

DRONE UNTUK DETEKSI HAMA DAN PENYEMPROTAN
PESTISIDA PADA TANAMAN PADI



SKRIPSI

Diajukan sebagai salah satu syarat untuk menyelesaikan
pendidikan diploma empat (D-4) Program Studi Teknik Mekatronika
Jurusan Teknik Mesin
Politeknik Negeri Ujung Pandang

FACHTURRAHMAN

444 17 017

HUTOMO FEBRI RICHARDO SUMBUNG

444 17 022

PROGRAM STUDI D-4 TEKNIK MEKATRONIKA
JURUSAN TEKNIK MESIN
POLITEKNIK NEGERI UJUNG PANDANG
MAKASSAR
2021

HALAMAN PENGESAHAN

Skripsi dengan judul “**Drone untuk Deteksi Hama dan Penyemprotan Pesticida pada Tanaman Padi**” oleh Fachturrahman NIM 444 17 017 dan Hutomo Febri Richardo Sumbang NIM 444 17 022 telah diterima dan disahkan sebagai salah satu syarat untuk memperoleh gelar Sarjana Terapan Teknik pada Program Studi Teknik Mekatronika Jurusan Teknik Mesin Politeknik Negeri Ujung Pandang.

Makassar, 7 September 2021

Pembimbing I,

Pembimbing II,



Dr.Eng. Akhmad Taufik, S.T., M.T.
19760413 200812 1 003



Imran Habriansyah, S.ST., M.T.
19881005 201903 1 009

Mengetahui

Koordinator Program Studi Teknik Mekatronika,



Dr. Ir. Simon Ka'ka, M.T.
19590913 198803 1 001

HALAMAN PENERIMAAN

Pada hari ini, Selasa tanggal 7 September 2021, tim penguji ujian sidang skripsi telah menerima skripsi mahasiswa: Fachturrahman NIM 444 17 017 dan Hutomo Febri Richardo Sumbung NIM 444 17 022 dengan judul: “**Drone untuk Deteksi Hama dan Penyemprotan Pestisida pada Tanaman Padi**”

Makassar, 7 September 2021

Tim Penguji Ujian Sidang Skripsi :

- | | | |
|--|------------|---|
| 1. Ir. Lewi, M.T. | Ketua |  |
| 2. Ir. Remigius Tandioga, M.Eng.Sc. | Sekretaris |  |
| 3. Dr.Eng. Abdul Kadir Muhammad, S.T.,
M.Eng. | Anggota |  |
| 4. Mukhtar, S.Pd., M.Eng. | Anggota |  |
| 5. Dr.Eng. Akhmad Taufik, S.T., M.T. | Anggota |  |
| 6. Imran Habriansyah, S.ST., M.T. | Anggota |  |

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah swt. karena berkat rahmat dan karunia-Nya, penulisan skripsi ini yang berjudul ”*Drone* untuk Deteksi Hama dan Penyemprotan Pestisida pada Tanaman Padi” dapat diselesaikan dengan baik.

Dalam penulisan skripsi ini tidak sedikit hambatan yang penulis alami. Namun, berkat bantuan berbagai pihak terutama pembimbing, hambatan tersebut dapat teratasi. Sehubungan dengan itu, pada kesempatan dan melalui lembaran ini penulis menyampaikan terima kasih dan penghargaan kepada:

1. Keluarga yang selalu memberikan dukungan dari segi materi maupun moril, mendoakan segala keselamatan dan kelancaran serta memberikan semangat motivasi sehingga skripsi ini dapat terselesaikan.
2. Bapak Prof. Dr. Muhammad Anshar, M.Si., Ph.D., selaku Direktur Politeknik Negeri Ujung Pandang.
3. Bapak Rusdi Nur S.ST., M.T., Ph.D., selaku Ketua Jurusan Teknik Mesin Politeknik Negeri Ujung Pandang.
4. Bapak Dr. Ir. Simon Ka'ka, M.T., selaku Koordinator Program Studi D4 Teknik Mekatronika Politeknik Negeri Ujung Pandang.
5. Bapak Dr.Eng. Akhmad Taufik, S.T., M.T., selaku Dosen Pembimbing I yang telah memberikan bimbingan, perhatian, dan kesempatannya untuk mengarahkan penulis dalam menyelesaikan skripsi ini.

6. Bapak Imran Habriansyah, S.ST., M.T., selaku Dosen Pembimbing II yang telah memberikan bimbingan, perhatian, dan kesempatannya untuk mengarahkan penulis dalam menyelesaikan skripsi ini.
7. Teman-teman seperjuangan kelas 4 (angkatan 2017) Teknik Mekatronika, Jurusan Teknik Mesin, Politeknik Negeri Ujung Pandang, yang telah banyak berdiskusi dan bekerjasama dengan penulis selama proses pembuatan skripsi.
8. Serta semua pihak yang telah banyak membantu dan memberikan masukan dalam penyelesaian skripsi ini yang tidak dapat disebutkan satu persatu.

Semoga Allah SWT membalas kebaikan siapa pun yang terlibat dalam penyusunan skripsi ini dengan nikmat dan berkah yang melimpah. Amin. Besar harapan penulis, dengan hadirnya skripsi ini dapat memberikan sumbangsih yang berarti demi kemajuan ilmu pengetahuan bangsa terutama pada bidang Mekatronika.

Penulis menyadari bahwa skripsi ini belum sempurna. Oleh karena itu, penulis mengharapkan kritikan dan saran bersifat membangun demi kesempurnaan skripsi ini dan demi perbaikan pada masa mendatang. Semoga skripsi ini bermanfaat bagi pembacanya.

Makassar, 7 September 2021

Penulis

DAFTAR ISI

	hlm.
HALAMAN SAMPUL.....	i
HALAMAN PENGESAHAN.....	ii
HALAMAN PENERIMAAN	iii
KATA PENGANTAR	iv
DAFTAR ISI.....	vi
DAFTAR TABEL	ix
DAFTAR GAMBAR.....	x
DAFTAR LAMPIRAN.....	xiii
SURAT PERNYATAAN.....	xiv
RINGKASAN.....	xvi
SUMMARY	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Ruang Lingkup Penelitian	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
BAB II TINJAUAN PUSTAKA	5
2.1 <i>Drone</i>	5
2.1.1 Jenis – jenis <i>drone</i>	5
2.1.2 Cara Kerja <i>Drone</i>	6
2.2 Komponen-Komponen <i>Drone</i>	7
2.2.1 <i>Flight Controller</i>	7
2.2.2 <i>Brushless dc Motor (BLDC)</i>	8
2.2.3 <i>Radio Controller</i>	9

2.2.4 Baterai <i>Lithium Polymer</i> (Li-Po)	10
2.2.5 <i>Frame Drone</i>	11
2.2.6 <i>Electronic Speed Controller</i> (ESC).....	11
2.2.7 Baling-baling (<i>Propeller</i>)	12
2.2.8 <i>Raspberry Pi</i>	12
2.2.9 <i>Global Positioning System</i> (GPS).....	13
2.2.10 <i>Router</i>	14
2.2.11 IP Webcam	14
2.3 <i>Image Processing</i>	15
2.3.1 Pengertian	15
2.3.2 <i>Color Filtering</i>	16
2.3.3 Model Warna HSV (<i>Hue Saturation Value</i>).....	17
2.4 OpenCV	17
2.5 <i>Drone</i> pada Aplikasi Pertanian	19
2.6 Perhitungan Parameter-Parameter pada <i>Drone</i>	20
2.6.1 Perhitungan Gaya Angkat <i>Drone</i> (<i>Thrust</i>)	20
2.6.2 Estimasi Waktu Terbang <i>Drone</i>	21
2.7 Penelitian Terkait <i>Drone</i> yang Diaplikasikan pada Bidang Pertanian	22
2.7.1 Penggunaan <i>Drone</i> dalam Mengaplikasikan Pestisida di Daerah Sungai Besar, Malaysia (Khoirunisa dan Fitrianingrum Kurniawati, 2019).....	22
2.7.2 Pengembangan Sistem Penyemprotan pada <i>Platform</i> Pesawat Tanpa Awak Berbasis <i>Quadcopter</i> untuk Membantu Petani Mengurangi Biaya Pertanian dalam Mendorong Konsep Pertanian Pintar atau <i>Smart Farming</i> (Santoso dan Kris Hariyanto, 2017)	24
2.7.3 Rancang Bangun Sistem Penyemprotan Pestisida dan Pupuk Pada Tanaman Padi Menggunakan <i>Mikrokontroler</i> (Arnizam, 2018).....	25
2.7.4 Prototipe Tabung Semprot Pupuk Cair Berbasis Wahana <i>Quadcopter</i> bagi Lahan Tanaman Padi (Akbar dan Anton Yudhana, 2016).....	26
2.7.5 Aplikasi <i>Drone</i> Wawasan Tani untuk Pertanian di Simpang Lima, Sungai Besar, Selangor (Ikhwana dan Dwi Retno Hapsari, 2019).....	27

2.8 <i>Road Map</i> Penelitian <i>Drone</i>	29
BAB III METODE PENELITIAN	30
3.1 Tempat dan Waktu Penelitian.....	30
3.2 Alat dan Bahan Penelitian.....	30
3.3 Diagram Blok Sistem	31
3.4 Tahapan Penelitian	33
3.4.1 Studi Literatur	34
3.4.2 Perancangan <i>Drone</i>	35
3.4.3 Pembuatan Mekanisme Penyemprotan Insektisida	41
3.4.4 Pembuatan Algoritma <i>Image Processing</i>	45
3.4.5 Pengujian Alat dan Pengambilan Data.....	46
3.4.6 Pengolahan Data.....	47
3.4.7 Penyusunan Laporan	47
3.5. Rencana Eksperimen	47
BAB IV HASIL DAN PEMBAHASAN	51
4.1 Pengujian Terbang <i>Drone</i>	51
4.2 Pengujian Mekanisme Penyemprotan	53
4.3 Pengujian Algoritma <i>Image Processing</i>	55
4.3.1 Uji Coba Algoritma <i>Image Processing</i>	55
4.3.2 Hasil Pengambilan Data	60
4.3.3 Hasil dan Analisis Pengujian Algoritma <i>Image Processing</i> dengan Mengaktifkan Mekanisme Penyemprotan Insektisida .	62
BAB V PENUTUP	64
5.1 Kesimpulan	64
5.2 Saran.....	64
DAFTAR PUSTAKA	66
LAMPIRAN	70

DAFTAR TABEL

	hlm.
Tabel 3.1 Alat dan Bahan Penelitian	30
Tabel 4.1 Hasil Eksperimen Terbang	52
Tabel 4.2 Hasil Eksperimen Mekanisme Penyemprotan	54



DAFTAR GAMBAR

	hlm.
Gambar 1.1 Tanaman Padi yang Mengalami Serangan Wereng Coklat.....	2
Gambar 2.1 <i>Fixed Wing Drone</i>	5
Gambar 2.2 <i>Multicopter Drone</i>	6
Gambar 2.3 Arah Putaran Baling-Baling pada <i>Drone</i>	7
Gambar 2.4 <i>Flight Controller</i>	8
Gambar 2.5 Motor <i>Brushless dc</i> (BLDC).....	9
Gambar 2.6 <i>Radio Controller</i>	10
Gambar 2.7 Baterai Li-Po.....	10
Gambar 2.8 <i>Frame Drone</i>	11
Gambar 2.9 <i>Electronic Speed Control</i> (ESC).....	11
Gambar 2.10 Baling-baling (<i>Propeller</i>).....	12
Gambar 2.11 <i>Raspberry Pi 3B+</i>	13
Gambar 2.12 <i>Global Positioning System</i> (GPS).....	13
Gambar 2.13 <i>Router</i>	14
Gambar 2.14 Logo IP Webcam	15
Gambar 2.15 Contoh Penggunaan <i>Image Processing</i> untuk Deteksi Objek	16
Gambar 2.16 Contoh Penggunaan <i>Color Filtering</i>	16
Gambar 2.17 Contoh Penggunaan <i>Image Processing</i> dengan Model Warna HSV	17
Gambar 2.18 Logo OpenCV.....	19
Gambar 2.19 Contoh Penggunaan <i>Drone</i> pada Bidang Pertanian.....	20

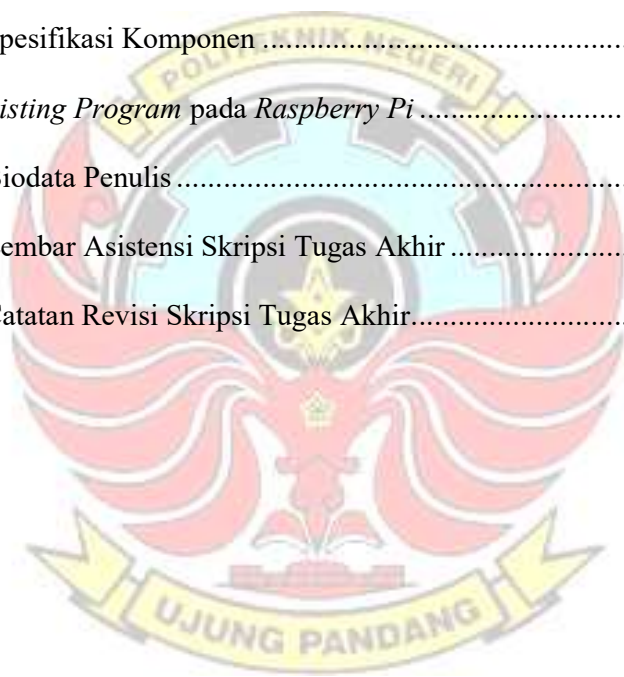
Gambar 2.20	Penggunaan <i>Drone</i> dalam Mengaplikasikan Pestisida di Daerah Sungai Besar, Malaysia.....	23
Gambar 2.21	Pengembangan Sistem Penyemprotan pada <i>Platform</i> Pesawat Tanpa Awak Berbasis <i>Quadcopter</i> untuk Membantu Petani Mengurangi Biaya Pertanian dalam Mendorong Konsep Pertanian Pintar atau <i>Smart Farming</i>	25
Gambar 2.22	Rancang Bangun Sistem Penyemprotan Pestisida dan Pupuk pada Tanaman Padi Menggunakan Mikrokontroler	26
Gambar 2.23	Tabung Semprot Pupuk Cair Berbasis Wahana <i>Quadcopter</i> bagi Lahan Tanaman Padi.....	27
Gambar 2.24	Aplikasi <i>Drone</i> Wawasan Tani untuk Pertanian di Simpang Lima, Sungai Besar, Selangor	28
Gambar 2.25	<i>Road Map</i> Penelitian <i>Drone</i> PNUP	29
Gambar 3.1	Diagram Blok Sistem.....	31
Gambar 3.2	Diagram Alir Penelitian	34
Gambar 3.3	Tampak Depan dari Hasil Rancangan Sistem Mekanik <i>Drone</i>	36
Gambar 3.4	Tampak Atas dari Hasil Rancangan Sistem Mekanik <i>Drone</i> . ..	37
Gambar 3.5	Rangkaian Hubungan antara <i>Flight Controller</i> , <i>Radio Controller</i> , ESC, Motor <i>Brushless</i> , PPM <i>Encoder</i> dan <i>Receiver RC</i>	38
Gambar 3.6	Rangkaian Hubungan antara <i>Flight Controller</i> dan <i>Safety Switch</i>	39
Gambar 3.7	Rangkaian Hubungan antara <i>Flight Controller</i> , Telemetri dan Laptop.....	40
Gambar 3.8	Rangkaian Hubungan antara <i>Flight Controller</i> dan <i>Buzzer</i> ...	40
Gambar 3.9	Rangkaian Hubungan antara <i>Flight Controller</i> dan GPS.....	41
Gambar 3.10	Uji Coba Pompa Wiper 12V.....	42

Gambar 3.11	Hasil Perancangan Mekanik untuk Mekanisme Penyemprotan	43
Gambar 3.12	Rangkaian Hubungan antara <i>Raspberry Pi</i> , Pompa Wiper, dan Modul Relay	44
Gambar 3.13	Tampilan Program Eksperimen Penyemprotan saat dijalankan	44
Gambar 3.14	Diagram Alir Perancangan <i>Software</i>	45
Gambar 3.15	Sketsa Rencana Eksperimen Terbang di Lapangan Kampus PNUP.....	48
Gambar 3.16	Sketsa Rencana Eksperimen Penyemprotan Cairan	48
Gambar 3.17	Sketsa Rencana Eksperimen <i>Real</i> di atas Sawah Petani	49
Gambar 3.18	Sketsa Rencana Eksperimen Penyemprotan Insektisida	50
Gambar 4.1	Pengujian Terbang <i>Drone</i>	51
Gambar 4.2	Proses Eksperimen Mekanisme Penyemprotan.....	53
Gambar 4.3	Hasil eksperimen mekanisme penyemprotan pada ketinggian $\pm 1,5$ meter.....	54
Gambar 4.4	Hasil Program <i>Colour Filtering</i> saat dijalankan.....	55
Gambar 4.5	Hasil Program <i>Colour Filtering</i> untuk Mengaktifkan Modul Relay saat dijalankan	56
Gambar 4.6	Program Perekaman Video.....	57
Gambar 4.7	Hasil Program Pemutar Video saat dijalankan	58
Gambar 4.8	Hasil Program <i>Color Filtering</i> pada Gambar Daun Padi.....	59
Gambar 4.9	Rangkaian Hubungan antara <i>Raspberry Pi 3</i> dan Kamera <i>Smartphone</i>	60
Gambar 4.10	Jalur Terbang yang <i>Drone</i> akan lewati untuk perekaman video ...	61
Gambar 4.11	Proses Pengambilan Data berupa Rekaman Video.....	61

Gambar 4.12 Proses <i>Color Filtering</i> pada Hasil Rekaman Video.....	62
Gambar 4.13 Tampilan Program pada saat Mendeteksi Nilai HSV	63

DAFTAR LAMPIRAN

	hlm.
Lampiran 1 Proses Pengoperasian Alat	71
Lampiran 2 Dokumentasi Pengerjaan Alat	75
Lampiran 3 Spesifikasi Komponen	77
Lampiran 4 <i>Listing Program</i> pada <i>Raspberry Pi</i>	86
Lampiran 5 Biodata Penulis	96
Lampiran 6 Lembar Asistensi Skripsi Tugas Akhir	97
Lampiran 7 Catatan Revisi Skripsi Tugas Akhir.....	99



SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Fachturrahman

NIM : 444 17 017

menyatakan dengan sebenar-benarnya bahwa segala pernyataan dalam skripsi ini yang berjudul “*Drone* untuk Deteksi Hama dan Penyemprotan Pestisida pada Tanaman Padi” merupakan gagasan dan hasil karya saya sendiri dengan arahan komisi pembimbing dan belum pernah diajukan dalam bentuk apapun pada perguruan tinggi dan instansi mana pun.

Semua data dan informasi yang digunakan telah dinyatakan secara jelas dan dapat diperiksa kebenarannya. Sumber informasi yang berasal atau dikutip dari karya penulis lain telah disebutkan dalam naskah dan dicantumkan dalam skripsi ini.

Jika pernyataan saya tersebut di atas tidak benar, saya siap menanggung risiko yang ditetapkan oleh Politeknik Negeri Ujung Pandang.

Makassar, 7 September 2021



Fachturrahman
NIM 444 17 017

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Hutomo Febri Richardo Sumbung

NIM : 444 17 022

menyatakan dengan sebenar-benarnya bahwa segala pernyataan dalam skripsi ini yang berjudul “*Drone* untuk Deteksi Hama dan Penyemprotan Pestisida pada Tanaman Padi” merupakan gagasan dan hasil karya saya sendiri dengan arahan komisi pembimbing dan belum pernah diajukan dalam bentuk apapun pada perguruan tinggi dan instansi mana pun.

Semua data dan informasi yang digunakan telah dinyatakan secara jelas dan dapat diperiksa kebenarannya. Sumber informasi yang berasal atau dikutip dari karya penulis lain telah disebutkan dalam naskah dan dicantumkan dalam skripsi ini.

Jika pernyataan saya tersebut di atas tidak benar, saya siap menanggung risiko yang ditetapkan oleh Politeknik Negeri Ujung Pandang.

Makassar, 7 September 2021



Hutomo Febri Richardo Sumbung
NIM 444 17 022

DRONE UNTUK DETEKSI HAMA DAN PENYEMPROTAN PESTISIDA PADA TANAMAN PADI

RINGKASAN

Seiring bertambahnya jumlah penduduk, konsumsi beras juga akan meningkat. Sehingga produktifitas padi harus ditingkatkan. Namun biasanya produktifitas padi menurun karena adanya hama wereng coklat. Warna merupakan indikator yang paling mudah untuk mendeteksi hama wereng coklat, sehingga petani mengandalkan penglihatannya untuk melihat warna tanaman padi satu per satu. Akan tetapi, persawahan yang luas akan membuat petani membutuhkan waktu yang lama dan tenaga yang ekstra untuk melihat hama tersebut. Oleh karena itu perlu sebuah teknologi untuk mempercepat pendeteksian hama wereng coklat. Salah satu teknologi yang dibutuhkan adalah dengan mengimplementasikan teknologi *drone* dengan fitur *image processing*.

Penelitian ini bertujuan untuk membuat algoritma *image processing* pada *drone* untuk mendeteksi warna daun pada tanaman padi yang mengalami serangan hama wereng coklat dan membuat mekanisme penyemprotan insektisida pada tanaman padi menggunakan wahana *drone*. Sehubungan dengan itu, penelitian ini dimulai dengan perancangan sistem mekanik, sistem elektronik, perancangan program *image processing* dan program mekanisme penyemprotan. Setelah itu, dilakukan dengan pengujian terbang *drone*, pengujian mekanisme penyemprotan, pengambilan nilai citra hsv warna daun padi, dan pengujian algoritma *image processing*.

Perancangan, perakitan, dan pengujian komponen-komponen *drone* telah dilakukan. Semua komponen dapat berfungsi, baik aktuator maupun komunikasi data antara *radio controller* dan *flight controller* serta *raspberry pi 3* dan laptop. Pengujian terbang dengan mode otomatis, program deteksi warna dan mekanisme penyemprotan telah dilakukan. Hasil pengujian menunjukkan bahwa *drone* dapat bergerak secara otomatis ke *waypoint* dengan kecepatan dan *altitude* yang telah ditentukan serta mekanisme penyemprotan bekerja dengan baik ketika *drone* mendeteksi adanya warna coklat sesuai dengan inputan nilai citra hsv.

DRONE FOR PEST DETECTION AND PESTICIDE SPRAYING ON RICE PLANT

SUMMARY

As the population increases, rice consumption will also increase. So the productivity of rice must be increased. However, rice productivity usually decreases because of the brown planthopper. Color is the easiest indicator to detect brown planthopper pests, so farmers rely on their eyesight to see the color of rice plants one by one. However, large rice fields will make farmers need a long time and extra effort to see these pests. Therefore we need a technology to accelerate the detection of brown planthopper pests. One of the technologies needed is to implement drone technology with image processing features.

This study aims to create an image processing algorithm on a drone to detect leaf color on rice plants that are attacked by brown planthoppers and create a mechanism for spraying insecticides on rice plants using drones. Accordingly, this research begins with the design of mechanical systems, electronic systems, image processing program designs and spraying mechanism programs. After that, it was carried out by drone flight testing, testing the spraying mechanism, taking the hsv image value of rice leaf color, and testing the image processing algorithm.

The design, assembly and testing of drone components has been carried out. All components can function, both actuator and data communication between radio controller and flight controller as well as raspberry pi 3 and laptop. Flight tests with automatic mode, color detection program and spraying mechanism have been carried out. The test results show that the drone can move automatically to the waypoint with a predetermined speed and altitude and the spraying mechanism works well when the drone detects a brown color according to the input hsv image value..

BAB I PENDAHULUAN

1.1 Latar Belakang

Padi (*Oryza sativa*) adalah tanaman penghasil beras yang merupakan salah satu komoditas tanaman pangan utama di Indonesia yang dijadikan makanan pokok bagi masyarakat Indonesia bahkan bagi setengah penduduk dunia. Padi juga merupakan kontributor terbesar bagi perekonomian para petani di Indonesia.

Kebutuhan beras akan semakin meningkat seiring bertambahnya jumlah penduduk, dikarenakan beras merupakan hasil pertanian yang sangat berpengaruh bagi kehidupan sehari-hari masyarakat. Sehubungan dengan hal tersebut, produktivitas padi perlu ditingkatkan.

Upaya untuk menjaga stabilitas dan peningkatan produksi padi telah dilakukan oleh pemerintah Indonesia sejak dahulu. Namun terjadinya penurunan hasil padi sawah, baik kuantitas ataupun kualitas disebabkan oleh beberapa faktor, diantaranya iklim yang selalu berubah, ketersediaan air, kesuburan tanah, varietas, sistem pengolahan tanaman, perkembangan hama dan penyakit (Siregar, 2007).

Salah satu hama utama tanaman padi di Indonesia adalah wereng coklat. Wereng coklat merupakan hama penusuk dan penghisap, terutama menghisap getah floem, mengurangi klorofil dan kandungan protein daun, serta mengurangi laju fotosintesis (Watanabe dan H. Kitagawa, 2000). Serangan hama ini dapat mengakibatkan kerugian yang sangat besar bagi petani, serangan mencapai lebih dari 90 persen dan mengakibatkan gagal panen atau puso (Baehaki dan A. Kartohardjono, 2005).



Gambar 1.1 Tanaman Padi yang Mengalami Serangan Wereng Coklat
(Sumber: Ridlo, 2017)

Warna daun merupakan indikator paling mudah digunakan untuk mengetahui serangan hama wereng coklat pada padi, sehingga petani hanya mengandalkan penglihatannya saja dengan cara mengamati warna daun pada padi satu persatu. Akan tetapi, untuk mengamati persawahan yang luas, petani membutuhkan waktu yang lama dan memerlukan banyak tenaga.

Di sisi lain, perkembangan teknologi informasi dan komputer di zaman revolusi industri 4.0 yang sangat pesat ini, mempengaruhi cara pandang orang terhadap teknologi secara keseluruhan. Beberapa hal yang dulunya dilakukan secara manual dan membutuhkan waktu yang lama, dapat dilakukan dengan lebih cepat secara digital ataupun otomatis. Salah satu contohnya adalah robot terbang yang sering disebut *drone*.

Dulu, pemanfaatan *drone* dilakukan secara terbatas, hanya dalam bidang militer. Seiring dengan perkembangan zaman, *drone* sudah menjadi produk teknologi yang bisa digunakan dalam bidang apa saja, tidak terkecuali dalam bidang pertanian. Penggunaan teknologi *drone* dalam bidang pertanian bisa memberikan banyak manfaat. Salah satu manfaat teknologi *drone* dalam bidang pertanian adalah pemberian nutrisi pupuk atau pestisida. Dengan menggunakan *drone*

memungkinkan proses penyiraman pestisida pada tanaman berlangsung dengan lebih cepat, hemat air dan merata di waktu yang sama juga berguna untuk mengurangi biaya untuk ongkos penanaman mencapai 85 persen (Harfian dan Yakob Bunga Tumanan, 2020). Berdasarkan uraian di atas, maka penulis mengajukan penelitian dengan judul ***Drone untuk Deteksi Hama dan Penyemprotan Pestisida pada Tanaman Padi.***

1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang di atas, maka rumusan masalah dalam penelitian ini adalah:

1. Bagaimana membuat mekanisme penyemprotan insektisida pada tanaman padi menggunakan wahana *drone*?
2. Bagaimana membuat algoritma *image processing* pada *drone* untuk mendeteksi warna daun pada tanaman padi yang mengalami serangan hama wereng coklat?

1.3 Ruang Lingkup Penelitian

Adapun ruang lingkup penelitian sebagai berikut:

1. Hama yang akan dideteksi adalah hama wereng coklat pada tanaman padi.
2. Warna daun pada tanaman padi yang mengalami serangan hama wereng coklat adalah warna coklat.
3. Metode *image processing* yang digunakan untuk mendeteksi warna daun pada tanaman padi yang mengalami serangan hama wereng coklat adalah metode HSV.

1.4 Tujuan Penelitian

Adapun tujuan penelitian dibuat yaitu:

1. Membuat mekanisme penyemprotan insektisida pada tanaman padi menggunakan wahana *drone*.
2. Membuat algoritma *image processing* pada *drone* untuk mendeteksi warna daun pada tanaman padi yang mengalami serangan hama wereng coklat.

1.5 Manfaat Penelitian

Adapun manfaat yang dapat diperoleh dari penelitian ini antara lain:

1. Mengembangkan dan meningkatkan kemampuan *drone* dengan menambahkan kemampuan untuk mendeteksi warna daun pada tanaman padi yang mengalami serangan hama wereng coklat.
2. Memudahkan petani untuk mendeteksi tanaman padi yang mengalami serangan hama wereng coklat.
3. Meningkatkan pengetahuan dan kompetensi penulis di bidang Mekanika, khususnya dalam *image processing*.

BAB II TINJAUAN PUSTAKA

2.1 *Drone*

Drone adalah pesawat tanpa awak yang dikendalikan dari jarak jauh dengan menggunakan komputer atau *remote control*, yang bisa digunakan untuk membawa muatan baik senjata maupun muatan lainnya. (Carafano dan Andrew Gudgel, 2007).

2.1.1 Jenis – jenis *drone*

Berdasarkan baling-balingnya, terdapat dua jenis *drone*, yaitu:

1) *Fixed Wing Drone (Tunggal)*

Drone jenis ini berbentuk seperti pesawat terbang biasa/komersial dan digunakan untuk proses yang cepat, daya jangkau lebih cepat serta lebih luas, biasanya untuk pemetaan (*mapping*) atau konsepnya seperti *scanning*. *Drone* jenis *Fixed wing* memiliki energi lebih irit baterai karena *single* baling-baling.



Gambar 2.1 *Fixed Wing Drone*
(Sumber: Saxon Aerospace, 2019)

2) *Multirotor Drone (Multi)*

Multirotor adalah jenis *drone* yang memanfaatkan putaran baling-baling untuk terbang. *Multirotor* dibagi lagi menjadi dua yaitu *single-rotor*

dan *multi-rotor*. Tipe *single-rotor* berbentuk seperti helikopter menggunakan baling-baling tunggal, sedangkan multi-rotor menggunakan 3 sampai 8 baling-baling. Keuntungan dari *multi-copter* bisa terbang vertikal hingga 300 meter sehingga cocok untuk pemetaan infrastruktur, lahan pertanian dan wilayah hutan. *Multicopter* dapat terbang selama 40 menit dengan area cover 100-400 hektare. Sedang untuk jenis *fixed-wing*, meski bisa meliputi area yang jauh lebih luas dan terbang hingga 1,5 jam, *drone* ini tidak bisa terbang secara vertikal.



Gambar 2.2 *Multicopter Drone*
(Sumber: Point of Beginning, 2019)

2.1.2 Cara Kerja *Drone*

Fixed wing drone, memiliki bentuk dan bekerja dengan prinsip-prinsip yang sama dengan pesawat terbang. Motornya diletakkan horizontal, sehingga baling-balingnya dapat menggerakkan badan *drone* secara horizontal pula. Sayap dan badan *drone* dibuat aerodinamis, untuk mendapat daya angkat optimal pada saat *drone* bergerak horizontal.

Multicopter drone bekerja dengan prinsip-prinsip yang sama dengan helikopter. Motornya diletakkan vertikal, sehingga baling-balingnya dapat membuat daya angkat (*thrust*) secara vertikal pula. Selain motor yang diletakkan

secara vertikal, kecepatan putaran motor dan baling-baling akan sangat berpengaruh dengan bagaimana gerakan *drone* selama mengudara.

Untuk bisa mengangkat kerangka *drone*, dua pasang baling-baling berputar dengan arah yang berbeda. Sepasang baling-baling akan bergerak searah jarum jam, sementara sepasang baling-baling lainnya bergerak melawan arah jarum jam. Namun, semua bergerak dengan kecepatan yang sama. Perbedaan arah putaran baling-baling menghasilkan adanya gaya tarik dan gaya dorong yang melawan prinsip gaya gravitasi. Hal inilah yang menyebabkan *multicopter* dapat melayang di udara.



Gambar 2.3 Arah Putaran Baling-Baling pada *Drone*
(Sumber: Ikhsan, 2014)

2.2 Komponen-Komponen *Drone*

2.2.1 *Flight Controller*

Flight Controller merupakan otak dari sebuah *drone*. Fungsi dari *flight controller* ini yaitu untuk mengatur kecepatan motor, stabilisasi *drone* dan mempertahankan ketinggian. Selain itu *flight controller* menerima semua perintah atau inputan dari pilot.

Biasanya *flight controller* ini terdiri dari beberapa komponen yang dilengkapi dengan sensor untuk mengetahui arah dari *drone*. *Flight controller* juga

menerima perintah dari pengguna, dan mengontrol motor (dinamo) agar *drone* tetap terbang di udara.



Gambar 2.4 *Flight Controller*
(Sumber: Saputra, 2018)

2.2.2 *Brushless dc Motor (BLDC)*

Motor *Brushless Direct Current* (BLDC) adalah motor yang tidak menggunakan sikat atau *brush* untuk pergantian medan magnet (komutasi) tetapi dilakukan secara komutasi elektronik.

Perbedaan utama antara motor dc Magnet Permanen (DC-MP) dengan motor *brushless* dc adalah terletak pada pembangkitan medan magnet untuk menghasilkan gaya gerak. Jika pada motor DC-MP medan magnet yang dikontrol berada di rotor dan medan magnet tetap berada di stator. Sebaliknya, motor *brushless* menggunakan pembangkitan medan magnet stator untuk mengontrol gerakannya sedang medan magnet tetap berada di rotor. (Pitowarno, 2006).

Motor BLDC mempunyai banyak keuntungan dibandingkan dengan motor dc lainnya (Yulistiyanto, 2013), yaitu:

- 1) Kecepatan yang lebih baik untuk melawan karakteristik tenaga putaran.

- 2) Tanggapan dinamis tinggi.
- 3) Efisiensi tinggi.
- 4) Tahan lama atau usia pakai lebih lama.
- 5) Nyaris tanpa suara bila dioperasikan.
- 6) *Speed range* yang lebih luas.

Perbandingan tenaga putaran lebih besar dibanding dengan ukuran motor, dengan ukuran motor yang relatif kecil dapat menghasilkan torsi yang cukup besar.



Gambar 2.5 Motor *Brushless dc* (BLDC)
(Sumber: Alselectro, 2014)

2.2.3 *Radio Controller*

Radio controller adalah sebuah alat elektronik yang digunakan untuk mengoperasikan sebuah mesin dari jarak jauh. Pada umumnya, *radio controller* digunakan untuk memberikan perintah dari kejauhan kepada *drone*. *Radio controller* memerlukan alat pemancar atau *transmitter* dan penerima atau *receiver*.

Radio controller yang diperlukan untuk mengendalikan *drone* membutuhkan minimal 4 kanal atau *channels* untuk berkomunikasi dengan *drone*. Masing-masing channel dipakai untuk mengendalikan: *throttle* atau *lift*, *yaw* atau *rudder*, *pitch* atau *elevator* dan *roll* atau *aileron*.

Throttle atau *lift* digunakan untuk mengangkat atau menurunkan ketinggian *drone*. *Yaw* atau *rudder* untuk menoleh atau memalingkan muka *drone* ke kiri dan kanan. *Pitch* atau *elevator* untuk membuat *drone* maju atau mundur. Sedangkan *roll* atau *aileron* membuat *drone* bergeser ke kiri atau kanan. Semua dilakukan mengikuti perintah dari *radio controller*.



Gambar 2.6 Radio Controller
(Sumber: Martinez, 2019)

2.2.4 Baterai Lithium Polymer (Li-Po)

Baterai *Lithium Polimer* (Li-Po) merupakan baterai tercanggih dan paling maju dalam dunia baterai saat ini. Keunggulan utamanya adalah rasio *Power to Weight*-nya dan memungkinkan baterai dicetak sesuai dengan keinginan. Selain memiliki keunggulan, Li-Po juga memiliki kelemahan/kekurangan. Sifatnya yang sensitif dan mudah rusak apabila diperlakukan dengan tidak tepat. Memiliki bahaya risiko *explosive* jika *overcharge*, ditusuk atau bocor.



Gambar 2.7 Baterai Li-Po
(Sumber: Swellpro, 2021)

2.2.5 Frame Drone

Frame drone merupakan tempat untuk meletakkan komponen lain dari *drone*, *frame* adalah badan dari *drone*. Desain pada *drone* haruslah proporsional karena jika tidak *drone* akan terbang tidak stabil, akibat beban yang tidak seimbang. Selain itu tata letak komponen juga harus dipikirkan untuk membuat quadcopter lebih rapi dalam instal peralatan.



Gambar 2.8 *Frame Drone*
(Sumber: Robolink, 2021)

2.2.6 Electronic Speed Controller (ESC)

ESC (*Electronic Speed Controller*) adalah bagian elektronik yang mengatur kecepatan perputaran motor *brushless* dengan cara mengatur suplai arus yang disesuaikan dengan kebutuhan *drone*, sehingga energi baterai dapat digunakan seefisien mungkin, dan memperlama waktu penerbangan. ESC bekerja secara cepat untuk menghidupkan atau mematikan pulsa ke motor, sehingga respon kendali motor cepat.



Gambar 2.9 *Electronic Speed Control (ESC)*
(Sumber: MAD Components Store, 2021)

2.2.7 Baling-baling (*Propeller*)

Baling-baling (*propeller*) adalah alat untuk menjalankan pesawat terbang. Baling-baling ini memindahkan tenaga dengan mengkonversi gerakan rotasi menjadi daya dorong untuk menggerakkan sebuah kendaraan seperti pesawat terbang, untuk melalui suatu massa seperti udara, dengan memutar dua atau lebih bilah kembar dari sebuah poros utama. Sebuah *propeller* berperan sebagai sayap berputar, dan memproduksi gaya yang mengaplikasikan prinsip Bernoulli dan hukum gerak Newton, menghasilkan sebuah perbedaan tekanan antara permukaan depan dan belakang.



Gambar 2.10 Baling-baling (*Propeller*)
(Sumber: Swellpro, 2021)

2.2.8 *Raspberry Pi*

Raspberry Pi adalah sebuah komputer papan tunggal atau SBC (*Single-Board Computer*) berukuran kartu kredit. *Raspberry Pi* telah dilengkapi dengan semua fungsi layaknya sebuah komputer lengkap, menggunakan SoC (*System-on-a-chip*) ARM yang dikemas dan diintegrasikan diatas PCB. Perangkat ini menggunakan kartu SD untuk *booting* dan penyimpanan jangka panjang. (Yuwono, dkk, 2015).



Gambar 2.11 *Raspberry Pi 3B+*
(Sumber: Shidiq, 2018)

2.2.9 *Global Positioning System (GPS)*

GPS (*Global Positioning System*) adalah sistem satelit navigasi dan penentu posisi yang memberikan posisi dan kecepatan tiga dimensi dan informasi waktu, secara kontinyu di seluruh dunia tanpa tergantung kepada waktu dan cuaca (Abidin, dkk, 2009).

GPS adalah sebuah sistem atau proses untuk menentukan posisi berdasarkan 4 faktor yaitu: *latitude* (garis lintang), *longitude* (garis bujur), *altitude* (ketinggian) dan waktu. GPS juga dapat untuk menentukan variable-variabel turunan seperti: kecepatan, percepatan/akselerasi, arah laju, dan ukuran interval (jarak, selang waktu). (Firdaus, 2010)



Gambar 2.12 *Global Positioning System (GPS)*
(Sumber: Swellpro, 2021)

2.2.10 Router

Router adalah suatu perangkat keras pada jaringan komputer yang berfungsi untuk menghubungkan beberapa jaringan, baik itu jaringan yang sama maupun jaringan yang berbeda dari sisi teknologinya. Ada juga yang menjelaskan bahwa pengertian router adalah suatu hardware jaringan komputer yang berfungsi untuk mengirimkan paket data melalui jaringan atau internet dari satu perangkat komputer ke perangkat lainnya, dimana proses tersebut disebut dengan *routing*. (Prawiro, 2018).



Gambar 2.13 Router
(Sumber: Mogu, 2015)

2.2.11 IP Webcam

Aplikasi IP Webcam dirilis oleh Pavel Khlebovich dengan keunggulan mampu merekam suara dan gerakan. IP Webcam sendiri merupakan suatu aplikasi yang bertujuan sebagai kamera pengintai atau CCTV tanpa kabel yang bekerja untuk merekam segala aktivitas di sekitarnya. Ada beragam fitur yang bisa dimanfaatkan dengan mudah oleh para penggunanya. Misalnya, fitur pengatur resolusi video, mode *audio* dan tampilan pada layar android. Aplikasi ini dapat diunduh secara gratis di Play Store. (Diminimalis, 2019)



Gambar 2.14 Logo IP Webcam
(Sumber: Khlebovich, 2020)

2.3 *Image Processing*

2.3.1 Pengertian

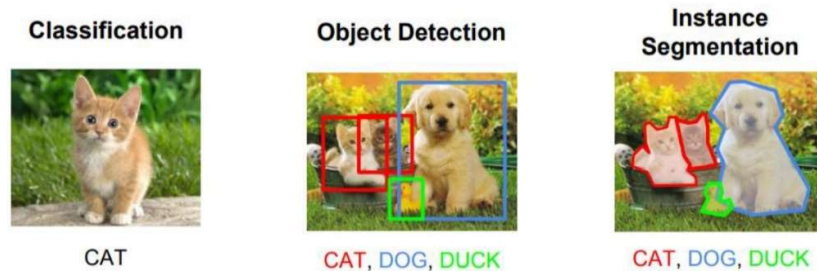
Pengolahan citra digital (*digital image processing*) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari *webcam*). Sedangkan digital disini mempunyai maksud bahwa pengolahan citra/gambar dilakukan secara digital menggunakan komputer.

Teknik-teknik pengolahan citra mentransformasikan citra ke citra yang lain. Jadi masukannya adalah citra dan keluarannya juga citra, namun citra keluaran atau hasil mempunyai kualitas lebih baik dari pada citra masukan.

Pengolahan citra bertujuan untuk :

- 1) Memperbaiki kualitas gambar, dilihat dari aspek *radiometric* dan aspek *geometric*. Aspek *radiometric* terdiri dari peningkatan kontras, restorasi citra, transformasi warna sedangkan aspek *geometric* terdiri dari rotasi, skala, translasi, transformasi geometric.
- 2) Melakukan proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung pada citra.

- 3) Melakukan pemilihan citra ciri (*feature images*) yang optimal untuk tujuan analisis.
- 4) Melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data.



Gambar 2.15 Contoh Penggunaan *Image Processing* untuk Deteksi Objek (Sumber: Patel, 2020)

2.3.2 *Color Filtering*

Color filtering adalah suatu teknik pengolahan citra yang dipakai untuk memanipulasi suatu citra berdasarkan warna spesifik. Cara kerjanya adalah dengan membandingkan komponen warna setiap piksel citra dengan warna spesifik.

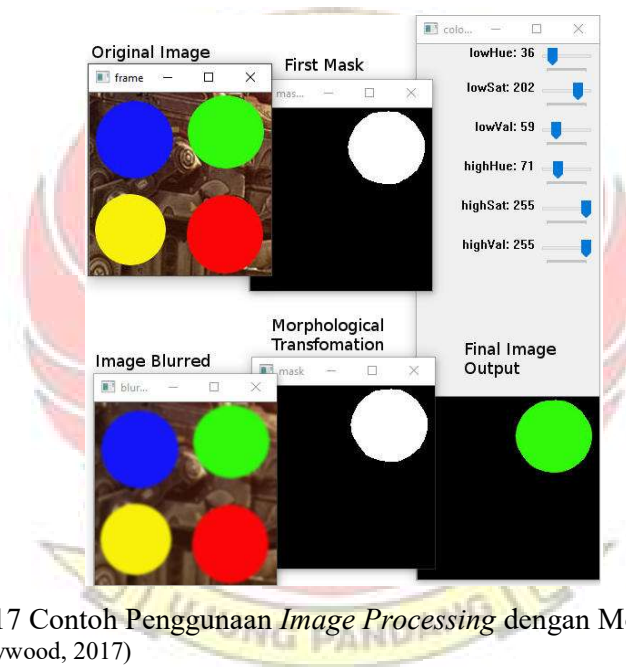
Warna yang digunakan dalam *Color Filtering* dapat direpresentasikan dalam berbagai ruang warna, antara lain RGB (*Red, Green, Blue*), HSV (*Hue, Saturation, Value*), YCbCr, dan sebagainya (Dhiemas, 2011).



Gambar 2.16 Contoh Penggunaan *Color Filtering* (Sumber: Keizer, 2008)

2.3.3 Model Warna HSV (*Hue Saturation Value*)

Model warna HSV mempunyai 3 variabel berhubungan antara *Hue*, *Saturation* dan *Value*. *Color model* atau ruang warna HSV berdasarkan pada koordinat polar. HSV adalah versi kubus RGB dengan transformasi non-linear (Deswal dan Neetu Sharma, 2014). *Hue* mewakili jenis warna, *Saturation* mewakili bagaimana jenuh atau pudarnya warna tersebut, dan *Value* mewakili tingkat penerangan. (Sholahuddin, 2017)



Gambar 2.17 Contoh Penggunaan *Image Processing* dengan Model Warna HSV (Sumber: Heywood, 2017)

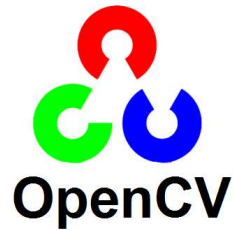
2.4 OpenCV

OpenCV adalah sebuah *library* yang berisi fungsi-fungsi pemrograman untuk teknologi *computer vision* secara *real time*. OpenCV bersifat *open source*, bebas digunakan untuk hal-hal yang bersifat akademis dan komersial. Di dalamnya terdapat *interface* untuk C++, C, Python, dan nantinya Java yang dapat berjalan pada Windows, Linux, Android, dan Mac. Terdapat lebih dari 2500 algoritma dalam OpenCV, digunakan di seluruh dunia, telah lebih dari 2.5 juta kali diunduh,

dan digunakan lebih dari 40 ribu orang. Penggunaannya antara lain pada seni interaktif, inspeksi tambang, menampilkan peta di web melalui teknologi robotik (Mostafa dan Quadrat-E-Alahy Ratul, 2008).

Fitur-fitur yang terdapat pada OpenCV antara lain: (Bradski dan Adrian Kaehler, 2008)

- 1) Manipulasi data *image* (alokasi, rilis, duplikasi, pengaturan, konversi)
- 2) Image dan I/O video (masukan berbasis *file* dan kamera, keluaran *image/video file*)
- 3) Manipulasi matriks dan vektor serta aljabar linear (produk, solusi, eigenvalues, SVD)
- 4) Beragam struktur data dinamis (daftar, baris, grafik)
- 5) Dasar pengolahan citra (filter, deteksi tepi, deteksi sudut, pengambilan sampel dan interpolasi, konversi warna, operasi morfologi, histogram)
- 6) Analisis struktur (komponen yang berhubungan, pengolahan kontur, transformasi jarak, variasi momen, transformasi Hough, perkiraan polygonal, menyesuaikan garis, delaunay triangulation)
- 7) Kalibrasi kamera (menemukan dan menelusuri pola kalibrasi, kalibrasi, dasar estimasi matriks, estimasi homografi, korespondensi stereo)
- 8) Analisis gerakan (*optical flow*, segmentasi gerakan, penelusuran)
- 9) Pengenalan objek (metode eigen, HMM)
- 10) Dasar *Graphical User Interface* atau GUI (menampilkan *image/video*, penanganan mouse dan keyboard, *scroll-bars*)
- 11) Pelabelan *image* (garis, poligon, gambar teks).



Gambar 2.18 Logo OpenCV
(Sumber: Setiawan, 2017)

2.5 *Drone* pada Aplikasi Pertanian

Drone untuk bidang pertanian mulai diciptakan untuk membantu melakukan perawatan terhadap lahan pertanian yang cukup luas. Sejak tahun 2013-2015, investasi dalam pengembangan *drone* untuk pertanian terus mengalami peningkatan.

Teknologi *drone* sendiri dapat memberikan banyak keuntungan serta manfaat bagi para petani. Beberapa diantaranya yaitu: memberikan gambaran tanaman mana saja yang sudah siap dibudidayakan, tanaman mana saja yang harus diberi pupuk, tanaman mana saja yang perlu diberikan perawatan khusus, antisipasi penyakit pada tanaman hingga mencegah kegagalan-kegagalan sebelum panen misalnya serangan OPT (Organisasi Pengganggu Tanaman) yang mengganggu pertumbuhan dari tanaman.

Tidak hanya itu, teknologi *drone* untuk pertanian juga mampu untuk menyemprotkan obat dan menebar pupuk ke tanaman yang terserang penyakit atau hama. Tentunya, dengan bantuan *drone* bagi petani akan membantu hasil produksi petani semakin meningkat dan efisiensi tenaga dan waktu bagi para petani.



Gambar 2.19 Contoh Penggunaan *Drone* pada Bidang Pertanian
(Sumber: Humas Fasilkom, 2020)

2.6 Perhitungan Parameter-Parameter pada *Drone*

2.6.1 Perhitungan Gaya Angkat *Drone* (*Thrust*)

Gaya angkat diperlukan oleh sebuah *quadcopter* untuk terbang (*hover*). Gaya sebesar F (*lift*) pada keempat motor diperlukan oleh *quadcopter* untuk dapat mengudara. Gaya yang dibutuhkan untuk menaikkan tinggi (*climb*) harus lebih besar dari massa dikali gravitasi dan gaya yang dibutuhkan untuk membuat *quadcopter* turun (*decline*) harus lebih kecil dari massa dikali gravitasi. Persamaan yang digunakan adalah (Khumairowati dan Bekti Yulianti, 2020):

$$F = (m \times g) / 4 \text{ (terbang)}$$

$$F > (m \times g) / 4 \text{ (bergerak naik)}$$

$$F < (m \times g) / 4 \text{ (bergerak turun)}$$

Dimana :

$$F = \text{gaya angkat pesawat oleh setiap motor (lift)} \left(\text{kg} \frac{\text{m}}{\text{s}^2} \right)$$

$$m = \text{massa total } \textit{quadcopter} \text{ (kg)}$$

$$g = \text{gravitasi bumi } (9,8 \text{ m/s}^2)$$

2.6.2 Estimasi Waktu Terbang *Drone*

Durasi waktu terbang yang bisa dicapai oleh *quadcopter* dipengaruhi oleh kapasitas baterai. Dimana dari besarnya kapasitas baterai tersebut bisa diketahui konsumsi arus tiap motor sehingga bisa dihitung durasi waktu terbang *quadcopter*. Durasi waktu terbang itu bisa dipakai oleh *quadcopter* untuk *take off*, *hover* dan *landing* (Khumairowati dan Bekti Yulianti, 2020).

Waktu terbang maksimum dan efektif untuk sebuah UAV dapat ditentukan dengan langkah-langkah berikut (Price, 2016):

- 1) Tentukan arus maksimum yang dapat disimpan atau disuplai baterai. Nilai ini biasanya diberikan dalam satuan mAh dibagian samping baterai. Jika yang digunakan dua atau lebih baterai yang dirangkai secara paralel, tambahkan nilai arus masing-masing baterai untuk mendapatkan arus total. Jika dirangkai secara seri, arus baterai terkecil yang digunakan.
- 2) Ubah satuan *miliAmpere per hour* (mAh) menjadi *Ampere per minute*, dengan membaginya dengan 1000 dan dikalikan dengan 60.
- 3) Tentukan berapa banyak arus yang digunakan UAV saat terbang dengan peralatan lengkap (seperti kamera, peralatan video, dan lain-lain). Nilai ini dapat ditentukan dengan mencatat aliran arus sesaat yang dipasang di *Ground Control Station* saat terbang. Jika pembacaan *Ground Control Station* tidak tersedia, perangkat pemantau arus dapat dipasang di UAV, atau *drone* dapat ditambatkan ke *ground* dengan memasang multimeter (catatan: nilai arus dengan *ground effect* mungkin sedikit berbeda dari nilai aktual arus saat terbang untuk UAV jenis *multicopter*). Catat konsumsi arus sesaat dalam Ampere (A).

- 4) Bagi hasil dari Langkah 2 dengan hasil Langkah 3. Hasil dari pembagian tersebut adalah waktu terbang maksimum UAV sebelum voltase baterai habis.
- 5) Waktu terbang efektif biasanya 40 persen hingga 80 persen dari waktu terbang UAV yang ditentukan di Langkah 4 untuk jenis *fixed-wing*, dan 40 hingga 70 persen dari nilai tersebut untuk jenis *multicopter* (catatan: UAV akan menggunakan 10 hingga 30 persen lebih banyak arus pada akhir penerbangan karena voltase baterai semakin rendah).

Dari langkah-langkah di atas dapat diketahui waktu terbang maksimum dan efektif untuk sebuah UAV dengan persamaan:

$$\text{waktu terbang}_{max} = \frac{\text{kapasitas baterai}}{\text{arus yang digunakan UAV saat terbang}}$$

$$\text{waktu terbang}_{eff} = \text{waktu terbang}_{max} \times \text{faktor keamanan}$$

Dimana :

- faktor keamanan = 0.4 – 0.8 (untuk jenis *fixed wing*)
= 0.4 – 0.7 (untuk jenis *multicopter*)
- waktu terbang_{max} (menit)
- waktu terbang_{eff} (menit)
- kapasitas baterai (*Ampere per minute*)
- arus yang digunakan UAV saat terbang (Ampere)

2.7 Penelitian Terkait *Drone* yang Diaplikasikan pada Bidang Pertanian

2.7.1 Penggunaan *Drone* dalam Mengaplikasikan Pestisida di Daerah Sungai Besar, Malaysia (Khoirunisa dan Fitrianingrum Kurniawati, 2019)

Penelitian ini bertujuan untuk membuat *quadcopter* atau *drone* yang dapat membantu petani dalam penyemprotan pestisida ke tanaman sesuai dengan yang

diinginkan dan membuat masa pengerjaannya bisa lebih cepat dua sampai dua puluh kali lipat dan juga menghindari kecelakaan akibat dampak negatif pestisida saat terjadi kontak secara langsung dalam artian terkena kulit secara, terhirup atau pun mengenai mata manusia karena pestisida mengandung bahan kimia yang berbahaya.

Adapun hasil pengujian dari penelitian yang menggunakan *spraying drone* jenis DJI ini yaitu *drone* mampu menyemprot 0.6 ha lahan dengan kapasitas 10 liter pestisida atau pupuk. Satu petak (1.2 ha) lahan sawah perlu disemprot pestisida dengan dua kali pengisian ulang tangki. Kapasitas baterai *drone* dapat diisi ulang/*charge* sebanyak 3 kali. Satu kali pengisian baterai dapat digunakan untuk pemakaian 20 kali penyemprotan. Dalam satu hari, lahan pertanian yang dapat diaplikasikan pestisida atau pupuk oleh *drone* seluas 5 ha. Untuk 1 ha lahan diperlukan waktu sekitar 10 menit dalam melakukan penyemprotan menggunakan *drone*. Sehingga, waktu yang diperlukan untuk menyemprot 5 ha lahan hanya berkisar 50 menit. Berbeda dengan penyemprotan konvensional yang akan memakan waktu 20 jam per 1 ha lahan.



Gambar 2.20 Penggunaan *Drone* dalam Mengaplikasikan Pestisida di Daerah Sungai Besar, Malaysia
(Sumber: Khoirunisa dan Fitrianingrum Kurniawati, 2019)

2.7.2 Pengembangan Sistem Penyemprotan pada Platform Pesawat Tanpa Awak Berbasis *Quadcopter* untuk Membantu Petani Mengurangi Biaya Pertanian dalam Mendorong Konsep Pertanian Pintar atau *Smart Farming* (Santoso dan Kris Hariyanto, 2017)

Penelitian ini bertujuan untuk mengembangkan teknologi pesawat tanpa awak untuk kepentingan pertanian dalam mewujudkan ketahanan pangan Indonesia. Salah satu permasalahan utama pertanian di Indonesia adalah tingginya biaya pertanian khususnya penggunaan bahan kimia, pupuk hingga tenaga kerja. Selama ini petani menghabiskan sumber daya seperti pupuk untuk ke semua tanaman tanpa porsi yang dibutuhkan, pemakaian tenaga kerja yang relatif besar untuk proses penyemprotan hama area yang luas. Sehingga harus diupayakan rekayasa yang mampu melakukan kegiatan pemupukan dan penyemprotan hama secara cepat, efisien dan akurat. Metode yang akan digunakan untuk permasalahan tersebut adalah membuat sebuah prototipe sistem penyemprotan hama dengan menggunakan media cair pada platform *Unmanned Aerial Vehicle* (UAV) dengan basis *quadcopter*.

Berdasarkan hasil penelitian yang sudah dilakukan platform UAV berbasis *quadcopter* mampu melaksanakan misi semua tugasnya yaitu penyemprotan dengan cairan pembasmi hama. Adapun dengan menggunakan fluida cair dengan massa 0.5 kg tersebut mampu melakukan penyemprotan dengan luasan area 2 meter persegi selama waktu 5 menit pada *altitude* 10 cm di atas tanaman padi, sementara *endurance* maksimum yang dimiliki baterai adalah 10 menit.



Gambar 2.21 Pengembangan Sistem Penyemprotan pada Platform Pesawat Tanpa Awak Berbasis *Quadcopter* untuk Membantu Petani Mengurangi Biaya Pertanian dalam Mendorong Konsep Pertanian Pintar atau *Smart Farming*
(Sumber: Santoso dan Kris Hariyanto, 2017)

2.7.3 Rancang Bangun Sistem Penyemprotan Pestisida dan Pupuk Pada Tanaman Padi Menggunakan *Mikrokontroler* (Arnizam, 2018)

Penelitian ini dibuat untuk memudahkan para petani di lahan penyemprotan pestisida dan pupuk pada tanaman padi. Alat ini memiliki sistem penyemprotan yang terkomputerisasi dengan menggunakan pengurutan dalam menyemprotkan pada tanaman, dimana petani hanya mengotrol alat ini di pematang sawah dan juga alat ini menggunakan sistem *drone*, dimana quadcopter bergerak atau terbang di atas tanaman padi dengan remote kontrol.

Berdasarkan hasil pengujian yang dilakukan koneksi *quadcopter* dengan *remote control* dianggap berhasil. Selain itu hasil pengujian alat penyemprot akan berposisi *ready* saat *quadcopter* melakukan proses penerbangan, apabila posisi *quadcopter* sudah mencapai 5 detik terbang *delay* dalam program akan berkerja dan alat penyemprotan saat itupun melakukan eksekusi. Sehingga alat dapat menjalankan misinya yaitu terbang di udara dengan menggunakan *remote control* kemudian akan menyemprot di area yang akan disemprot.



Gambar 2.22 Rancang Bangun Sistem Penyemprotan Pestisida dan Pupuk pada Tanaman Padi Menggunakan Mikrokontroler
(Sumber: Arnizam, 2018)

2.7.4 Prototipe Tabung Semprot Pupuk Cair Berbasis Wahana *Quadcopter* bagi Lahan Tanaman Padi (Akbar dan Anton Yudhana, 2016)

Penelitian ini memberikan solusi terbaru dengan penerapan teknologi modern berbasis wahana pesawat UAV jenis *multirotor (quadcopter)*. Wahana ini membawa alat tabung semprot berisi pupuk cair yang dapat menyebarkan pupuk melalui udara tanpa menyentuh area persawahan. Sistem yang saling terintegrasi antara alat tabung semprot dan wahana *multirotor (quadcopter)* dapat bekerja secara *autonomous* yang dikontrol dari jarak jauh. Perbandingan waktu dan debit cairan pupuk menjadi tolak ukur hasil seberapa efektifitas perangkat ini bekerja pada setiap area lahan persawahan.

Berdasarkan hasil pengujian yang dilakukan, pada dua pengujian luaran dari selang semprot yang dihasilkan menunjukkan bahwa dengan menggunakan dua buah selang semprot, estimasi waktu dan jumlah debit cairan pada proses penyebaran pupuk berbanding lebih cepat dan lebih banyak dari pada dengan pengujian menggunakan luaran satu buah selang semprot. Pengujian yang dilakukan dengan dua kondisi penerbangan yakni stabil dan tidak stabil (goncangan) juga menunjukkan luaran berbeda yang disebabkan pengaruh faktor

alam seperti halnya kondisi angin yang tidak beraturan. Penelitian selanjutnya akan melakukan pengujian dengan menambah volume alat tabung semprot dan jumlah luaran selang semprot.



Gambar 2.23 Tabung Semprot Pupuk Cair Berbasis Wahana *Quadcopter* bagi Lahan Tanaman Padi
(Sumber: Akbar dan Anton Yudhana, 2016)

2.7.5 Aplikasi *Drone* Wawasan Tani untuk Pertanian di Simpang Lima, Sungai Besar, Selangor (Ikhwana dan Dwi Retno Hapsari, 2019)

Pengabdian masyarakat ini bertujuan untuk mengetahui pengaplikasian *drone* di bidang pertanian dan pengambilan keputusan penerapan *Drone* Wawasan Tani. Metode pengabdian ini menggunakan data kualitatif berupa data primer dari wawancara dan pengamatan lapangan serta sekunder dari literatur jurnal dan buku. Berdasarkan data, *drone* Wawasan Tani telah berhasil diterapkan oleh 70% masyarakat di kawasan ini. Sebagian masyarakat belum mengetahui inovasi ini karena akses informasi belum mencapai seluruh masyarakat. Inovasi ini akan terus diperluas sebagai bentuk peran koperasi dalam mengayomi masyarakat.

Dari hasil penyuluhan yang dilakukan *Drone* Wawasan Tani diterima baik oleh masyarakat karena mendukung efisiensi pertanian yang ada. *Drone* Wawasan Tani saat ini dikembangkan oleh enam pilot. *Drone* ini mampu menyemprot 0.6

hektare dengan kapasitas tangki sepuluh liter pupuk dan dalam penyemprotan per petak (1.2 hektare) membutuhkan pengisian ulang karena kapasitas maksimal tabung *drone* sepuluh liter.

Penggunaan baterai pada *drone* dapat diisi ulang/charge maksimal sebanyak tiga kali. Sekali pengisian dapat digunakan dua puluh kali penyemprotan. Satu hari *drone* biasa dipakai menyemprot seluas lima hektare. Penyemprotan ini dalam sekali musim dapat dilakukan lima kali penyemprotan atau dua minggu sekali. *Drone* ini anti air dan anti korosi, inilah yang membuat *drone* efisien lebih dari empat puluh kali lipat dari pada penyemprot secara manual. Setiap jamnya, *drone* dapat menjangkau lahan seluas empat hektare atau kira-kira sepuluh menit per hektare jika tanpa ada gangguan seperti angin dengan kecepatan maksimum delapan meter per jam.

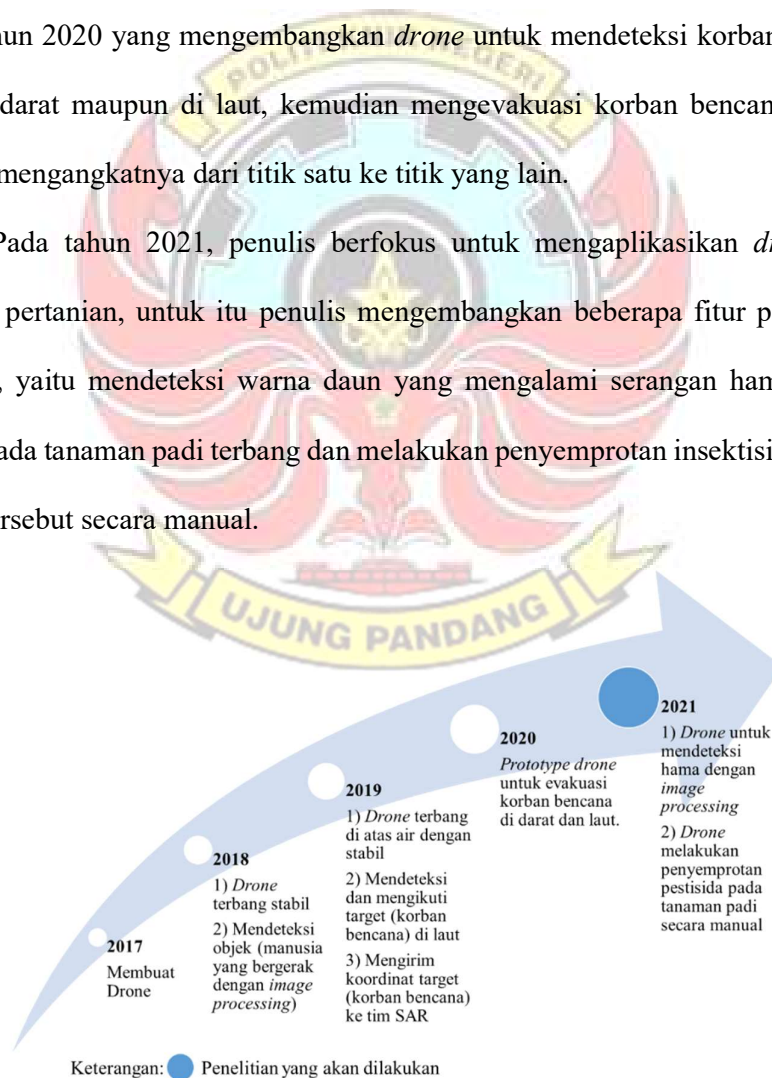


Gambar 2.24 Aplikasi *Drone* Wawasan Tani untuk Pertanian di Simpang Lima, Sungai Besar, Selangor
(Sumber: Ikhwana dan Dwi Retno Hapsari, 2019)

2.8 Road Map Penelitian Drone

Drone ini adalah pengembangan dari hasil penelitian-penelitian sebelumnya, dimana pada tahun 2017 *drone* ini dibuat oleh Steven Arif dkk. Dilanjutkan pada tahun 2018 untuk menstabilkan terbang *drone* dan menambahkan fitur deteksi objek, oleh Ivan Nugraha dkk. Dilanjutkan pada tahun 2019 yang memfokuskan *drone* pada operasi SAR di laut yang mendeteksi korban dan mengirim lokasi target ke tim SAR (Hasan dan Adam Palilu, 2020). Dilanjutkan pada tahun 2020 yang mengembangkan *drone* untuk mendeteksi korban bencana, baik di darat maupun di laut, kemudian mengevakuasi korban bencana tersebut dengan mengangkatnya dari titik satu ke titik yang lain.

Pada tahun 2021, penulis berfokus untuk mengaplikasikan *drone* pada aplikasi pertanian, untuk itu penulis mengembangkan beberapa fitur pada *drone* tersebut, yaitu mendeteksi warna daun yang mengalami serangan hama wereng coklat pada tanaman padi terbang dan melakukan penyemprotan insektisida dengan *drone* tersebut secara manual.



Gambar 2.25 Road Map Penelitian Drone PNUP

BAB III METODE PENELITIAN

3.1 Tempat dan Waktu Penelitian

Lokasi penelitian akan dilakukan di kampus Politeknik Negeri Ujung Pandang dan di areal persawahan di Kecamatan Bontomarannu, Kabupaten Gowa. Adapun waktu pelaksanaan penelitian dimulai dari bulan Januari 2021 sampai dengan September 2021.

3.2 Alat dan Bahan Penelitian

Untuk mendukung pelaksanaan dalam penelitian ini, diperlukan beberapa alat, bahan dan *software* untuk membuat tugas akhir sesuai dengan tujuan yang diinginkan. Adapun alat dan bahan yang akan digunakan dalam pelaksanaan penelitian ditunjukkan pada Tabel 3.1.

Tabel 3.1 Alat dan Bahan Penelitian

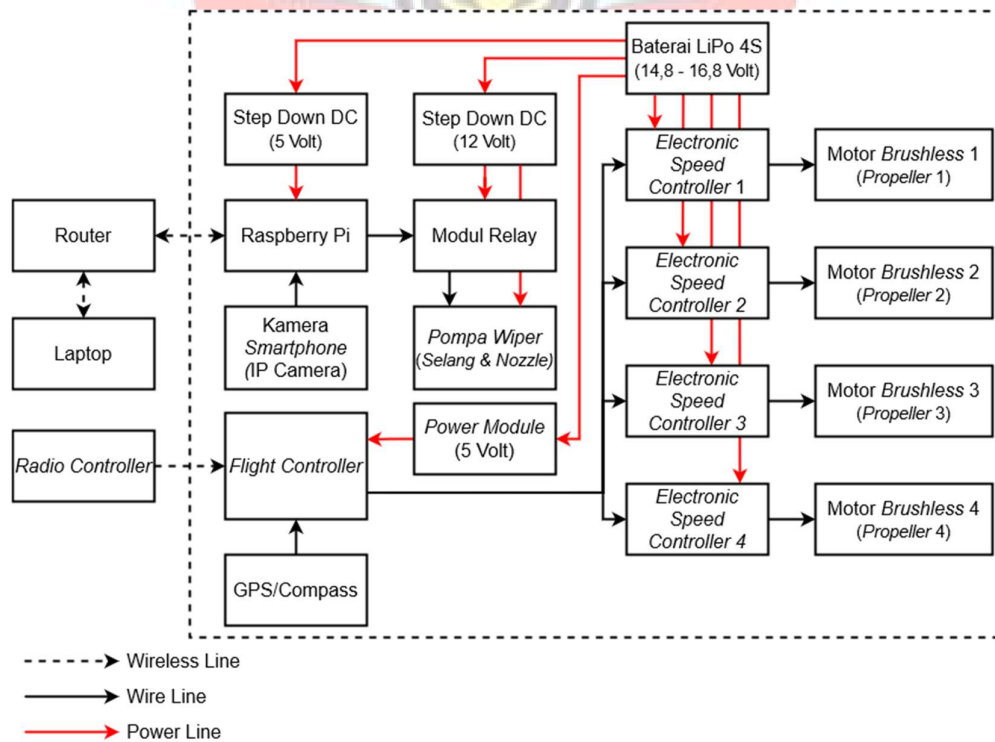
No	Alat	Bahan
1	Komputer/Laptop	<i>Flight Control</i>
2	Penggaris	Motor BLDC
3	Solder	Baterai LiPo
4	Tang potong	<i>Drone frame</i>
5	Tang jepit	ESC
6	Gunting	<i>Propeller</i>
7	Obeng (+) dan (-)	Kamera <i>Smartphone</i>
8	Palu	<i>Raspberry Pi</i>
9	Multimeter	Pompa Wiper
10	<i>Cutter</i>	Selang
11	Penghisap timah	Kabel tis
12	Timbangan	Timah
13	Meteran	Lem Araldite

Adapun beberapa perangkat lunak (*software*) yang akan digunakan dalam penelitian ini, yaitu:

- 1) Windows 10
- 2) Raspbian Jessie 4.9
- 3) OpenCV
- 4) Remote Desktop Connection

3.3 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 3.1



Gambar 3.1 Diagram Blok Sistem

Sistem ini dimulai dari laptop yang berada di *Ground Station* dan berfungsi untuk menjalankan program python yang tersimpan di *raspberry pi*. Laptop dan *raspberry pi* terhubung ke jaringan LAN (*Local Area Network*) dengan menggunakan router sebagai penghubungnya. Dalam jaringan LAN tersebut, laptop dapat melakukan *remote desktop* tanpa harus terkoneksi ke internet, sehingga laptop dapat mengontrol *raspberry pi* dan menjalankan program yang tersimpan di dalamnya.

Selanjutnya dilakukan penambatan/*tethering* USB dari *smartphone* ke *raspberry pi* menggunakan kabel *micro* USB. Hal ini dilakukan agar *smartphone* dan *raspberry pi* terhubung dalam jaringan yang sama. Kemudian pada *smartphone*, dijalankan aplikasi IP Webcam, dimana aplikasi ini akan menjadikan kamera *smartphone* dapat diakses melalui IP.

Setelah menjalankan aplikasi IP Webcam, selanjutnya menjalankan program *image processing* yang tersimpan dalam *raspberry pi*, sehingga kamera *smartphone* akan berfungsi sebagai media penangkap gambar bergerak (*webcam*) secara *realtime*. Kemudian gambar bergerak tersebut akan dijadikan *input* untuk *image processing* oleh *raspberry pi* dengan *library* OpenCV, sehingga dapat diketahui tanaman padi yang mengalami serangan hama.

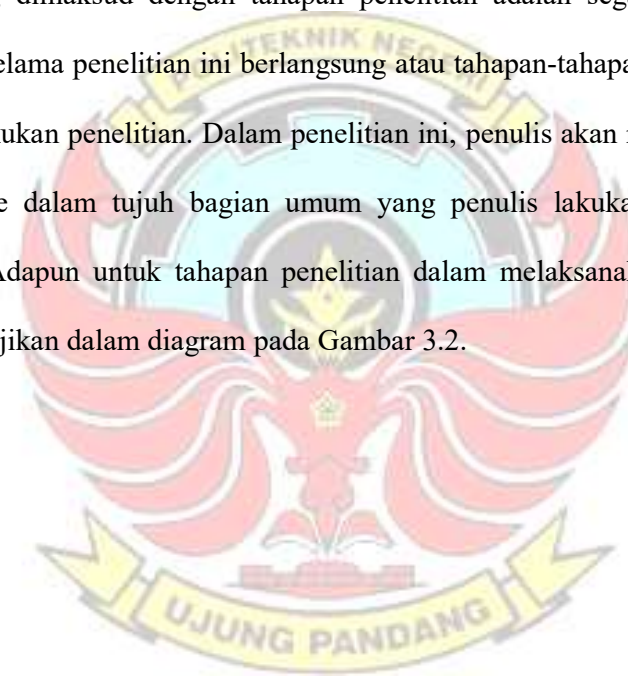
Setelah warna daun padi yang mengalami serangan hama terdeteksi maka modul relay akan mengaktifkan pompa wiper sehingga terjadi penyemprotan cairan ke tanaman padi tersebut.

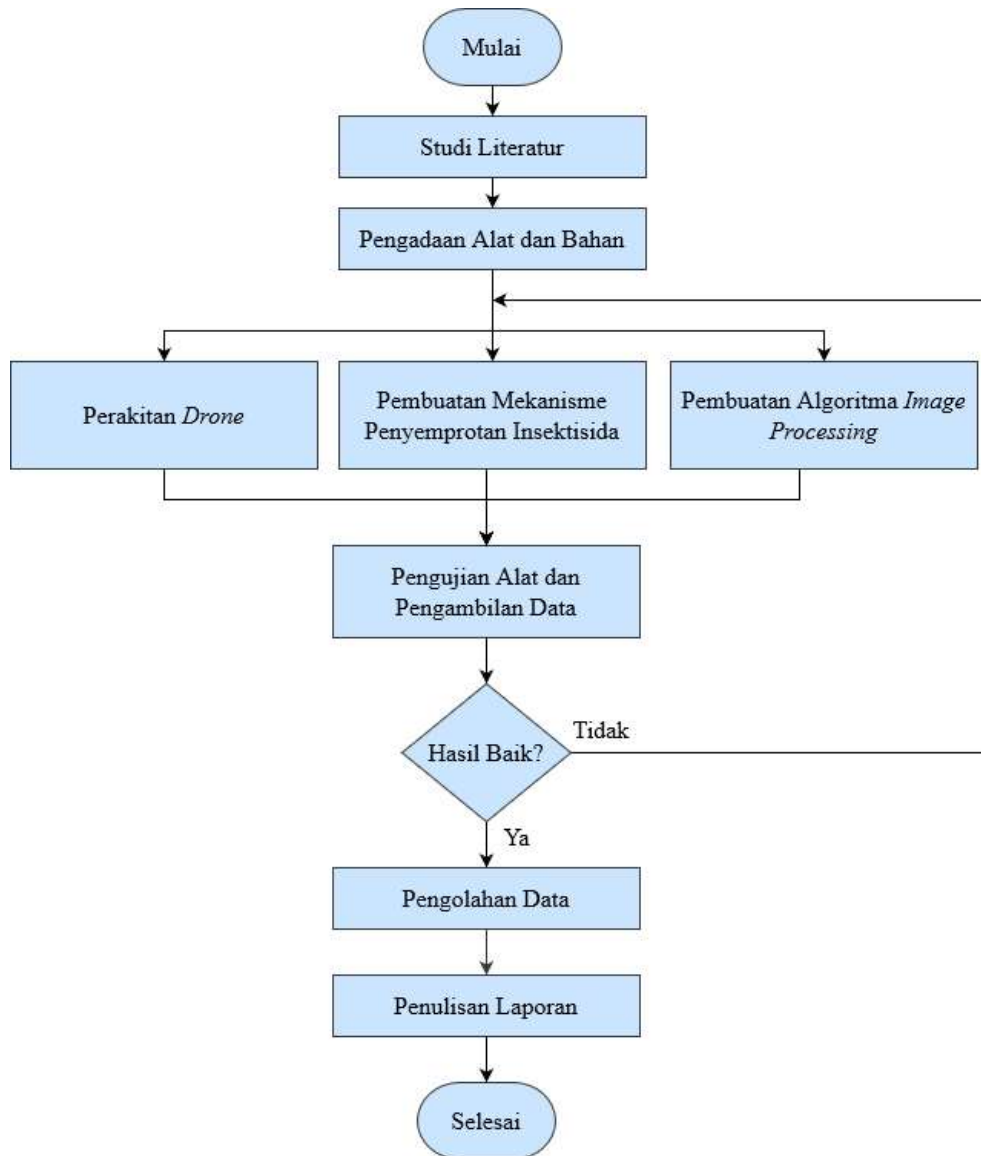
Adapun *drone* dikendalikan menggunakan *radio controller*. *Radio controller* akan mengirimkan *input* berupa frekuensi sinyal radio. Kemudian sinyal

radio tersebut akan diproses oleh *flight controller* menjadi *output* berupa sinyal kecepatan PWM (*Pulse With Modulation*). Sinyal PWM kemudian dikirimkan ke komponen ESC sebagai input, komponen ESC tersebut berfungsi untuk mengatur kecepatan putaran motor *drone* dengan memberikan *input* tegangan sesuai dengan sinyal PWM yang diterima dari *flight controller*.

3.4 Tahapan Penelitian

Yang dimaksud dengan tahapan penelitian adalah segala sesuatu yang dikerjakan selama penelitian ini berlangsung atau tahapan-tahapan yang ditempuh dalam melakukan penelitian. Dalam penelitian ini, penulis akan membagi tahapan penelitian ke dalam tujuh bagian umum yang penulis lakukan selama proses penelitian. Adapun untuk tahapan penelitian dalam melaksanakan Tugas Akhir ini telah disajikan dalam diagram pada Gambar 3.2.





Gambar 3.2 Diagram Alir Penelitian

3.4.1 Studi Literatur

Dalam perancangan alat ini, langkah awal yang dilakukan adalah studi literatur, yang dimaksudkan untuk mencari data serta informasi berupa skripsi, *paper*, jurnal dan buku, dimana informasi tersebut harus relevan dengan alat yang akan dibuat.

Setelah itu dilakukan pembuatan suatu rancangan penelitian dalam bentuk proposal, berisi tentang kerangka dasar yang menjadi acuan bagi penulis dalam melaksanakan penelitian.

3.4.2 Perancangan *Drone*

Pada tahap ini, *drone* akan dirakit menggunakan komponen-komponen yang telah diperoleh. Perancangan *drone* terbagi atas dua, yaitu perancangan sistem mekanik dan perancangan sistem elektronik.

1) Perancangan Sistem Mekanik *Drone*

Perancangan sistem mekanik *drone* meliputi pemasangan *frame drone*, peredam anti getaran *flight controller*,udukan *raspberry pi*, dan *holder smartphone*.

a) *Frame Drone*

Frame drone terdiri dari lengan *drone*, *landing gear*, dan *board* bagian atas dan bawah. *Board* bagian atas berfungsi sebagai penghubung antar lengan *drone*, sedangkan *board* bagian bawah berfungsi sebagai *power distribution*.

b) Peredam anti getaran *flight controller*

Peredam anti getaran *flight controller* terletak di bagian atas *frame drone*, berfungsi sebagaiudukan *flight controller* dan juga mengurangi tekanan ketika menghubungkan komponen ke *flight controller* agar tidak terjadi kerusakan.

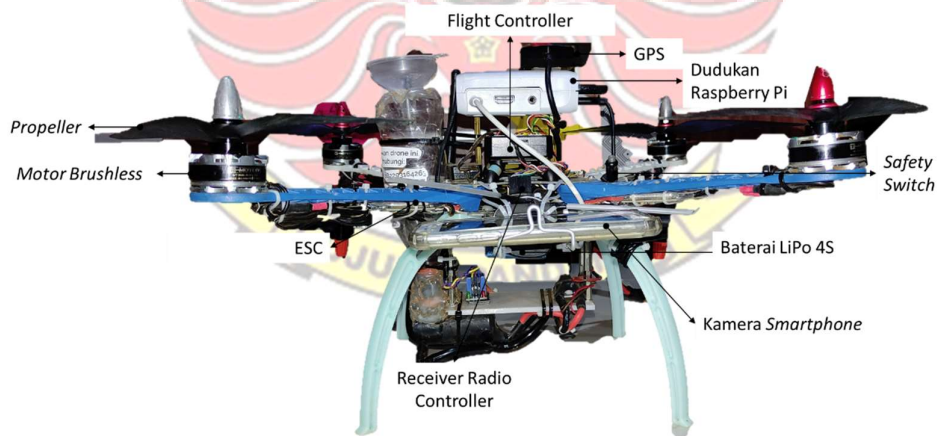
c) Dudukan *raspberry pi*

Dudukan *raspberry pi* terletak di atas *flight controller*, dimana dudukan ini memiliki dua kaki dengan panjang masing-masing 50mm yang dihubungkan langsung ke *board* bagian atas *frame drone*. Dudukan ini berfungsi sebagai *cover raspberry pi*.

d) *Holder smartphone*

Holder smartphone terletak di bagian depan *frame drone*, berfungsi sebagai dudukan atau pegangan *smartphone*, agar *smartphone* tidak gampang jatuh pada saat *drone* diterbangkan.

Setelah melakukan perancangan maka penulis mendapatkan hasil rancangan sistem mekanik *drone* yang tampak depannya ditunjukkan pada Gambar 3.3.



Gambar 3.3 Tampak Depan dari Hasil Rancangan Sistem Mekanik *Drone*

Adapun tampak atas dari hasil rancangan sistem mekanik *drone* ditunjukkan pada Gambar 3.4.



Gambar 3.4 Tampak Atas dari Hasil Rancangan Sistem Mekanik *Drone*

2) Perancangan Sistem Elektronik *Drone*

Perancangan sistem elektronik digunakan Pixhawk sebagai *flight controller drone*, *raspberry pi 3* sebagai kontroler untuk mekanisme penyemprotan dan *image processing*, motor *brushless* sebagai aktuator *drone* dan pompa wiper sebagai aktuator untuk mekanisme penyemprotan, serta kamera *smartphone* untuk menangkap dan atau merekam video.

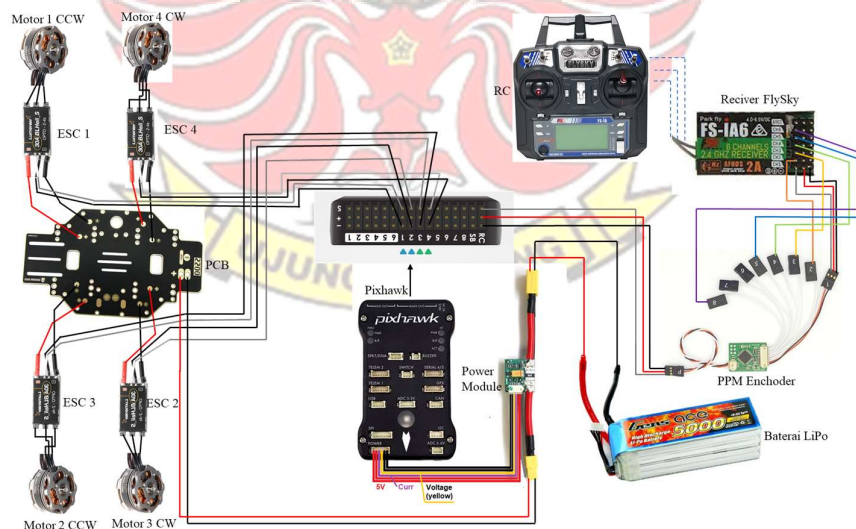
Kecepatan pada motor *brushless* diatur oleh *Electric Speed Control (ESC)*. Sumber tegangan menggunakan baterai lithium polymer (LiPo) 4s dengan kapasitas 5000mAh yang berfungsi menyuplay tegangan masuk ke *step down dc (5 Volt)* untuk *raspberry pi 3* ke *step down dc (12 Volt)* untuk pompa wiper, dan ke *power module (5 Volt)* untuk *flight controller*. Baterai juga berfungsi

untuk menyuplai tegangan ke ESC yang terhubung langsung ke *power distribution board*.

Adapun hubungan-hubungan elektronik pada *drone* dapat dijabarkan sebagai berikut:

- a) Rangkaian hubungan antara *flight controller*, RC (*Radio Controller*), ESC (*Electrical Speed Control*), Motor *Brushless*, PPM (*Pulse Position Modulation*) Encoder dan Receiver RC.

Rangkaian ini merupakan rangkaian mekanisme gerakan *yaw/throttle* dan *roll/pitch* yang terdiri dari komponen *flight controller*, *radio controller*, ESC, motor brushless, PPM dan receiver RC. Gambar 3.5 menunjukkan rangkaian hubungan antara *flight controller*, *radio controller*, ESC, motor *brushless*, PPM encoder dan receiver RC.



Gambar 3.5 Rangkaian Hubungan antara *Flight Controller*, *Radio Controller*, ESC, Motor *Brushless*, PPM Encoder dan Receiver RC

b) Rangkaian hubungan antara *flight controller* dan *safety switch*

Rangkaian ini merupakan rangkaian untuk mengaktifkan atau menonaktifkan output ke komponen motor *brushless*. Gambar 3.6 menunjukkan rangkaian hubungan antara *flight controller* dan *safety switch*.



Gambar 3.6 Rangkaian Hubungan antara *Flight Controller* dan *Safety Switch*

c) Rangkaian hubungan antara *flight controller*, telemetri dan laptop

Rangkaian ini merupakan rangkaian yang digunakan untuk menghubungkan laptop dan *flight controller* dengan menggunakan telemetri sebagai transfer datanya. Gambar 3.7 menunjukkan rangkaian hubungan antara *flight controller*, telemetri dan laptop.



Gambar 3.7 Rangkaian Hubungan antara *Flight Controller*, Telemetri dan Laptop

d) Rangkaian hubungan antara *flight controller* dan *buzzer*

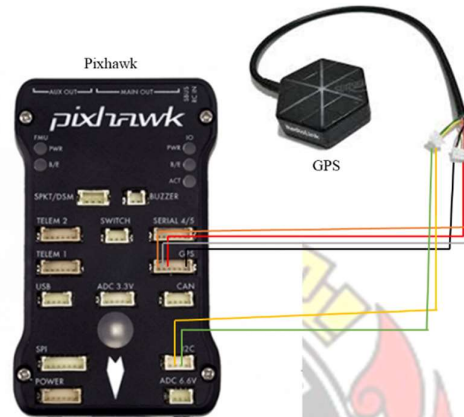
Rangkaian ini merupakan rangkaian yang digunakan untuk menunjukkan perubahan status *drone* dengan suara/nada yang berbeda-beda tergantung status *dronenya*. Gambar 3.8 menunjukkan rangkaian hubungan antara *flight controller* dan *buzzer*.



Gambar 3.8 Rangkaian Hubungan antara *Flight Controller* dan *Buzzer*

e) Rangkaian hubungan antara *flight controller* dan GPS

Rangkaian ini merupakan rangkaian untuk menentukan letak atau posisi *drone* di permukaan bumi dengan bantuan sinyal satelit. Gambar 3.9 menunjukkan rangkaian hubungan antara *flight controller* dan GPS.



Gambar 3.9 Rangkaian Hubungan antara *Flight Controller* dan GPS

3.4.3 Pembuatan Mekanisme Penyemprotan Insektisida

Mekanisme penyemprotan insektisida dibuat dengan tujuan agar memudahkan petani dalam mengatasi hama wereng coklat pada persawahan yang luas, sehingga prosesnya menjadi lebih cepat dan efisien.

Adapun langkah pertama yang dilakukan dalam pembuatan mekanisme penyemprotan insektisida adalah melakukan pemilihan pompa yang dapat meneruskan cairan ke *nozzle* sehingga cairan yang disemprotkan berbentuk embun atau kabut.

Dalam hal ini, penulis melakukan uji coba menggunakan pompa celup 5V, akan tetapi pompa tersebut tidak menghasilkan tekanan yang cukup untuk meneruskan cairan ke *nozzle*, maka penulis mencoba pompa wiper 12V, dan hasilnya pompa tersebut berhasil meneruskan cairan ke *nozzle*, sehingga cairan

yang disemprotkan dapat berbentuk seperti embun atau kabut. Adapun proses uji coba pompa wiper 12V ditunjukkan pada Gambar 3.10.

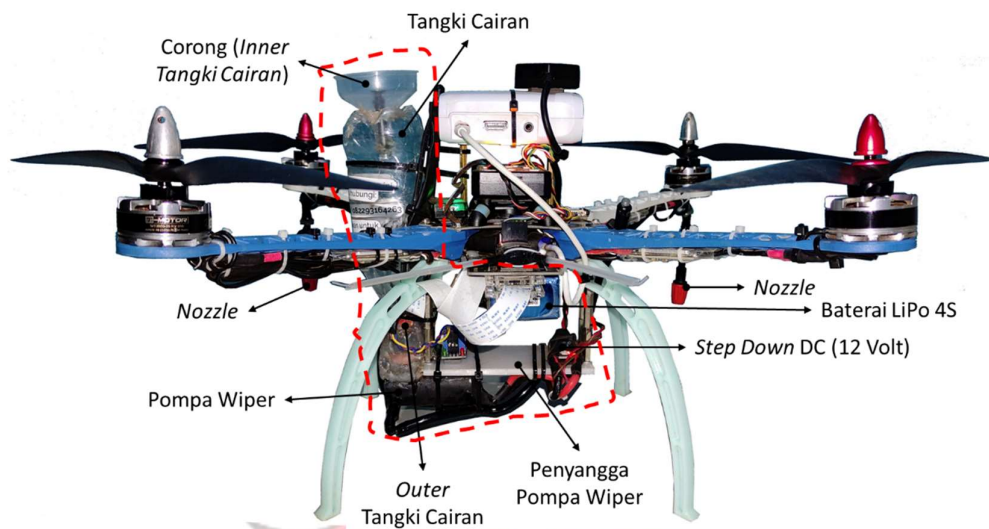


Gambar 3.10 Uji Coba Pompa Wiper 12V

Dikarenakan pompa celup memiliki tegangan kerja 5V dan pompa wiper memiliki tegangan kerja 12V, maka dalam melakukan uji coba pompa, penulis menghubungkan sumber tegangan yaitu baterai *lithium polymer* (LiPo) 4s yang memiliki total tegangan sampai dengan 16.8V ke *stepdown* dc terlebih dahulu, sehingga besarnya tegangan yang masuk ke pompa-pompa tersebut sesuai dengan tegangan kerjanya masing-masing.

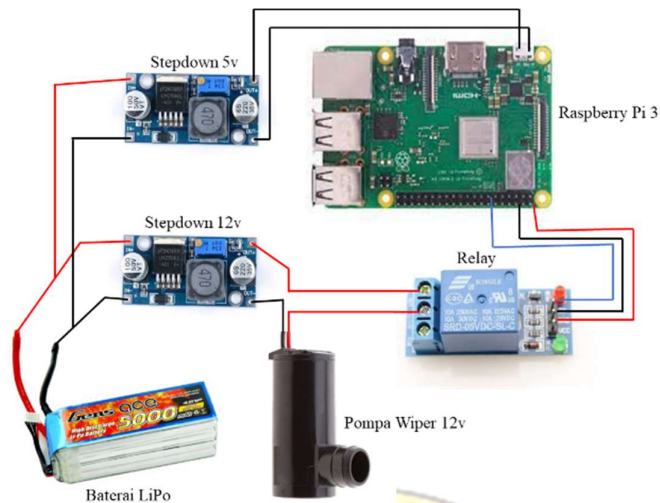
Setelah menemukan pompa yang cocok, selanjutnya penulis melakukan perancangan mekanik untuk mekanisme penyemprotan, yang terdiri dari botol plastik sebagai tangki cairan penyemprotan yang memiliki volume 230ml, pompa wiper 12 Volt dan 2 *nozzle*. Bagian tutup botol plastik dihubungkan dengan *inlet* pompa wiper, dikarenakan pompa wiper dapat berfungsi ketika *inlet*nya terendam cairan dan bagian pangkal botol plastik dilubangi dan dihubungkan dengan corong yang dijadikan sebagai *inlet* untuk masuknya cairan penyemprotan.

Untuk dua *nozzle* diletakkan masing-masing dibagian belakang *drone*, tepatnya dibawah motor *brushless* dan untuk penyangga pompa wiper terletak di bawah frame *drone* atau di antara *landing gear*. Penyangga ini berfungsi sebagai tempat melekatnya pompa wiper dan dudukan untuk modul relay dan *step down* dc (12 Volt). Adapun hasil perancangan mekanik untuk mekanisme penyemprotan ditunjukkan pada Gambar 3.11.



Gambar 3.11 Hasil Perancangan Mekanik untuk Mekanisme Penyemprotan

Setelah perancangan mekanik selesai, selanjutnya penulis melakukan perancangan elektronik dengan menghubungkan *raspberry pi*, pompa wiper dan modul relay. Adapun rangkaian hubungan antara *raspberry pi*, pompa wiper, dan modul relay ditunjukkan pada Gambar 3.12.



Gambar 3.12 Rangkaian Hubungan antara *Raspberry Pi*, Pompa Wiper, dan Modul Relay

Setelah perancangan mekanik dan elektronik selesai, selanjutnya penulis melakukan pembuatan perancangan perangkat lunak, dimana program yang dibuat nantinya akan digunakan untuk mengatur nyala modul relay yang terhubung dengan pompa wiper pada saat pengujian mekanisme penyemprotan. Hal ini dimaksudkan agar pompa wiper bisa menyemprotkan cairan selama waktu yang ditentukan ketika program dijalankan. Adapun tampilan program eksperimen penyemprotan saat dijalankan ditunjukkan pada Gambar 3.13.

```

Shell
Python 3.5.3 (/usr/bin/python3)
>>> %Run EksperimenPenyemprotan.py
*****
* PROGRAM EKSPERIMEN PENYEMPROTAN *
*****
Lama Penyemprotan (detik): 40
[Relay ON] 40 detik
[Relay OFF]
Apakah Anda masih ingin melakukan eksperimen penyemprotan (y/n)?

```

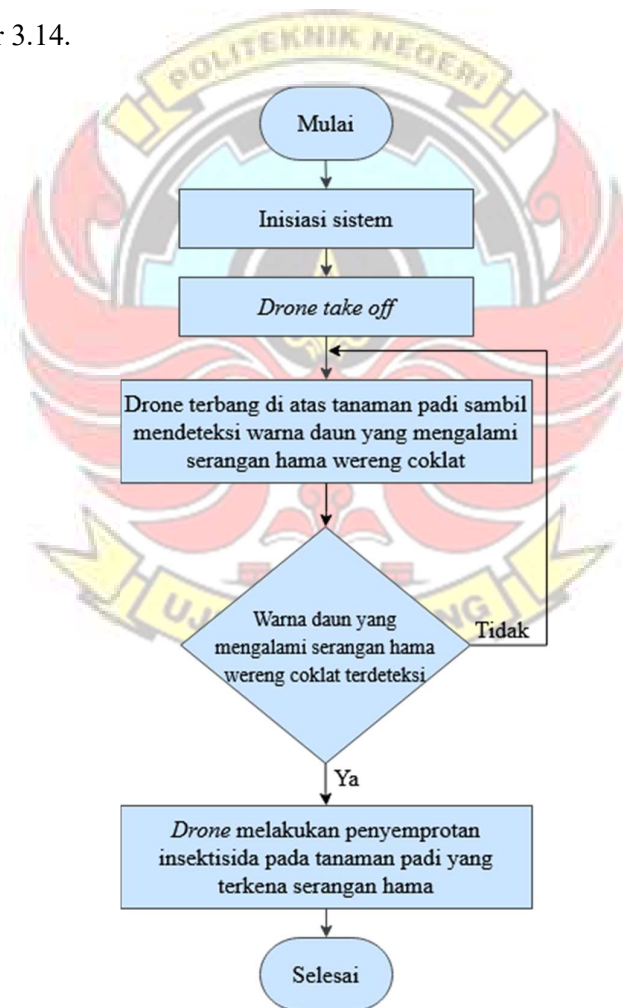
Gambar 3.13 Tampilan Program Eksperimen Penyemprotan saat dijalankan

Setelah perancangan perangkat lunak selesai dibuat, maka mekanisme penyemprotan sudah berhasil dibuat.

3.4.4 Pembuatan Algoritma *Image Processing*

Pembuatan algoritma *image processing* dibuat dengan tujuan agar memudahkan petani dalam mendeteksi hama wereng coklat di areal persawahan yang luas, sehingga jika terdapat tanaman padi yang mengalami serangan hama wereng coklat, maka petani dapat melakukan strategi pengendalian hama wereng coklat dengan lebih cepat.

Perancangan ini diawali dengan membuat diagram alir yang telah disajikan pada Gambar 3.14.



Gambar 3.14 Diagram Alir Perancangan *Software*

Sebelum dilakukan *image processing*, dilakukan pengambilan data berupa rekaman video, yang akan dilakukan dengan menerbangkan *drone* di areal persawahan, sementara *drone* terbang maka kamera *smartphone* melakukan perekaman video.

Setelah proses perekaman video selesai, maka yang dilakukan selanjutnya adalah melakukan *color filtering* pada rekaman video tersebut untuk mengambil nilai HSV. Kemudian, nilai HSV tersebut akan digunakan sebagai nilai standar yang dijadikan acuan untuk eksperimen real.

Proses pengambilan data dan eksperimen real akan dilaksanakan di salah satu areal persawahan yang berada di Kecamatan Bontomarannu, Kabupaten Gowa. Pembuatan *image processing* dilakukan dengan *library* OpenCV menggunakan bahasa pemrograman python.

3.4.5 Pengujian Alat dan Pengambilan Data

Pada proses ini akan dilakukan tiga tahapan pengujian. Pengujian pertama adalah untuk menguji apakah *drone* dapat melakukan misi terbang mengikuti *way-point* yang telah ditentukan dengan tepat dan mendarat secara otomatis.

Pengujian kedua adalah untuk mengetahui luas area mekanisme penyemprotan cairan dengan menerbangkan *drone* pada ketinggian ± 1 meter sampai dengan $\pm 1,5$ meter dari atas tanah. Pengujian ketiga adalah untuk mengetahui apakah kamera *drone* dapat mendeteksi warna daun pada tanaman padi yang mengalami serangan wereng coklat dengan tepat.

Adapun data yang diambil pada tahap pengujian adalah sebagai berikut:

- 1) Data koordinat GPS yang telah ditentukan untuk misi terbang dan data koordinat GPS saat menjalankan misi terbang tersebut pada aplikasi *mission planner*.
- 2) Luas area penyemprotan cairan pada saat *drone* diterbangkan dengan ketinggian 1 sampai 1,5 meter dari atas tanah.
- 3) Nilai HSV dari warna daun pada tanaman padi yang mengalami serangan hama wereng coklat.

3.4.6 Pengolahan Data

Pengolahan data dilakukan dengan mengumpulkan semua data hasil penelitian baik data kualitatif maupun data kuantitatif. Setelah data terkumpul, maka dilakukan pengolahan serta penganalisisan data hasil penelitian dengan tujuan untuk menjawab rumusan masalah yang telah ditentukan kemudian ditarik kesimpulan.

3.4.7 Penyusunan Laporan

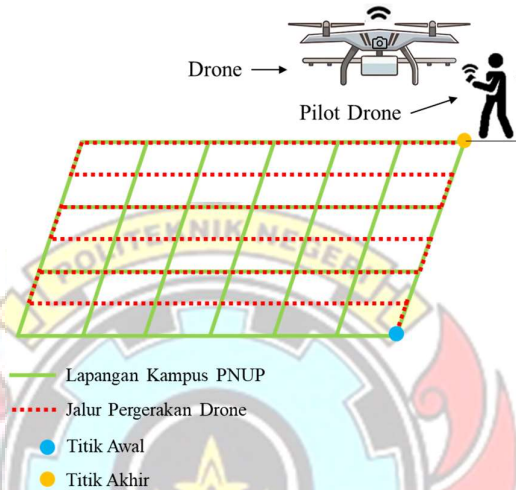
Penyusunan laporan merupakan bukti tertulis menilai kualitas dan ketepatan penelitian dalam menyelesaikan rumusan masalah secara real. Maka dari itu penyusunan laporan penelitian harus dibuat dengan sebaik-baiknya dan disesuaikan berdasarkan buku pedoman.

3.5. Rencana Eksperimen

Sebelum melakukan eksperimen real di lapangan, maka terlebih dahulu dilakukan beberapa eksperimen pendahuluan di lapangan kampus PNUP.

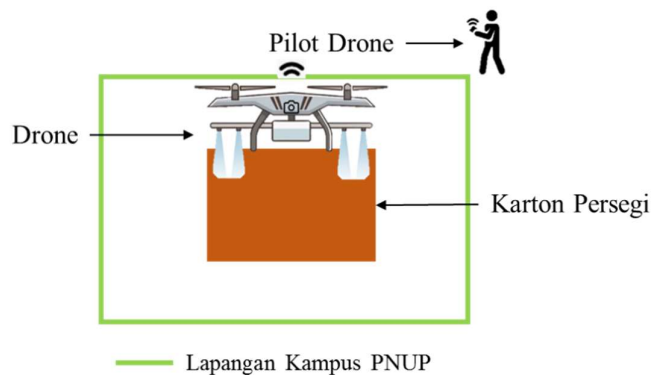
Eksperimen pendahuluan tersebut terdiri atas:

1. Eksperimen terbang otomatis dengan *waypoint* yang sudah ditentukan dengan ketinggian 1,5 meter dari atas lapangan. Adapun sketsa rencana eksperimen terbang ditunjukkan pada Gambar 3.15.



Gambar 3.15 Sketsa Rencana Eksperimen Terbang di Lapangan Kampus PNUP

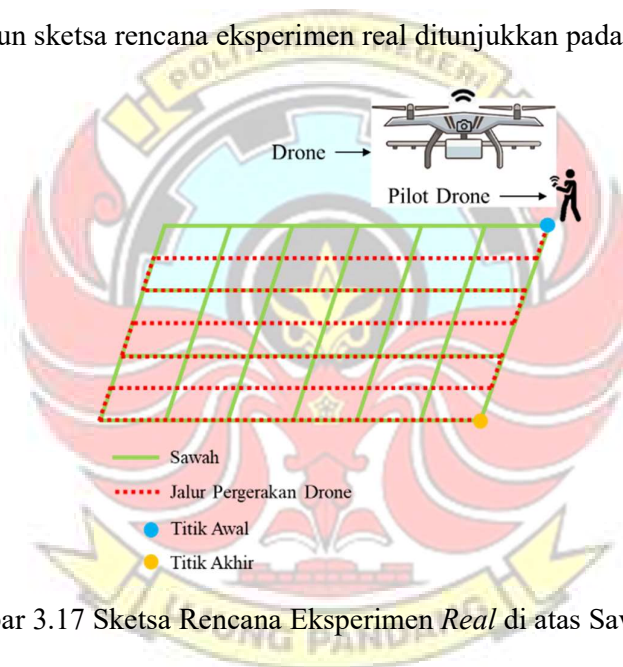
2. Eksperimen penyemprotan cairan untuk mendapatkan luas area penyemprotan dengan ketinggian *drone* 1 meter sampai dengan 1,5 meter dari permukaan lapangan. Adapun sketsa rencana eksperimen penyemprotan cairan ditunjukkan pada Gambar 3.16.



Gambar 3.16 Sketsa Rencana Eksperimen Penyemprotan Cairan

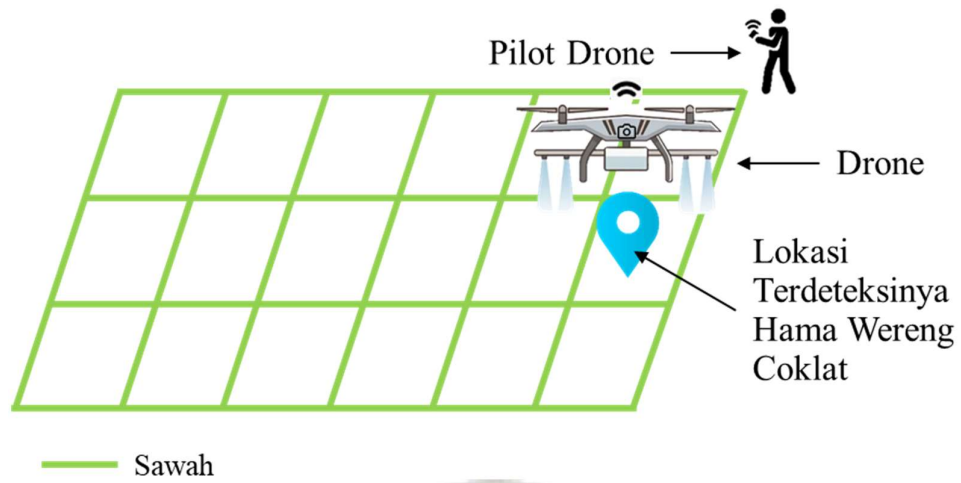
Dalam rencana eksperimen *real* di areal sawah petani, *drone* akan diterbangkan dan selama *drone* tersebut terbang, maka kamera *drone* akan melakukan pengambilan gambar bergerak (*webcam*) secara *realtime* pada ketinggian 20-60 cm di atas tanaman padi.

Kemudian dari gambar bergerak tersebut akan dilakukan *color filtering* oleh *raspberry pi* dengan *library* OpenCV menggunakan bahasa pemrograman python. Sehingga dapat diketahui manakah tanaman padi yang mengalami serangan wereng coklat. Adapun sketsa rencana eksperimen *real* ditunjukkan pada Gambar 3.17.



Gambar 3.17 Sketsa Rencana Eksperimen *Real* di atas Sawah Petani

Ketika kamera *drone* berhasil mendeteksi warna daun pada tanaman padi yang mengalami serangan hama wereng coklat, maka *drone* akan melakukan penyemprotan cairan secara otomatis di atas tanaman padi yang mengalami serangan hama wereng coklat tersebut. Adapun sketsa rencana eksperimen penyemprotan insektisida pada tanaman padi disajikan pada Gambar 3.18.



Gambar 3.18 Sketsa Rencana Eksperimen Penyemprotan Insektisida

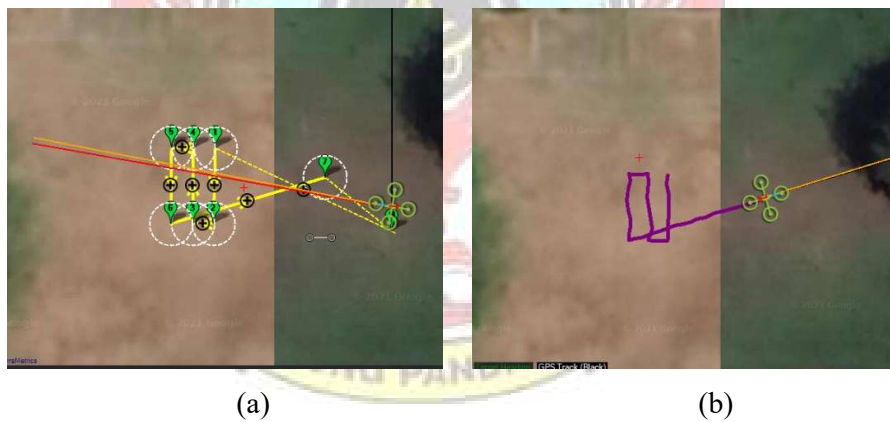


BAB IV HASIL DAN PEMBAHASAN

4.1 Pengujian Terbang *Drone*

Pengujian ini bertujuan untuk mengetahui apakah *drone* dapat terbang secara stabil dengan melewati titik jalur (*waypoint*) dan mendarat secara otomatis sesuai dengan *waypoint* yang telah ditentukan.

Pada pengujian ini digunakan aplikasi *mission planner* untuk membuat misi terbang. Jalur yang akan dibuat terdiri dari enam *waypoint* (WP) yang akan ditempuh oleh *drone* dan satu titik *land*, dimana nantinya *drone* akan mendarat dengan otomatis di titik tersebut. Adapun hasil eksperimen terbang ditunjukkan pada Gambar 4.1 dan Tabel 4.1.



Gambar 4.1 Pengujian Terbang *Drone*
(a) Rencana Misi Terbang *Drone*
(b) Hasil Pengujian Terbang *Drone*

Tabel 4.1 Hasil Eksperimen Terbang

Perintah Terbang	Rencana Misi Terbang		Hasil Eksperimen Terbang		Error	
	<i>Latitude</i>	<i>Longitude</i>	<i>Latitude</i>	<i>Longitude</i>	<i>Latitude</i>	<i>Longitude</i>
	(°S)	(°E)	(°S)	(°E)	(%)	(%)
WP #1	5,1290838	119,4815916	5,1290850	119,4815954	0	0
WP #2	5,129146	119,4815905	5,129144	119,481591	0	0
WP #3	5,129146	119,4815725	5,1291456	119,481573	0	0
WP #4	5,1290836	119,4815735	5,1290918	119,4815727	0	0
WP #5	5,1290834	119,4815555	5,1290843	119,4815545	0	0
WP #6	5,129146	119,4815544	5,1291402	119,4815549	0	0
<i>Land</i>	5,1291079	119,4816821	5,1291061	119,4816808	0	0

Berdasarkan hasil eksperimen terbang, diperoleh error koordinat GPS rata-rata antara rencana misi terbang dan hasil eksperimen terbang sebesar 0%. Berikut rumus dalam mencari nilai error koordinat GPS:

$$error = \frac{\text{koordinat hasil eksperimen terbang} - \text{koordinat rencana misi terbang}}{\text{koordinat rencana misi terbang}} \times 100\%$$

$$error = \left| \frac{\text{koordinat hasil eksperimen terbang}}{\text{koordinat rencana misi terbang}} - 1 \right| \times 100\%$$

Sehingga dari Tabel 4.1 dapat diketahui bahwa *drone* telah berhasil terbang secara stabil dengan melewati titik jalur (*waypoint*) dan mendarat secara otomatis sesuai dengan titik yang telah ditentukan tanpa mengalami hambatan, dengan tingkat error sebesar 0%.

4.2 Pengujian Mekanisme Penyemprotan

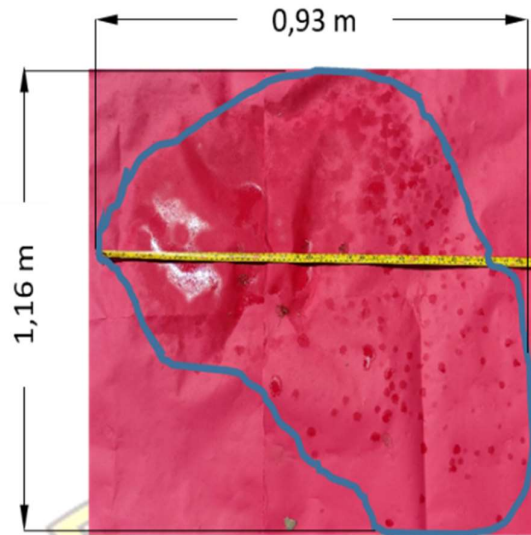
Pengujian ini bertujuan untuk mengetahui luas area penyemprotan *drone*. Sebelum melakukan pengujian mekanisme penyemprotan, langkah awal yang penulis lakukan adalah melakukan pengisian ke tangki cairan dengan air sebanyak ± 220 ml. Selanjutnya pengujian dilakukan dengan menerbangkan *drone* di atas karton yaitu pada ketinggian ± 1 meter hingga $\pm 1,5$ meter.

Kemudian modul relay akan diaktifkan selama 10 detik dengan menjalankan program yang telah dibuat pada *raspberry pi* sebelumnya, sehingga pompa wiper akan menyemprotkan cairan. Adapun proses eksperimen mekanisme penyemprotan ditunjukkan pada Gambar 4.2.



Gambar 4.2 Proses Eksperimen Mekanisme Penyemprotan

Adapun hasil pengujian mekanisme penyemprotan ditunjukkan pada Gambar 4.3 dan Tabel 4.2.



Gambar 4.3 Hasil eksperimen mekanisme penyemprotan pada ketinggian $\pm 1,5$ meter.

Tabel 4.2 Hasil Eksperimen Mekanisme Penyemprotan

Ketinggian Terbang <i>Drone</i> (meter)	Luas Penyemprotan (m ²)	Durasi Waktu Penyemprotan (detik)
± 1	$\pm 1,026$	10
$\pm 1,25$	$\pm 1,086$	10
$\pm 1,5$	$\pm 1,079$	10

Berdasarkan hasil pengujian mekanisme penyemprotan pada Tabel 4.2, ditunjukkan bahwa mekanisme penyemprotan yang dirancang sudah mampu melakukan penyemprotan cairan dengan kondisi *drone* diterbangkan, akan tetapi pada saat *drone* berada pada ketinggian $\pm 1,25$ meter, jangkauan penyemprotannya lebih luas dibandingkan pada saat *drone* berada pada ketinggian $\pm 1,5$ meter. Hal ini dikarenakan kencangnya angin pada saat pengujian sehingga membuat cairan yang keluar dari *nozzle* lebih menyebar dan tidak merata. Selain itu, posisi *nozzle* yang

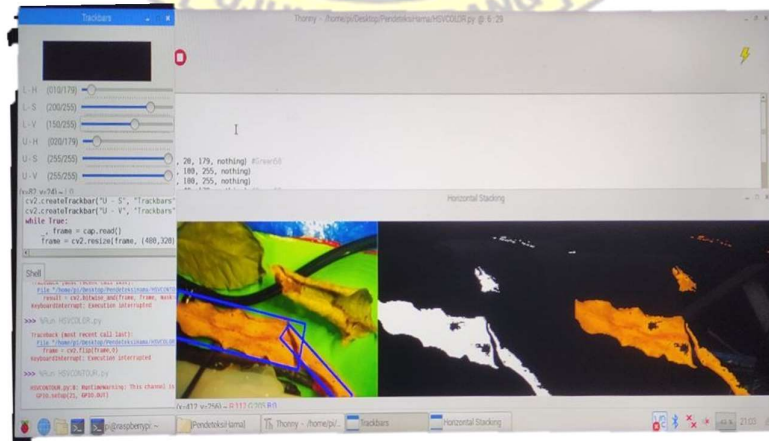
berada di bawah motor *brushless* mengakibatkan hasil penyemprotan terpengaruh oleh hembusan udara yang dihasilkan dari *propeller drone*.

4.3 Pengujian Algoritma *Image Processing*

4.3.1 Uji Coba Algoritma *Image Processing*

Sebelum melakukan pengujian algoritma *image processing* untuk mendeteksi warna daun pada tanaman padi yang mengalami serangan hama wereng coklat, langkah awal yang penulis lakukan adalah melakukan uji coba program *color filtering* dengan menggunakan daun yang berwarna coklat kekuning-kuningan sebagai objek yang akan dideteksi, proses deteksi warna dilakukan dengan cara mengatur rentang nilai HSV sampai warna daun yang diinginkan terdeteksi.

Adapun rentang nilai yang digunakan pada saat pembuatan program *color filtering* yaitu rentang nilai *hue* adalah 10 sampai 20, rentang nilai *saturation* adalah 200 sampai 255, dan nilai *value* adalah 150 sampai 255. Hasil program *color filtering* saat dijalankan ditunjukkan pada Gambar 4.4.



Gambar 4.4 Hasil Program *Colour Filtering* saat dijalankan

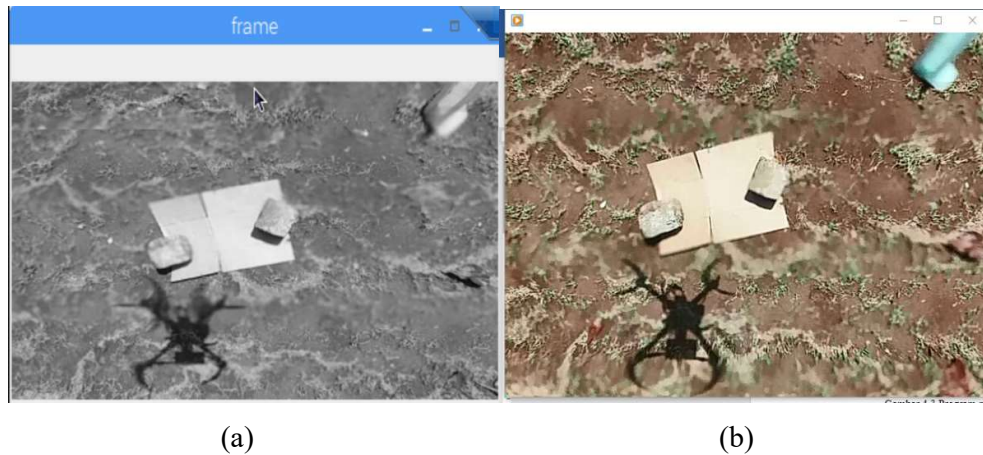
Setelah melakukan uji coba program *color filtering* dengan mendeteksi warna daun yang berwarna coklat kekuning-kuningan, selanjutnya penulis melakukan uji coba program *color filtering* untuk mengaktifkan modul relay saat objek yang diinginkan terdeteksi, dengan cara mengatur rentang nilai HSV-nya. Pada proses pembuatannya, penulis menghubungkan modul relay dengan LED dan menggunakan rentang nilai HSV dari warna merah, sehingga pada saat ada objek yang berwarna merah terdeteksi, maka LED akan menyala.

Adapun rentang nilai yang digunakan pada saat pembuatan program *color filtering* untuk mengaktifkan modul relay yaitu rentang nilai *hue* adalah 161 sampai 179, rentang nilai *saturation* adalah 155 sampai 255, dan nilai *value* adalah 84 sampai 255. Hasil program *colour filtering* untuk mengaktifkan modul relay saat dijalankan ditunjukkan pada Gambar 4.5.



Gambar 4.5 Hasil Program *Colour Filtering* untuk Mengaktifkan Modul Relay saat dijalankan

Pada langkah selanjutnya, penulis melakukan pembuatan program perekaman video, dimana program ini berfungsi merekam video yang ditangkap oleh kamera ketika *drone* diterbangkan, nantinya *output* dari program ini dapat dijadikan *input* untuk program pemutar video. Adapun hasil program perekaman video ditunjukkan pada Gambar 4.6.



Gambar 4.6 Program Perekaman Video
 (a) Tampilan Kamera saat Program dijalankan dan Melakukan Perekaman Video
 (b) Hasil Rekaman Video saat diputar Menggunakan Aplikasi Pemutar Video di Windows

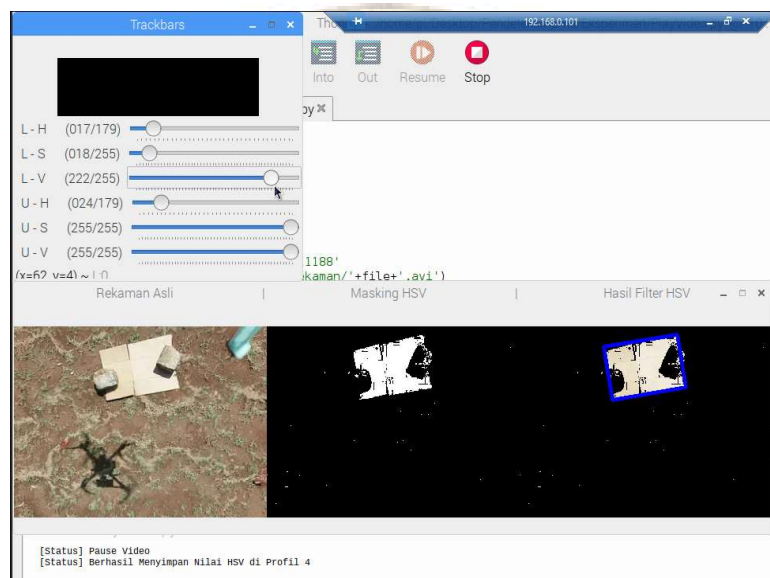
Dari Gambar 4.6 ditunjukkan bahwa pada saat program dijalankan terlihat tangkapan dari kamera berwarna abu-abu, akan tetapi pada saat video hasil rekaman diputar menggunakan aplikasi pemutar video, maka video terlihat berwarna sesuai dengan kondisi asli di lapangan. Hal ini dilakukan untuk meminimalisir *delay* yang terjadi pada saat perekaman video.

Setelah program perekaman video berhasil dibuat, selanjutnya penulis melakukan pembuatan program untuk melakukan pemutaran video yang telah direkam sebelumnya, dimana program ini dapat melakukan *play* dan *pause* pada video yang sedang diputar, sehingga penulis dapat melakukan *color filtering* untuk mencari nilai HSV pada objek yang berada di rekaman video yang sedang diputar.

Ketika nilai HSV sudah sesuai dengan objek yang diinginkan, maka program ini dapat menyimpan nilai HSV, nantinya nilai HSV tersebut dapat dijadikan *input* untuk program *color filtering* yang telah dibuat sebelumnya, dimana

program tersebut akan digunakan pada “pengujian algoritma *image processing* dengan mengaktifkan mekanisme penyemprotan”.

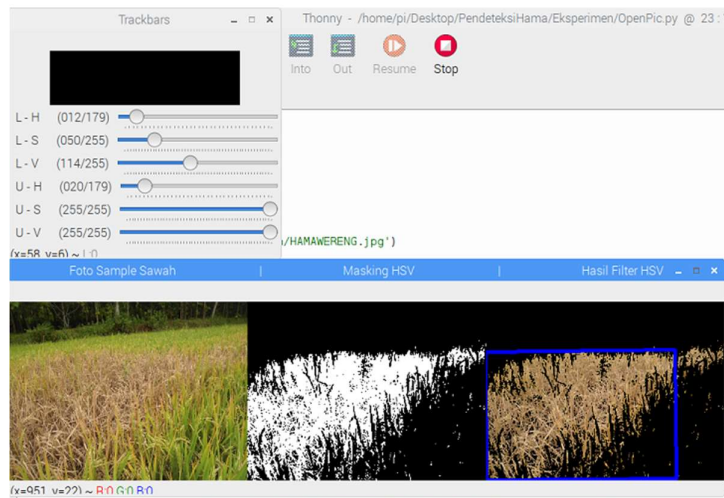
Pada proses pembuatannya, penulis mencari nilai HSV dari warna kardus. Adapun rentang nilai HSV yang digunakan untuk mendeteksi warna kardus yaitu rentang nilai *hue* adalah 17 sampai 24, rentang nilai *saturation* adalah 18 sampai 255, dan nilai *value* adalah 222 sampai 255. Hasil program pemutar video saat dijalankan ditunjukkan pada Gambar 4.7.



Gambar 4.7 Hasil Program Pemutar Video saat dijalankan

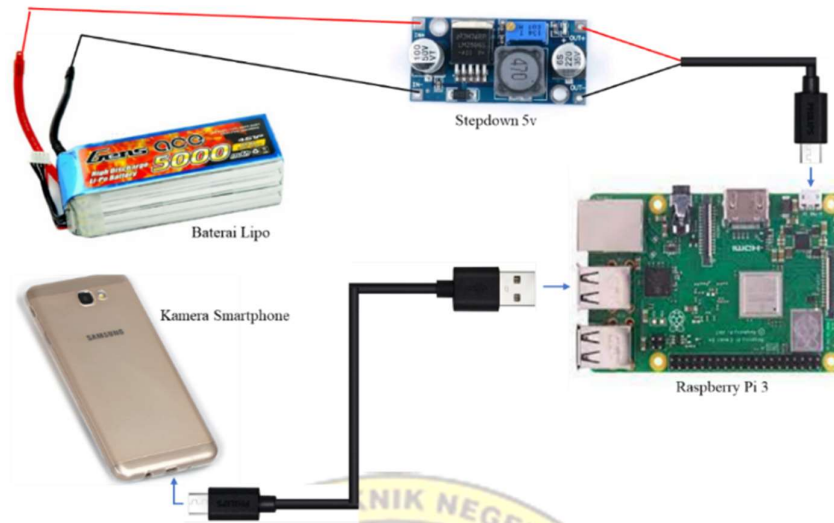
Setelah program perekaman video sudah berhasil dibuat, selanjutnya penulis melakukan pembuatan program *color filtering* untuk mencari nilai HSV pada gambar daun padi yang terkena hama wereng coklat, nantinya nilai HSV dari gambar tersebut digunakan sebagai data tambahan dalam menentukan rentang nilai *hue* pada pengujian algoritma *image processing* dengan mengaktifkan mekanisme penyemprotan.

Adapun rentang nilai HSV yang digunakan untuk mendeteksi warna daun padi yang terkena hama wereng coklat pada gambar yaitu rentang nilai *hue* adalah 12 sampai 20, rentang nilai *saturation* adalah 50 sampai 255, dan nilai *value* adalah 114 sampai 255. Hasil program pemutar *color filtering* pada gambar daun padi ditunjukkan pada Gambar 4.8.



Gambar 4.8 Hasil Program *Color Filtering* pada Gambar Daun Padi

Setelah program *color filtering* pada gambar sudah berhasil dibuat, maka pengujian untuk algoritma *image processing* siap untuk dilakukan, baik “pengujian pendahuluan algoritma *image processing*” maupun “pengujian algoritma *image processing* dengan mengaktifkan mekanisme penyemprotan”. Adapun rangkaian hubungan antara *raspberry pi 3* dan kamera *smartphone* ditunjukkan pada Gambar 4.9.



Gambar 4.9 Rangkaian Hubungan antara *Raspberry Pi 3* dan Kamera *Smartphone*

4.3.2 Hasil Pengambilan Data

Pengambilan data ini bertujuan untuk mencari nilai HSV dari warna daun pada tanaman padi yang mengalami serangan hama wereng coklat. Nilai HSV ini akan digunakan sebagai nilai standar yang dijadikan acuan atau *inputan* untuk pengujian algoritma *image processing* dengan mengaktifkan mekanisme penyemprotan.

Dalam pelaksanaannya, dikarenakan penulis tidak mendapatkan areal persawahan yang mengalami serangan hama wereng coklat, maka penulis mencari nilai HSV dari daun padi yang berwarna coklat, dimana warna ini merupakan salah satu ciri khas apabila tanaman padi mengalami serangan hama wereng coklat.

Dalam proses mencari nilai HSV pada daun padi, penulis menemukan bahwa besarnya nilai *hue* untuk warna coklat pada daun kering, warna coklat pada kardus, dan warna coklat pada gambar daun padi yang terkena hama wereng coklat berada pada rentang nilai 10 sampai 24, sehingga dalam mencari nilai *hue* yang ada

pada daun padi yang berwarna coklat, penulis mencari nilai *hue* pada rentang nilai tersebut.

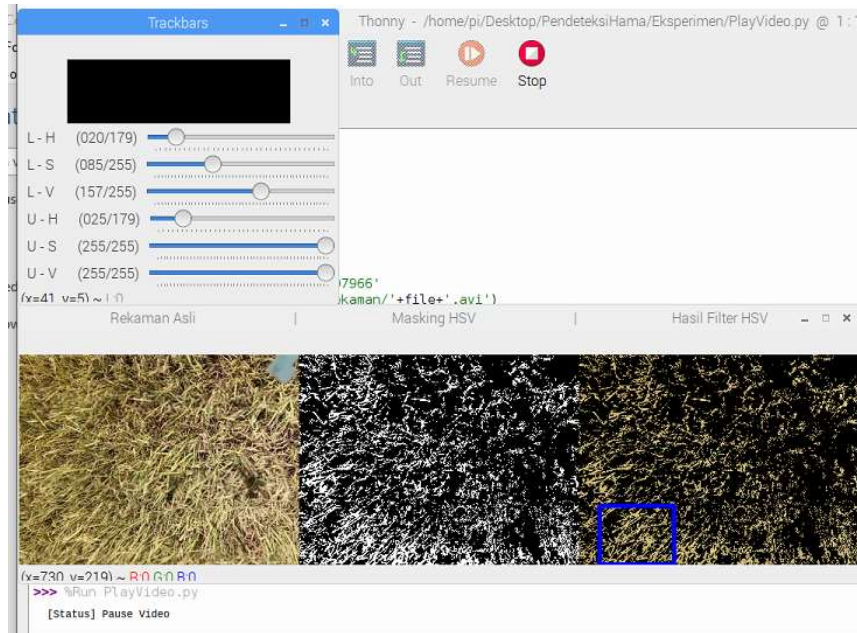
Adapun rentang nilai HSV yang didapatkan dari hasil pengambilan data yaitu: rentang nilai *hue* adalah 20 sampai 25, rentang nilai *saturation* adalah 85 sampai 255, dan rentang nilai *value* adalah 157 sampai 255. Proses pengambilan data berupa rekaman video ditunjukkan pada Gambar 4.10, Gambar 4.11, dan Gambar 4.12.



Gambar 4.10 Jalur Terbang yang *Drone* akan lewati untuk perekaman video



Gambar 4.11 Proses Pengambilan Data berupa Rekaman Video

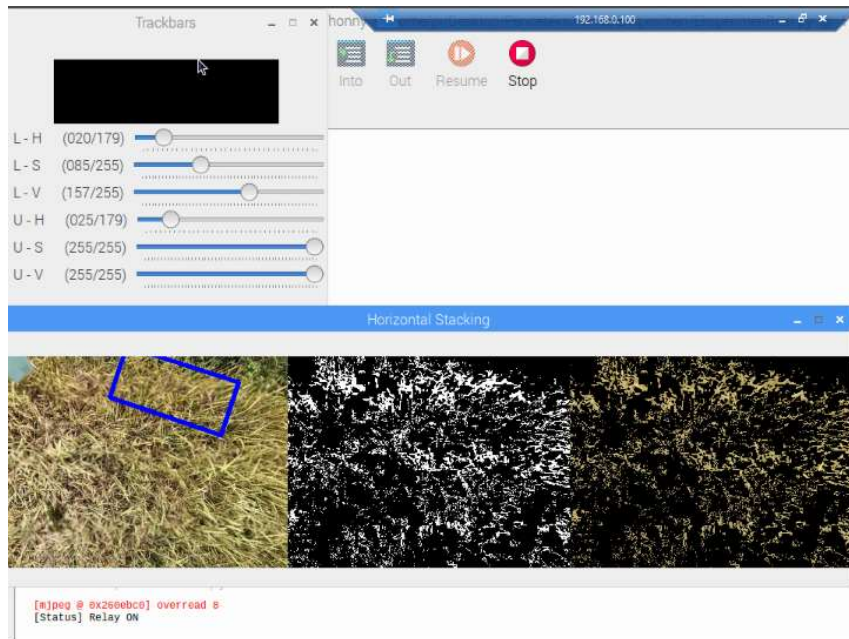


Gambar 4.12 Proses *Color Filtering* pada Hasil Rekaman Video

4.3.3 Hasil dan Analisis Pengujian Algoritma *Image Processing* dengan Mengaktifkan Mekanisme Penyemprotan Insektisida

Pengujian ini bertujuan untuk menguji kemampuan algoritma *image processing* yang telah dibuat, apakah algoritma yang dibuat sudah dapat mendeteksi nilai HSV sesuai dengan nilai HSV yang telah didapatkan pada pengambilan data sebelumnya, dan ketika objek dari nilai HSV tersebut sudah terdeteksi, apakah algoritma yang dibuat sudah dapat mengaktifkan mekanisme penyemprotan, sehingga *drone* dapat melakukan penyemprotan cairan secara otomatis.

Dalam pelaksanaannya, dikarenakan penulis tidak mendapatkan areal persawahan yang mengalami serangan hama wereng coklat untuk disemprotkan insektisida, maka penulis mengganti cairan penyemprotan dengan air. Adapun tampilan program pada saat mendeteksi nilai HSV yang telah ditentukan, ditunjukkan pada Gambar 4.13.



Gambar 4.13 Tampilan Program pada saat Mendeteksi Nilai HSV

Dari Gambar 4.13 dapat dilihat bahwa algoritma *image processing* yang dirancang sudah mampu mendeteksi warna daun padi yang berwarna coklat atau sesuai dengan nilai HSV yang telah ditentukan dan juga mampu mengaktifkan modul relay yang terhubung ke mekanisme penyemprotan ketika warna objek tersebut terdeteksi. Dengan kata lain, sistem *drone* yang dirancang sudah mampu melakukan penyemprotan cairan secara otomatis ketika mendeteksi warna yang ditentukan.

BAB V PENUTUP

5.1 Kesimpulan

Dari hasil penelitian yang dilakukan, maka dapat disimpulkan bahwa:

1. Mekanisme penyemprotan yang dirancang telah dibuat dan sudah mampu menyemprotkan cairan dalam kondisi *drone* diterbangkan. Pada uji coba dengan ketinggian terbang $\pm 1,5$ m dan durasi waktu penyemprotan 10 detik, luas area penyemprotan *drone* sekitar $\pm 1,079$ m².
2. Algoritma *image processing* yang dirancang sudah mampu mendeteksi warna daun padi yang berwarna coklat dan juga mampu melakukan penyemprotan cairan secara otomatis ketika warna objek yang ditentukan terdeteksi.

5.2 Saran

Drone untuk deteksi hama dan penyemprotan pestisida pada tanaman padi ini masih memiliki banyak kekurangan. Oleh sebab itu, saran untuk pengembangan penelitian ini kedepannya adalah sebagai berikut:

1. Disarankan menggunakan kamera yang memiliki resolusi tinggi dan memiliki fitur *auto focus* dan *stabilizer*, sehingga gambar yang ditangkap lebih jelas dan stabil, serta warnanya lebih cerah dan lebih tajam.
2. Dikarenakan koneksi antara laptop dan *raspberry pi* melalui sambungan *Wi-Fi* dan menggunakan aplikasi *remote desktop connection*, mengakibatkan jangkauan jaringan yang terbatas dan tampilan *raspberry pi* pada laptop mengalami *delay*, maka dari itu disarankan menggunakan koneksi melalui *video sender*.

3. Disarankan melakukan penggantian untuk komponen *receiver radio controller* dan PPM *encoder*, dikarenakan umur komponennya sudah lama, sehingga koneksi antara *radio controller* dan *receivernya* sering terputus.
4. Disarankan untuk menggunakan *frame drone* yang memiliki dimensi lebih luas untuk membuat terbang *drone* lebih stabil dan mampu membawa beban yang lebih berat.



DAFTAR PUSTAKA

- Abidin, Hassanuddin Z, dkk. 2009. Deformasi Koseismik dan Pascaseismik Gempa Yogyakarta 2006 dari Hasil Survei GPS. Dalam *Jurnal Geologi Indonesia*, IV (4): 275-284.
- Akbar, Son Ali dan Anton Yudhana. 2016. Prototipe Tabung Semprot Pupuk Cair Berbasis Wahana *Quadcopter* bagi Lahan Tanaman Padi. Skripsi. Yogyakarta: Universitas Ahmad Dahlan.
- Alselectro. 2014. *Quadcopter-1.Things you need to build your own copter.* (Online). (<https://alselectro.wordpress.com/2014/10/13/quadcopter1-things-you-need-to-build-your-own-copter/>). Diakses 1 Februari 2021.
- Arnizam. 2018. Rancang Bangun Sistem Penyemprotan Pestisida dan Pupuk Pada Tanaman Padi Menggunakan Mikrokontroler. Skripsi. Makassar: Universitas Islam Negeri Alauddin Makassar.
- Baehaki dan A. Kartohardjono. 2005. Penilaian Penurunan Hasil Berdasar Skor Kerusakan akibat wereng coklat dan Wereng Punggung Putih. Prosiding *Seminar Nasional dan Kongres Biologi XIII*. Yogyakarta.
- Bradski, Gary dan Adrian Kaehler. 2008. *Learning OpenCV*. O'Reilly Media, Inc., Sebastopol: 1005 *Gravenstein Highway North*.
- Carafano, James dan Andrew Gudgel. 2007. *The Pentagon's Robots: Arming the Future. The Heritage Foundation: Background*, 2096: 1-6.
- Deswal, Monika dan Nectu Sharma. 2014. *A Fast HSV Image Color and Texture Detection and Image Conversion Algorithm*. Dalam *International Journal of Science and Research (IJSR)*, III (6): 1279-1284.
- Dhiemas, R.Y.S. 2011. Pencarian Posisi Robot: Studi Kasus Pencarian Sumber Kebocoran Gas. Dalam *Jurnal Ilmu Komputer dan Informasi*, III (2): 55-64.
- Diminimalis, Lutfi. 2019. Beginiilah Cara Menggunakan *IP Camera* di Android dengan Benar. (Online). (<https://www.diminimalis.com/cara-menggunakan-ip-camera-di-android/>). Diakses 30 Agustus 2021.
- Firdaus, Oktri Mohammad. 2010. Analisis Implementasi *Global Positioning System* (GPS) pada Moda Transportasi di PT.X. Yogyakarta: UGM Yogyakarta.
- Harfian, Irwan dan Yakob Bunga Tumanan. 2020. Teknologi *Drone* Mendukung Kostratani di Sulawesi Tengah. (Online).

(<https://sulteng.litbang.pertanian.go.id/ind/index.php/info-teknologi/1402-teknologi-drone-mendukung-kostratani-di-sulawesi-tengah>). Diakses 6 Februari 2021.

Hasan, Achmad Syahrizal Fahmi dan Adam Palilu. 2020. Pengembangan Prototipe *Drone* untuk Keperluan Evakuasi Korban Bencana. Skripsi. Makassar: Politeknik Negeri Ujung Pandang.

Heywood, Mark. 2017. OpenCV Color Detection and Filtering with Python. (*Online*). (<https://www.bluetin.io/opencv/opencv-color-detection-filtering-python/>). Diakses 31 Agustus 2021.

Humas Fasilkom. 2020. Metode Jalur Penyemprotan Mandiri Drone tanpa Awak Pembasmi Hama Tanaman. (*Online*). (<https://cs.ui.ac.id/2020/06/21/metode-jalur-penyemprotan-mandiri-drone-tanpa-awak-pembasmi-hama-tanaman/>). Diakses 1 Februari 2021.

Ikhsan, Firman. 2014. Mengenal Dasar-dasar *Quadcopter*. (*Online*). (<https://firmanikhsan.com/mengenal-quadcopter/>). Diakses 1 Februari 2021.

Ikhwana, Nuri dan Dwi Retno Hapsari. 2019. Aplikasi *Drone* Wawasan Tani untuk Pertanian di Simpang Lima, Sungai Besar, Selangor. Jurnal. Bogor: Institut Pertanian Bogor.

Keizer, Kannan. 2008. *Color Filtering*. (<https://www.mathworks.com/matlabcentral/fileexchange/20477-color-filtering>). (*Online*). Diakses 22 Februari 2021.

Khlebovich, Pavel. 2020. IP Webcam (*Online*). (<https://play.google.com/store/apps/details?id=com.pas.webcam>). Diakses 30 Agustus 2021.

Khoirunisa, Hana dan Fitrianingrum Kurniawati. 2019. Penggunaan *Drone* dalam Mengaplikasikan Pestisida di Daerah Sungai Besar, Malaysia. Skripsi, Bogor: Institut Pertanian Bogor.

Khumairowati dan Bektu Yulianti. 2020. Dalam *Jurnal Mitra Manajemen*, IX (1): 122-127.

MAD Components Store. 2021. AMPX ESC 30A. (*Online*). (<https://ozrobotics.com/shop/ampx-esc-30a/>). Diakses 13 September 2021.

Martinez, Kennedy. 2019. *Remote Control Drones [Updated Holidays 2019] – The Top Remote Control Quadcopters On The Market*. (*Online*). (<https://www.dronethusiast.com/remote-control-drones/>). Diakses 1 Februari 2021.

- Mogu. 2015. Pengertian Router, Fungsi Router, dan Cara Kerja Router. (Online). (<https://woocara.blogspot.com/2015/05/pengertian-router-fungsi-router-dan-cara-kerja-router.html>). Diakses 13 September 2021.
- Mostafa, Khan Muhammad Nafee dan Quadrat-E-Alahy Ratul. 2008. *Traffic Jam Detection System. CSE 3199 Report on Software Development Project-II*. Bangladesh: Khulna University of Engineering and Technology.
- Patel, Ashish. 2020. *What is Object Detection?* (Online). (<https://medium.com/ml-research-lab/what-is-object-detection-51f9d872ece7>). Diakses 13 September 2021.
- Pitowarno, Endra. 2006. *ROBOTIKA: Desain, Kontrol, dan Kecerdasan Buatan*. Yogyakarta : Andi Offset.
- Point of Beginning. 2019. *Multicopter Drone Flies for 45 Minutes*. (Online). (<https://www.pobonline.com/articles/101729-multicopter-drone-flies-for-45-minutes>). Diakses 13 September 2021.
- Prawiro, M. 2018. Pengertian Router: Fungsi, Cara Kerja, dan Jenis-Jenis Router (Online). (<https://www.maxmanroe.com/vid/teknologi/internet/pengertian-router.html>). Diakses 30 Agustus 2021.
- Price, Randy R. 2016. *How to Estimate The Maximum and Recommended Flight Times of a UAS, UAV, or Drone System*. (Online). (<https://www.lsuagcenter.com/NR/rdonlyres/24FEF5D5-FD37-4DF2-8323-974243633CC7/105242/3469Drones.pdf>). Diakses 22 Februari 2021.
- Ridlo, Muhammad. 2017. Wereng Coklat Mengamuk di Lahan 1.000 Hektare. (Online). (<https://www.liputan6.com/regional/read/3020180/wereng-cokelat-mengamuk-di-lahan-1000-hektare>). Diakses 6 Februari 2021.
- Robolink. 2021. *CoDrone Pro- A Programmable Drone That Was Designed To Teach You Programming*. (Online). (<https://ozrobotics.com/shop/codrone-pro-a-programmable-drone-that-was-designed-to-teach-you-programming/>). Diakses 13 September 2021.
- Santoso, Djarot Wahyu dan Kris Hariyanto. 2017. Pengembangan Sistem Penyemprotan Pada Platform Pesawat Tanpa Awak Berbasis *Quadcopter* untuk Membantu Petani Mengurangi Biaya Pertanian dalam Mendorong Konsep Pertanian Pintar (*Smart Farming*). Dalam *Jurnal Ilmiah Bidang Teknologi, ANGKASA*, IX (2): 49-56.
- Saputra, Dimas. 2018. *Skywalker-X8 UAV Mapping and Monitoring*. (Online). (<http://saputraptr.blogspot.com/2018/12/general.html>). Diakses 13 September 2021.

- Saxon Aerospace. 2019. *Saxon SRV Futura Mapping Drone*. (Online). (<https://ozrobotics.com/shop/saxon-m-6-mapping-drone/>). Diakses 13 September 2021.
- Setiawan, Alik. 2017. *Tutorial WebCam for OpenCV 2.410 and Visual Basic 2012*. (Online). (<https://aliketia012.blogspot.com/2017/03/tutorial-webcam-for-opencv-2410-and.html>). Diakses 22 Februari 2021.
- Shidiq, Maruf. 2018. *Raspberry Pi*. (Online). (<https://otomasi.sv.ugm.ac.id/2018/06/03/raspberry-pi/>). Diakses 13 September 2021.
- Sholahuddin, Muhammad Bahru. 2017. Analisa Perubahan Warna HSV pada Pengolahan Citra Terhadap Intensitas Cahaya Sebagai Dasar Penerapan Masukan Kontrol *Automatic Stacking Crane*. Skripsi. Surabaya: Institut Teknologi Sepuluh Nopember.
- Siregar, Ameilia Zuliyanti. 2007. Hama-hama Tanaman Padi. Medan: Fakultas Pertanian, Universitas Sumatera Utara.
- Swellpro. 2021. *Carbon Fiber Propeller (1 Pair)*. (Online). (<https://ozrobotics.com/shop/carbon-fiber-propeller-1-pair/>). Diakses 13 September 2021.
- Swellpro. 2021. *Splashdrone 3+ Top GPS Cover (with GPS module)*. (Online). ([https://ozrobotics.com/shop/splashdrone-3-top-gps-cover \(with-gps-module\) /](https://ozrobotics.com/shop/splashdrone-3-top-gps-cover-with-gps-module/)). Diakses 13 September 2021.
- Swellpro. 2021. *SwellPro®Remote Controller Battery (2S 2300mAh)*. (Online). (<https://ozrobotics.com/shop/swellproremote-controller-battery-2s-2300mah/>). Diakses pada 13 September 2021.
- Watanabe, T., dan H. Kitagawa. 2000. *Photosynthesis and Translocation of Assimilates in Rice Plants Following Phloem Feeding by The Planthopper Nilaparvata Lugens* (Homoptera: Delphacidae). *Jurnal Entomologi Indonesia*.
- Yulistiyanto, Andry. 2013. Pembuatan *Quadcopter* Sebagai Pemantau Area yang Dikendalikan Jarak Jauh dan Diakses Melalui *Web*. Semarang.
- Yuwono, Bambang. dkk. 2015. Pengembangan Model *Public Monitoring System* Menggunakan *Raspberry Pi*. Dalam *Jurnal Telematika*, XII (2): 123-133.

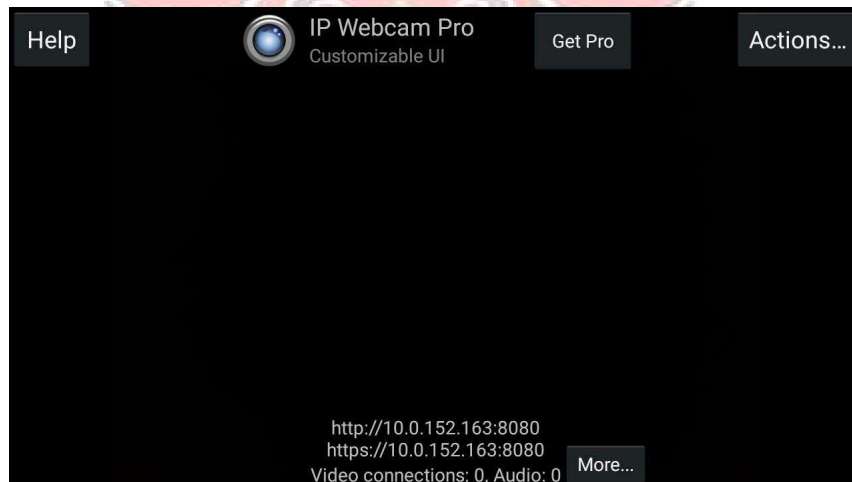


LAMPIRAN

Lampiran 1 Proses Pengoperasian Alat

Proses Pengoperasian *Drone*

- 1) Melakukan pemasangan *propeller* pada motor *brushless*.
- 2) Melakukan pengisian cairan penyemprotan ke tangki cairan.
- 3) Menghubungkan baterai dengan rangkaian *drone*.
- 4) Melakukan pemasangan *smartphone* ke *holder smartphone*
- 5) Menghubungkan *smartphone* dan *raspberry pi* menggunakan kabel micro USB dan melakukan penambatan/*tethering* USB.
- 6) Membuka aplikasi IP Webcam pada *smartphone* dan menjalankan fitur “*Start Server*” untuk memulai *video stream* dan untuk mendapatkan IP dari *smartphone*, yang nantinya akan digunakan untuk menghubungkan *kamera smartphone* dan program *image processing*.



Tampilan aplikasi IP Webcam setelah fitur *start server* dijalankan

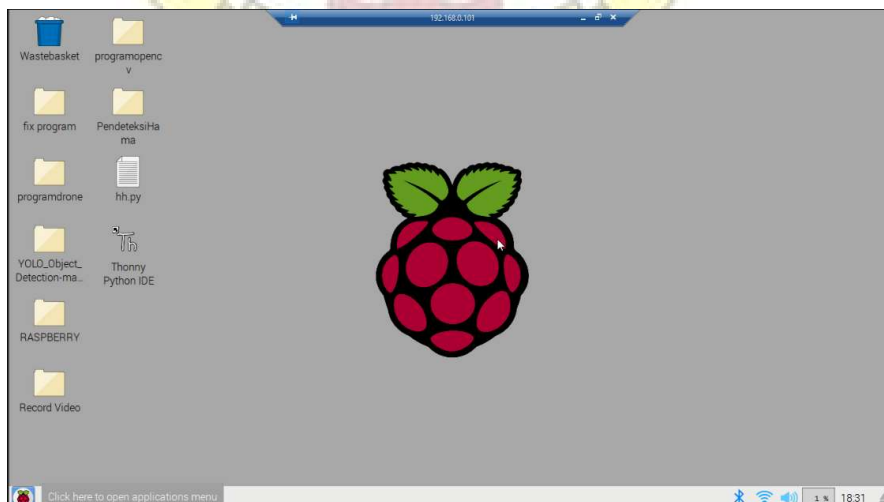
- 7) Mengaktifkan *radio controller*.

- 8) Membuka *software* Mission Planner pada laptop dan menghubungkannya dengan *flight controller* melalui telemetri, untuk mengatur *waypoint* yang hendak dilintasi oleh *drone* dan menerima data terbang *drone*.



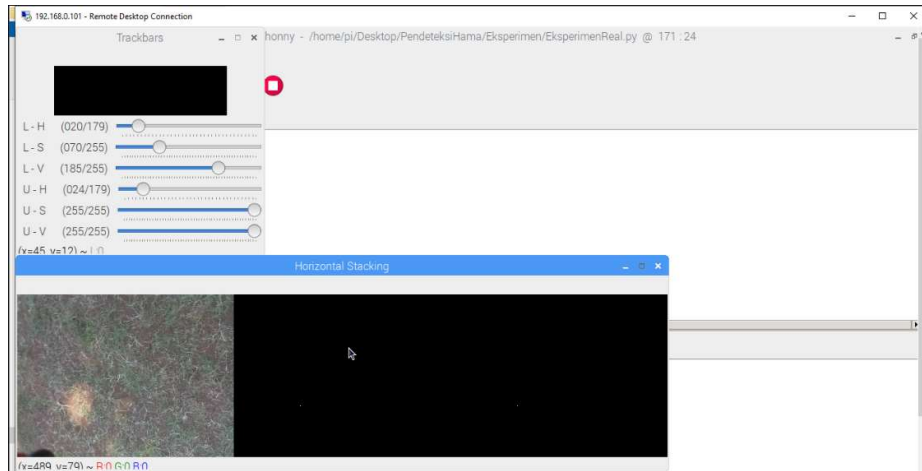
Tampilan *Software* Mission Planner setelah Mengatur *Waypoint*

- 9) Membuka *software* Remote Desktop Connection pada laptop kemudian memasukkan *IP address raspberry pi* untuk dapat mengontrol dan memonitoring *raspberry pi* secara *wireless*. Pastikan *raspberry pi* dan laptop terhubung di jaringan *Wi-Fi* yang sama.



Tampilan *Software* Remote Desktop Connection saat Terhubung ke *Raspberry Pi*

- 10) Membuka program “*EksperimenReal.py*” menggunakan *software* Thonny Python IDE, kemudian menjalankan program tersebut.



Program Deteksi Warna pada *Raspberry Pi*

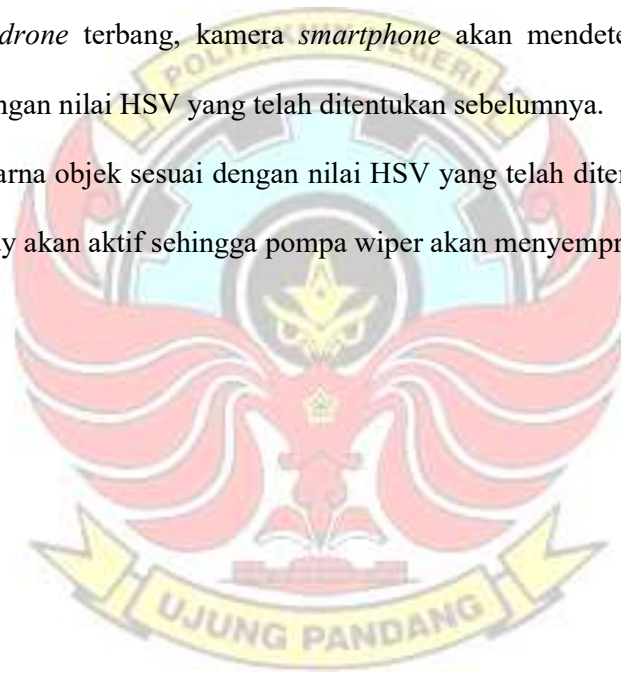
- 11) Selanjutnya melakukan *arming* motor untuk mengaktifkan motor *drone*, dengan cara menggerakkan tuas kiri dan kanan seperti pada Gambar berikut.



Arah Gerakan Tuas Kiri dan Kanan *Radio Controller*

- 12) Setelah itu *drone* siap diterbangkan, dengan mengerakkan tuas kiri ke atas (*throttle up*), maka *drone* akan terbang.

- 13) Pada saat ketinggian terbang *drone* sudah mencapai 1 meter. Maka pilot *drone* sudah dapat mengganti mode terbang *drone* dari manual (mode *stabilize*) menjadi mode *loiter*, dimana *drone* akan mempertahankan ketinggian dan posisinya saat terbang dengan memanfaatkan GPS.
- 14) Selanjutnya pilot *drone* sudah dapat mengganti mode terbang menjadi mode *auto*, dimana *drone* akan terbang melintasi *waypoint* yang telah dibuat sebelumnya.
- 15) Sembari *drone* terbang, kamera *smartphone* akan mendeteksi warna objek sesuai dengan nilai HSV yang telah ditentukan sebelumnya.
- 16) Ketika warna objek sesuai dengan nilai HSV yang telah ditentukan terdeteksi, maka relay akan aktif sehingga pompa wiper akan menyemprotkan cairan.



Lampiran 2 Dokumentasi Pengerjaan Alat



Pengecekan dan Pengujian Komponen *Drone* Lama



Pengerjaan Sistem Elektronik *Drone*



Proses Kalibrasi Motor *Brushless*



Proses Uji Coba Terbang *Drone*



Proses Eksperimen Real di Areal Persawahan

Lampiran 3 Spesifikasi Komponen

1. Flight Controller (Pixhawk)

Specifications:

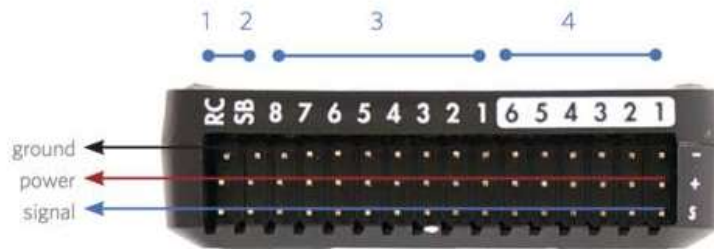
- Processor
 - 32-bit ARM Cortex M4 core with FPU
 - 168 Mhz/256 KB RAM/2 MB Flash
 - 32-bit failsafe co-processor
- Sensors
 - MPU6000 as main accel and gyro
 - ST Micro 16-bit gyroscope
 - ST Micro 14-bit accelerometer/compass (magnetometer)
 - MEAS barometer
- Power
 - Ideal diode controller with automatic failover
 - Servo rail high-power (7 V) and high-current ready
 - All peripheral outputs over-current protected, all inputs ESD protected
- Interfaces
 - 5x UART serial ports, 1 high-power capable, 2 with HW flow control
 - Spektrum DSM/DSM2/DSM-X Satellite input
 - Futaba S.BUS input (output not yet implemented)
 - PPM sum signal
 - RSSI (PWM or voltage) input
 - I2C, SPI, 2x CAN, USB
 - 3.3V and 6.6V ADC inputs
- Dimensions
 - Weight 38 g (1.3 oz)
 - Width 50 mm (2.0")
 - Height 15.5 mm (.6")
 - Length 81.5 mm (3.2")



- 1 Spektrum DSM receiver
- 2 Telemetry (on-screen display)
- 3 Telemetry (radio telemetry)
- 4 USB
- 5 SPI (serial peripheral interface) bus
- 6 Power module
- 7 Safety switch button
- 8 Buzzer
- 9 Serial
- 10 GPS module
- 11 CAN (controller area network) bus
- 12 I²C splitter or compass module
- 13 Analog to digital converter 6.6 V
- 14 Analog to digital converter 3.3 V
- 15 LED indicator



- 1 Input/output reset button
- 2 SD card
- 3 Flight management reset button
- 4 Micro-USB port



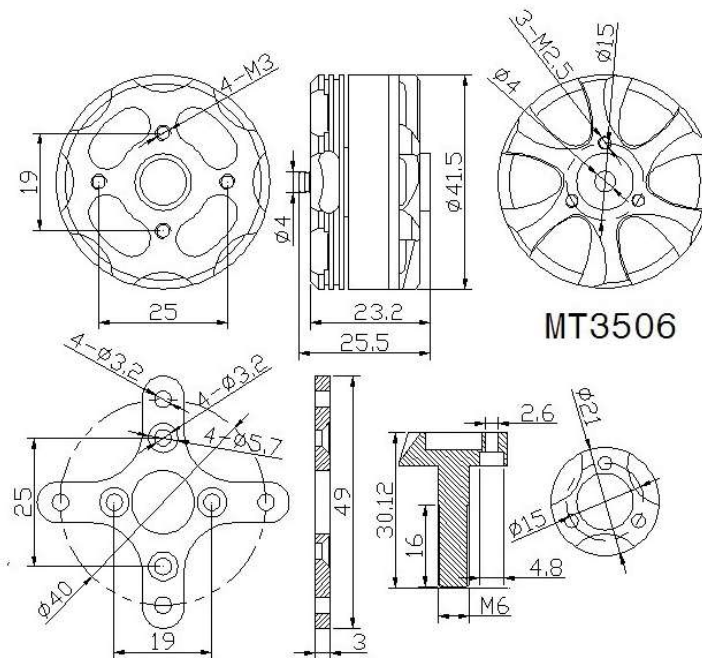
- 1 Radio control receiver input
- 2 SBus output
- 3 Main outputs
- 4 Auxiliary outputs

2. Motor Brushless dc (T-Motor MT3506-25 KV650)



Specification:

- KV : 650
- Configuration : 12N14P
- Stator Diameter : 35mm
- Stator Length : 6mm
- Shaft Diameter : 4.0mm
- Motor Dimensions (Diameter x Length) : $\Phi 41.5 \times 23.2$ mm
- Weight (g) : 70g
- No.of Cells(Lipo) : 3-4S
- Max Continuous current(A) : 14.5A
- Max Continuous Power(W) : 260W
- Internal Resistance : 155m Ω



Item No.	Volts (V)	Prop	Throttle	Amps (A)	Watts (W)	Thrust (g)	RPM	Efficiency (g/W)	Operating temperature(°C)
MT3506 KV650	11.1	T-MOTOR 11*3.7CF	50%	1.4	15	195	3500	13.00	42
			65%	2.6	28	285	4500	10.18	
			75%	3.6	40	380	5000	9.50	
			85%	5	54	472	5650	8.74	
			100%	5.9	65	540	6000	8.31	
		T-MOTOR 12*4CF	50%	1.6	18	230	3300	12.95	45
			65%	3.2	36	334	4200	9.40	
			75%	4.5	50	439	4800	8.79	
			85%	6.1	68	540	5350	7.98	
			100%	7.3	81	645	5700	7.96	
		T-MOTOR 13*4.4CF	50%	1.9	21	260	3200	12.33	47
			65%	3.5	39	400	4100	10.30	
	75%		5.3	59	520	4600	8.84		
	85%		7	78	636	5100	8.19		
	100%		8.3	92	723	5400	7.85		
	T-MOTOR 14*4.8CF	50%	2.5	28	340	2950	12.25	54	
		65%	5.1	57	499	3700	8.81		
		75%	7.1	79	626	4150	7.94		
		85%	9.3	103	736	4550	7.13		
		100%	11	122	848	4700	6.95		
	14.8	T-MOTOR 11*3.7CF	50%	2	30	290	4400	9.67	46
			65%	3.9	57	451	5600	7.91	
			75%	5.5	80	570	6400	7.13	
			85%	7.4	108	720	7100	6.67	
100%			8.7	127	800	7450	6.30		
T-MOTOR 12*4CF		50%	2.5	38	350	4100	9.21	52	
		65%	4.9	71	570	5300	8.03		
		75%	6.9	100	720	6000	7.20		
		85%	9.2	135	880	6600	6.52		
		100%	10.8	158	1000	6900	6.33		
T-MOTOR 13*4.4CF		50%	2.7	40	386	4000	9.66	60	
		65%	5.6	83	651	5100	7.85		
	75%	7.8	115	822	5800	7.12			
	85%	10.3	152	970	6300	6.36			
	100%	12	178	1053	6600	5.93			

Notes: The test condition of temperature is motor surface temperature in 100% throttle while the motor run 10 min.

3. Frame Drone (S500)



Specification:

- Motor Centers : 480mm
- Height : 170mm
- Weight : 405g (frame only)
- Motor Mount Bolt Holes : 16/19/25mm

4. Electronic Speed Control (ZTW Spider Series 2-6S 30A)



Specification:

- Length : 43mm / 1.69in
- Width : 25mm / 0.98in
- Height : 10mm / 0.39in
- Weight : 26g
- MAX Output : Continuous Current 30 A
- Burst Output : 40A
- Battery : 5-18NC/2-6Lipo
- BEC Output : No BEC
- Power Connector : Bare Leads
- Motor Connector : Bare Leads

5. GPS Compass (RadioLink SE100)



Specification:

- Height : 15.3mm
- Width : 48.5mm
- Weight : 34.9mm
- Antena : 300mm
- Connect to PIX : 4pin to I2C, 6Pin to GPS
- Accuracy : 50cm
- Positioning : 20 Satelites in 6 Second

6. Radio Controller (RC FLYSKY FS-I6)



Specification:

- Bandwidth (KHz) : 500.
- RF Range (GHz) : 2.40 ~ 2.48.
- No. of channels : 6.
- Remote controller weight (gm) : 400.
- Power : 6V (1.5V AA*4).
- Antenna Length : 26mm * 2 (dual antenna).
- Transmitting Power : ≤ 20 dBm.
- RF Receiver Sensitivity : -105dbm.

7. Battery Li-Po (Gens Ace 5000mah 4s 45c)



Specification:

- Battery Capacity : 5000mah
- Voltage : 14.8V / 4S / 4 Cell
- Continuous Discharge: 45C
- Burst Discharge : 90C (520A)
- Weight : 480g
- Dimensions : 154mm x 46mm x 30mm

8. Telemetri 433mhz (3DR Radio Telemetry 433MHZ Module)



Specification:

- Item Name : 3DR Radio Telemetry Module
- Band : 433MHz
- Antenna connectors : RP-SMA connector
- Output Power : 100mW (20dBm), adjustable between 1-20dBm
- Sensitivity : -117dBm sensitivity
- Interface : Standard TTL UART
- Connection status : LED indicators

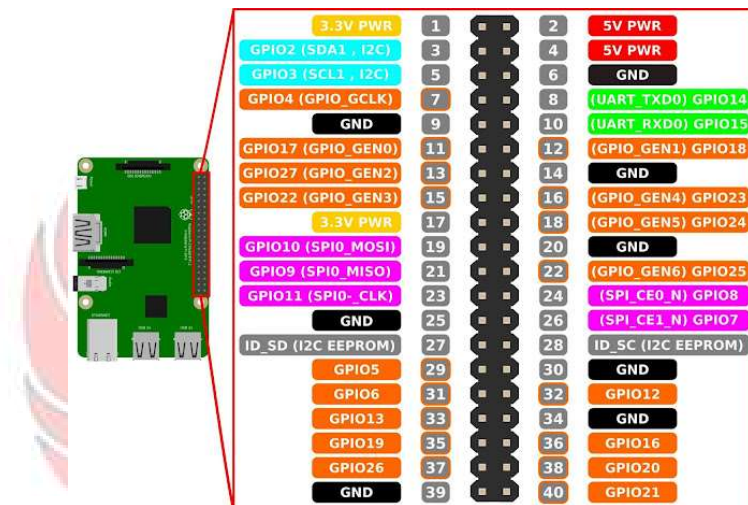
9. Raspberry Pi 3



Specification:

- Processor: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- Memory: 1GB LPDDR2 SDRAM
- Power Supply: 5V/2.5A dc power input (microUSB)
- Wireless: 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Ethernet: Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- GPIOs: Extended 40-pin GPIO header
- Video Output: Full-size HDMI

- Audio Output: 4-pole stereo output and composite video port
- USB Port: 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Power-over-Ethernet (PoE) support (requires separate PoE HAT)
- Operating Temperature: 0~50°C
- Dimension: 120mm x 75mm x 34mm
- Weight: 75g
- Raspberry Pi 3 GPIO Pin



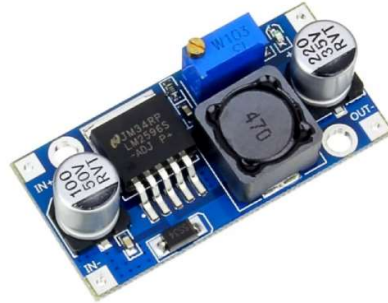
10. Pompa Wiper 12V



Specification:

- Voltage: 12 Volt DC
- Body plastik
- Warna hitam

11. Stepdown LM2596



Specification:

- Tegangan input : 3-40V DC
- Tegangan output : 1.25-35V DC
- Selisih input dan output : Minimal 1.5V DC
- Arus maksimal : 3A (Rekomendasi 2.5A untuk pemakaian jangka panjang)
- Efisiensi step down : 92%
- Ukuran : 43 x 24 x 14mm
- Operating Temperature : - 40 to + 85
- Range input tegangan (DC) : 3,2 - 40 V (tegangan input harus lebih tinggi daripada tegangan output lebih dari 1,5 V)
- Range output tegangan (DC) : 1,25 - 35 V tegangan yang dapat diatur, efisiensi tinggi (92%) arus output maksimum terbesar adalah 3A.

Lampiran 4 Listing Program pada Raspberry Pi

a) Program Mekanisme Penyemprotan

```
import RPi.GPIO as GPIO
import time

pinIO = 27
GPIO.setmode(GPIO.BCM)
GPIO.setup(pinIO, GPIO.OUT)
GPIO.output(pinIO, 1)
relayOn = 0

print("*****")
print("* PROGRAM EKSPERIMEN PENYEMPROTAN *")
print("*****")
pilihan = "y"
try:
    while (pilihan != "n"):
        durasi = int(input("Lama Penyemprotan (detik): "))
        print ("[Relay ON]", durasi, "detik")
        GPIO.output(pinIO, 0)
        time.sleep(durasi)
        GPIO.output(pinIO, 1)
        print ("[Relay OFF]")
        pilihan = input("Apakah Anda masih ingin melakukan
eksperimen penyemprotan (y/n)? ")
        print()
except KeyboardInterrupt:
    GPIO.output(pinIO, 1)
```

b) Program Perekam Video

```
import numpy as np
import os
import cv2
import time

timestr = time.strftime("%Y%m%d_%H%M%S_%s")
b = 'REC_'+timestr+'.avi'
filename = '../HasilRekaman/'+b
frames_per_second = 12
res = '480p'

def change_res(cap, width, height):
    cap.set(3, width)
    cap.set(4, height)

STD_DIMENSIONS = {
    "360p": (480, 360),
    "480p": (640, 480),
    "720p": (1280, 720),
    "1080p": (1920, 1080),
    "4k": (3840, 2160),
```

```

}

def get_dims(cap, res='1080p'):
    width, height = STD_DIMENSIONS["360p"]
    if res in STD_DIMENSIONS:
        width,height = STD_DIMENSIONS[res]
        ## change the current capture device to the resulting
        resolution
        change_res(cap, width, height)
        return width, height

VIDEO_TYPE = {
    'avi': cv2.VideoWriter_fourcc(*'XVID')
}

def get_video_type(filename):
    filename, ext = os.path.splitext(filename)
    if ext in VIDEO_TYPE:
        return VIDEO_TYPE[ext]
    return VIDEO_TYPE['avi']

cap = cv2.VideoCapture('http://192.168.42.129:8080/video')
out = cv2.VideoWriter(filename, get_video_type(filename),
frames_per_second, get_dims(cap, res))
frames_per_second, get_dims(cap, res))

try:
    while True:
        ret, frame = cap.read()
        display = cv2.resize(frame, (426,240))
        gray_display = cv2.cvtColor(display,
cv2.COLOR_BGR2GRAY)
        out.write(frame)
        cv2.imshow('frame',gray_display)
        key = cv2.waitKey(1)
        if key == ord('q'):
            break

except KeyboardInterrupt:
    pass

print("SELESAI MEREKAM")
cap.release()
out.release()
cv2.destroyAllWindows()

```

c) Program Pemutar Video

```

import numpy as np
import os
import cv2
import datetime

def nothing(x):
    pass

```

```

#Select Video
file = 'REC_20210829_113246_1630207966'
cap = cv2.VideoCapture('../HasilRekaman/'+file+'.avi')
#ReadLastValueHSV
loadHSV = open("NilaiHSV/NilaiHSV.txt", "r")
nilaiHSV = loadHSV.read().split("\n")
l_h = int(nilaiHSV[0])
l_s = int(nilaiHSV[1])
l_v = int(nilaiHSV[2])
u_h = int(nilaiHSV[3])
u_s = int(nilaiHSV[4])
u_v = int(nilaiHSV[5])
loadHSV.close()

#TrackBars
cv2.namedWindow("Trackbars")
cv2.createTrackbar("L - H", "Trackbars", l_h, 179, nothing)
cv2.createTrackbar("L - S", "Trackbars", l_s, 255, nothing)
cv2.createTrackbar("L - V", "Trackbars", l_v, 255, nothing)
cv2.createTrackbar("U - H", "Trackbars", u_h, 179, nothing)
cv2.createTrackbar("U - S", "Trackbars", u_s, 255, nothing)
cv2.createTrackbar("U - V", "Trackbars", u_v, 255, nothing)
status = 'play'

try:
    while True:
        key = cv2.waitKey(33)

        if status == 'play':
            ret, frame = cap.read()
            if key == ord('2'):
                status = 'pause'
                print("[Status] Pause Video")

            if status == 'pause':
                if key == ord('1'):
                    status = 'play'
                    print("[Status] Play Video")

            if not ret:
                break
            frame = cv2.resize(frame, (320,240))

            hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
            l_h = cv2.getTrackbarPos("L - H", "Trackbars")
            l_s = cv2.getTrackbarPos("L - S", "Trackbars")
            l_v = cv2.getTrackbarPos("L - V", "Trackbars")
            u_h = cv2.getTrackbarPos("U - H", "Trackbars")
            u_s = cv2.getTrackbarPos("U - S", "Trackbars")
            u_v = cv2.getTrackbarPos("U - V", "Trackbars")

            nilaiHSV = "\n".join([str(l_h), str(l_s), str(l_v),
str(u_h), str(u_s), str(u_v)])
            saveHSV = open('NilaiHSV/NilaiHSV.txt','w')
            saveHSV.write(nilaiHSV)
            saveHSV.close()

```

```

lower_blue = np.array([l_h, l_s, l_v])
upper_blue = np.array([u_h, u_s, u_v])
mask = cv2.inRange(hsv, lower_blue, upper_blue)
result = cv2.bitwise_and(frame, frame, mask=mask)

_, cnts, _ = cv2.findContours(mask, cv2.RETR_TREE,
cv2.CHAIN_APPROX_NONE)

for contour in cnts:
    area = cv2.contourArea(contour)
    if area > 1000:
        try:
            rect = cv2.minAreaRect(contour)
            box = cv2.boxPoints(rect)
            box = np.int0(box)
            cv2.drawContours(result, [box], -
1, (255,0,0), 3)
        except Exception as e:
            print(e)

mask = cv2.cvtColor(mask, cv2.COLOR_GRAY2BGR)
hStack = np.hstack([frame, mask, result])
spasi = " "
cv2.imshow("Rekaman Asli"+spasi+" | "+spasi+"Masking
HSV"+spasi+" | "+spasi+"Hasil Filter HSV", hStack)

#Save - Load Nilai HSV (Profil 1)
if key == ord('3'):
    nilaiHSV_profil = "\n".join([str(l_h), str(l_s),
str(l_v), str(u_h), str(u_s), str(u_v)])
    saveHSV_profil =
open('NilaiHSV/NilaiHSV_Profil1.txt', 'w')
    saveHSV_profil.write(nilaiHSV)
    saveHSV_profil.close()
    print("[Status] Berhasil Menyimpan Nilai HSV di
Profil 1")

if key == ord('7'):
    loadHSV = open("NilaiHSV/NilaiHSV_Profil1.txt",
"r")

    nilaiHSV = loadHSV.read().split("\n")
    l_h = int(nilaiHSV[0])
    l_s = int(nilaiHSV[1])
    l_v = int(nilaiHSV[2])
    u_h = int(nilaiHSV[3])
    u_s = int(nilaiHSV[4])
    u_v = int(nilaiHSV[5])
    loadHSV.close()
    cv2.setTrackbarPos("L - H", "Trackbars", l_h)
    cv2.setTrackbarPos("L - S", "Trackbars", l_s)
    cv2.setTrackbarPos("L - V", "Trackbars", l_v)
    cv2.setTrackbarPos("U - H", "Trackbars", u_h)
    cv2.setTrackbarPos("U - S", "Trackbars", u_s)
    cv2.setTrackbarPos("U - V", "Trackbars", u_v)
    print("[Status] Berhasil Memuat Nilai HSV di Profil
1")

```



```

#Save - Load Nilai HSV (Profil 2)
if key == ord('4'):
    nilaiHSV_profil = "\n".join([str(l_h), str(l_s),
str(l_v), str(u_h), str(u_s), str(u_v)])
    saveHSV_profil =
open('NilaiHSV/NilaiHSV_Profil2.txt','w')
    saveHSV_profil.write(nilaiHSV)
    saveHSV_profil.close()
    print("[Status] Berhasil Menyimpan Nilai HSV di
Profil 2")

if key == ord('8'):
    loadHSV = open("NilaiHSV/NilaiHSV_Profil2.txt",
"r")

    nilaiHSV = loadHSV.read().split("\n")
    l_h = int(nilaiHSV[0])
    l_s = int(nilaiHSV[1])
    l_v = int(nilaiHSV[2])
    u_h = int(nilaiHSV[3])
    u_s = int(nilaiHSV[4])
    u_v = int(nilaiHSV[5])
    loadHSV.close()
    cv2.setTrackbarPos("L - H", "Trackbars", l_h)
    cv2.setTrackbarPos("L - S", "Trackbars", l_s)
    cv2.setTrackbarPos("L - V", "Trackbars", l_v)
    cv2.setTrackbarPos("U - H", "Trackbars", u_h)
    cv2.setTrackbarPos("U - S", "Trackbars", u_s)
    cv2.setTrackbarPos("U - V", "Trackbars", u_v)
    print("[Status] Berhasil Memuat Nilai HSV di Profil
2")

#Save - Load Nilai HSV (Profil 3)
if key == ord('5'):
    nilaiHSV_profil = "\n".join([str(l_h), str(l_s),
str(l_v), str(u_h), str(u_s), str(u_v)])
    saveHSV_profil =
open('NilaiHSV/NilaiHSV_Profil3.txt','w')
    saveHSV_profil.write(nilaiHSV)
    saveHSV_profil.close()
    print("[Status] Berhasil Menyimpan Nilai HSV di
Profil 3")

if key == ord('9'):
    loadHSV = open("NilaiHSV/NilaiHSV_Profil3.txt",
"r")

    nilaiHSV = loadHSV.read().split("\n")
    l_h = int(nilaiHSV[0])
    l_s = int(nilaiHSV[1])
    l_v = int(nilaiHSV[2])
    u_h = int(nilaiHSV[3])
    u_s = int(nilaiHSV[4])
    u_v = int(nilaiHSV[5])
    loadHSV.close()
    cv2.setTrackbarPos("L - H", "Trackbars", l_h)
    cv2.setTrackbarPos("L - S", "Trackbars", l_s)

```

```

cv2.setTrackbarPos("L - V", "Trackbars", l_v)
cv2.setTrackbarPos("U - H", "Trackbars", u_h)
cv2.setTrackbarPos("U - S", "Trackbars", u_s)
cv2.setTrackbarPos("U - V", "Trackbars", u_v)
print("[Status] Berhasil Memuat Nilai HSV di Profil
3")

#Save - Load Nilai HSV (Profil 4)
if key == ord('6'):
    nilaiHSV_profil = "\n".join([str(l_h), str(l_s),
str(l_v), str(u_h), str(u_s), str(u_v)])
    saveHSV_profil =
open('NilaiHSV/NilaiHSV_Profil4.txt','w')
    saveHSV_profil.write(nilaiHSV)
    saveHSV_profil.close()
    print("[Status] Berhasil Menyimpan Nilai HSV di
Profil 4")

if key == ord('0'):
    loadHSV = open("NilaiHSV/NilaiHSV_Profil4.txt",
"r")
    nilaiHSV = loadHSV.read().split("\n")
    l_h = int(nilaiHSV[0])
    l_s = int(nilaiHSV[1])
    l_v = int(nilaiHSV[2])
    u_h = int(nilaiHSV[3])
    u_s = int(nilaiHSV[4])
    u_v = int(nilaiHSV[5])
    loadHSV.close()
    cv2.setTrackbarPos("L - H", "Trackbars", l_h)
    cv2.setTrackbarPos("L - S", "Trackbars", l_s)
    cv2.setTrackbarPos("L - V", "Trackbars", l_v)
    cv2.setTrackbarPos("U - H", "Trackbars", u_h)
    cv2.setTrackbarPos("U - S", "Trackbars", u_s)
    cv2.setTrackbarPos("U - V", "Trackbars", u_v)
    print("[Status] Berhasil Memuat Nilai HSV di Profil
4")

if 0xFF == ord('q'):
    break

except KeyboardInterrupt:
    pass

print("[Status] Stop Video")
cap.release()
cv2.destroyAllWindows()

```

d) Program Algoritma *Image Processing* dengan Mengaktifkan Mekanisme Penyemprotan

```

import cv2
import numpy as np
import time
import RPi.GPIO as GPIO

```

```

pinIO = 27
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(pinIO, GPIO.OUT)
GPIO.output(pinIO, 1)
relayOn = 0

def nothing(x):
    pass

cap =
cv2.VideoCapture('http://192.168.42.129:8080/video')
#ReadLastValueHSV
loadHSV = open("NilaiHSV/NilaiHSV.txt", "r")
nilaiHSV = loadHSV.read().split("\n")
l_h = int(nilaiHSV[0])
l_s = int(nilaiHSV[1])
l_v = int(nilaiHSV[2])
u_h = int(nilaiHSV[3])
u_s = int(nilaiHSV[4])
u_v = int(nilaiHSV[5])
loadHSV.close()

cv2.namedWindow("Trackbars")
cv2.createTrackbar("L - H", "Trackbars", l_h, 179,
nothing)
cv2.createTrackbar("L - S", "Trackbars", l_s, 255,
nothing)
cv2.createTrackbar("L - V", "Trackbars", l_v, 255,
nothing)
cv2.createTrackbar("U - H", "Trackbars", u_h, 179,
nothing)
cv2.createTrackbar("U - S", "Trackbars", u_s, 255,
nothing)
cv2.createTrackbar("U - V", "Trackbars", u_v, 255,
nothing)
try:
    while True:
        key = cv2.waitKey(33)
        _, frame = cap.read()
        frame = cv2.resize(frame, (320,240))
        frame = cv2.flip(frame,90)
        hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
        l_h = cv2.getTrackbarPos("L - H", "Trackbars")
        l_s = cv2.getTrackbarPos("L - S", "Trackbars")
        l_v = cv2.getTrackbarPos("L - V", "Trackbars")
        u_h = cv2.getTrackbarPos("U - H", "Trackbars")
        u_s = cv2.getTrackbarPos("U - S", "Trackbars")
        u_v = cv2.getTrackbarPos("U - V", "Trackbars")
        lower_blue = np.array([l_h, l_s, l_v])

```

```

upper_blue = np.array([u_h, u_s, u_v])
mask = cv2.inRange(hsv, lower_blue, upper_blue)
result = cv2.bitwise_and(frame, frame, mask=mask)

_, cnts, _ = cv2.findContours(mask,
cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)

for contour in cnts:
    area = cv2.contourArea(contour)
    if area > 1000:
        try:
            rect = cv2.minAreaRect(contour)
            box = cv2.boxPoints(rect)
            box = np.int0(box)
            cv2.drawContours(frame, [box], -
1, (255,0,0), 3)
            if (relayStatus == 1):
                GPIO.output(pinIO, 0)
                print('[Status] Relay ON')
                relayOn = time.time()
            except Exception as e:
                print(e)

relayStatus = GPIO.input(pinIO)
relayOff = time.time()
relayTime = relayOff - relayOn
if (relayStatus == 0):
    if ((relayOff - relayOn) > 3):
        GPIO.output(pinIO, 1)
        print('[Status] Relay OFF')
mask = cv2.cvtColor(mask, cv2.COLOR_GRAY2BGR)
hStack = np.hstack([frame,mask,result])

cv2.imshow("Horizontal Stacking", hStack)

#Load Nilai HSV (Profil 1)
if key == ord('7'):
    loadHSV =
open("NilaiHSV/NilaiHSV_Profil1.txt", "r")
    nilaiHSV = loadHSV.read().split("\n")
    l_h = int(nilaiHSV[0])
    l_s = int(nilaiHSV[1])
    l_v = int(nilaiHSV[2])
    u_h = int(nilaiHSV[3])
    u_s = int(nilaiHSV[4])
    u_v = int(nilaiHSV[5])
    loadHSV.close()
    cv2.setTrackbarPos("L - H", "Trackbars", l_h)
    cv2.setTrackbarPos("L - S", "Trackbars", l_s)
    cv2.setTrackbarPos("L - V", "Trackbars", l_v)
    cv2.setTrackbarPos("U - H", "Trackbars", u_h)

```

```
        cv2.setTrackbarPos("U - S", "Trackbars", u_s)
        cv2.setTrackbarPos("U - V", "Trackbars", u_v)
        print("[Status] Berhasil Memuat Nilai HSV di
Profil 1")
```

```
        #Load Nilai HSV (Profil 2)
        if key == ord('8'):
            loadHSV =
open("NilaiHSV/NilaiHSV_Profil2.txt", "r")
            nilaiHSV = loadHSV.read().split("\n")
            l_h = int(nilaiHSV[0])
            l_s = int(nilaiHSV[1])
            l_v = int(nilaiHSV[2])
            u_h = int(nilaiHSV[3])
            u_s = int(nilaiHSV[4])
            u_v = int(nilaiHSV[5])
            loadHSV.close()
            cv2.setTrackbarPos("L - H", "Trackbars", l_h)
            cv2.setTrackbarPos("L - S", "Trackbars", l_s)
            cv2.setTrackbarPos("L - V", "Trackbars", l_v)
            cv2.setTrackbarPos("U - H", "Trackbars", u_h)
            cv2.setTrackbarPos("U - S", "Trackbars", u_s)
            cv2.setTrackbarPos("U - V", "Trackbars", u_v)
            print("[Status] Berhasil Memuat Nilai HSV di
Profil 2")
```

```
        #Load Nilai HSV (Profil 3)
        if key == ord('9'):
            loadHSV =
open("NilaiHSV/NilaiHSV_Profil3.txt", "r")
            nilaiHSV = loadHSV.read().split("\n")
            l_h = int(nilaiHSV[0])
            l_s = int(nilaiHSV[1])
            l_v = int(nilaiHSV[2])
            u_h = int(nilaiHSV[3])
            u_s = int(nilaiHSV[4])
            u_v = int(nilaiHSV[5])
            loadHSV.close()
            cv2.setTrackbarPos("L - H", "Trackbars", l_h)
            cv2.setTrackbarPos("L - S", "Trackbars", l_s)
            cv2.setTrackbarPos("L - V", "Trackbars", l_v)
            cv2.setTrackbarPos("U - H", "Trackbars", u_h)
            cv2.setTrackbarPos("U - S", "Trackbars", u_s)
            cv2.setTrackbarPos("U - V", "Trackbars", u_v)
            print("[Status] Berhasil Memuat Nilai HSV di
Profil 3")
```

```
        #Load Nilai HSV (Profil 4)
        if key == ord('0'):
            loadHSV =
open("NilaiHSV/NilaiHSV_Profil4.txt", "r")
```

```

nilaiHSV = loadHSV.read().split("\n")
l_h = int(nilaiHSV[0])
l_s = int(nilaiHSV[1])
l_v = int(nilaiHSV[2])
u_h = int(nilaiHSV[3])
u_s = int(nilaiHSV[4])
u_v = int(nilaiHSV[5])
loadHSV.close()
cv2.setTrackbarPos("L - H", "Trackbars", l_h)
cv2.setTrackbarPos("L - S", "Trackbars", l_s)
cv2.setTrackbarPos("L - V", "Trackbars", l_v)
cv2.setTrackbarPos("U - H", "Trackbars", u_h)
cv2.setTrackbarPos("U - S", "Trackbars", u_s)
cv2.setTrackbarPos("U - V", "Trackbars", u_v)
print("[Status] Berhasil Memuat Nilai HSV di
Profil 4")

    if key == ord('q'):
        break

except KeyboardInterrupt:
    GPIO.output(pinIO, 1)
    pass

print('[Status] Menutup Program')
cap.release()
cv2.destroyAllWindows()

```



Lampiran 5 Biodata Penulis

Biodata Penulis Skripsi “*Drone untuk Deteksi Hama dan Penyemprotan Pestisida pada Tanaman Padi*”



Fachturrahman. Lahir di Makassar pada tanggal 23 April 1999 dari ayah Abdul Kadir dan ibu Asriani. Penulis adalah anak ketiga dari empat bersaudara. Tahun 2011 penulis menyelesaikan Pendidikan Sekolah Dasar di SDN 60 Inpres Perumnas Tumalia, Maros. Pada tahun itu juga penulis melanjutkan Pendidikan di SMPN 2 Unggulan Maros dan tamat pada tahun 2014, kemudian melanjutkan Sekolah Menengah Kejuruan di SMK Telkom Makassar, Jurusan Teknik Komputer dan Jaringan dan lulus pada tahun 2017. Pada tahun 2017 penulis diterima sebagai mahasiswa program studi Teknik Mekatronika, Jurusan Teknik Mesin di Politeknik Negeri Ujung Pandang. Penulis pernah mengikuti Praktek Kerja Lapangan di PT. PLN (Persero) Unit Layanan PLTG Tello, Kota Makassar, Provinsi Sulawesi Selatan pada tahun 2020.



Hutomo Febri Richardo Sumbung. Lahir di Makale pada tanggal 06 Februari 2000 dari ayah Tato'Sumbung dan ibu Paulina. Penulis adalah anak Kedua dari dua bersaudara. Tahun 2011 penulis menyelesaikan Pendidikan Sekolah Dasar di SD Kristen 1 Makale. Pada tahun itu juga penulis melanjutkan Pendidikan di SMP Katolik Makale dan tamat pada tahun 2014 kemudian melanjutkan Sekolah Menengah Atas di SMA Negeri 1 Makale dengan mengambil jurusan IPA dan lulus pada tahun 2017. Pada tahun 2017 penulis diterima sebagai mahasiswa program studi Teknik Mekatronika Jurusan Teknik Mesin di Politeknik Negeri Ujung Pandang. Penulis pernah mengikuti Praktek Kerja Lapangan di PT. PLN (Persero) Unit Layanan PLTG Tello, Kota Makassar, Provinsi Sulawesi Selatan pada tahun 2020.

Lampiran 6 Lembar Asistensi Skripsi Tugas Akhir



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

POLITEKNIK NEGERI UJUNG PANDANG

Jln. Perintis Kemerdekaan Km 10, Makassar 90245

☎ (0411) 585365, 585367, 585368 Fax (586043)

E-mail: pnup@poliupg.ac.id

Home page: <http://www.poliupg.ac.id>

LEMBAR ASISTENSI SKRIPSI TUGAS AKHIR

Nama Mahasiswa : 1. Fachturrahman (444 17 017)

2. Hutomo Febri Richardo Sumbung (444 17 022)

Judul Skripsi : **DRONE UNTUK DETEKSI HAMA DAN PENYEMPROTAN
PESTISIDA PADA TANAMAN PADI**

No	Hari/Tanggal	Uraian Revisi	Paraf
1	Kamis/18 Maret 2021	Memperhatikan komponen yang ingin dibeli, dikarenakan penelitian bersifat berkelanjutan	Ahmad
2	Kamis/29 April 2021	Fokus menerbangkan <i>drone</i> terlebih dahulu	Ahmad
3	Kamis/27 Mei 2021	Dimaksimalkan pengerjaan tugas akhirnya	Ahmad
4	Kamis/10 Juni 2021	Memaksimalkan penggunaan dana untuk pengadaan alat dan bahan	Ahmad
5	Kamis/10 Juni 2021	Berkoordinasi dengan pembimbing saat ingin melakukan eksperimen	Ahmad
6	Kamis/8 Juli 2021	Memaksimalkan eksperimen di bulan Juli	Ahmad
7	Kamis/22 Juli 2021	Dalam mengerjakan alat, harus lebih memperhatikan SOP	Ahmad
8	Rabu/27 Juli 2021	Ketinggian terbang drone pada eksperimen mekanisme penyemprotan disesuaikan dengan jarak rencana saat eksperimen real di sawah.	Ahmad

Makassar, | September 2021

Pembimbing I,

Dr.Eng. Akhmad Taufik, S.T., M.T.
NIP. 19760413 200812 1 003



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI UJUNG PANDANG
Jln. Perintis Kemerdekaan Km 10, Makassar 90245
☎ (0411) 585365, 585367, 585368 Fax (586043)
E-mail: pnup@poliupg.ac.id
Home page: <http://www.poliupg.ac.id>

LEMBAR ASISTENSI SKRIPSI TUGAS AKHIR

Nama Mahasiswa : 1. Fachturrahman (444 17 017)
2. Hutomo Febri Richardo Sumbung (444 17 022)

Judul Skripsi : **DRONE UNTUK DETEKSI HAMA DAN PENYEMPROTAN PESTISIDA PADA TANAMAN PADI**

No	Hari/Tanggal	Uraian Revisi	Paraf
1	Senin/9 Agustus 2021	Bercerita data pendukung untuk mencapai tujuan penelitian	
2	Senin/9 Agustus 2021	Masukkan skenario cara mendapatkan datanya	
3	Senin/9 Agustus 2021	Masukkan hasil dan analisisnya dihubungkan dengan tujuan penelitian	
4	Minggu/29 Agustus 2021	Melakukan perbaikan diagram blok	
5	Rabu/1 September 2021	Melakukan penulisan skripsi sesuai dengan pedoman penulisan	
6	Rabu/1 September 2021	Menjelaskan hubungan data-data yang diperoleh	
7	Rabu/1 September 2021	Memindahkan data yang kurang penting	
8	Rabu/1 September 2021	Memindahkan bagian perancangan sistem dari bab 4 ke bab 3	

Makassar, September 2021

Pembimbing II

Imran Habria Syah, S.ST., M.T.
NIP. 19881005 201903 1 009





Lampiran 7 Catatan Revisi Skripsi Tugas Akhir

LAMPIRAN BERITA ACARA PELAKSANAAN
UJIAN SIDANG SKRIPSI

Nama Mahasiswa : Fachrurrahman / Hutomo Febri Richardo Sumbang

NIM : 444 17 017 / 444 17 022

Catatan/Daftar Revisi Penguji:

No.	Nama	Uraian	Tanda Tangan
1	P' Kadir	- Periksa cara pengoperasian, pengontrolan dan sistem pembacaan gambar lama.	
2	P' Mukhtar	- Tambahkan konl citra warna coblat yang asli kemudian di bandingkan dengan data perbandingan pengujian.	
3	P' Lewi	- Melengkapi data hal 54	
4	P' Remi	- Koreksi ada pada skripsi	
	S. P. R.		

Makassar, 7 September 2021
Sekretaris Penguji



Ir. Remigius Tandioaga, M.Eng.Sc.
NIP. 19621210 199003 1 005

Catatan: Jika ada perubahan Judul Tugas Akhir konfirmasi secepatnya ke bagian Akademik.