON	Tegangan Kondisi OFF
GND	0V	0V
VCC	5,25V	0V
IN	3,29V	0V

Perhitungan kecepatan penyemprotan dilakukan dengan cara mengisi tangki air mencapai kapasitas maksimum tangki yaitu sebesar 1600mL. Ketika waterpump DC diberikan tegangan listrik sebesar 12V maka air pada tangki akan dialirkan melalui slang dan *nozzle spray* yang digunakan. Sehingga menghasilkan disepersi penyebaran air seperti Gambar 28 berikut.



*Gambar 30 Aliran Penyemprotan Cairan*

Waterpump dinyalakan saat drone ditempatkan di atas permukaan tanah, lebar dispersi air yang disemprotan mampu mencapai hingga 40cm diameter lingkaran. Kecepatan debit air yang dihasilkan ketika melakukan semprotan seperti pada Tabel 4 berikut :

*Tabel 8 Hasil Pengukuran Debit Air*

Tegangan Waterpump (V)	Volume Air (mL)	Waktu Semprot (detik)	Debit Air (mL/detik)
12,16	200	8	25,00
12,16	400	15	26,67
12,16	600	23	26,09
12,16	800	31	25,81
12,16	1000	38	26,32
12,16	1200	46	26,09
12,16	1400	53	26,42
12,16	1600	61	26,23
Rata-rata debit air			26,08

Berdasarkan Tabel 4 diatas, debit air yang dihasilkan rata-rata sebesar 26 mL/detik dan waktu yang diperlukan ketika tangki dalam keadaan penuh adalah 61 detik hingga air dalam tangki dalam keadaan kosong. Maka dapat disimpulkan maksimal kemampuan drone melakukan penyemprotan adalah selama 1 menit atau sekitar 61 detik penerbangan.

## 6. Pengujian Daya Angkat Drone

Pengujian daya angkat merupakan serangkaian pengujian yang bertujuan untuk mengukur berapa maksimal beban yang mampu diangkat oleh drone, dan mengukur konsumsi daya. Pengujian dilakukan dengan cara membuat misi take off dan delay selama 60 detik diudara dengan membawa beban seperti pada Gambar 33 dan Tabel 9 berikut.



*Gambar 31 Pengujian Daya Angkat Drone hexacopter*

*Tabel 9 Hasil Pengujian Daya Angkat Drone Hexacopter*

Berat Drone (gr)	Beban Angkut (gr)	Durasi Terbang (s)	Tegangan Awal (V)	Tegangan Akhir (V)	Selisih Tegangan ( $\Delta V$ )
2800	0	60s	16,7	16,4	0,3
2800	400	60s	16,7	16,3	0,4
2800	800	60s	16,7	16,1	0,6
2800	1200	60s	16,7	15,8	0,9
2800	1600	60s	16,7	15,5	1,2

Berdasarkan data pada Tabel 15, ketika *drone* terbang tanpa membawa beban kondisi tegangan baterai sebelum take off adalah sebesar 16,7V ketika setelah rentang waktu 60 detik tegangan baterai berkurang hingga 16,4V, yang berarti besar konsumsi daya baterai ketika drone tanpa membawa beban selama 60 detik adalah sebesar 0,3 V. Ketika drone tersebut membawa beban maksimum hingga 1,6kg besar konsumsi daya baterai selama 60 detik sebesar 1,2V. Maka

dapat disimpulkan bahwa semakin besar beban angkut yang diberikan maka semakin besar konsumsi daya baterai yang digunakan

## 7. Pengujian Mekanisme Sistem Kontrol Penyemprotan

Pengujian mekanisme sistem kontrol penyemprotan dilakukan untuk memastikan bahwa seluruh komponen sistem berfungsi sesuai dengan desain yang direncanakan. Tahap ini mencakup pengujian kinerja perangkat keras, respons perangkat lunak, serta keandalan sistem dalam mengontrol proses penyemprotan secara otomatis. Gambar 32 ini bertujuan untuk memberikan gambaran visual yang jelas mengenai mekanisme kerja sistem secara keseluruhan.



*Gambar 32 Pengujian Mekanisme Penyemprotan*

Gambar 32 merupakan sebuah pengujian mekanisme penyemprotan yang sudah terintegrasi dengan aplikasi web kontrol. Pengujian dilakukan di lapangan hijau dengan luas sekitar  $60\text{m}^2$  dengan jalur semprot sejauh 60m. pengujian dilakukan dengan menguji keberhasilan suatu sistem kontrol yang telah dirancang. Berikut merupakan hasil dari pengujian mekanisme sistem penyemprotan.

Berdasarkan pada desain aplikasi kontrol pada Gambar 24 perancangan software, kontrol yang diuji adalah tombol connect, armed, input jarak, tombol navigasi, dan tombol spraying. Saat tombol connect ditekan tulisan pada tombol tersebut akan berubah menjadi disconnect, hal tersebut menandakan sistem/server mini komputer telah terhubung dengan drone. Disamping tombol connect terdapat tombol dengan tulisan armed yang difungsikan sebagai menyalakan mesin/motor drone, ketika tombol *armed* ditekan tulisan pada tombol tersebut akan berubah menjadi *disarmed*. Hal ini menandakan bahwa motor drone sudah dinyalakan. Form input yang bertuliskan “Masukkan jarak” merupakan form yang berfungsi untuk menuliskan jarak awal semprotan mulai dinyalakan, saat tombol kirim ditekan drone mulai lepas landas pada jarak yang telah ditentukan maka penyemprotan akan otomatis dinyalakan. Tombol berikutnya yang berada di bawah form jarak semprotan merupakan tombol navigasi drone yang berfungsi untuk menentukan arah bergerak drone dengan manual. Terakhir terdapat tombol spray yang berfungsi untuk mengendalikan penyemprotan secara manual, diatas tombol juga terdapat status penyemprotan yang bertujuan untuk mengetahui apakah drone sedang melakukan penyemprotan.

## **BAB V**

### **PENUTUP**

#### **A. Kesimpulan**

Berdasarkan hasil pengujian dan analisis yang telah dilakukan pada sistem kontrol penyemprotan cairan pestisida menggunakan *drone UAV Hexacopter*. Maka dapat disimpulkan sebagai berikut:

1. Sistem kontrol penyemprotan cairan pestisida otomatis ini berbasis website realtime menggunakan *framework Flask python* dengan sebuah mini komputer yang terpasang pada *drone* yang berperan sebagai server utama kontrol *drone*. Semua data yang diperoleh dari sensor-sensor ditampilkan pada halaman website kontrol melalui jaringan wifi lokal
2. Berbagai sensor yang digunakan seperti GPS, dan sensor pendukung lainnya di integrasikan dengan cara menghubungkan kabel atau socket komponen ke pin GPIO jetson nano menggunakan protokol komunikasi serial seperti SPI dan I2C dan UART.
3. Mekanisme sistem penyemprotan cairan pestisida secara otomatis melakukan penyemprotan ketika drone mulai lepas landas pada waktu yang sudah di inputkan ke dalam web kontrol. Penyemprotan juga bisa dilakukan secara manual dengan cara menekan tombol spray pada web kontrol

#### **B. Saran**

Selama proses pengujian sistem kontrol penyemprotan cairan pestisida dengan menggunakan *hexacopter* masih memiliki kekurangan. Oleh karena itu saran untuk pengembang penelitian ini berikutnya yaitu :



1. Aplikasi sistem kontrol masih terbilang kurang baik dikarenakan masih menggunakan jaringan wifi lokal yang mengakibatkan jangkauan maksimum kendali drone terbatas maka disarankan untuk pengembang selanjutnya untuk menggunakan kontrol dengan jangkauan yang lebih jauh seperti telemetri dan perangkat radio frequency.
2. Tampilan aplikasi sistem kontrol masih menggunakan *tag style* CSS pada template aplikasi website. Disarankan untuk menggunakan framework CSS seperti *Bootstrap* atau *TailwindCSS*.
3. Perangkat kontrol mini komputer jetson nano memiliki kapasitas RAM yang kecil yang mengakibatkan terjadinya jeda waktu saat menjalankan server website. Disarankan untuk menggunakan perangkat mini komputer seperti raspberry pi dengan kapasitas RAM yang lebih besar untuk mengatasi jeda saat menjalankan server aplikasi kontrol
4. Maksimum penyimpanan tangki yang digunakan sebesar 1,5 liter menyebabkan penyemprotan tidak sepenuhnya mengenai lahan. Untuk itu disarankan menggunakan tangki dengan kapasitas penyimpanan cairan yang lebih besar.

## DAFTAR PUSTAKA

- Abdul Wahid, A. (2020). Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi. *Jurnal Ilmu-Ilmu Informatika Dan Manajemen STMIK*, November, 1–5.
- Ahiarwar, S., Swarnkar, R., Bhukya, S., & Namwade, G. (2019). Application of drone in agriculture: A review. *International Journal of Chemical Studies*, 8(5), 181–187. <https://doi.org/10.22271/chemi.2020.v8.i5d.10529>
- Almetania, F., & Huda, Y. (2022). Rancangan Sistem Antenna Tracker pada Ground Station UAV Sebagai Media Pantau Pasca Bencana. *Voteteknika (Vocational Teknik Elektronika Dan Informatika)*, 10(3), 67. <https://doi.org/10.24036/voteteknika.v10i3.118803>
- Astuti, W., & Widyastuti, C. R. (2016). Pestisida Organik Ramah Lingkungan Pembasmi Hama Tanaman Sayur. *Rekayasa*, 14(2), 117.
- Dewi, A. Y., Harinita, D., & Bachtiar, A. (2018). Korelasi Gaya Angkat dengan Kecepatan Putaran Rotor pada Sistem Pesawat Quadcopter. *Teknik Elektro Institut Teknologi Padang*.
- Febri, H., Sumbung, R., Mesin, J. T., Negeri, P., & Pandang, U. (2021). *DRONE UNTUK DETEKSI HAMA DAN PENYEMPROTAN PESTISIDA PADA TANAMAN PADI*.
- Feriadi, D. (2017). *Aplikasi Sistem Pendukung Keputusan Untuk Simulasi Diagnosa Hama dan Penyakit Pada Tanaman Cabai*. 106–113.
- Hidayat, R., Muhaimin, & Aidi, F. (2019). Rancang Bangun Prototype Drone Penyemprot Pestisida Untuk Pertanian Padi Secara Otomatis. *Jurnal Tektro*,

- 3(2), 86–94. <http://e-jurnal.pnl.ac.id/index.php/TEKTRO/article/view/1550>
- Hidayati, N. (2019). Penggunaan Metode Waterfall Dalam Rancang Bangun Sistem Informasi Penjualan. *Generation Journal*, 3(1), 1–10.
- Indreswari Suroso. (2021). *Institut teknologi nasional, PENGERTIAN DRONE*. 4–11.
- Khumairowati, & B. Yulianti. (2020). Analisis Motor pada Quadcopter. *Jurnal Mitra Manajemen*, 8(2), 121–127. <https://journal.universitassuryadarma.ac.id/index.php/jmm/article/view/519>
- Kusumaningrum, S. I. (2019). Pemanfaatan Sektor Pertanian Sebagai Penunjang Pertumbuhan Perekonomian Indonesia. *Jurnal Transaksi*, 11(1), 80–89. <http://ejournal.atmajaya.ac.id/index.php/transaksi/article/view/477>
- Maulana Ichsan, I., & Rizianiza, I. (2021). Perbandingan Uji Kestabilan Attitude Hexacopter Pada Metode Root Locus dan Routh Hurwitz. *Jurnal METTEK*, 7(2), 57. <https://doi.org/10.24843/mettek.2021.v07.i02.p01>
- Perdana, A. R., & Rizianiza, I. (2021). Perancangan Sistem Pengendalian Menggunakan Kontroler PID pada Gerakan Pitch dan Roll untuk Stabilitas Attitude Hexacopter. *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, 11(2), 386–398. <https://doi.org/10.24176/simet.v11i2.5172>
- Rahmadiansyah, A., Orlanda, E., Wijaya, M., Nugroho, H. W., & Firmansyah, R. (2017). Perancangan Sistem Telemetry Untuk Mengukur Intensitas Cahaya Berbasis Sensor Light Dependent Resistor Dan Arduino Uno. *JEEE-U (Journal of Electrical and Electronic Engineering-UMSIDA)*, 1(1), 15–21. <https://doi.org/10.21070/jeee-u.v1i1.760>

- Scott, J. E., & Scott, C. H. (2017). Drone delivery models for healthcare. *Proceedings of the Annual Hawaii International Conference on System Sciences, 2017-Janua*, 3297–3304. <https://doi.org/10.24251/hicss.2017.399>
- Septian, A. D., & Huda, Y. (2022). *Perancangan Sistem Tiltrotor pada UAV Sebagai Media Pantau Pasca Bencana* Flowchart merupakan diagram alir dari proses kerja sebuah alat, proses ini memberikan gambaran dari tahapan yang akan dilalui mulai dari menghidupkan alat hingga alat dimatikan kemb. 10(3).
- Taufik, A., Habriansyah, I., Fachturrahman, F., & Sumbung, H. F. R. (2022). Drone Untuk Deteksi Hama dan Penyemprotan Pestisida pada Tanaman Padi. *Jurnal Teknik Mesin Sinergi*, 20(2), 139. <https://doi.org/10.31963/sinergi.v20i2.3737>
- Widiasari, C., & Este Dulan Agustinus S., R. (2020). Rancang Bangun Drone Quadcopter Tanpa Awak Penyiram Pupuk Tanaman. *Jurnal ELEMENTER*, 6(2), 81–90.
- Yallappa, D., Veerangouda, M., & Maski, D., Palled, V., & Bheemanna, M. (2017). Development and evaluation of drone mounted sprayer for pesticide applications to crops. *Development and Evaluation of Drone Mounted Sprayer for Pesticide Applications to Crops*, 16.1.2015. <https://ieeexplore.ieee.org/abstract/document/8239330/>

## LAMPIRAN

### Lampiran 1. Program Kontrol Sistem (*Python*) :

```
from flask import Flask, render_template, request
from flask_socketio import SocketIO, emit
from dronekit import connect, VehicleMode
from pymavlink import mavutil
from threading import Thread
import Jetson.GPIO as GPIO
import time
from smbus2 import SMBus
from bmp280 import BMP280

connection_string = '/dev/ttyAMA0'
baud_rate = 921600

app = Flask(__name__)
app.config['SECRET_KEY'] = 'secret!'
socketio = SocketIO(app)

relay_pin = 18
GPIO.setmode(GPIO.BOARD)
GPIO.setup(relay_pin, GPIO.OUT)

koneksi = "Connect"
armStatus = "Disarmed"
status = "OFF"

logicConnect = False
```

```

logicArm = False
logicStatus = False

bus = SMBus(1)

def relay_on():
    GPIO.output(relay_pin, GPIO.LOW)

def relay_off():
    GPIO.output(relay_pin, GPIO.HIGH)

def generate_temperature():
    bmp280 = BMP280(i2c_dev=bus)

    while True:
        temperature = round(bmp280.get_temperature(), 2)
        pressure = round(bmp280.get_pressure(), 2)
        altitude = round(44330 * (1 - (pressure / 1013.25) **
0.1903), 2)
        yaw = "180°"
        groundSpeed = 1.8
        socketio.emit('temperature_update',
                        {'temperature': temperature,
                         "pressure" : pressure,
                         "yaw": yaw,
                         "altitude": altitude,
                         "groundSpeed" : groundSpeed,
                        })
        time.sleep(1)

@app.route('/')
def index():

```

```

        return render_template('index.html', status=status,
                                koneksi=koneksi, armStatus=armStatus)

```

```

@app.route('/connect', methods=['POST'])

```

```

def command():

```

```

    global koneksi

```

```

    global logicConnect

```

```

    if request.form['connect'] == 'connect':

```

```

        logicConnect = not logicConnect

```

```

        if logicConnect == True :

```

```

            koneksi = "Connect"

```

```

        else :

```

```

            koneksi = "Disconnect"

```

```

        return render_template('index.html', status=status,
                                koneksi=koneksi, armStatus=armStatus)

```

```

@app.route('/armed', methods=['POST'])

```

```

def armed():

```

```

    global armStatus

```

```

    global logicArm

```

```

    if request.form['armed'] == 'armed':

```

```

        logicArm = not logicArm

```

```

        if logicArm == True :

```

```

            armStatus = "Armed"

```

```

        else :

```

```

            armStatus = "Disarmed"

```

```

        return render_template('index.html', status=status,
                                koneksi=koneksi, armStatus=armStatus)

```

```

@app.route('/spray', methods=['POST'])

```

```

def control_relay():

```

```

global logicStatus
global status

if request.form['sprayer'] == 'on':
    logicStatus = not logicStatus
    if logicStatus==True :
        status = "ON"
        relay_on()
    else :
        status = "OFF"
        relay_off()

    return render_template('index.html', status=status,
koneksi=koneksi, armStatus=armStatus)

@app.route('/ kirim', methods=['POST'])
def kirim():
    global logicStatus
    global status

    if request.form[' kirim'] == 'on':
        logicStatus = not logicStatus
        time.sleep(int(request.form['jarak']))
        if logicStatus==True :
            status = "ON"
            relay_on()
        else :
            status = "OFF"
            relay_off()

    return render_template('index.html', status=status,
koneksi=koneksi, armStatus=armStatus)

@socketio.on('connect')

```



```

def handle_connect():
    print('Client connected')

if __name__ == '__main__':
    thread = Thread(target=generate_temperature)
    thread.start()
    socketio.run(app, host='0.0.0.0', port=5000, debug=True)

```

## Lampiran 2. Program Tampilan Website Kontrol (*HTML*) :

```

<!DOCTYPE HTML>

<html lang="en">

<head>

    <title>Drone Control Web Server</title>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <link rel="icon" href="data:,">

                                                                    <script
src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/4.0.0/socket
.io.min.js"></script>

<style>

    html {

        font-family: Arial, Helvetica, sans-serif;

        text-align: center;

    }

```

```
h1 {  
    font-size: 1.5rem;  
    color: white;  
}  
  
h2 {  
    font-size: 1.5rem;  
    font-weight: bold;  
    color: #143642;  
}  
  
body {  
    margin: 0;  
}  
  
.topnav {  
    overflow: hidden;  
    background-color: #143642;  
}  
  
.content {  
    padding: 30px;  
    max-width: 600px;  
    margin: 0 auto;  
    display: block;  
    justify-content: center;  
    align-items: center;  
}  
  
.card {  
    background-color: #F8F7F9;  
    box-shadow: 2px 2px 12px 1px rgba(172, 159, 159, 0.5);  
    padding-top: 10px;
```

```

padding-bottom: 20px;

border-radius: 20px;
}

.block {

display: flex;

gap: 10px;

justify-content: center;

margin-bottom: 10px;

margin-top: 20px;
}

.connect button {

background-color: #00e1f1;

outline: auto;

width: 120px;

height: 50px;

font-size: 20px;

font-family: "Lucida Console", "Courier New", monospace;
}

.connect button:hover {

cursor: pointer;

background-color: #0f8b8d;
}

.connect button:active {

box-shadow: 2 2px #CDCDCD;

transform: translateY(2px);

background-color: #00b0bd;
}

.armed button {

```

```
background-color: #00e1f1;

outline: auto;

width: 120px;

height: 50px;

font-size: 20px;

font-family: "Lucida Console", "Courier New", monospace;
}

.armed button:hover {

    cursor: pointer;

    background-color: #0f8b8d;
}

.armed button:active {

    box-shadow: 2 2px #CDCDCD;

    transform: translateY(2px);

    background-color: #00b0bd;
}

.state {

    font-size: 1.5rem;

    color: #8c8c8c;

    font-weight: bold;
}

.sprayer {

    padding: 15px 50px;

    font-size: 24px;

    text-align: center;

    outline: none;

    color: #fff;

    background-color: #0f8b8d;
```

```

border: none;

border-radius: 10px;

-webkit-touch-callout: none;

-webkit-user-select: none;

-khtml-user-select: none;

-moz-user-select: none;

-ms-user-select: none;

user-select: none;
}

.sprayer:hover {

    background-color: #04d1d4;

    cursor: pointer;
}

.sprayer:active {

    background-color: #0f8b8d;

    box-shadow: 2 2px #CDCDCD;

    transform: translateY(2px);
}

.flight-status {

    display: grid;

    margin: auto;

    margin-top: 20px;

    background-color: #bcffc1;

    box-shadow: 2px 2px 12px 1px rgba(172, 159, 159, 0.5);

    padding-top: 10px;

    padding-bottom: 20px;

    grid-template-columns: auto auto;

    padding: 10px;

```

```

        gap: 10px;

        font-size: 20px;

        border-radius: 20px;
    }

    .flight-status div span {

        font-size: 30px;
    }

    p.status {

        font-family: 'Courier New', Courier, monospace;
    }

    /* tombol */

    .container {

        display: flex;

        flex-direction: column;

        align-items: center;

        gap: 10px;
    }

    .btn {

        background-color: #007bff;

        color: white;

        border: none;

        border-radius: 5px;

        padding: 10px 20px;

        font-size: 20px;

        cursor: pointer;

        transition: background-color 0.3s ease;
    }

```

```

.btn:hover {
    background-color: #0056b3;
}

.middle-row {
    display: flex;
    justify-content: space-between;
    gap: 10px;
}

/* form untuk jarak semprot */

.form-container {
    background-color: #ffffff;
    border-radius: 10px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    padding: 20px;
    width: 300px;
    justify-content: center;
    align-items: center;
    margin-left: 10px;
    margin-bottom: 10px;
}

.form-container h1 {
    font-size: 20px;
    color: #333;
    margin-bottom: 20px;
    text-align: center;
}

.form-row {
    display: flex;

```

```
        justify-content: space-between;
        align-items: center;
    }
    .form-container input[type="number"] {
        flex: 1;
        padding: 10px;
        border: 1px solid #ccc;
        border-radius: 5px;
        font-size: 16px;
    }
    .form-container button {
        margin-left: 10px;
        background-color: #007bff;
        color: white;
        border: none;
        border-radius: 5px;
        padding: 10px 15px;
        font-size: 16px;
        cursor: pointer;
        transition: background-color 0.3s ease;
    }
    .form-container button:hover {
        background-color: #0056b3;
    }
</style>
</head>
<body>
    <div class="topnav">
```



```

        <h1 class="judul">Sistem Kontrol Penyemprotan Cairan
Pestisida Otomatis Menggunakan Drone UAV Hexacopter</h1>

    </div>

    <div class="content">

        <div class="card">

            <div class="block">

                <form method="post" action="/connect"
class="connect">

                    <button type="submit" name="connect"
class="connect" id="connect" value="connect">{{ koneksi
                    }}</button>

                </form>

                <form method="post" action="/armed" class="armed">

                    <button type="submit" name="armed" class="armed"
id="armed" value="armed">{{ armStatus }}</button>

                </form>

            </div>

            <p class=status>

                Mode      :      <strong><span
id="Modes">GUIDED</span></strong><br>

                Status: <strong><span id="status"> {{ koneksi
                }}</span></strong><br>

                Altitude  :      <strong><span
id="altitude">0</span>m</strong>

            </p>

            <div class="form-container">

                <form action="/ kirim" method="POST">

                    <div class="form-row">

```

```

<input type="number" name="jarak"
placeholder="Masukkan jarak (meter)" required>

<button type="submit" name=" kirim"
value="on">Kirim</button>

</div>

</form>

</div>

<div class="container">

<button class="btn" id="up">↑</button>

<div class="middle-row">

<button class="btn" id="left">←</button>

<button class="btn" id="right">→</button>

</div>

<button class="btn" id="down">↓</button>

</div>

<h2 class="gpio2">Output - GPIO 18</h2>

<p class="state">State: <span id="state">{{ status
}}</span></p>

<form method="post" action="/spray">

<button type="submit" id="sprayer" class="sprayer"
name="sprayer" value="on">Spray</button>

</form>

</div>

<div class="flight-status">

<div>Altitude <br><span id="alt">??</span>m</div>

<div>Ground Speed <br><span id="gs">??</span>m/s</div>

<div>Sprayer<br><span
id="spray">{{status}}</span></div>

```

```

        <div>Yaw<br><span id="yaw"> ?? </span>deg</div>

        <div>Temperature<br><span
id="temperature">__</span>°C</div>

        <div>Pressure<br><span id="press">??</span>hPa</div>

    </div>

</div>

<script type="text/javascript">
    var socket = io();

    socket.on('temperature_update', function (data) {

        document.getElementById('temperature').innerText =
data.temperature;

        document.getElementById('press').innerText =
data.pressure;

        document.getElementById('yaw').innerText = data.yaw;

        document.getElementById('alt').innerText =
data.altitude;

        document.getElementById('altitude').innerText =
data.altitude;

        document.getElementById('gs').innerText =
data.groundSpeed;

    });

    socket.on('connect', function () {

        console.log('Connected to server');

    });

</script>

</body>

</html>

```