

RANCANG BANGUN UNMANNED AERIAL VEHICLE BERBASIS IMAGE PROCESSING UNTUK ESTIMASI HASIL PANEN PADA LAHAN PERKEBUNAN

Muhammad Giriarda Abrari

Dr. Aris Budiarto, S.T., M.T. . M.T. , Siti Aminah, S.T., M.T.

Jurusan Teknik Mesin Dan Manufaktur
Politeknik Manufaktur Negeri Bandung
Jalan Kanayakan No. 21 Bandung, Indonesia

ABSTRAK

Beberapa tahun kebelakang ini perkembangan teknologi pertanian khususnya di Indonesia telah berkembang menuju pertanian yang lebih modern, cerdas, dan presisi. Salah satu teknologi yang digunakan adalah pengaplikasian Unmanned Aerial Vehicle (UAV) atau sering juga disebut drones sebagai alat untuk membantu manusia dalam mengolah lahan pertanian. Berdasarkan penelitian sebelumnya yang mengestimasi produktivitas lahan dengan menggunakan metode perhitungan luasan lahan, hasil perhitungan tidak akurat karena terdapatnya tanaman yang tidak tumbuh pada beberapa bagian lahan namun tetap terhitung. Perhitungan estimasi produktivitas lahan dengan menggunakan UAV dapat menghasilkan hasil perhitungan yang lebih akurat. Penelitian ini merancang dan membangun sebuah UAV yang dapat digunakan untuk melihat kondisi lahan dari angkasa berbasis image processing, data berupa gambar yang didapat digunakan untuk mengestimasi hasil panen pada suatu rumpun tanaman. Perangkat UAV ini menerapkan sistem pergerakan otomatis berbasis image processing. UAV akan memposisikan diri pada lahan atau rumpun yang terdapat tanda berwarna kemudian user dapat mengambil gambar lahan tersebut dari udara melalui antarmuka yang tersedia. Gambar lahan yang telah diambil diproses dengan metode filterasi dan object counting untuk menghitung jumlah tanaman pada gambar tersebut. Hasil dari penelitian ini adalah sistem otomatis yang dapat berfungsi sesuai dengan rancangan sistem dan kesimpulan bahwa metode object counting pada penelitian ini tidak efektif dengan error perhitungan paling kecil 33.69% terhadap perhitungan tanaman secara manual.

Kata kunci— pertanian, unmanned aerial vehicle, image processing

1. PENDAHULUAN

1.1 Latar Belakang

Indonesia adalah negara agraris, data dan informasi lahan pertanian merupakan elemen penting untuk pemantauan di bidang pertanian, seperti pola tanam dan kalender tanam, agar perencanaan pengelolaan pertanian dapat dilakukan tepat sasaran dan bijaksana (Shofiyanti, 2011). Oleh karena itu, diperlukan pelayanan informasi data sumber daya lahan yang cepat dan akurat.

Teknologi penginderaan jauh (inderaja) dapat digunakan untuk mendapatkan data spasial dengan cepat dan akurat, sehingga mampu menjawab kebutuhan informasi sumber daya lahan (Shofiyanti, 2011). Aplikasi teknologi inderaja dapat dimanfaatkan untuk memprediksi luas area panen dan

produktivitasnya, sehingga jika dikaitkan dengan kebutuhan pangan nasional, tingkat ketersediaan sembako dapat diprediksi tiap musim panen nya. Data hasil identifikasi dan pemantauan lahan dapat digunakan sebagai dasar menyusun arahan pengembangan komoditas tanaman.

Metodologi inderaja banyak menggunakan data potret udara yang dihasilkan oleh satelit optik, yang saat ini masih terkendala oleh awan, terutama pada saat musim hujan. Disamping itu penggunaan satelit membutuhkan biaya besar serta pengadaan data yang lambat (Kushardono, 2014).

Salah satu teknologi alternatif untuk inderaja adalah *unmanned aerial vehicle* (UAV) atau pesawat tanpa awak. Pesawat tanpa awak dapat mengambil data lebih detail, *real*

time, cepat, dan lebih murah (Shofiyanti, 2011). Teknologi UAV telah dikembangkan dan banyak digunakan untuk mengidentifikasi dan memonitor kondisi sumber daya lahan pertanian dan pertumbuhan tanaman pada lahan tersebut.

Penelitian ini ingin mengimplementasikan UAV sebagai alat untuk menghitung jumlah tanaman yang ada pada suatu lahan dan memproses data tersebut lebih lanjut untuk mengestimasi hasil panen pada lahan tersebut.

1.1 Perumusan Masalah

Secara garis besar, tahapan yang perlu dilakukan UAV sebelum menghitung jumlah tanaman adalah mendeteksi adanya suatu lahan, memposisikan UAV tepat berada di atas lahan, dan kemudian mengambil gambar dari lahan tersebut. Dibutuhkan suatu metode untuk dapat mengenali suatu lahan dan kemudian mengambil gambar dari lahan tersebut dengan mempertimbangkan kualitas gambar dan kepresisian data yang dipengaruhi oleh jarak kamera dengan subjek yang nantinya akan dipengaruhi oleh cakupan luas lahan (Prasad et al. 1992). Diperlukan suatu algoritma untuk UAV dapat mempertahankan posisi tepat di atas lahan yang akan diambil gambarnya.

1.2 Batasan Masalah

Agar penelitian ini lebih terarah dan memudahkan dalam pembahasan sehingga tujuan penelitian akan tercapai. Beberapa batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Lahan pertanian yang digunakan adalah perkebunan sayur yang memiliki satu rumpun tanaman yang memiliki perlakuan pemeliharaan yang sama tiap rumpun nya.
2. Lahan pertanian yang digunakan harus memiliki jarak antar lahan yang terlihat dari penglihatan udara dan galengan yang memiliki perbedaan warna yang signifikan dengan tanaman.
3. Daerah pengoperasian dilakukan di perkebunan sayur Kp, Maleber, Desa Ciharang, Kec. Pacet, Kab. Cianjur.
4. UAV menggunakan konfigurasi *Quad-copter*.
5. UAV menggunakan *flight-controller* yang tersedia di pasaran, sehingga penelitian ini tidak membuat kendali pergerakan dasar dari UAV tersebut.

6. Pengendalian pergerakan UAV secara otomatis, dibatasi hanya mengambil alih kendali maneuver mengangguk (*Pitch*) dan berguling (*Roll*) pada kendali dasar yang telah ada.
7. UAV dioperasikan pada kondisi angin kurang dari 1.54 m/s

1.3 Tujuan

Tujuan penulisan karya tulis penelitian ini sebagai berikut:

1. Sebagai solusi untuk menjawab kebutuhan informasi lahan pertanian yang cepat dan akurat.
2. Untuk membangun sistem otomatis pada UAV dalam pengambilan keputusan pergerakan.
3. Untuk memprediksi produktivitas pada suatu lahan pertanian dengan sebuah UAV yang memiliki pengendalian secara otomatis.

1.4 Kegunaan

Kegunaan penulisan karya tulis penelitian ini sebagai berikut:

- a. Bagi masyarakat:
 1. Memberi informasi pemanfaatan teknologi inderaja untuk lahan pertanian.
 2. Memudahkan masyarakat dalam merencanakan pengelolaan lahan pertanian.
- b. Bagi pemerintah:
 1. Mensosialisasikan salah satu teknologi pertanian pada masyarakat.
 2. Dapat menjadi alternatif yang baik untuk memonitor tingkat ketersediaan pangan nasional.

2. TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka dari Penelitian Terdahulu

Beberapa penelitian mengenai pengendalian UAV dengan berdasarkan visual atau *image processing* telah dilakukan, diantaranya penelitian pada (Venugopalan et al., 2012) yang mengaplikasikan UAV untuk memonitoring kondisi laut. Peneliti tersebut membuat sistem kendali otomatis pada UAV *quad-copter* untuk dapat mendarat pada kapal tanpa awak menggunakan informasi visual dari kamera yang ditempatkan di bawah UAV.

Penelitian lainnya yaitu pada (Warren et al.,2015), penulis mencari solusi dari permasalahan UAV yang menggunakan *stereo visual odometry* pada jarak jauh. Warren et al., membuktikan bahwa navigasi dengan menggunakan visual mengurangi ketergantungan pada GPS dan satelit navigasi lainnya, juga meningkatkan ketahanan sistem navigasi pada aplikasi penerbangan rendah (< 120m) di tengah keberadaan *jamming*, *spoofing*, dan *long dropouts*.

Pada (Sotomayor et al.,2014) mengaplikasikan UAV berbasis kendali visual untuk menginspeksi lahan pertanian. UAV terbang mengikuti garis-garis tekstur pada lahan dengan cara terbang parallel dengan lahan pertanian, sudut yang didapat dari sistem visual menjadi patokan untuk mengarahkan pergerakan UAV.

Penelitian lain yang menjadi inspirasi adalah pada (Putri A., 2014) yang mengaplikasikan drone dengan kendali manual untuk menghitung produktivitas lahan padi dengan metode luasan lahan. Metode ini akan dibandingkan dengan metode yang diproposalkan pada penelitian ini.

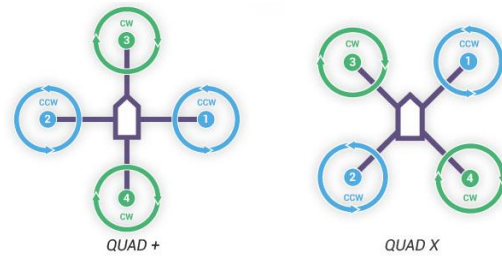
2.2 Unmanned Aerial Vehicle (UAV)

UAV merupakan pesawat sistem tanpa awak (*Unmanned System*), yaitu sistem berbasis elektro-mekanik yang dapat melakukan misi-misi terprogram, dengan karakteristik: (i) tanpa awak pesawat, (ii) beroperasi pada mode mandiri baik secara penuh atau sebagian, (iii) Sistem ini dirancang untuk dapat dipergunakan secara berulang (Department of Defence, 2007, dalam Shofiyanti, 2011).



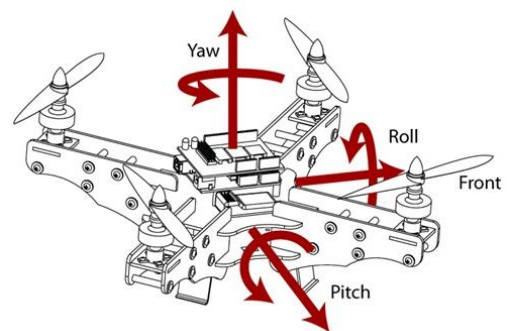
Gambar 2. 1 UAV jenis *quadcopter* sedang terbang membawa kamera (sumber: open-electronics.org, 2011)

Pada penelitian ini, konfigurasi UAV yang digunakan adalah berjenis *quadcopter*. UAV berjenis *quadcopter* termasuk jenis *helicopter* yang mempunyai empat rotor yang arah putarannya saling menyilang. Dua rotor berputar searah jarum jam sedangkan dua rotor lagi berputar berkebalikan arah jarum jam.



Gambar 2. 2 Arah rotasi baling-baling motor (rotor) pada *quadcopter* (sumber: 3DRobotics)

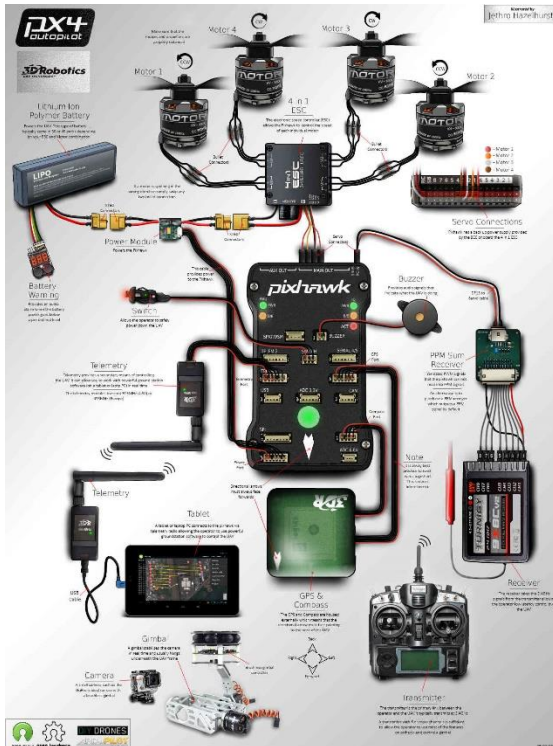
Pergerakan *quadcopter* dikontrol dengan mengatur kecepatan dari masing-masing rotor. Variasi kecepatan rotor ini menghasilkan pergerakan membelok, miring, dan berputar ke semua arah pada bidang referensi.



Gambar 2. 3 Mekanika terbang *quadcopter* (sumber: Salaskar, Prathamesh, et al. 2014)

- *Yaw* mengubah kecepatan rotor yang arah putarnya sama
- *Pitch* mengubah kecepatan rotor di bagian depan atau belakang
- *Roll* mengubah kecepatan rotor di bagian samping bodi

Komponen utama pada *quadcopter* dibagi menjadi dua bagian yaitu, komponen elektrik dan mekanik. Komponen mekanik secara sederhana adalah kerangka *quadcopter* dan aktuator berupa *brushless motor* dengan baling-balingnya. Kerangka *quadcopter* merupakan tempat terpasangnya keempat rotor dan komponen-komponen elektrik yang berhubungan dengan aktuator secara umum dapat digambarkan pada Gambar 2.4.



Gambar 2. 4 Hubungan antar komponen pada quadcopter (sumber: 3DRobotics)

2.3 Image Processing

Image Processing adalah suatu metode yang digunakan untuk memproses atau memanipulasi gambar dalam bentuk 2 dimensi (Ahmad, 2005, dalam Jatmika, 2014). *Image processing* dilakukan untuk memperbaiki kesalahan data sinyal gambar yang terjadi akibat transmisi dan selama akuisisi sinyal, serta untuk meningkatkan kualitas penampakan gambar agar lebih mudah diinterpretasi oleh sistem penglihatan manusia baik dengan melakukan manipulasi dan juga penganalisisan terhadap gambar. *Image processing* dapat juga dikatakan sebagai segala operasi untuk memperbaiki, menganalisa, atau mengubah suatu gambar (Jatmika, 2014).

2.4 Citra Digital

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek (Sutoyo, 2009, dalam Jatmika, 2014). Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optic berupa foto, bersifat analog berupa sinyal – sinyal video seperti gambar pada monitor televisi, atau bersifat

digital yang dapat langsung disimpan pada suatu media penyimpanan. Citra dapat dikelompokkan menjadi dua bagian yaitu citra diam yaitu citra tunggal yang tidak bergerak dan citra bergerak yaitu rangkaian citra diam yang ditampilkan secara beruntun, sehingga memberi kesan pada mata sebagai gambar yang bergerak. Setiap citra didalam rangkaian itu disebut frame. Gambar – gambar yang tampak pada film layar lebar atau televisi yaitu terdiri dari ribuan sampai ratusan ribu frame (Jatmika, 2014).

Telah banyak penelitian untuk UAV dapat mendeteksi objek darat dengan metode pengambilan citra adapaun beberapa metodenya adalah: Camera Image Processing, IR-LED sensor, Laser+Retoreflectors, Kinect Radar, GPS, dan lain-lain. Dari beberapa metode yang telah dilakukan pada penelitian sebelumnya, pendeteksian objek berbasis kamera menjadi metode yang paling mudah dan murah dengan akurasi yang cukup yang menjadikan metode Camera Image Processing dipilih pada penelitian ini (Venugopalan et al.,2012).

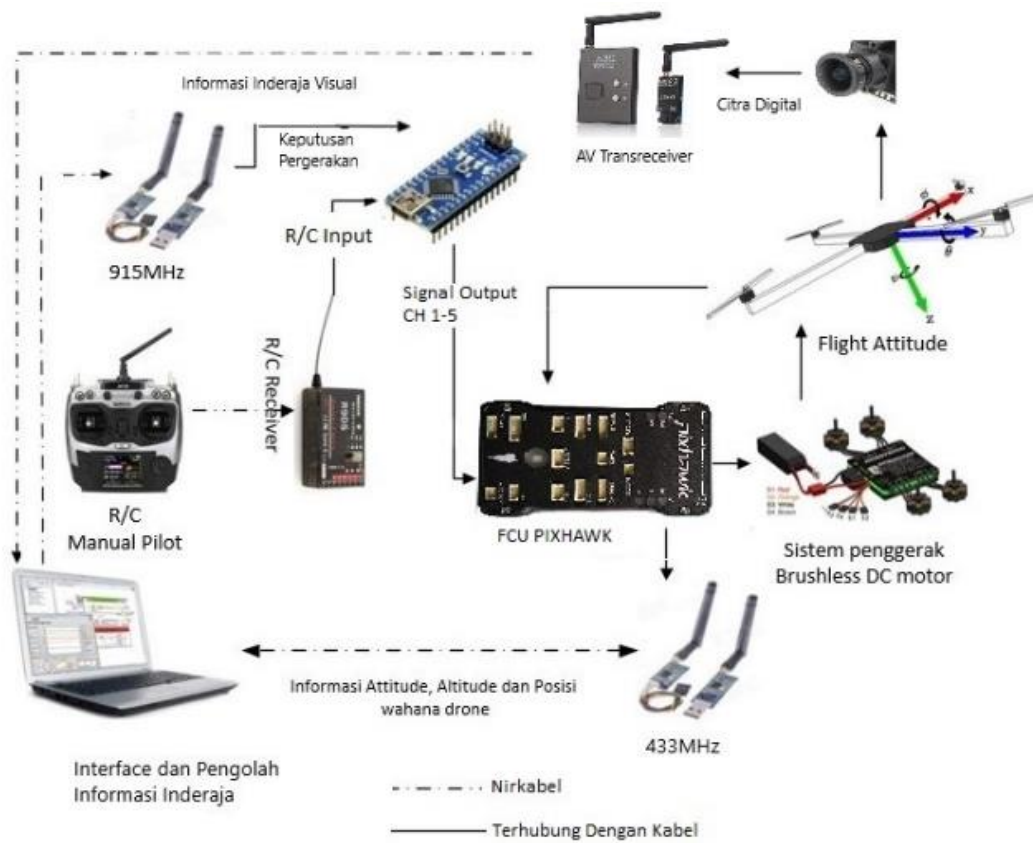
Terinspirasi penelitian pada Venugopalan et al.,2012, penelitian ini menggunakan MATLAB sebagai pengolah citra. Pada MATLAB, sebuah gambar akan disimpan sebagai sebuah matriks $m \times n \times 3$, di mana m adalah jumlah *pixel* dalam sumbu x , n adalah jumlah *pixel* pada sumbu y , dan 3 adalah jumlah *layer* (R, G dan B) (Sanjaya, M.,2016).

Berbeda dengan penelitian yang dilakukan Putri, A dan Sumiharto R. pada (Putri, A, 2014) yang menggunakan metode luasan lahan, metode yang digunakan pada penelitian ini adalah metode *object counting*. Metode *object counting* merupakan proses menghitung objek berdasarkan konektivitasnya terhadap *pixel* di sekitarnya.

3 PERANCANGAN SISTEM

3.1 Gambaran umum sistem

Gambaran umum sistem UAV yang menerapkan GPS dan *image processing* untuk estimasi hasil panen pada lahan pertanian diperlihatkan pada Gambar 3.1.



Gambar 3. 1 Gambaran umum sistem

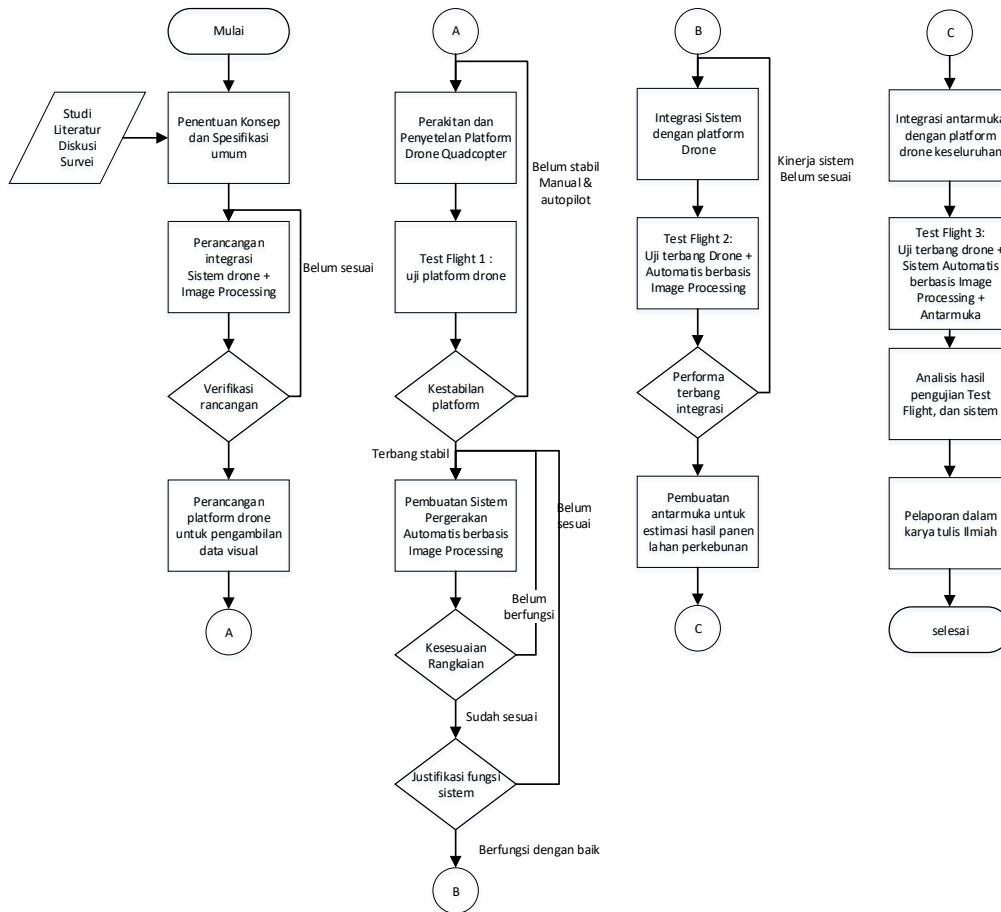
Unsur kendali melibatkan pemrograman pada mikrokontroler dimana pada UAV terdapat dua pengendali yakni *Flight Controller unit* (FCU) dan pergerakan otomatis berbasis *image processing*. Program dan perangkat dari FCU menggunakan dari yang sudah ada. Oleh karena itu wahana *Drone* tanpa kendali otomatis berbasis *image processing* merupakan sebuah ‘platform’ sehingga pemrograman kendali otomatis bukan pada FCU namun pada perangkat Arduino dan komputer/laptop sebagai pemroses gambar.

User dapat mengontrol pergerakan wahana *Drone* dengan menggunakan jalur komunikasi R/C dan *interface* pada komputer. Kedua jalur komunikasi pergerakan tersebut diterima oleh Arduino yang terpasang pada *Drone* dan diprogram untuk dapat meng-interrupt komunikasi satu dengan yang lain. Komunikasi pergerakan dibagi menjadi dua yaitu, auto dan manual. Pergerakan Auto tersedia pada *interface* komputer dengan

berdasarkan informasi visual yang didapat dari kamera pada wahana *Drone*, sedangkan pergerakan manual tersedia pada komputer dan juga R/C.

3.2 Alur Proses pengerjaan

Secara garis besar, proses penyelesaian Tugas Akhir secara menyeluruh dijelaskan dalam diagram alir pada Gambar 3.2. Diawali dengan penentuan konsep perancangan arsitektur sistem baik dalam penentuan aspek mekanik, aspek elektrik maupun perancangan program. Kemudian aspek-aspek tersebut diintegrasikan supaya menjadi satu sistem utuh. Lalu selanjutnya baik sistem maupun sub-sistem tersebut diuji untuk mendapatkan data yang kemudian diolah. Data data tersebut diperoleh dari pengujian baik pengujian darat sistem maupun pengujian terbang *quad-copter* (*test-flight*). Hasil pengujian berupa data yang telah diolah kemudian dipelajari dan dianalisis untuk memperoleh jawaban atas rumusan masalah.

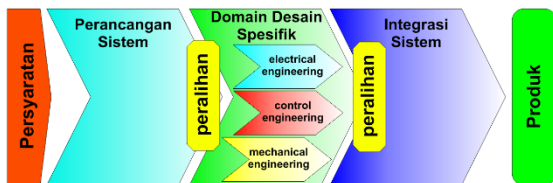


Gambar 3. 2 Diagram Alir Penyelesaian Tugas Akhir

3.3 Studi Pertimbangan Rancang Bangun

3.3.1 Penerapan Standar Perancangan model V dari VDI 2206

Dalam tugas akhir ini, model V dalam VDI 2206 digunakan sebagai acuan untuk membangun prototip sistem. model ini mencakup langkah penerapan yang diilustrasikan pada Gambar 3.3



Gambar 3. 3 Gambaran umum standar VDI 2206

Dalam model ini, terdapat 3 tahap pengerjaan berupa perancangan sistem, Desain spesifik, dan Integrasi Sistem. Sebelum dieksekusi pada pengerjaan, berdasarkan subproses tersebut dibuat daftar tuntutan (DRO – *Design Requirement & Objective*) sebagai acuan dan batasan pengerjaan.

3.3.2 Tuntutan umum

DRO digunakan untuk memperjelas batasan-batasan yang perlu diperhatikan serta mempermudah tahapan yang harus dilakukan. Adapun kriteria tuntutan yang dibuat berdasarkan sistem yang digunakan dan ekspektasi fungsi dari alat yang dirancang pada Tabel 3.1, DRO secara umum sesuai dengan model V dijelaskan.

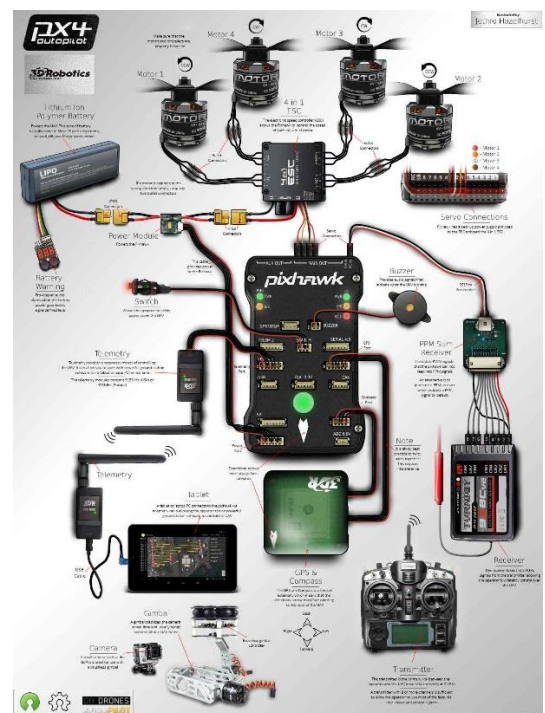
Tabel 3. 1 Daftar Tuntutan umum

Tujuan	Tuntutan Utama	Tuntutan tambahan
Wahana <i>Drone Quad-Copter</i>	<ul style="list-style-type: none"> Mampu Mengangkut bobot wahana beserta beban perangkat pengendali <i>avoidance</i> dengan Thrust-to-Weight 3:1 Mampu melaksanakan mode otomatis dasar sesuai <i>firmware</i> yang <i>existing</i>: Mengunci ketinggian. Dapat menerima input kendali baik dari pilot maupun pengendali otomatis. Mampu terbang hingga minimal 10 menit di udara 	<ul style="list-style-type: none"> Konstruksi dan perangkat mekanik sederhana dan mudah di-<i>maintenance</i>. Konstruksi Kokoh Menggunakan struktur yang umum digunakan oleh <i>user</i>.. Komponen rakitan mudah ditemukan di pasaran Menggunakan kontroler <i>drone</i> yang <i>existing</i> di masyarakat
Perangkat Sistem Otomatis berbasis Image Processing	<ul style="list-style-type: none"> Dapat mendeteksi penanda lahan. Dapat mengambil alih kendali pilot manual. 	<ul style="list-style-type: none"> Algoritma yang sederhana dan cepat diproses oleh pengendali Menggunakan perangkat yang mudah ditemukan di pasaran Perangkat mudah diintegrasikan pada <i>Drone</i>
Program Antarmuka Pengguna	<ul style="list-style-type: none"> Pemantauan informasi visual secara nir-kabel. Dapat memproses informasi lahan berupa jumlah populasi tanaman untuk kemudian di estimasi produktivitasnya. 	<ul style="list-style-type: none"> Antarmuka yang mudah dipahami dan digunakan oleh pengguna

Daftar tuntutan umum diatas menjadi acuan dalam memilih secara detail periferil komponen dan modul sistem yang digunakan. Tuntutan secara spesifik harus dilakukan dengan pertimbangan rekomendasi dari komunitas wahana *Drone* dan ketersediaan sumber daya.

3.4 Perancangan Sistem

Sistem *drone quadcopter* yang menerapkan sistem auto pilot berbasis *image processing* secara dasar memiliki unsur mekanik, elektrik dan kendali untuk menjadi sebuah wahana terbang tanpa awak. Unsur kendali melibatkan pemrograman pada mikrokontroler dimana pada wahana sendiri terdapat dua pengendali yakni *Flight Controller unit (FCU)* dan *controller* untuk sistem otomatis berbasis *image processing*. *Drone Quad-copter* sebagai *platform* sendiri memiliki periferil perangkat keras yang direkomendasikan dari FCU merk Pixhawk yang ditunjukkan pada Gambar 3.4.

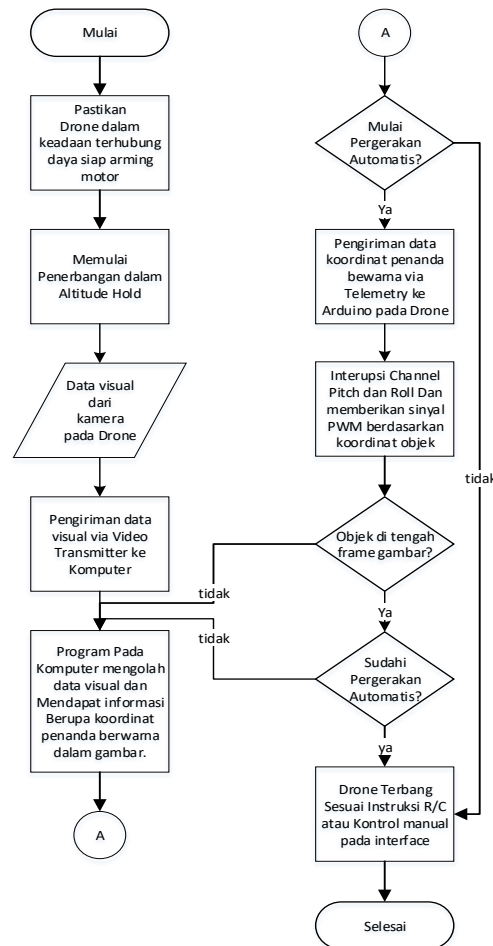


Gambar 3. 4 Gambaran platform Drone

Dari gambaran *platform drone* sistem pada Gambar 3.4, dapat dibuat arsitektur sistem yang melibatkan perangkat dan antar hubungannya dalam sistem *drone*. Arsitektur konsep sistem yang dibuat berdasarkan gambaran umum tersebut diperlihatkan pada Gambar 3.1.

Diagram pada Gambar 3.1 menunjukkan pada *platform drone* terdapat tambahan sistem

otomatis berbasis *image processing* yang bertugas menginterupsi keluaran sinyal PWM dari input R/C. Adapun sinyal yang diinterupsi tersebut adalah untuk manuver *Roll, pitch, Yaw, Throttle*, dan *Mode*. Sistem otomatis berbasis *image processing* ini dapat bekerja pada mode *Altitude Holding*. Alur sistem *drone* dengan kemampuan ini berdasarkan Gambar 3.1 dapat dijelaskan pada diagram alir Gambar 3.5.



Gambar 3.5 Diagram alir kerja drone

Dari diagram alir Gambar 3.5 menjelaskan kerja interupsi yang dimulai ketika *drone* sedang terbang. Program pada sistem pergerakan otomatis berbasis *image processing* mengolah data visual. Kemudian diputuskan pergerakan *drone* yang harus diambil alih *roll* dan *pitch*-nya secara otomatis dari kontrol manual. Jika sistem pergerakan otomatis belum dijalankan atau objek belum terdeteksi, maka kontroler (Arduino) akan meneruskan sinyal perintah *pwm* dari user langsung ke input pada FCU.

3.5 Domain Desain Spesifik

3.5.1 Pemilihan Perangkat Inti Drone

Pemilihan perangkat ini mengacu pada rekomendasi komunitas wahana *Drone*, rekomendasi pabrikan, dan tabel pemilihan Motor untuk *Drone* tipe *Multicopter* yang dibuat oleh Alf dari komunitas Flite Test. Penulis mengacu pada waktu terbang, jenis *Drone(frame)* yang digunakan dan estimasi beban yang akan dibawa.

Tabel 3. 2 Tabel Pemilihan Motor

Motor calculator

Fill in 'Inputs' below

Setup		
# rotor	4	
A	1,6	
<i>Use calculated AUW</i>		
Mass	2	kg
Battery		
Voltage	11,1	V
#	1	
Capacity	6200	mAh
Rating	14	C
Propeller		
Diameter	13	in
Pitch	4	in
Thrust-to-Weight		
<i>Desired Thrust-to-Weight (usually 3)</i>		
Desired	3	
Req Tot T	6	kg
Req T Per	1,5	kg

Max RPM calculated using Prop D

Propeller Correction Factors		
Prop	α	Max RPM
Aeront E	0,93	TBC
APC E	1,3	11154
APC SF	1,9	5000
APC MR	1,6	8077
Graupner E	1,45	10385
MA-K	1,3	12692
MA Scim	0,95	12692
Zinger	1,1	TBC

Required Motor		
Required	6535	RPM
Required KV	589	KV
Efficiency	80	%
Req Nom KV	706	KV
Req Power	204	W
Req ESC	22	A

Estimator		
100% Throttle		
Required	6535	RPM
P-out	170	W
P-in	204	W
A-motor	18,4	A
A-tot	73,4	A
Hover		
RPM	4133	RPM
P-out	43	W
P-in	52	W
A-motor	4,6	A
A-tot	18,6	A
Endurance		
Hover	16,0	Min
Average	10,0	Min
100% Throt	4,1	Min

Tabel 3. 3 Komponen dan spesifikasi platform Drone yang dipilih

Frame	Tarot Iron Man 650
Propeller	RCTimer Carbon Fibre 13x4
Motor	Tarot 2814 700KV
ESC	HobbyWing 30A
Baterai	Li-Po 11.1 6200 mAh
Pengendali	Pixhawk
R/C	FlySky FS-i6 6 Channel

Model Iron Man 650 memiliki ruang yang cukup untuk membawa kamera beserta gimbal dengan panjang motor ke motor 650mm, model ini dapat dipasang propeller yang panjangnya hingga 17inch yang dapat disesuaikan untuk kebutuhan membawa beban yang berat. Konstruksi Iron Man 650 yang modular ini membuat wahana mudah untuk *maintenance*. Material dari Iron Man 650 menggunakan Toray 3K carbon fibre yang membuat frame kokoh.

3.5.2 Pemilihan Perangkat Sistem Pergerakan Otomatis Berbasis Image Processing

Pemilihan perangkat sistem berdasarkan observasi, kemampuan integrasi dengan wahana *Drone*, dan ketersediannya di pasaran. Dengan *ground clearance drone* 180mm ruang yang tersedia pada bagian bawah *Drone* cukup untuk sebuah gimbal dan sebuah kamera aksi dengan bobot kurang dari 500gr agar tidak melebihi perhitungan pemilihan motor.

Gimbal yang digunakan adalah jenis 3-axis. Berdasarkan analisis penulis, gimbal 3-axis lebih minim getaran dan goyangan dibandingkan

jenis 2-axis atau tidak memakai gimbal sama sekali. dengan menggunakan gimbal 3-axis, gambar yang dihasilkan menjadi lebih baik untuk hasil pemrosesan gambar yang lebih akurat.

Kamera yang digunakan merupakan kamera aksi dengan fitur pengiriman gambar via usb agar kompatibel dengan video transmitter yang ada di pasaran, kamera juga kompatibel dengan gimbal yang tersedia di pasaran.




Ada beberapa variasi Video Receiver yang ada di pasaran, namun yang penulis akan gunakan adalah yang mempunyai jalur komunikasi langsung via usb, sehingga *receiver* dapat langsung terhubung ke komputer melalui sambungan USB.

Tabel 3. 4 Komponen dan spesifikasi sistem pergerakan otomatis berbasis image processing yang dipilih

Kamera	YI Action Camera
Gimbal	3-axis Strom32 BGC
Video Transmitter	Eachine TX805
Video Receiver	Eachine ROTG01
Kontroler	Arduino Nano

Arduino Nano dipilih berdasarkan jumlah I/O yang dibutuhkan dan sumber listrik yang tersedia pada wahana *Drone* yaitu 12v dan 5v. Nano mampu menerima sinyal PWM pada tiap I/O nya dengan fitur PulseIn. Dan dapat berkomunikasi dengan komputer (ground) secara serial menggunakan Software Serial. Berikut spesifikasi Arduino Nano dibandingkan dengan mikrokontroler sekelasnya.

Tabel 3. 5 Perbandingan perangkat mikrokontroler

Type	Arduino Nano 328P	Arduino Uno	Arduino Mega2560
Tampilan			
Prosesor	ATmega328	ATmega328	ATmega2560
EEPROM	1kB	1kB	4 kB
Analog I/O	8/0	6/0 (DIP)	16/0
Digital I/O	14	14	54
Dimensi (mm)	45 x 18 x 1.5	68.6 x 53.4 x 1.5	101.5 x 53.4 x 1.5

Dimensi menjadi pertimbangan dalam pemilihan mikrokontroler dikarenakan keterbatasan ruang pada badan drone.

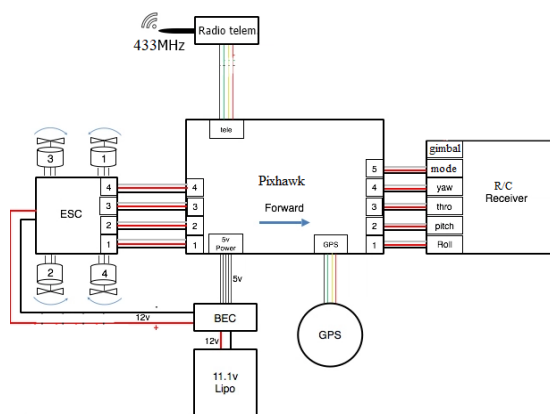
MATLAB dipilih sebagai antarmuka dan pemroses gambar dikarenakan jika dibandingkan dengan *software* pengolah data sekelasnya seperti LabView, MATLAB memiliki *library* yang memiliki fungsi dengan level lebih tinggi yang dibutuhkan dalam pemrosesan gambar. Antarmuka yang dapat dibuat pada MATLAB memang terbatas namun pada penelitian ini tidak dibutuhkan antarmuka dengan fungsi yang kompleks.

3.5.3 Wiring Diagram

Secara fisik, terdapat dua sistem pengendali pada wahana *drone* yang memiliki kemampuan pergerakan otomatis berbasis *image processing*. Tiap sistem memiliki periferan komponennya sendiri. Setiap komponen dalam periferan terhubung secara fisik melalui *wiring* sesuai dengan skema.

1. Wiring Periferan FCU Pixhawk

Drone sebagai *Platform* yang telah ada untuk sistem keseluruhan memiliki periferan tersendiri dimana Pixhawk sebagai pusat pengendali *Drone*. Selain built-in sensors, Periferan komponen pada *Drone* ini terdiri dari unit *powerplant* (sumber tenaga dan penggerak), Sensor GPS, dan Komunikasi Nirkabel (Telemetri dan R/C).



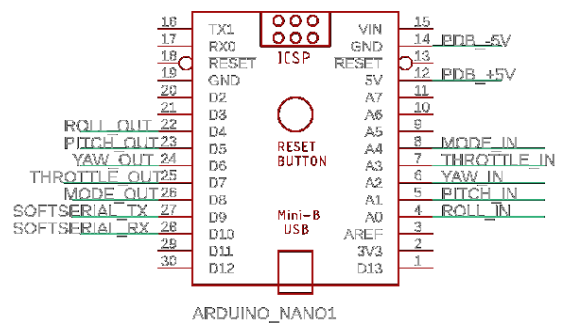
Gambar 3. 6 Periferan Wiring Diagram pada platform drone

Pada Gambar 3.6, manajemen sumber tenaga terdiri dari Baterai Li-Po 11.1V yang tersambung ke Sensor Baterai dan PCB Distribusi Daya untuk memberi tenaga listrik pada sistem propulsi (Motor + ESC). Sensor Baterai yang ditenagai baterai juga

menyalurkan listrik 5V untuk menyalakan perangkat elektronik lain.

2. Wiring Periferan perangkat kendali pergerakan otomatis berbasis *image processing*

Pada Kontroler periferan ini yaitu Arduino Nano, terpasang perangkat komunikasi nirkabel. Adapun perangkat periferan ini menerima input PWM dari perangkat *Receiver* R/C dan mengeluarkan sinyal PWM kepada FCU. Berikut merupakan *wiring diagram* dari Arduino ke perangkat yang digunakan pada Gambar 3.7.

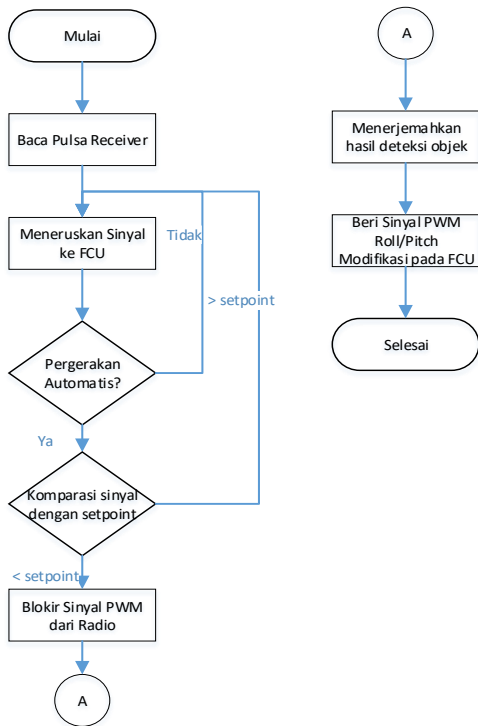


Gambar 3. 7 Sambungan perangkat kontroler Arduino

3.5.4 Desain Kendali

Unit kendali otomatis berbasis *image processing* secara perangkat menggunakan mikrokontroler Arduino Nano. *Drone platform* yang dimodifikasi dengan perangkat kendali otomatis berbasis *image processing* bekerja berdasarkan hasil pemrosesan data visual yang didapat dari *Drone*. Untuk mengolah hasil deteksi objek secara visual menjadi pergerakan *Drone* secara otomatis, diperlukan perancangan informatika untuk memprogram kontroler.

Pemrograman pada unit Arduino dilakukan dengan menggunakan Arduino IDE. Pada Program ini terdiri dari pembacaan input R/C, penerjemahan hasil deteksi objek yang dikirim melalui *Telemetry* dari pengolah gambar (MATLAB/Ground), meneruskan perintah kendali manual dari antarmuka serta pengambilalihan kendali Roll dan Pitch pada pergerakan otomatis.



Gambar 3. 8 Diagram alir program kendali

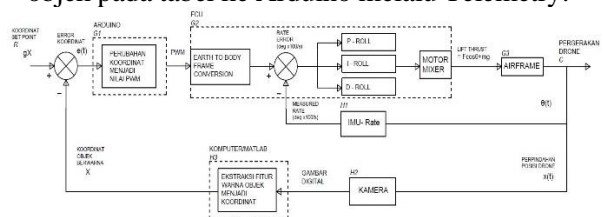
Data mengenai pulsa Receiver yang diperoleh berupa nilai PWM yang diberikan oleh user baik itu dari *interface* maupun dari R/C. Pada dasarnya user memberi input kendali *Throttle*, *Yaw*, *Roll* dan *Pitch* berupa PWM. Namun pada program kendali otomatis, input *Roll* dan *Pitch* tidak langsung diteruskan kedalam input pada FCU melainkan diubah terlebih dahulu nilainya oleh Arduino sesuai dengan keputusan pergerakan yang dihasilkan dari data visual.

Citra digital dibentuk oleh kumpulan titik yang dinamakan pixel (picture element). Setiap pixel digambarkan sebagai sebuah kotak kecil. Setiap pixel mempunyai koordinat posisi. Berikut adalah koordinat pixel yang akan digunakan dari frame gambar berukuran 320 x 420. Koordinat ini dinyatakan oleh (X, Y), di mana X menyatakan posisi kolom dan Y menyatakan posisi baris. Pixel pojok kiri atas mempunyai koordinat (0, 0) dan pixel pada kanan bawah mempunyai koordinat (320, 240).

Tabel 3. 6 Tabel keputusan pergerakan berdasarkan posisi objek pada frame gambar

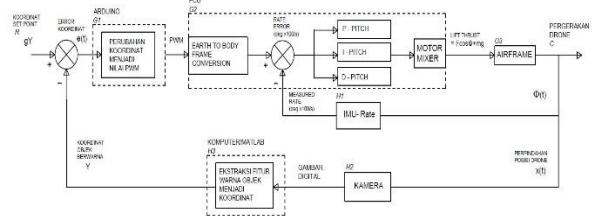
			106, 7		213, 3		32 0	X
					d			
					e			
80					f			
					g		k	l
16	0	a	b	c	h			m
					i			
					j			
24	0							
0								
Y								

MATLAB sebagai pemroses citra digital mengirimkan karakter sesuai dengan posisi objek pada tabel ke Arduino melalui Telemetry.



Gambar 3. 9 Diagram blok kendali roll

Kendali berbasis pixel koordinat dapat digambarkan dalam diagram blok kendali sebagai berikut, dengan (gX,gY) adalah koordinat tengah dari frame.



Gambar 3. 10 Diagram blok kendali pitch

3.6 Integrasi Sistem

Integrasi dimulai ketika MATLAB telah selesai memproses gambar yang dikirim dari kamera pada *Drone* dan merubah informasi koordinat objek pada frame gambar menjadi karakter sesuai dengan Tabel 3.5. Karakter yang diterima kemudian diterjemahkan menjadi

sinyal PWM yang akan diteruskan ke FCU. Sinyal PWM yang diterima FCU berada pada rentang 1975 hingga 995 dengan nilai tengah/idle 1485 yang merupakan nilai *Time-On* dalam satuan *microseconds*. Berikut adalah tabel penerjemahan karakter menjadi nilai-nilai PWM yang akan diteruskan ke FCU.

Tabel 3. 7 Tabel nilai PWM terhadap karakter yang diterima Arduino

	a	b	c	k	l	m		
d	1975	1812	1648	1975	1322	1158	995	PITCH MAJU
e				1812				
f				1648				
g				1485				
h				1322				
i				1158				
j	995							PITCH MUNDUR
	ROLL KIRI			ROLL KANAN				

Nilai PWM tersebut kemudian diteruskan ke input sinyal FCU sesuai *channel*-nya (roll/pitch). Input PWM terhadap karakter selalu dapat diinterupsi oleh kontrol manual, jika nilai PWM dari kontrol manual baik itu R/C atau antarmuka > 1485 atau < 1485.

3.7 Pengujian

Tempat yang akan digunakan untuk pengujian sistem adalah di perkebunan sayur Kp, Maleber, Desa Ciherang, Kec. Pacet, Kab. Cianjur. Perkebunan sayur ini memiliki beberapa komoditas tanaman yang masih aktif memenuhi kebutuhan pasar daerah, beberapa komoditas nya yaitu: bawang putih, daun bawang, brokoli, sawi dan kol.



Gambar 3. 11 Lahan dengan komoditas daun bawang

3.8 Tahapan Analisis dalam Penyusunan laporan

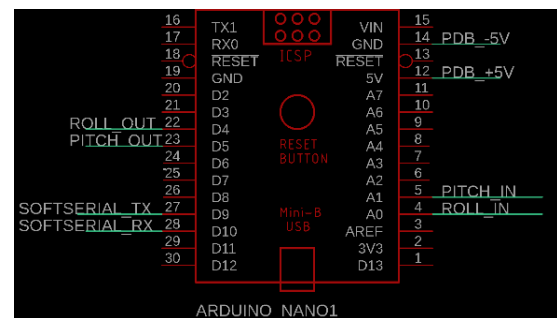
Pada tahap akhir dari metodologi ini ialah analisis hasil pengujian yang dilakukan serta penyusunan laporan. Laporan ini berisi hasil pengujian dan analisa performa sistem dalam menjalankan misinya untuk mengestimasi produktivitas suatu lahan perkebunan.

4 ANALISIS HASIL IMPLEMENTASI

4.1 Pengujian Kendali Otomatis Berbasis Image Processing

4.1.1 Rangkaian Perangkat Kendali

Kontrol pergerakan diprogram pada Arduino Nano dengan rangkaian seperti pada Gambar 4.1.



Gambar 4. 1 Wiring pada perangkat Arduino

Terdapat dua input dan dua output pada perangkat, berbeda dengan perancangan sistem yang memiliki lima input dan lima output. Dua input adalah input sinyal PWM dari Radio Receiver untuk *channel* Roll dan Pitch, sedangkan untuk *channel* lainnya seperti *mode*, *yaw*, dan *throttle* tidak diinputkan pada Arduino melainkan sinyal langsung diterima FCU dari Receiver. Pengurangan *channel* yang diinterupsi bertujuan untuk mempercepat proses penerimaan data dari Telemetry. User mengendalikan UAV secara manual melalui

remot kontrol yang terhubung dengan Radio Receiver. Dua output adalah output sinyal PWM untuk Roll dan Pitch yang merupakan sinyal yang diteruskan dari Radio Receiver, ataupun sinyal yang dihasilkan dari keputusan pergerakan otomatis berbasis *image processing*. Sinyal PWM output ini terhubung dengan pin-pin input pada FCU sesuai dengan peruntukan pinnya (*roll* dan *pitch*).

Pin D9 dan D10 merupakan jalur komunikasi serial yang terhubung dengan Telemetry pada UAV. Telemetry pada UAV terhubung dengan Telemetry yang ada pada pengendali otomatis (laptop) secara nirkabel. Telemetry yang digunakan adalah SiC Digital Telemetry dengan frekuensi 915MHz.

4.1.2 Pemrograman Kendali MATLAB

Pengendalian berbasis *image processing* dilakukan di MATLAB. Keputusan pergerakan dihasilkan dari koordinat posisi objek pada *frame* gambar. Objek yang menjadi patokan pergerakan adalah lingkaran berwarna kuning. Untuk pengujian pergerakan, objek merupakan sebuah lingkaran kuning yang dicetak pada kertas HVS.

Kamera yang digunakan untuk mengambil gambar secara *real-time* merupakan kamera aksi Yi Action Camera 16MP.



Gambar 4. 2 Yi Action Camera [yitechnology.com]

Gambar dari kamera tersebut dihubungkan dengan Video Transmitter pada UAV melalui USB Port pada kamera. Video Transmitter terhubung dengan Video Receiver yang ada pada pengendali otomatis (laptop). Video Transmitter dan Receiver yang digunakan berfrekuensi 5.8GHz.



Gambar 4. 3 Video Transmitter TX805 dan Video Receiver ROTG01

Format yang didukung oleh Video Receiver adalah MJPG_352x288 dan MJPG_640x480. Format yang digunakan untuk pengujian kendali otomatis adalah MJPG_352x288 yang di-*crop* menjadi 320x240 pixel.

Gambar dari Video Receiver diolah dengan MATLAB untuk mendeteksi posisi lingkaran kuning pada *frame* dengan spesifikasi warna kombinasi merah, hijau, dan biru sebagai berikut: Red > 130, Green > 130, dan Blue < 100 yang dituliskan menjadi program pada MATLAB seperti berikut.

```
for i=1:m
    for j=1:n
        if(im(i,j,1)>130 &&
            im(i,j,2)>130 && im(i,j,3)<100)
            bw2(i,j)=1;
        end
    end
end
```

Setelah pendeteksian warna objek, MATLAB diprogram untuk mendeteksi bentuk lingkaran dengan radius 6-30 pixel yang dapat dituliskan menjadi program pada MATLAB seperti berikut.

```
img=bw2;
diff_im = bwareaopen(img,6);
[centers,radii,metric] =
imfindcircles(diff_im,[6 30]);

if length(imfindcircles(diff_im,[6 30])) >
0
    centersStrong5 = centers(1:1,:);
```

```

radiiStrong5 = radii(1:1);
metricStrong5 = metric(1:1);

set(handles.radii,'String',num2str(radiiStrong5)
);

axes(handles.axes2);

hold on

Jika objek terdeteksi, koordinat objek
dapat diketahui dengan mengeluarkan koordinat
dari tengah lingkaran. disebutkan dengan bc (1)
dan bc (2) sebagai koordinat X dan koordinat Y.

for object = 1:length(centersStrong5)
px = centersStrong5(1)-radiiStrong5;
py = centersStrong5(2)-radiiStrong5;
d = radiiStrong5*2;

rectangle('Position',[px py d
d],'EdgeColor','r','LineWidth',2)

bc= centersStrong5;

plot(bc(1),bc(2),'-m+')

```

Berdasarkan koordinat tersebut, keputusan pergerakan dapat dihasilkan sesuai dengan tabel keputusan pergerakan yang telah dibuat pada bab perancangan sistem. Berikut adalah cuplikan program keputusan pergerakan pada MATLAB.

```

if bc(1)>=0 && bc(1)<=45.714
valauto = get(handles.auto,'value');

if valauto == 1
fprintf(s,'a');

end

set(myform.arah,'String','roll kiri
curam');

set(myform.karakter,'String','a');

elseif bc(1)>45.714 &&
bc(1)<=91.428

valauto = get(handles.auto,'value');

if valauto == 1
fprintf(s,'b');

end

```

```

set(myform.arah,'String','roll kiri
sedang');

set(myform.karakter,'String','b');

elseif bc(1)>91.428 &&
bc(1)<=137.142

valauto = get(handles.auto,'value');

if valauto == 1
fprintf(s,'c');

end

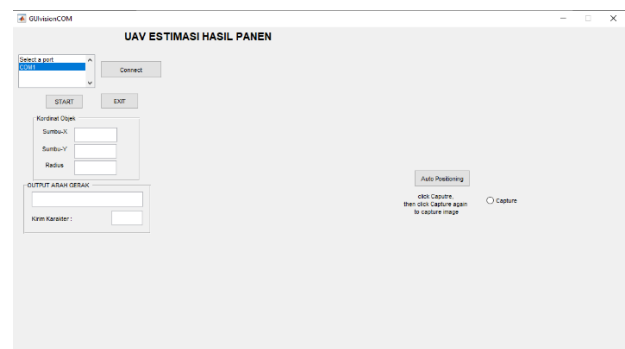
set(myform.arah,'String','roll kiri
sedikit');

set(myform.karakter,'String','c');

...

```

Karakter untuk mewakili keputusan pergerakan dikirimkan ke Arduino melalui Telemetry secara serial. Tombol 'Auto Positioning' yang disebutkan pada program dengan 'valauto' berfungsi untuk memulai dan menghentikan pengiriman karakter ke Arduino.



Gambar 4. 4 Tampilan Antarmuka Pengujian Kendali Otomatis

Pengiriman karakter menggunakan fungsi 'fprintf' yang ada pada MATLAB dengan konfigurasi komunikasi serial sebagai berikut.

```

serPort = serList{serPortn};

serConn = serial(serPort, 'TimeOut', 1, ...
'BAUDRate', 9600);

try
fopen(serConn);

handles.serConn = serConn;

set(hObject, 'String','Disconnect')

catch e

```

```
errorldg(e.message);
end
```

4.1.3 Pemrograman Kendali Arduino

Karakter yang dikirim dari MATLAB melalui Telemetry diproses sesuai dengan tabel nilai PWM terhadap karakter yang diterima. Keputusan nilai PWM kemudian di-outputkan ke FCU sehingga UAV bergerak sesuai dengan arah pergerakan yang telah diputuskan.

Program komunikasi menggunakan *library* SoftwareSerial dengan konfigurasi komunikasi sebagai berikut.

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(9,10); //rx,tx
void setup() {
  mySerial.begin(9600);
```

Pemrosesan sinyal PWM menggunakan *library* Servo dengan konfigurasi sebagai berikut.

```
#include <Servo.h>
Servo myroll;
Servo mypitch;
Servo myyaw;
Servo mytrot;
Servo mymode;
void setup() {
  myroll.attach(4);
  mypitch.attach(5);
  myyaw.attach(6);
  mytrot.attach(7);
  mymode.attach(8);
```

Pembacaan sinyal PWM dari Radio Receiver untuk penerbangan UAV secara manual menggunakan fungsi 'PulseIn' dan dapat dituliskan sebagai program seperti berikut.

```
roll=pulseIn(14,HIGH);
pitch=pulseIn(15,HIGH);
yaw=pulseIn(16,HIGH);
trot=pulseIn(17,HIGH);
mode=pulseIn(18,HIGH);
```

Ketika terdapat karakter yang dikirimkan dari MATLAB, sinyal PWM yang diteruskan ke FCU untuk kanal *Roll* dan *Pitch* tidak lagi sinyal dari Radio Receiver melainkan dari nilai PWM berdasarkan tabel nilai PWM terhadap karakter yang diterima. Program penerimaan karakter dan pengiriman PWM ke FCU dapat dituliskan seperti cuplikan program berikut.

```
if(mySerial.available()>0){
  c = mySerial.read();
}
else {
  c = 'o';
}
switch (c){
case 'o':
  myroll.writeMicroseconds(roll);
  mypitch.writeMicroseconds(pitch);
  myyaw.writeMicroseconds(yaw);
  mytrot.writeMicroseconds(trot);
  mymode.writeMicroseconds(mode);
  break;
case 'a':
  myroll.writeMicroseconds(1975);
  mypitch.writeMicroseconds(pitch);
  myyaw.writeMicroseconds(yaw);
  mytrot.writeMicroseconds(trot);
  mymode.writeMicroseconds(mode);
  break;
case 'b':
  myroll.writeMicroseconds(1812);
  mypitch.writeMicroseconds(pitch);
  myyaw.writeMicroseconds(yaw);
  mytrot.writeMicroseconds(trot);
  mymode.writeMicroseconds(mode);
  break;
...
case 'g':
  mypitch.writeMicroseconds(1485);
  myroll.writeMicroseconds(1485);
  myyaw.writeMicroseconds(yaw);
  mytrot.writeMicroseconds(trot);
  mymode.writeMicroseconds(mode);
  break;
```

...
Karakter 'o' disebutkan sebagai perintah interupsi dari remot kontrol manual. PWM yang di-inputkan ke FCU ketika kondisi interupsi remot adalah PWM yang dibaca langsung dari Radio Receiver. User dapat menginterupsi pergerakan otomatis dengan menggerakkan tuas *pitch* atau *yaw* pada saat apapun. Program interupsi ini dapat dituliskan sebagai berikut.


```

if(roll > 1500 || roll < 1300){
    c = 'o';
}
if(pitch > 1500 || pitch < 1300){
    c = 'o';
}

```

4.1.4 Pengujian Kendali Otomatis



Gambar 4. 5 Tampilan Antarmuka saat Pengujian

Pengujian dilakukan dengan menaruh gambar lingkaran kuning berdiameter 13mm di lantai tepat di bawah kamera yang menghadap ke bawah pada UAV. Jarak dari kamera ke lantai kurang lebih 13cm. Gambar lingkaran tersebut digeser-geser untuk menghasilkan karakter yang berbeda.

Hasil dari pergerakan tersebut dimonitor melalui aplikasi Mission Planner yang terhubung langsung ke UAV dan serial monitor pada Arduino IDE.



Gambar 4. 6 Tampilan Mission Planner untuk Memonitor Kanal Input FCU

Pengujian responsivitas dari kendali otomatis dilakukan dengan melihat jentang waktu perubahan posisi penanda dengan perubahan perintah pada Serial Monitor Arduino.

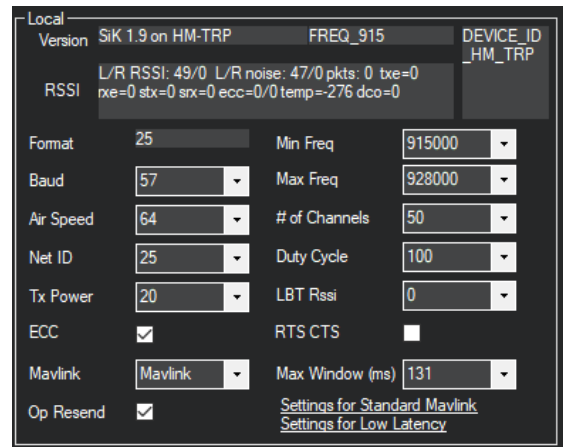
Pengujian ini dilakukan di dalam ruangan dengan kondisi UAV dan pengendali otomatis berada dalam satu ruangan tanpa terhalang apapun dengan CPU Background Usage pengendali < 15%. Berikut adalah

spesifikasi komputer/pengendali yang digunakan:

- CPU : i7-7500U 2.9GHz
- GPU : Intel HD 620 300MHz
- RAM : 8GB
- OS : Windows 10 Pro ver.1903

Perlu diketahui bahwa penggunaan telemetry untuk pengiriman data pengendali memperlambat sistem secara keseluruhan. Terdapat kemungkinan terjadinya ‘bottleneck’ ketika menggunakan pengendali/komputer yang jauh lebih cepat dari kemampuan pengiriman data Telemetry.

Berikut adalah konfigurasi Telemetry yang digunakan. Konfigurasi yang digunakan berdasarkan default pabrik, hanya pada ‘Tx Power’ saja yang diubah menjadi maksimum.



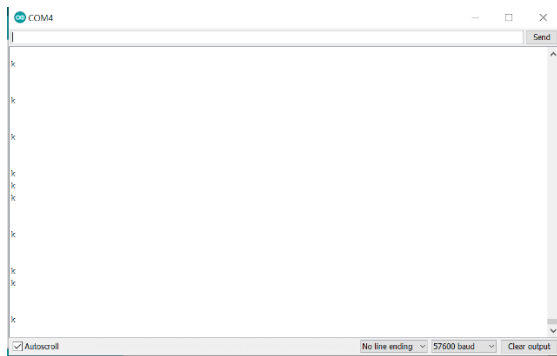
Gambar 4. 7 Konfigurasi Telemetry Pengendali Otomatis

Dari hasil pengukuran waktu perubahan karakter pada MATLAB dan waktu perubahan karakter pada Arduino, didapatkan tabel berikut.

Tabel 4. 1 Pengujian Responsivitas

Pengujian ke-	Waktu (mili detik)
	Baud Rate 57600
1	35
2	104
3	69
4	69
5	68
Rata-rata	69

Didapatkan waktu respon rata-rata 69 mili detik pada konfigurasi sistem ini.



Gambar 4. 8 Tampilan Serial Monitor Arduino saat Penerimaan Karakter

4.1.5 Pengujian Kendali Otomatis di Lapangan

Setelah dipastikan bahwa kendali otomatis berjalan pada sistem, drone

diterbangkan di lapangan terbuka untuk pengujian integrasi kendali otomatis dengan sistem drone ketika terbang. Pengujian dilakukan di lapangan *mini-soccer* milik POLMAN pada 21 Oktober 2019, pukul 16.15 hingga pukul 17.00 dengan kondisi cuaca cerah berangin.

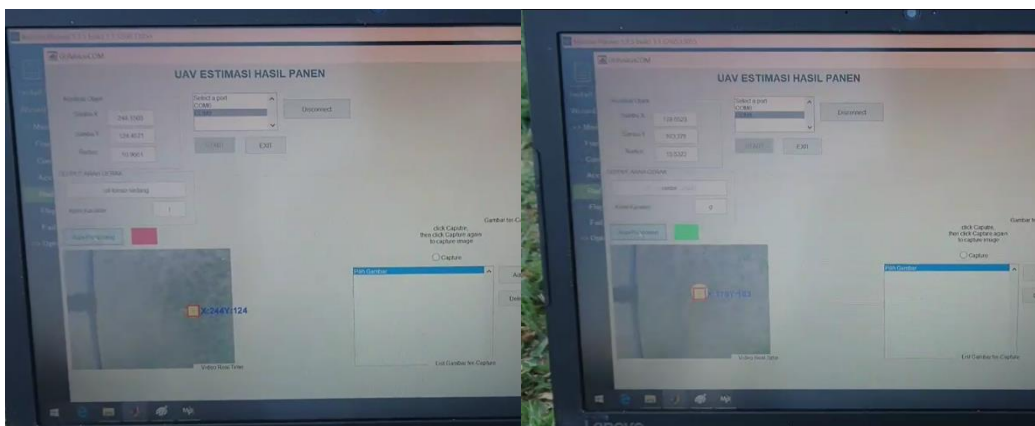
Konfigurasi mode terbang yang digunakan adalah mode Loiter atau mode Altitude Hold + GPS Hold, pada mode ini drone menjadi sulit terbawa angin sehingga ketika pengendali otomatis tidak dapat melawan gaya dari angin, drone akan diam di tempat dan tidak terbawa angin.

GPS Hold dipastikan terlebih dahulu berfungsi dengan baik dengan percobaan *take-off* kemudian *landing* dari beberapa ketinggian tiga kali berturut-turut hanya dengan menaikkan *throttle* dan mengukur jarak dari tempat *take-off* ke tempat *landing*.

Tabel 4. 2 Pengujian GPS Hold

Tinggi (m)	2.5		5		10		20		30	
Jarak TO-Landing (m)	I	0.4	I	0.6	I	1.0	I	0.8	I	0.8
	II	0.2	II	0.8	II	2.5	II	1.5	II	1.2
	III	0.4	III	0.7	III	1.6	III	1.4	III	0.8

Hasil percobaan menunjukkan bahwa mode GPS Hold dapat berfungsi dengan baik dengan toleransi 2.5meter.



(a)

(b)

Gambar 4. 9 (a) Tampilan Penglihatan Kamera drone Sebelum kendali otomatis dijalankan. (b) Penglihatan kamera drone setelah kendali otomatis dijalankan

Tabel 4. 3 Pengujian Kecepatan Pergerakan Drone dengan Kendali Otomatis

Kondisi Angin	Waktu dari titik terjauh ke center
< 1.54 m/s	7 s
> 1.54 m/s	12 s

Dapat diketahui dari hasil pengujian bahwa kendali otomatis dapat terintegrasi dengan sistem drone ketika terbang. Angin menjadi faktor yang mempengaruhi kecepatan pergerakan dari drone mengingat bahwa drone akan diam ketika kendali otomatis tidak dapat melawan gaya dari angin.

4.2 Pengujian Perhitungan Populasi Tanaman

4.2.1 Pengujian Kualitas Gambar Lahan

Pengujian kualitas gambar lahan dilakukan untuk mengetahui kemampuan kamera untuk mengambil seluruh lahan dalam satu kali terbang yang bertujuan untuk efisiensi waktu dan baterai dari drone.

Pengambilan gambar lahan dilakukan di perkebunan sayur kampung Maleber, Cianjur. Pengambilan gambar dilakukan pada pukul 10.41 hingga pukul 11.46 pada hari Sabtu, 10 Agustus 2019, dengan kondisi cuaca cerah berawan. Lahan yang diambil gambarnya adalah lahan Tomat dan Kol dengan spesifikasi lahan sebagai berikut.

Tabel 4. 4 Spesifikasi Lahan Pengujian

Lahan	Panjang Lahan	Lebar Lahan	Jarak Antar Tanaman	Mulsa
Kol	30 m	8 m	30 cm	Ya
Tomat	48.5 m	35 m	50 cm	Tidak

Pengambilan gambar **tidak** menggunakan kendali otomatis berbasis *image processing*, melainkan menggunakan *positioning* berbasis GPS yang sudah tersedia pada FCU dikarenakan ketinggian yang diambil drone terlalu tinggi untuk dapat mengambil gambar

keseluruhan lahan dan sekaligus mendeteksi penanda lahan.

Berikut adalah gambar lahan kol yang diambil dari gambar yang didapat dari Video Transmitter dengan resolusi 640x480. Lahan yang ditanami Kol ditunjukkan di dalam gambar kotak merah.



Gambar 4. 10 Foto Lahan Kol dari Ketinggian 28.35m Terhadap Lahan

Berikut adalah gambar lahan tomat yang diambil dari gambar yang didapat dari Video Transmitter dengan resolusi 640x480.



Gambar 4. 11 Foto Lahan Tomat dari Ketinggian 42.26m Terhadap Lahan

Pemrosesan gambar dilakukan pada MATLAB dengan tahapan proses sebagai berikut:

1. Pembacaan gambar
2. Crop gambar
3. Filterasi warna dengan mendeteksi warna pada rentang RGB yang diinginkan

- a. untuk tanaman kol rentang RGB nya adalah $R < 112$, $G < 145$, dan $B > 106$
 - b. untuk tanaman tomat rentang RGB nya adalah $R < 110$, $G > 110$, dan $B < 115$
4. Perhitungan objek yang berada pada rentang warna

Rentang RGB didapatkan dari mengetahui nilai parameter RGB pada beberapa titik sampel pada gambar.



Gambar 4. 12 Pengambilan Sampel Parameter RGB

Hasil pemrosesan gambar lahan kol ditunjukkan pada gambar berikut.



(a)



(b)



(c)

Gambar 4. 13 (a) Foto Lahan Kol Setelah Di-crop Menjadi 400x201 pixel

(b) Foto Lahan Kol Setelah Diseleksi Berdasarkan Warna dan Dirubah Menjadi Gambar Biner

(c) Foto Lahan Kol dengan Penanda Letak Objek pada Frame

Dari hasil pemrosesan gambar lahan kol dapat terlihat terdapat warna tanaman yang menyatu dengan tanaman sekitarnya dan terdapat tanaman yang tidak terdeteksi oleh seleksi warna. Pada lahan yang sama terdapat populasi tanaman bawang merah yang seharusnya tidak terdeteksi pada seleksi warna namun masuk pada seleksi.

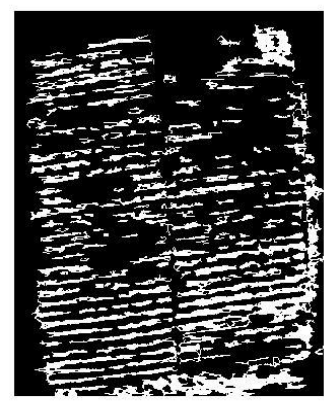
Hasil pemrosesan gambar lahan tomat ditunjukkan pada gambar berikut.



(a)



(b)



(c)

Gambar 4. 14 (a) Foto Lahan Tomat Setelah Di-crop Menjadi 305x374 pixel

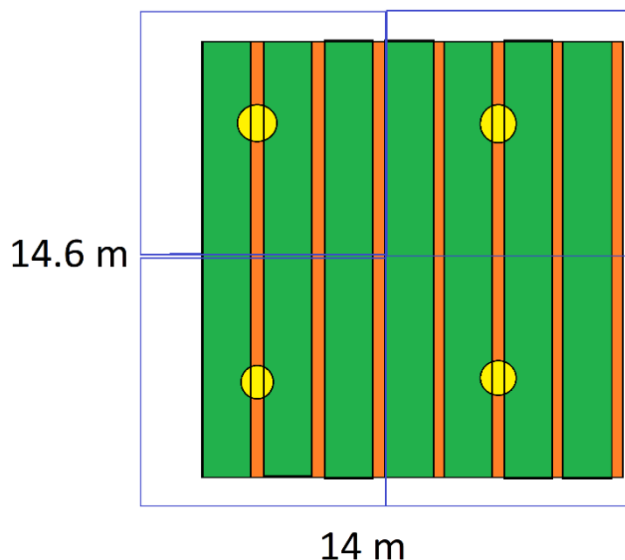
b) Foto Lahan Tomat Setelah Diseleksi Berdasarkan Warna dan Dirubah Menjadi Gambar Biner

(c) Foto Lahan Tomat dengan Penanda Letak Objek pada Frame

Dari hasil pemrosesan gambar lahan tomat dapat terlihat terdapat warna tanaman yang menyatu dengan tanaman sekitarnya dan terdapat tanaman yang tidak terdeteksi oleh seleksi warna. Terdapat tanaman di luar lahan yang terdeteksi oleh seleksi warna yang seharusnya tidak.

4.2.2 Pengujian Perhitungan dengan Penanda Lahan

Setelah diketahui bahwa kamera tidak mampu menghasilkan gambar lahan keseluruhan yang dapat diproses, maka dilakukan pengambilan gambar dengan pembagian lahan. Pengujian dilakukan pada pukul 12.30 hingga pukul 14.30 hari Sabtu, 26 Oktober 2019, dengan kondisi cuaca cerah berawan. Lahan yang digunakan adalah lahan brokoli dengan luasan 14 m x 14.6 m. Lahan brokoli digunakan karena memiliki spesifikasi yang mirip dengan kol yaitu jarak antar tanaman 30 cm dan daun yang lebar. Skema pengambilan gambar pada lahan kol adalah sebagai berikut.



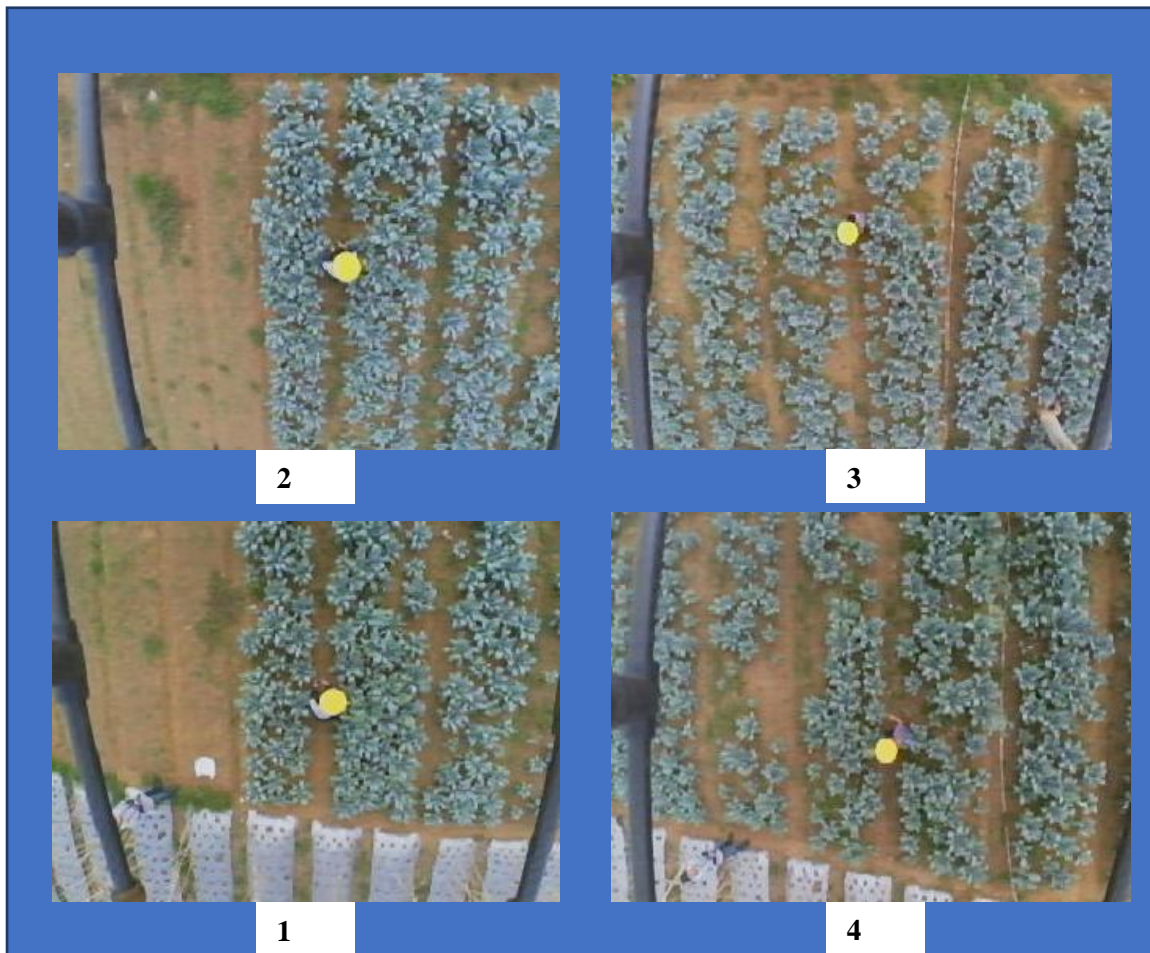
Gambar 4. 15 Skema pengambilan gambar lahan brokoli.

Ketinggian pengambilan gambar yaitu 2.5 m. Ketinggian didapatkan dari ratio *Field of View* dari kamera yang digunakan. Dari pengujian ratio ketinggian terhadap cakupan

kamera terhadap luasan gambar didapatkan ratio 2:5, yaitu 2 m ketinggian kamera terhadap objek mencakup 5 m luasan objek. Dengan demikian pada pengujian pengambilan gambar lahan brokoli mencakup 6.25 m tiap bagian lahan.



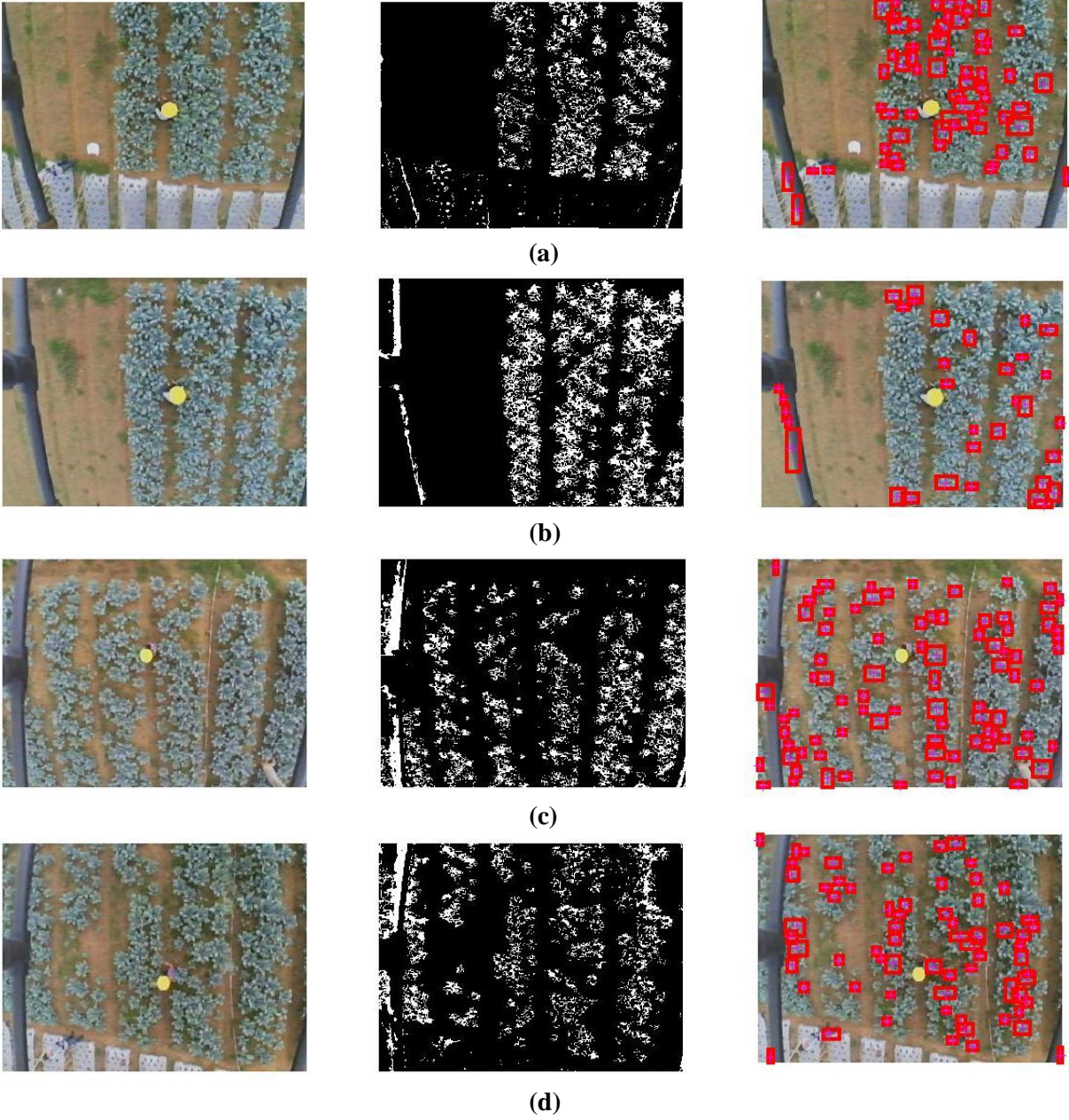
Gambar 4. 16 Pengukuran ratio cakupan kamera dengan ubin 30 x 30 cm sebagai objek.



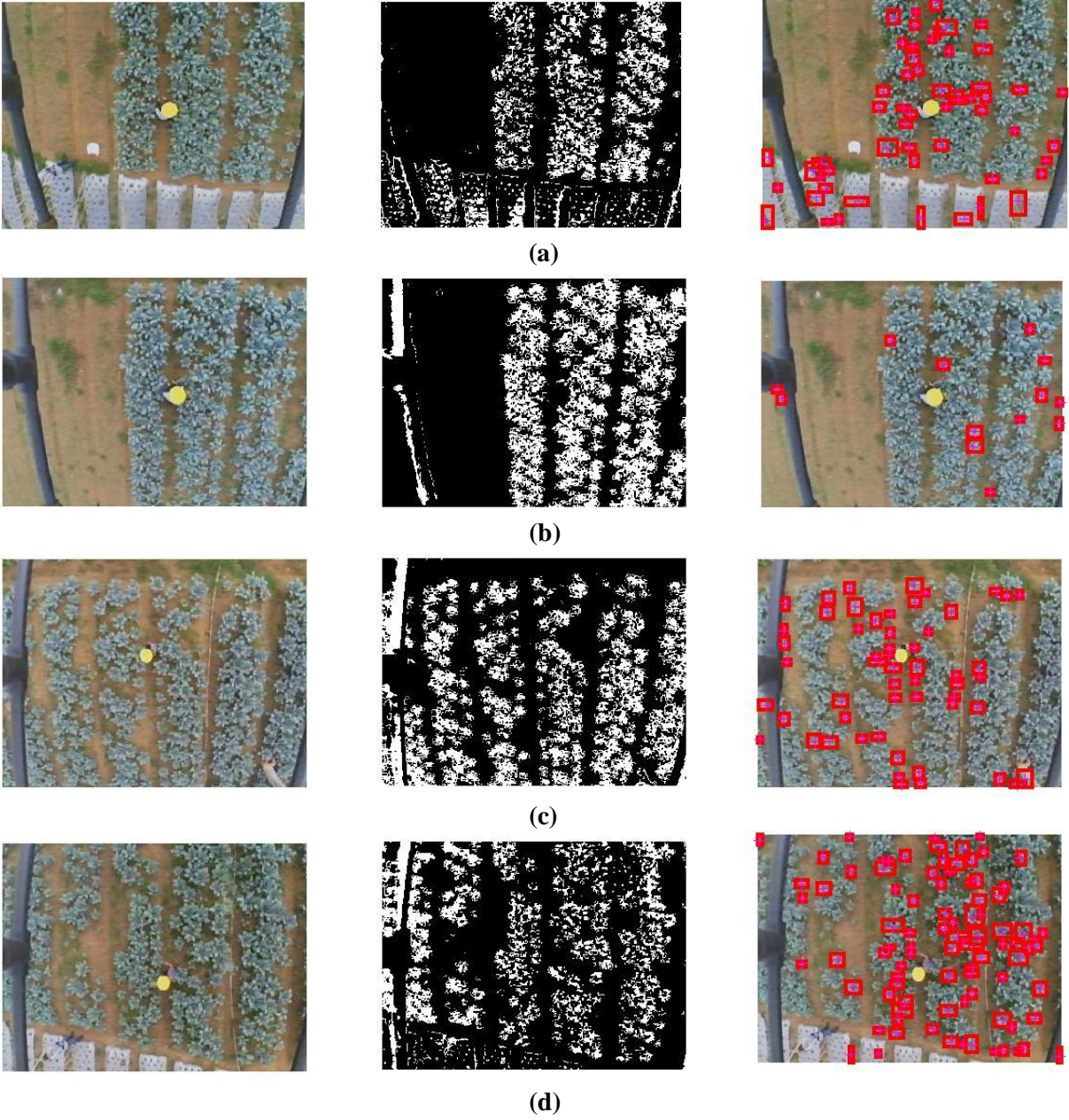
Gambar 4. 17 Gambar lahan brokoli diambil sesuai skema.

Pemrosesan gambar dilakukan pada MATLAB dengan tahapan proses sebagai berikut:

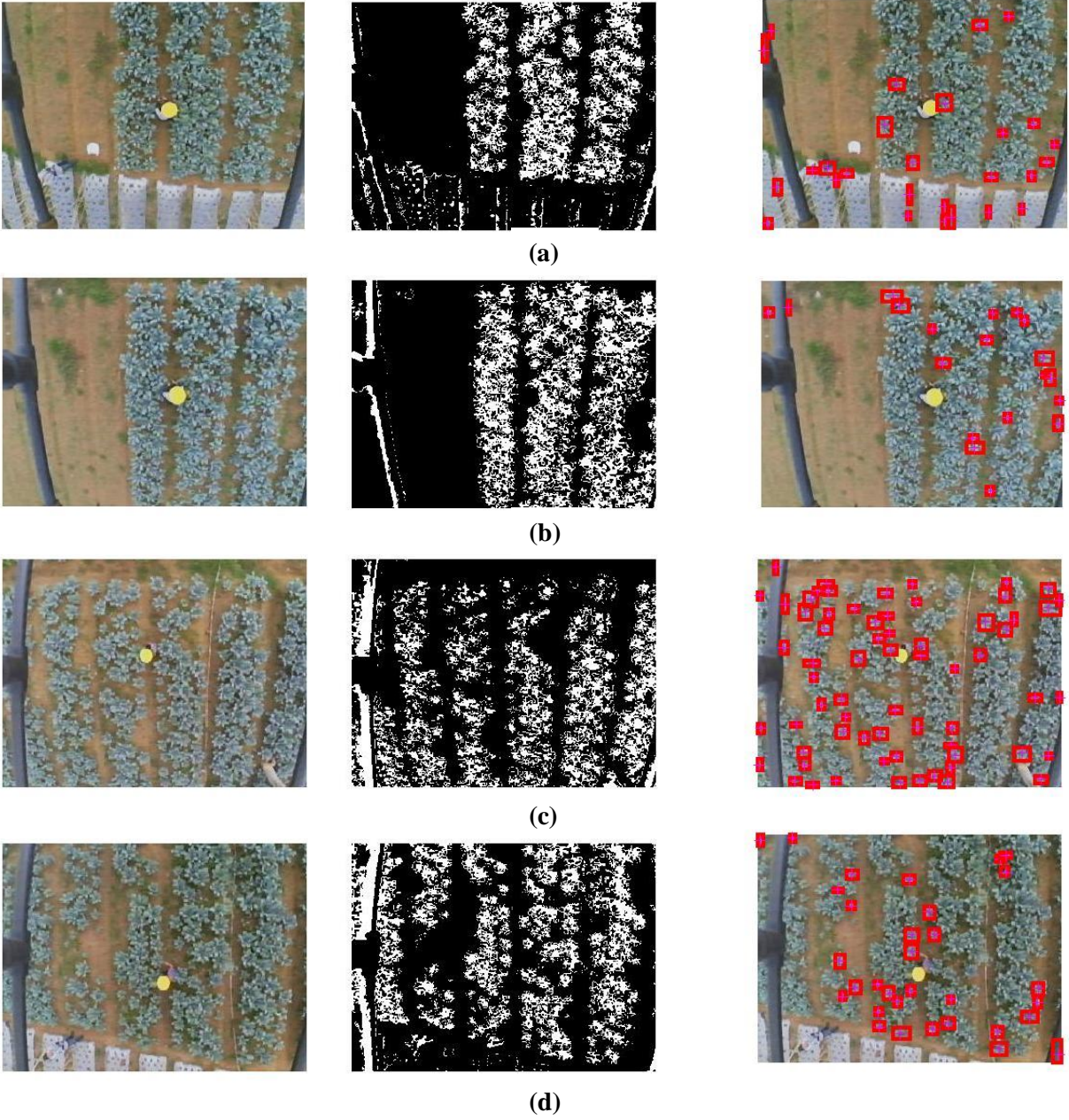
1. Pembacaan gambar
2. Filterasi warna dengan mendeteksi warna pada rentang RGB yang diinginkan
 - a. Rentang RGB percobaan I adalah $R < 130$, $G > 95$, dan $B > 120$
 - b. Rentang RGB percobaan II adalah $R < 150$, $G > 95$, dan $B > 120$
(memperbanyak objek dengan unsur merah)
 - c. Rentang RGB percobaan III adalah $R < 130$, $G > 95$, dan $B > 100$
(memperbanyak objek dengan unsur biru)
3. Filterasi objek terdeteksi dengan $\text{pixel} < 30$ pixel
4. Perhitungan objek yang berada pada rentang warna



Gambar 4. 18 (a,b,c,d) Pemrosesan gambar 1, 2, 4, dan 4 dari raw, biner, dan hasil secara berurutan



Gambar 4. 19 (a,b,c,d) Pemrosesan gambar 1, 2, 4, dan 4 dari raw, biner, dan hasil secara berurutan (memperbanyak objek dengan unsur merah)



Gambar 4. 20 (a,b,c,d) Pemrosesan gambar 1, 2, 4, dan 4 dari raw, biner, dan hasil secara berurutan (memperbanyak objek dengan unsur biru)

Tabel 4. 5 Hasil perhitungan jumlah objek dari pemrosesan gambar dan error dengan hasil hitung manual.

Gambar	Jumlah dari Pemrosesan Gambar		
	I	II	II I
1	64	56	27
2	33	13	19
3	85	55	66
4	68	90	32
Total	250	214	144
Hasil Perhitungan Manual	659		
	Error		
	I	II	III
	62.06%	67.53%	78.15%

Dari hasil perhitungan objek dari pemrosesan gambar dan dari kondisi gambar biner pada masing-masing bagian, ada beberapa fenomena yang terlihat:

1. Sistem tidak dapat membedakan tanaman yang daunnya berhimpit dengan tanaman lain seperti yang selalu terlihat pada Gambar 2 dan perhitungan dengan menaikkan jumlah objek yang memiliki unsur warna biru.
2. Sistem mendeteksi objek lain yang bukan tanaman seperti yang terlihat pada percobaan perhitungan dengan menaikkan jumlah objek yang memiliki unsur warna merah.
3. Sistem dapat membedakan warna tanaman dengan warna lingkungannya hanya saja tidak dapat mengidentifikasi objek tersebut sebagai 'tanaman'.

4.2.3 Pengujian Perhitungan dengan Pencarian Pusat Tanaman

Fenomena pada bab 4.2.2 menunjukkan bahwa sistem tidak dapat mengidentifikasi objek sebagai sebuah tanaman, maka dari itu pada bab ini percobaan dilakukan dengan mencari pusat dari tanaman atau mendeteksi lingkaran pada tanaman.



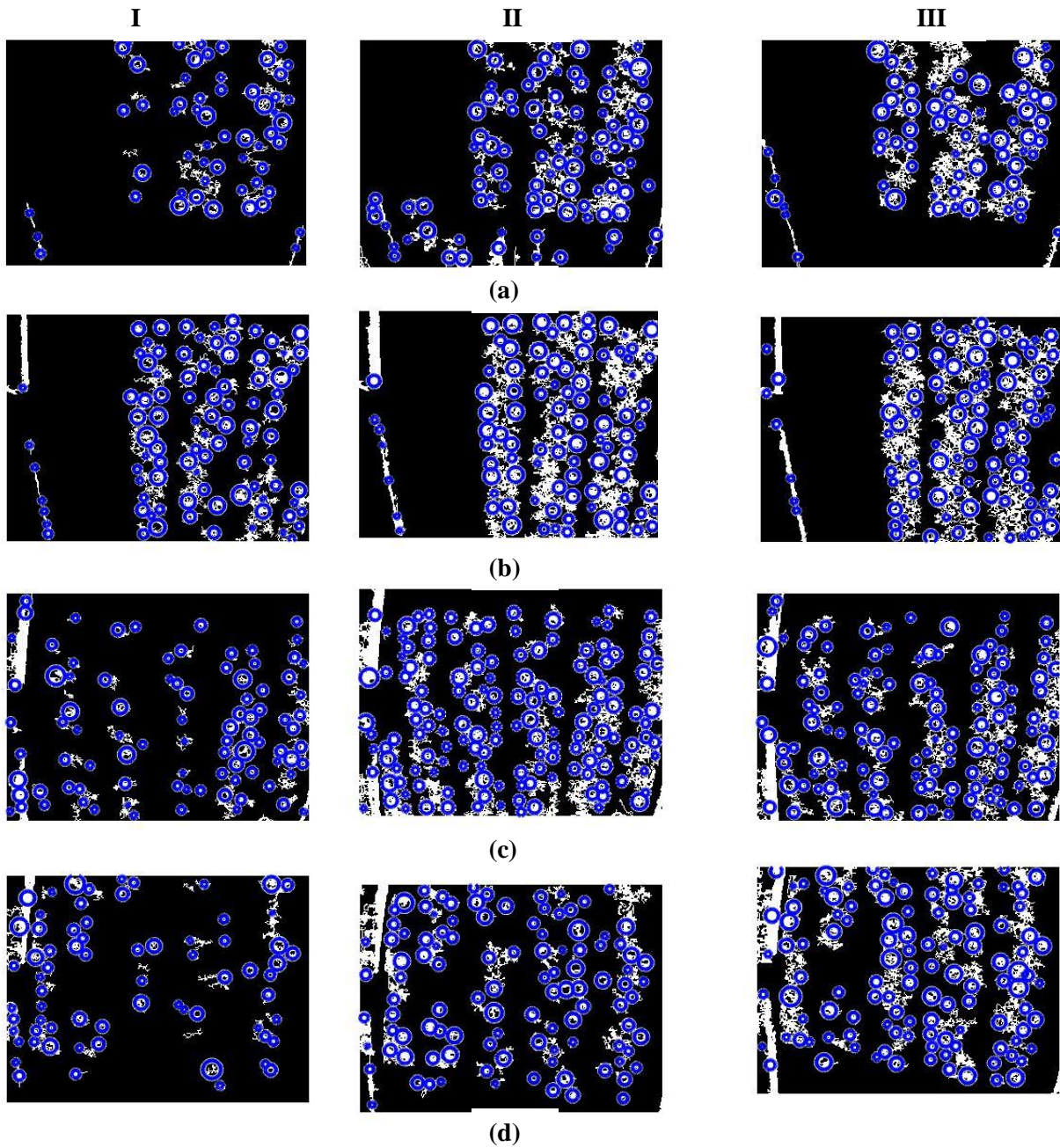
Gambar 4. 21 Pusat dari tanaman brokoli ditandai dengan lingkaran merah.

Pemrosesan gambar dilakukan pada MATLAB dengan tahapan proses sebagai berikut:

1. Pembacaan gambar
2. Filterasi warna dengan mendeteksi warna pada rentang RGB yang diinginkan
 - a. Rentang RGB percobaan I adalah $R < 130$, $G > 95$, dan $B > 120$
 - b. Rentang RGB percobaan II adalah $R < 150$, $G > 95$, dan $B > 120$
(memperbanyak objek dengan unsur merah)
 - c. Rentang RGB percobaan III adalah $R < 130$, $G > 95$, dan $B > 100$
(memperbanyak objek dengan unsur biru)
3. Filterasi objek terdeteksi dengan pixel < 30 pixel
4. Pencarian pusat lingkaran dengan fungsi *imfindcircles*
5. Perhitungan pusat lingkaran tanaman yang terdeteksi

Tabel 4. 6 Hasil perhitungan jumlah lingkaran (pusat tanaman) dari pemrosesan gambar dan error dengan hasil hitung manual.

Gambar	Jumlah dari Pemrosesan Gambar				
	I	II	II I		
1	56	91	84		
2	91	93	96		
3	90	14 6	12 9		
4	69	10 7	11 6		
Total	30 6	43 7	42 5		
Hasil Perhitungan Manual	659		Error		
			I	II	III
			53. 57 %	33. 69 %	35. 51 %



Gambar 4. 22 (a) Deteksi lingkaran pada gambar 1 dengan percobaan I, II, dan III (b) Deteksi lingkaran pada gambar 2 dengan percobaan I, II, dan III (c) Deteksi lingkaran pada gambar 3 dengan percobaan I, II, dan III (d) Deteksi lingkaran pada gambar 4 dengan percobaan I, II, dan III

Dari hasil perhitungan lingkaran dari pemrosesan gambar dan dari kondisi gambar biner pada masing-masing bagian, ada beberapa fenomena yang terlihat:

1. Penerapan pendeteksian lingkaran untuk mendeteksi pusat tanaman mengurangi error dari percobaan sebelumnya (tanpa lingkaran) secara signifikan.
2. Sistem masih mendeteksi objek lain yang bukan tanaman, paling terlihat pada percobaan II gambar 1 ketika objek dengan unsur warna merah diperbanyak.
3. Pada percobaan perhitungan pusat tanaman, dengan menambah objek yang memiliki unsur warna biru ataupun merah, tanaman yang terdeteksi bertambah banyak.
4. Masih terdapat tanaman yang tidak terdeteksi dikarenakan terlalu rapat satu dengan yang lainnya.









4.3 Ketidakpastian Sistem





4.3.1 Hasil Seleksi Gambar Tidak Tepat Sasaran

Penggunaan Video Transmitter dan Video Receiver membuat kualitas gambar yang

dihasilkan menurun. Kualitas gambar mempengaruhi hasil dari pemrosesan gambar. Semakin jauh jarak objek (tanaman) terhadap kamera yang terpasang pada UAV semakin menurun kualitas gambar yang dihasilkan.

Sebagai pembanding, metode untuk mengestimasi produktivitas lahan dengan menggunakan luasan lahan seperti pada (Putri. A, 2014) lebih efektif dibandingkan dengan metode Object Counting yang diuji dengan konfigurasi sistem yang ada pada penelitian ini.

Foto Masukan	Hasil	Perhitungan Manual (m ²)	Perhitungan Sistem (m ²)	Prediksi Hasil Panen (kg)
		7898,85	8150	52160
		4590,04	4990	31936
		6078,04	6377	40812
		7172,51	24965	158048

Hasil	Perhitungan Manual (m ²)	Perhitungan Sistem (m ²)	$\frac{ perhitungan\ sistem - perhitungan\ manual }{perhitungan\ manual} \times 100\ %$
	7898,85	8150	3,1 %
	4590,04	4990	8,7 %
	6078,04	6377	4,9 %
	7172,51	24965	248 %

Gambar 4. 23 Estimasi Produktivitas Lahan Pertanian dengan Metode Luasan Lahan (Putri. A, 2014)

5. PENUTUP

5.1 Kesimpulan

Sistem UAV atau drone untuk mengestimasi hasil produksi lahan perkebunan dengan metoda berbasis image processing telah direalisasikan dan diintegrasikan dengan drone tipe quad-copter yang menggunakan FCU APM 2.6 sebagai pengendali dasar drone. Kendali

otomatis yang telah dirancang dan dibangun telah dapat menggerakkan drone dari satu titik ke titik tengah dari frame pada kamera mengikuti penanda bulat berwarna kuning dalam waktu 7 detik dengan kecepatan angin kurang dari 1.54 m/s dan 12 detik ketika kecepatan angin lebih dari 1.54 m/s.

Pengujian kualitas kamera menunjukkan bahwa kualitas resolusi kamera yang menurun

akibat penggunaan video transmitter sebagai media pengiriman gambar dari kamera pada drone ke pengendali yang ada di darat. Akibat dari penurunan kualitas gambar adalah menyatunya spektrum warna dari tanaman satu dengan yang lainnya bahkan pada ketinggian 2.5 m dari tanah. Perhitungan populasi tanaman mendapatkan error paling kecil 62.06% pada percobaan pendeteksian warna tanaman. Pengujian perhitungan populasi tanaman dengan menggunakan pendeteksian pusat tanaman atau lingkaran tanaman mengurangi error pada pengujian sebelumnya yaitu dengan error sebesar 33.69% dengan catatan bahwa masih terdapat objek selain tanaman yang terdeteksi oleh sistem.

Estimasi produktivitas tanaman memiliki beberapa metode dalam pencapaiannya, dalam penelitian ini metode yang digunakan adalah object counting yaitu menghitung populasi tanaman yang ada pada lahan. Metode yang telah dan masih digunakan hingga saat ini adalah metode luasan lahan yang dalam penelitian ini memperlihatkan bahwa metode tersebut masih lebih efektif dibandingkan metode object counting yang dengan konfigurasi diuji pada penelitian ini.

5.2 Saran pada penilitan selanjutnya

1. Pengendalian otomatis dapat dikembangkan algoritma programnya agar dapat menahan gaya angin yang lebih besar dari yang ada pada penelitian ini.
2. Pengendalian otomatis pada drone berbasis *image processing* dengan pemroses gambar yang ikut terbang dengan drone dapat menghilangkan kebutuhan video transmitter sehingga tidak mengurangi kualitas gambar kamera.
3. Perlu adanya algoritma pengidentifikasi tanaman seperti Artificial Neural Network untuk menambah ke akuratan perhitungan populasi tanaman sehingga mengurangi error perhitungan yang ada pada penelitian ini.

DAFTAR PUSTAKA

1. Argaputra, S. (2018). Penerapan Sistem *Automatic Collision Avoidance* pada *Drone Quadcopter* Sebagai Alat Didaktik Mekatronika. Polman Bandung.
2. García, Pedro Castillo, Hernandez, Laura Elena Munoz dan Gil, Pedro García. *Indoor Navigation Strategies for Aerial Autonomous Systems*. Oxford : Butterworth-Heinemann publications, 2017. ISBN: 978-0-12-805189-4.
3. Hidayat, R. Mardiyanto, R. (2016). Pengembangan Sistem Navigasi Otomatis Pada UAV (*Unmanned Aerial Vehicle*) dengan GPS (Global Positioning System) Waypoint. *Jurnal Teknik ITS, Surabaya* Vol 5 No.2, 2016.
4. Sanjaya, M. (2016). Panduan Praktis Pemrograman Robot Vision Menggunakan MATLAB dan IDE Arduino. Yogyakarta: Andi. ISBN: 978-979-29-5180-6.
5. Staff. (2016). *Dronecode, the Open Source Unmanned Aerial Vehicle Project*. Tersedia di: www.open-electronics.org (diakses pada: 27 November 2018).
6. Sotomayor, J. Gómez, A. Castillo, A. (2014). *Visual Control of an Autonomous Aerial Vehicle for Crop Inspection*. *Revista Politécnica*, 33(1).
7. Putri, A. Sumiharto, R. Purwarupa Sistem Prediksi Luas dan Hasil Panen Padi suatu Wilayah menggunakan Pengolahan Citra Digital dengan Metode Sobel dan Otsu. *IJEIS*, Vol 6 No 2, Oktober 2014, pp 187-198. ISSN: 2088-3714.
8. Kushardono, D. (2014). Teknologi Akuisisi Data Pesawat Tanpa Awak Dan Pemanfaatannya Untuk Mendukung Produksi Informasi Penginderaan Jauh. *INDERAJA* Vol V No.7, Juli 2014.
9. Jatmika, S. Andiko, I. (2014). Simulasi Pengaturan Lampu Lalu Lintas Berdasarkan Data Image Processing Kepadatan Kendaraan Berbasis Mikrokontroler Atmega16. *Jurnal Imiah*

Teknologi dan Informasi ASIA, Malang Vol 8 No.2, Agustus 2014.

10. Venugopalan, T. Taher, T. Barbastathis, G. (2012). *Autonomous Landing of an Unmanned Aerial Vehicle on an Autonomous Marine Vehicle. Oceans*, 2012.
11. Shofyati, R. (2011). Teknologi Pesawat Tanpa Awak untuk Pemetaan dan

Pemantauan Tanaman dan Lahan Pertanian. *Informatika Pertanian, Badan Litbang Pertanian, Bogor* Vol 20 No.2, 58-64.

12. Theia Technologies. The Trade-off between Image Resolution and Field of View: the Influence of Lens Selection. Tersedia di: *Squarespace*