

IMPLEMENTASI *PROOF-OF-CONCEPT* SISTEM VISI KOMPUTER BERBASIS YOLO11n PADA *AUTONOMOUS QUADCOPTER* UNTUK MISI TERBANG PENDETEKSIAN MANUSIA

Yasmin Fitria Amanda^{1*}, Budi Hartono²

^{1,2} Jurusan Teknik Mesin, Program Studi Teknik Aeronautika, Politeknik Negeri Bandung
Jl. Gegerkalong Hilir, Ds. Ciwaruga, Kec. Parongpong, Kab. Bandung Barat, Jawa Barat

Email korespondensi: yasmin.fitria.aer23@polban.ac.id

Abstrak. Perkembangan *quadcopter* membuka peluang untuk mendukung misi pencarian dan penyelamatan karena mampu menjangkau area yang sulit dan berisiko bagi manusia. Penelitian ini bertujuan merancang dan mengevaluasi *proof-of-concept* sistem *autonomous quadcopter* yang terintegrasi dengan sistem visi komputer untuk mendeteksi manusia menggunakan algoritma *You Only Look Once* (YOLO). Sistem menggunakan DJI F450 konfigurasi X, *flight controller* Pixhawk 2.4.8, Mission Planner sebagai *Ground Control Station* (GCS), dan Raspberry Pi 5 sebagai unit komputasi. Deteksi manusia dilakukan menggunakan model YOLO11n *pretrained* COCO dalam format NCNN, sedangkan OpenCV digunakan untuk akuisisi, pemrosesan, dan visualisasi *frame*. Pengujian meliputi penerbangan manual, penyetelan PID, *waypoint*, *autonomous takeoff-landing*, uji statis, dan misi deteksi manusia. Pada satu skenario misi terbang dengan satu target manusia diam pada ketinggian 3 m dan kecepatan 0,3 m/s, *quadcopter* mencapai ketinggian aktual rata-rata 2,76 m dan kecepatan 0,34 m/s selama 70 detik. Hasil pendeteksian menunjukkan rata-rata *confidence score* 90,55%, *recall* 86,52%, *precision* 99,18%, FPS log sebesar 12,47, dan *overall throughput* 11,14 FPS. Hasil ini menunjukkan fungsi dasar sistem berhasil dijalankan, tetapi belum menggambarkan kinerja pada variasi target, lingkungan, pencahayaan, ketinggian, dan kondisi gerak.

Kata kunci: *quadcopter*, sistem otonom, visi komputer, deteksi manusia, YOLO11n

Abstract. The development of *quadcopters* opens up opportunities to support search and rescue missions because they can reach areas that are difficult and risky for humans. This research aims to design and evaluate a *proof-of-concept* *autonomous quadcopter* system integrated with a computer vision system to detect humans using the *You Only Look Once* (YOLO) algorithm. The system uses a DJI F450 X configuration, a Pixhawk 2.4.8 *flight controller*, Mission Planner as the *Ground Control Station* (GCS), and a Raspberry Pi 5 as the computing unit. Human detection is performed using the YOLO11n *pretrained* COCO model in NCNN format, while OpenCV is used for frame acquisition, processing, and visualisation. Testing includes manual flight, PID tuning, *waypoints*, *autonomous takeoff-landing*, static tests, and human detection missions. In one flight mission scenario with a stationary human target at a height of 3 m and a speed of 0.3 m/s, the *quadcopter* reached an average actual height of 2.76 m and a speed of 0.34 m/s over 70 seconds. The detection results show an average *confidence score* of 90.55%, *recall* of 86.52%, *precision* of 99.18%, a log FPS of 12.47, and an *overall throughput* of 11.14 FPS. These results indicate that the basic functions of the system have been successfully executed, but they do not yet reflect performance under variations in target, environment, lighting, altitude, and motion conditions.

Keywords: *quadcopter*, *autonomous system*, *computer vision*, *human detection*, YOLO11n

*Penulis Korespondensi

Diterima : 21 Juli 2026, Disetujui : 09 Agustus 2026, Dipublikasikan : 30 September 2026

1. Pendahuluan

Unmanned Aerial Vehicle (UAV) multirotor, khususnya *quadcopter*, memiliki kemampuan *vertical takeoff and landing*

(VTOL), *hovering*, serta bermanuver pada ruang terbatas. Kemampuan tersebut membuat *quadcopter* sesuai untuk pemantauan, inspeksi, serta pencarian dan

penyelamatan di area sulit dijangkau (Rafiq et al., 2020). Dalam skenario pencarian dan penyelamatan (*Search and Rescue*), quadcopter dapat memperluas pengamatan dan mengurangi risiko personel memasuki area berbahaya.

Penerbangan *autonomous* belum cukup apabila hanya mengikuti waypoint tanpa memperoleh informasi visual. Oleh karena itu, diperlukan sistem visi komputer untuk memproses citra kamera dan mendeteksi manusia secara *real-time* (Voulodimos et al., 2018). Salah satu algoritma deteksi objek yang banyak digunakan adalah *You Only Look Once* (YOLO), yang menghasilkan lokasi objek, kelas objek, dan *confidence score* dalam satu tahap inferensi (Redmon et al., 2016).

Sugandi mengembangkan quadcopter berbasis YOLOv5 pada Raspberry Pi 4 untuk mendeteksi manusia. Quadcopter mampu terbang stabil pada ketinggian 5 meter dan nilai mAP@0.5 yang diperoleh dari hasil training pengolahan citra adalah 86,7% (Sugandi & Hartono, 2022). Namun, sistem hanya menghasilkan kecepatan pemrosesan sebesar 4 hingga 6 *frames per second* (FPS). Misi *autonomous* juga masih dijalankan melalui *Ground Control Station* (GCS) dan belum menggunakan skrip Python untuk mengendalikan seluruh misi (Sugandi, 2022).

Kontribusi penelitian ini terletak pada implementasi *proof-of-concept* YOLO11n dalam format NCNN pada Raspberry Pi 5 yang terintegrasi dengan Pixhawk melalui MAVLink untuk menjalankan takeoff, navigasi waypoint, pemindaian target, kembali ke titik awal, dan landing secara *autonomous*. Perbandingan dengan penelitian terdahulu hanya digunakan sebagai referensi implementasi karena terdapat perbedaan model, perangkat keras, resolusi, format inferensi, dan kondisi pengujian.

Penelitian ini bertujuan merancang dan menguji *autonomous* quadcopter berbasis YOLO11n pada Raspberry Pi 5 untuk mendeteksi manusia secara *real-time*

(Nurharlita et al., 2025). Pengujian meliputi penerbangan manual, penyetelan nilai *Proportional-Integral-Derivative* (PID), navigasi waypoints, lepas landas dan pendaratan otomatis, serta misi deteksi manusia. Kinerja sistem dievaluasi berdasarkan error ketinggian, error kecepatan, keberhasilan satu misi, *confidence score*, FPS, *throughput frame*, serta vibrasi.

2. Metodologi

2.1. Waktu dan Tempat

Penelitian ini dilaksanakan pada bulan Februari hingga Juli 2026 di area terbuka dengan permukaan yang rata dan dilakukan pada waktu siang hari, yaitu Hanggar Teknik Aeronautika, Lapangan Depan Hanggar, dan Lapangan Belakang Hanggar. Seluruh kegiatan penelitian dilakukan di Kampus Politeknik Negeri Bandung, Kecamatan Parongpong, Kabupaten Bandung Barat.

2.2. Rancangan Sistem

Sistem terdiri atas sistem *autonomous quadcopter* dan sistem visi komputer. Sistem *autonomous quadcopter* menggunakan *frame* DJI F450 konfigurasi X, empat motor *brushless*, *Electronic Speed Controller* (ESC), Pixhawk 2.4.8, *Global Positioning System* (GPS), radio telemetri, baterai Li-Po 4S 5000 mAh, dan sensor jarak TF Mini LiDAR. Sistem visi komputer menggunakan Raspberry Pi 5 RAM 8 GB, webcam Logitech C270, serta catu daya 5 Volt dan 5 Ampere melalui modul *step-down*. Pemilihan komponen mempertimbangkan kompatibilitas antarmuka, kebutuhan daya, kemampuan navigasi, dan kapasitas komputasi untuk inferensi citra.

Frame konfigurasi X dipilih untuk menghasilkan respons *roll* dan *pitch* yang seimbang. Dua *propeller* berputar *clockwise* dan dua lainnya *counter-clockwise* untuk menyeimbangkan momen putar. Perubahan kecepatan motor menghasilkan gerak quadcopter, sedangkan

Pixhawk dan *Electronic Speed Controller* (ESC) mengatur putaran motor melalui sinyal *Pulse Width Modulation* (PWM). Pixhawk dipasang di pusat *frame* menggunakan *rubber damper* untuk mereduksi vibrasi, sedangkan Global Positioning System (GPS) ditempatkan

pada stand di atas wahana agar lebih jauh dari ESC dan kabel daya untuk menghindari gangguan elektromagnetik. Tabel 1 menunjukkan daftar komponen quadcopter beserta spesifikasinya, sedangkan Gambar 1 menunjukkan skematik rancangan rakitan sistem quadcopter

Tabel 1. Konfigurasi perangkat autonomous quadcopter

Komponen	Spesifikasi
Rangka	DJI F450, konfigurasi X
Gimbal	Tarot GOPRO T4-3D Dual suspension 3 axis Gimbal TL3D02
<i>Flight Controller</i>	Pixhawk 2.4.8 PX4 32-bit
<i>Motor Brushless</i>	Sunnysky X2216 1400 kV
Komputer onboard	Raspberry Pi 5, RAM 8 GB, OS 64-bit
ESC	Diatone Mamba 50A ESC 4 in 1
<i>Propeller</i>	12×4.5-inch pitch Pair CW CCW blade nylon fiber propeller
Kamera	Logitech C270 HD Webcam, 640 x 480 piksel
Navigasi	GPS Ublox Neo M7N dan Mission Planner
Sensor tinggi	TF Mini LiDAR
Model deteksi	YOLO11n <i>pretrained</i> COCO
Perangkat lunak	Python 3.11, OpenCV 4.12, Ultralytics, Flask
Catu daya	Li-Po 4S 5000 mAh; <i>step-down</i> 5V 5A
<i>Power Module</i>	3DR Ardupilot Power Module 6-18v 60 A
Radio Telemetri	RFD 900X
<i>Radio Control</i>	Flysky FS-i6X 10 CH plus FS-IA10B transmitter
<i>Firmware</i>	Quadcopter, <i>STABLE-4.6.3</i>
Komputasi	Python 3.11.9, Ultralytics 8.4.50, OpenCV 4.13.0.92
Performa	15 menit



Gambar 1. Skematik rancangan rakitan sistem quadcopter

2.3. Konfigurasi Sistem *Autonomous Quadcopter*

Pixhawk dikonfigurasi melalui Mission Planner dengan firmware ArduCopter, *frame type X*, kalibrasi *accelerometer*, *compass*, *radio control*, ESC, *flight mode*, dan *failsafe*. Penerbangan awal dilakukan pada mode *Stabilize* untuk memeriksa arah putaran motor, respons kendali, dan kestabilan dasar.

Raspberry Pi dihubungkan ke Pixhawk melalui kabel JST pada port /dev/ttyACM0 dengan *baud rate* 115200. Komunikasi perintah dan telemetri menggunakan protokol MAVLink yang dikelola melalui MAVProxy, DroneKit, dan PyMAVLink. Program Python mengaktifkan mode *GUIDED*, melakukan *arming*, *takeoff*, mengirim koordinat *waypoint*, memantau posisi, dan menjalankan perintah *LAND* setelah misi selesai. Mission Planner digunakan sebagai *Ground Control Station* untuk memantau kondisi penerbangan. (Pulungan et al., 2024).

Aspek keselamatan dikonfigurasi melalui *battery failsafe* dan *RC failsafe*. *Battery failsafe* memerintahkan pendaratan apabila tegangan mencapai batas bawah 14,0 V, sedangkan *Radio Control (RC) failsafe* menangani kehilangan sinyal radio dengan perintah *return to launch*. Tahap *autonomous* dilakukan setelah penerbangan manual dan *waypoint* berbasis Mission Planner dinyatakan bagus.

2.4. Sistem Visi Komputer Berbasis YOLO11n

Sebagai referensi, pengujian resmi pada COCO menunjukkan YOLO11n memperoleh mAP@50–95 sebesar 39,5, lebih tinggi daripada YOLOv8n sebesar 37,3 (Ultralytics, 2025) dan YOLOv5n sebesar 28,0. YOLO11n juga memiliki 2,6 juta parameter dan 6,5 GFLOPs, lebih ringan daripada YOLOv8n yang memiliki 3,2 juta parameter dan 8,7 GFLOPs, meskipun masih lebih besar daripada YOLOv5n. Dengan demikian, keunggulan YOLO11n terutama ditunjukkan melalui

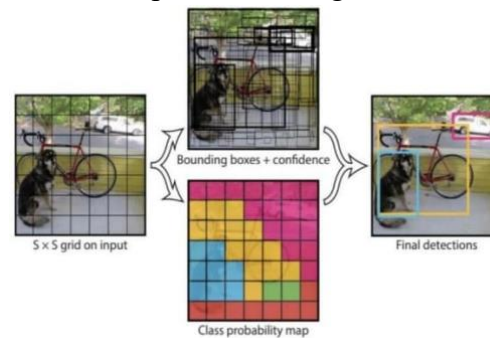
keseimbangan antara akurasi dan efisiensi komputasi. Penelitian Harminto et al. (2025) yang membandingkan YOLOv5, YOLOv8, dan YOLOv11 pada segmentasi citra reptil menunjukkan bahwa YOLOv11 memperoleh recall tertinggi sebesar 0,725 dan mAP@50 tertinggi sebesar 0,706. Nilai tersebut lebih tinggi daripada mAP@50 YOLOv5 sebesar 0,543 dan YOLOv8 sebesar 0,617 (Harminto et al., 2025).

Raspberry Pi 5 menggunakan OS Lite 64-bit agar sumber daya sistem lebih banyak tersedia untuk proses inferensi. Lingkungan virtual Python memuat Ultralytics, NCNN, OpenCV-headless, NumPy, dan Flask. Model YOLO11n *pretrained* COCO, yaitu model yang telah dilatih oleh Ultralytics menggunakan *dataset* COCO untuk tugas *object detection*. Model yolol1n.pt diekspor ke format NCNN menggunakan Ultralytics versi 8.4.50 dengan perintah “yolo export model=yolol1n.pt format=ncnn”, kemudian dijalankan untuk kelas *person* pada program. Inferensi menggunakan *imgsz* berukuran 640 × 480 piksel. Model akan memproses inputan untuk menghasilkan *bounding box*, label kelas, dan *confidence score*.

YOLO mendeteksi objek *input* dari citra yang diperoleh kamera melalui satu lintasan inferensi untuk menghasilkan *bounding box*, probabilitas kelas, dan *confidence score* (Sarosa & Muna, 2021). Pada penelitian ini digunakan YOLO11 yang memiliki tiga bagian utama, yaitu *backbone* untuk mengekstraksi fitur citra, *neck* untuk menggabungkan fitur pada berbagai skala, serta *detection head* berbasis *anchor-free* untuk memprediksi lokasi dan kelas objek. Ketepatan posisi *bounding box* dievaluasi menggunakan Intersection over Union (IoU), yaitu perbandingan antara luas irisan dan luas gabungan kotak prediksi dengan *ground truth*. Sementara itu, pada proses *Non-Maximum Suppression* (NMS), IoU digunakan untuk membandingkan sesama kotak prediksi dan menghapus kotak yang

bertumpang tindih agar tidak terjadi deteksi ganda. Sistem kemudian hanya mempertahankan kelas person dengan *class ID* 0 agar hasil deteksi berfokus pada

manusia dan mengurangi keluaran yang tidak relevan (Khairunnas et al., 2021). Gambar 2 menunjukkan tahapan kerja algoritma YOLO.



Gambar 2. Tahapan kerja algoritma YOLO

Pada program, flask menyediakan *video streaming* melalui jaringan lokal, sedangkan modul *Comma-Separated-Values* (CSV) dan Matplotlib menyimpan log serta grafik evaluasi. Pemrosesan dilakukan secara *onboard* sehingga keputusan deteksi tidak bergantung pada koneksi internet. Jaringan *router* hanya digunakan untuk akses SSH, pemantauan, dan pengambilan data.

2.5. Teknik Pengumpulan Data

Data penelitian dikumpulkan dari dua sumber, yaitu data penerbangan yang direkam melalui Mission Planner dan data hasil deteksi YOLO11 yang direkam oleh Raspberry Pi. Data penerbangan diperoleh dari log Pixhawk yang diunduh melalui Mission Planner dalam bentuk *DataFlash log* (.bin) atau *telemetry log* (.tlog). Data tersebut mencakup *timestamp*, posisi GPS berupa *latitude* dan *longitude*, ketinggian,

kecepatan, mode penerbangan, status waypoint, *attitude* aktual berupa *roll*, *pitch*, dan *yaw*, *desired attitude*, nilai vibrasi pada sumbu X, Y, dan Z.

Data deteksi dicatat oleh Raspberry Pi pada setiap frame yang berhasil diproses. Field yang disimpan meliputi *timestamp*, nomor frame, FPS, ID dan nama kelas, *confidence score*, titik pusat objek, koordinat *bounding box*, status deteksi, serta gambar tiap frame. Model hanya mendeteksi kelas *person* dengan *class ID* 0 yaitu manusia. Setelah data terkumpul, selanjutnya akan dilakukan pengolahan data yaitu dengan membuat *ground truth annotation* pada setiap frame yang terproses atau memberi label secara manual pada data asli untuk mengetahui apakah hasil deteksi model benar atau salah. Setiap frame akan dikelompokkan sesuai kondisinya seperti pada Tabel 2.

Tabel 2. Pengelompokkan frame berdasarkan kondisi

Kondisi Frame	Status
Tidak ada manusia dalam frame	Tidak termasuk evaluasi
Ada manusia, terdeteksi $confidence < threshold$	FN (<i>False Negative</i>)
Ada manusia, terdeteksi $confidence \geq threshold$	TP (<i>True Positif</i>)
Tidak ada manusia, terdeteksi $confidence \geq threshold$	FP (<i>False Positif</i>)

Setelah pengelompokkan frame, dilakukan perhitungan nilai *detection rate* atau *recall* dan *precision* dengan rumus:

$$Recall = \frac{TP}{TP + FN} \times 100\%$$

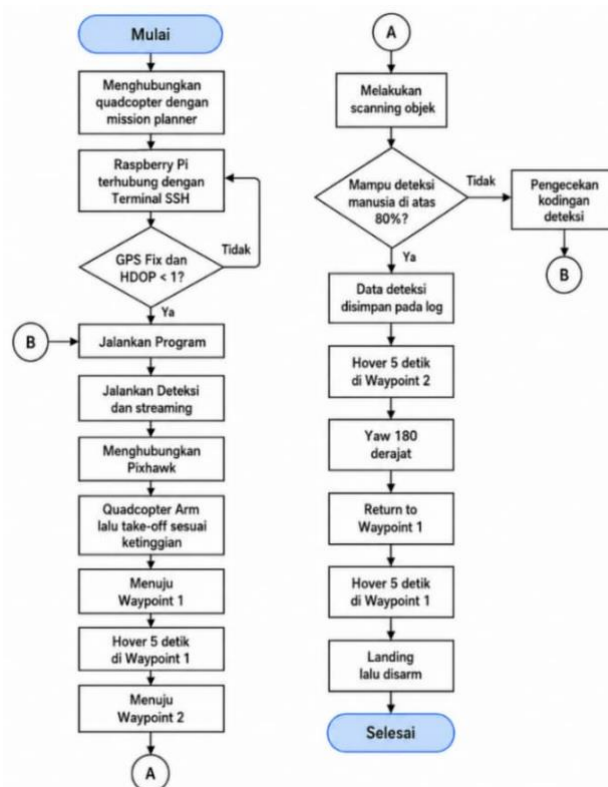
$$Precision = \frac{TP}{TP + FP} \times 100\%$$

Recall menunjukkan persentase manusia yang berhasil dideteksi dari seluruh manusia yang sebenarnya terdapat pada citra. *Precision* menunjukkan persentase prediksi manusia yang benar dari seluruh hasil deteksi yang dihasilkan model.

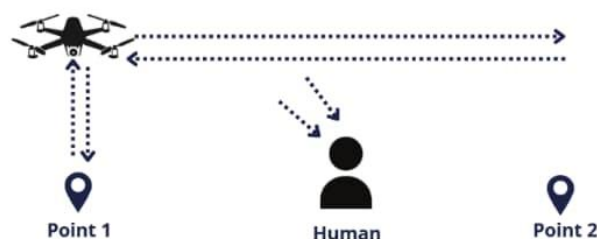
2.6. Prosedur Pengujian

Pengujian dilaksanakan bertahap: (1) Sekali uji terbang perdana pada ketinggian 1 hingga 6 m; (2) Sekali uji terbang setelah penyetelan Nilai PID pada ketinggian 1 hingga 6 m; (3) Sekali uji terbang *waypoint* menggunakan Mission Planner pada ketinggian 10 m; (4) Sekali uji terbang *autonomous takeoff-landing* pada

ketinggian 1 m dengan *hover* 10 detik; (5) Sekali *ground test* model selama 20 detik pada jarak 4 m; dan (6) Sekali uji misi terbang deteksi manusia pada ketinggian input 3 m. Pada uji misi terbang, quadcopter bergerak dari titik *takeoff* menuju *waypoint* 1, kemudian *waypoint* 2, lalu kembali ke *waypoint* 1, dan *landing*. Kecepatan input terbang adalah 0,3 m/s. Target berupa seorang manusia dalam keadaan diam yang ditempatkan di tengah jalur pemindaian, sedangkan kamera diarahkan ke bawah. Gambar 3 dan Gambar 4 menunjukkan diagram alir dan gambaran skema uji misi terbang deteksi manusia



Gambar 3. Diagram alir uji misi terbang deteksi manusia



Gambar 4. Skema uji misi terbang deteksi manusia

3. Hasil dan Pembahasan

3.1. Penerbangan Perdana dan Penyetelan Nilai PID

Quadcopter berhasil dirakit tanpa indikasi *short circuit* pada rangkaian dan

semua komponen terintegrasi dengan baik. Gambar 5 menunjukkan hasil rakitan quadcopter.



Gambar 5. Hasil rakitan quadcopter

Setelah quadcopter selesai dirakit dan integrasi dengan mission planner, kemudian dilakukan uji terbang perdana secara manual untuk melihat *attitude* terbang quadcopter pertama kali sebelum dilakukan penyetelan Nilai PID. Hasil pengujian menunjukkan bahwa grafik *roll*, *pitch*, dan *yaw* masih mengalami *overshoot* ketika menerima *input* pilot dan masih belum sepenuhnya halus dan terlambat dalam mengikuti perintah yang diberikan. Kondisi ini mengindikasikan bahwa sistem kontrol belum stabil, yaitu sistem belum mampu meredam perubahan gerak secara optimal. Dan pada grafik *yaw* menjelang akhir pengujian terdapat lonjakan nilai *yaw* hingga mendekati 350° sampai 360° . Lonjakan ini menunjukkan adanya perubahan arah hadap atau putaran quadcopter. Sudut *yaw* biasanya dibaca dalam rentang 0° sampai 360° , sehingga perubahan dari sekitar 0° menuju 350° dapat terlihat seperti lonjakan besar, padahal secara arah bisa berarti quadcopter berputar mendekati batas sudut 360° . Nilai vibrasi rata-rata tercatat $3,46 \text{ m/s}^2$ pada sumbu X, $5,62 \text{ m/s}^2$ pada sumbu Y, dan $13,84 \text{ m/s}^2$ pada sumbu Z.

Untuk terbang yang lebih stabil, parameter PID kemudian disetel secara manual hingga respons aktual *roll*, *pitch*, dan *yaw* dapat mengikuti nilai target dengan *overshoot* yang lebih kecil. Karena berdasarkan data, gerakan *roll*, *pitch*, dan

yaw quadcopter kurang halus mengikuti gerakan yang diinput RC maka nilai P dan I diturunkan. Nilai akhir *rate roll* adalah $P=0,030$, $I=0,030$, $D=0,0022$; *rate pitch* $P=0,020$, $I=0,020$, $D=0,0022$; dan *rate yaw* $P=0,180$, $I=0,018$, $D=0,000$. Hasil uji terbang setelah penyetelan menunjukkan respons *attitude* menjadi lebih halus dan nilai aktual lebih dekat dengan nilai yang diinginkan. Secara umum, perubahan target *roll*, *pitch*, dan *yaw* diikuti lebih cepat dan *overshoot* tidak berlangsung lama. Artinya, respons gerakan sudah membaik dan sistem tidak lagi terlalu lambat dalam merespons perubahan gerak.

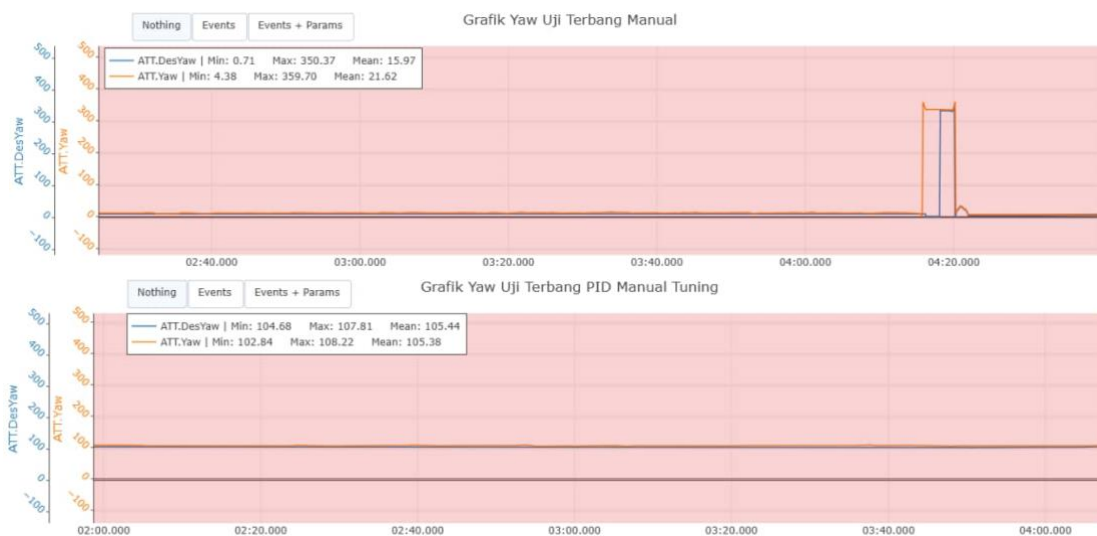
Nilai vibrasi rata-rata setelah penyetelan tercatat $3,46 \text{ m/s}^2$ pada sumbu X, $5,62 \text{ m/s}^2$ pada sumbu Y, dan $13,84 \text{ m/s}^2$ pada sumbu Z. Nilai tersebut masih berada dalam batas toleransi berdasarkan panduan pengukuran getaran ArduPilot (Ardupilot, n.d.). Nilai vibrasi rata-rata berada di bawah 30 m/s^2 . Namun, vibrasi maksimum sumbu Z mencapai $32,87 \text{ m/s}^2$ pada fase tertentu seperti pada saat *take-off* dan *landing* sehingga terdapat lonjakan sesaat yang melampaui batas rekomendasi tersebut. Gambar 6 menunjukkan perbedaan respon gerak *roll* sebelum penyetelan PID dan setelah penyetelan. Gambar 7 menunjukkan perbedaan respon gerak *pitch* dan Gambar 8 menunjukkan perbedaan respon gerak *yaw*.



Gambar 6. Grafik roll (derajat) terhadap waktu (detik) sebelum dan setelah pengaturan PID



Gambar 7. Grafik pitch (derajat) terhadap waktu (detik) sebelum dan setelah pengaturan PID

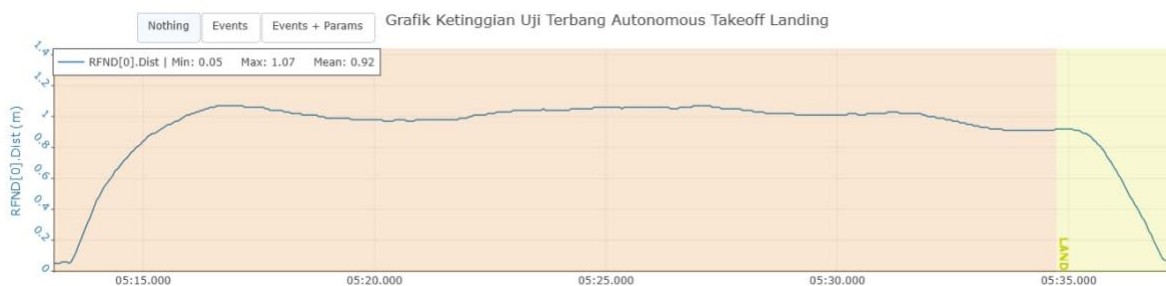


Gambar 8. Grafik yaw (derajat) terhadap waktu (detik) sebelum dan setelah pengaturan PID

3.2. Penerbangan *Waypoints* dan *Takeoff Landing*

Pengujian waypoint dilakukan untuk mengetahui kemampuan sistem navigasi quadcopter dalam mengikuti koordinat yang telah ditentukan. Pengujian dilaksanakan pada ketinggian 10 m dengan empat titik waypoint berbentuk persegi panjang yang dibuat melalui Mission Planner. Quadcopter diperintahkan menuju setiap waypoint secara berurutan, kemudian menjalankan perintah *Return-to-Launch* setelah seluruh titik selesai dilewati. Hasil pengujian menunjukkan bahwa quadcopter mampu menuju keempat *waypoints* dan kembali ke titik awal penerbangan. Temuan ini menunjukkan bahwa GPS dan *flight controller* mampu menjalankan rute sesuai misi yang dirancang. Namun, tingkat keakuratan GPS belum dapat ditentukan hanya berdasarkan keberhasilan mencapai *waypoints* karena diperlukan perhitungan selisih antara koordinat target dan posisi aktual.

Pengujian autonomous *takeoff-landing* dilakukan untuk mengetahui kemampuan quadcopter dalam melakukan lepas landas, mempertahankan ketinggian, dan mendarat secara otomatis pada mode GUIDED. Ketinggian target ditetapkan sebesar 1 m, kemudian quadcopter melakukan *hover* selama 10 detik sebelum menjalankan proses pendaratan. Berdasarkan Gambar 9, menunjukkan data hasil pembacaan sensor LiDAR yang mengikuti perubahan ketinggian dalam meter selama fase naik, *hover*, dan turun. Hasil pengujian menunjukkan bahwa quadcopter mampu *takeoff* hingga ketinggian 1,07 meter, mempertahankan ketinggian atau *hover* selama 10 detik dengan rata-rata ketinggian 0,92 meter, kemudian *landing* secara otomatis. Error ketinggian sebesar 0,08 m dari 1 m atau sebesar 8%. Gambar 9 menunjukkan grafik ketinggian terbang dalam meter terhadap waktu penerbangan dalam detik.



Gambar 9. Grafik ketinggian terbang (m) terhadap waktu (detik) uji terbang takeoff landing

3.3. Uji Misi Terbang Deteksi Manusia

Pada uji misi terbang deteksi manusia untuk simulasi *Search and Rescue* (SAR), pengujian dilakukan sekali pada ketinggian 3 m dengan *waypoint* 1 pada titik koordinat (-6.873543, 107.573186) dan *waypoint* 2 pada titik koordinat (-6.873552, 107.573106) kemudian *landing* pada *waypoint* 1. Waypoints tersebut ditentukan menggunakan posisi quadcopter yang diletakkan pada 2 titik dengan jarak 10 m diukur menggunakan meteran dan titik koordinat dilihat pada mission planner. Hasil pengujian menunjukkan bahwa quadcopter dapat menjalankan misi terbang

dari *waypoint* 1 ke 2 hingga kembali lagi ke *waypoint* 1. Dengan program disetting pada proses *takeoff*, drone dianggap telah mencapai ketinggian target ketika ketinggian aktual mencapai minimal 95% dari nilai yang ditentukan yaitu target 3 m dianggap tercapai pada ketinggian minimal 2,85 m. Sementara itu, *waypoint* dianggap tercapai apabila selisih gabungan antara koordinat aktual drone dan koordinat tujuan lebih kecil dari 0,00003 derajat, yang secara kasar setara dengan jarak sekitar 3,3 m di lokasi pengujian. Penggunaan toleransi tersebut bertujuan mencegah sistem terus menunggu akibat fluktuasi sensor GPS dan ketinggian, sehingga misi dapat dilanjutkan

meskipun posisi drone tidak berada tepat pada nilai target. Selama misi terbang, quadcopter memiliki ketinggian rata-rata 2,76 m dari target 3 m dan kecepatan rata-rata 0,34 m/s dari target 0,3 m/s. Error ketinggian sebesar 0,24 m atau 8%. Error kecepatan sebesar 0,04 m/s atau 13,33%. Perbedaan ini dapat terjadi karena faktor eksternal seperti pengaruh angin dan ketelitian GPS, sehingga quadcopter melakukan koreksi kontrol. Gambar 10

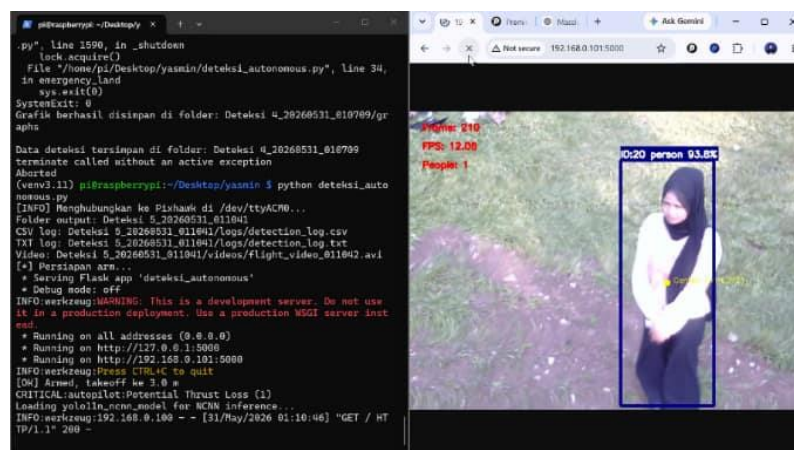
memperlihatkan perubahan ketinggian selama fase *takeoff*, penerbangan pada lintasan *waypoints*, dan *landing* secara otomatis. Tabel 3 menunjukkan ringkasan data hasil uji misi terbang. Model untuk deteksi yaitu YOLO11n juga dapat berjalan dan berhasil mendeteksi objek dengan tampilan streaming video yang dapat dilihat pada Gambar 11.



Gambar 10. Grafik ketinggian terbang (m) terhadap waktu (detik) uji misi terbang

Tabel 3. Data terbang hasil uji misi terbang deteksi manusia

Parameter	Hasil
Ketinggian target	3,00 m
Ketinggian aktual maksimum	3,19 m
Ketinggian aktual rata-rata	2,76 m
Kecepatan target	0,30 m/s
Kcepatan aktual maksimum	0,19 m/s
Kecepatan aktual rata-rata	0,34 m/s
Jarak antar-waypoint	10 m
Durasi misi	70 detik
Vibrasi sumbu X	2,09 m/s ²
Vibrasi sumbu Y	4,52 m/s ²
Vibrasi sumbu Z	14,81 m/s ²



Gambar 11. Tampilan layar selama uji misi terbang

3.4.

3.5. Kinerja Pendeteksian

Uji deteksi statis atau *Ground test* dilakukan untuk mengevaluasi kinerja sistem deteksi ketika objek dan quadcopter berada dalam kondisi diam. Pengujian dilaksanakan selama 20 detik pada jarak 4 m dengan nilai *confidence threshold* sebesar 70%. Pengujian menghasilkan 230 *frame* dan menunjukkan bahwa YOLO11n mampu mendeteksi manusia dengan rata-rata *confidence score* sebesar 89,27%

dengan nilai *detection rate* atau *recall* sebesar 100% serta *precision* sebesar 100%. Nilai tersebut menunjukkan bahwa seluruh objek manusia berhasil terdeteksi tanpa menghasilkan *False Negative* dan *False Positive* selama pengujian. Hasil ini juga menunjukkan bahwa Raspberry Pi 5 mampu menjalankan model YOLO11n secara *real-time* pada resolusi 640×480 piksel dalam kondisi pengujian statis. Tabel 4 menunjukkan performa deteksi manusia pada uji statis atau *ground test*.

Tabel 4. Performa deteksi manusia pada uji statis

Parameter	Nilai
Total <i>frame</i>	230
<i>Frame</i> berisi manusia	230
Jumlah <i>frame</i> terdeteksi	230
<i>True Positive</i>	230
<i>False Negative</i>	0
<i>False Positive</i>	0
<i>Recall</i>	100%
<i>Precision</i>	100%

Kemudian misi terbang dilakukan pada objek yang diam dan juga mengevaluasi kinerja pendeteksian pada sistem yang bergerak dengan *threshold* 80%, karena saat misi terbang citra yang dihasilkan kamera lebih rentan terhadap getaran, *motion blur*, perubahan sudut pandang, jarak objek, dan latar belakang yang bergerak. Ambang yang lebih tinggi membuat sistem hanya mempertahankan deteksi dengan tingkat keyakinan kuat sehingga *False Positive* dapat dikurangi dan hasil deteksi lebih andal.

Pengujian berlangsung selama 70 detik dan sistem visi komputer memproses 780 *frame*. Model bekerja baik saat wahana menuju *waypoint* 2 maupun ketika kembali ke *waypoint* 1. Hasil menunjukkan bahwa model memiliki tingkat keyakinan deteksi yang tinggi, dengan rata-rata *confidence score* sebesar 90,55%. Nilai *precision* sebesar 99,18% menunjukkan bahwa

hampir seluruh prediksi manusia yang dihasilkan model merupakan deteksi yang benar, sehingga jumlah *False Positive* sangat rendah. Sementara itu, nilai *recall* sebesar 86,52% menunjukkan bahwa model berhasil mendeteksi sebagian besar objek manusia, tetapi masih melewatkan sekitar 13,48% objek atau 19 *frame* yang seharusnya terdeteksi. Manusia dengan *confidence* di bawah *threshold* 80% tetap dianggap terdeteksi oleh model pada level *raw output*, tetapi dianggap *False Negative* pada evaluasi sistem karena tidak menghasilkan deteksi valid setelah *threshold* diterapkan. Dengan demikian, model memiliki ketepatan deteksi yang sangat baik saat quadcopter ditrbangkan atau dalam keadaan sistem yang bergerak. Tabel 5 menunjukkan performa deteksi manusia pada uji misi terbang pendeteksian manusia

Tabel 5. Performa deteksi manusia pada uji misi terbang

Parameter	Nilai
Total <i>frame</i>	780
<i>Frame</i> berisi manusia	141
Jumlah <i>frame</i> terdeteksi	123
<i>True Positive</i>	122
<i>False Negative</i>	19
<i>False Positive</i>	1
<i>Recall</i>	86,52%
<i>Precision</i>	99,18%

Kemudian kecepatan pemrosesan dievaluasi menggunakan dua pendekatan. Rata-rata FPS yang tercatat pada log program menunjukkan nilai rata-rata sebesar 12,42 FPS pada *ground test* dan 12,47 FPS pada misi terbang. Sementara itu, berdasarkan perbandingan jumlah

frame yang diproses terhadap durasi pengujian, rata-rata *overall throughput* sistem sebesar 11,50 FPS pada *ground test* dan 11,14 FPS pada misi terbang. Tabel 6 menunjukkan beberapa *frame* hasil pendeteksian objek saat *ground test* dan saat uji misi terbang.

Tabel 6. Tampilan frame hasil pendeteksian

<i>Ground Test</i>		Misi Terbang	
<i>Frame Ke-</i>	Hasil Deteksi	<i>Frame Ke-</i>	Hasil Deteksi
50	 Confidence Score: 89,5% FPS: 12,39	226	 Confidence Score: 91,1% FPS: 12,35
100	 Confidence Score: 89,4% FPS: 12,34	229	 Confidence Score: 92,4% FPS: 12,18

150	 Confidence Score: 89,4% FPS: 12,36	457	 Confidence Score: 93,1% FPS: 12,34
200	 Confidence Score: 89,4% FPS: 12,42	459	 Confidence Score: 94,0% FPS: 12,41

Pada *ground test*, contoh tampilan hasil deteksi diambil pada interval 50 *frame*, yaitu setiap 50 *frame* yaitu pada *frame* ke-50, 100, 150, dan 200. Kemudian tampilan *frame* 226 dan 229 pada uji misi terbang yaitu hasil pendeteksian ketika quadcopter menuju ke *waypoint* 2. Sedangkan *frame* 457 dan 459 yaitu hasil pendeteksian ketika quadcopter kembali ke *waypoint* 1.

Penggunaan ambang 70% pada pengujian statis meningkatkan sensitivitas sehingga objek yang terdeteksi dengan keyakinan sedang tetap dicatat untuk evaluasi kinerja deteksi. Pada uji misi terbang, ambang 80% digunakan untuk menyaring prediksi yang muncul akibat getaran, *motion blur*, perubahan sudut kamera, dan latar belakang yang bergerak. Konsekuensinya, ambang 80% memang dapat mengurangi *false positive*, tetapi juga berpotensi meningkatkan *false negative*, yaitu manusia sebenarnya ada tetapi tidak terdeteksi karena nilai *confidence* berada di bawah 80%. Dengan demikian, penetapan nilai *threshold* perlu didahului oleh pengujian pada beberapa variasi ambang

confidence dan kondisi untuk mengetahui pengaruhnya terhadap jumlah *True Positive*, *False Positive*, dan *False Negative*. Nilai *threshold* kemudian dipilih berdasarkan keseimbangan antara sensitivitas deteksi dan risiko *False Positive*, sehingga sistem tidak terlalu mudah menerima prediksi yang salah maupun terlalu ketat hingga melewati objek yang seharusnya terdeteksi (Affandi & Hartono, 2023).

4. Simpulan dan Saran

4.1. Simpulan

Penelitian ini menunjukkan *proof-of-concept* integrasi *autonomous quadcopter* berbasis Pixhawk 2.4.8, Raspberry Pi 5, dan YOLO11n untuk menjalankan tahapan misi yang meliputi *takeoff*, navigasi menuju *waypoint*, pemindaian target, deteksi manusia, kembali ke titik awal, dan *landing* secara otomatis menggunakan skrip Python. Penyetelan parameter PID meningkatkan kestabilan gerak karena sistem dapat mengoreksi perubahan sudut dan gangguan

pada sumbu *roll*, *pitch*, dan *yaw*, sehingga orientasi serta tingkat vibrasi wahana berada dalam batas yang mendukung pengambilan citra. Pada satu skenario pengujian misi terbang dengan satu target manusia dalam keadaan diam dengan ketinggian target 3 m dan kecepatan target 0,3 m/s, quadcopter mencapai ketinggian aktual rata-rata sebesar 2,76 m dan kecepatan aktual rata-rata sebesar 0,34 m/s.

Sistem visi komputer menggunakan YOLO11 *pretrained* COCO dalam format NCNN yang dijalankan pada Raspberry Pi 5 menghasilkan 122 frame deteksi valid dari total 780 frame selama misi 70 detik. Hasil menunjukkan rata-rata *confidence score* sebesar 90,55%, *detection rate* atau *recall* sebesar 86,52%, dan *precision* sebesar 99,18%. Rata-rata kecepatan pemrosesan pada data deteksi tercatat sebesar 12,47 FPS, sedangkan *overall throughput* berdasarkan keseluruhan *frame* selama misi sebesar 11,14 FPS. Hasil tersebut menunjukkan bahwa fungsi dasar integrasi sistem telah berhasil dijalankan, tetapi belum cukup untuk menggambarkan kinerja sistem pada variasi jumlah dan kondisi target, lingkungan, pencahayaan, ketinggian, serta kondisi gerak yang berbeda.

4.2. Saran

Pengembangan penelitian selanjutnya perlu dilakukan melalui pengujian berulang pada setiap skenario untuk mengevaluasi *repeatability* dan meningkatkan validitas hasil. Pengujian juga perlu diperluas dengan menggunakan beberapa target manusia, target bergerak, variasi pakaian, variasi ketinggian terbang, serta kondisi pencahayaan yang berbeda untuk mengetahui pengaruh faktor lingkungan terhadap kinerja model (Jeremia Zophie, 2022). Evaluasi sistem juga perlu dilengkapi dengan pengukuran *latency end-to-end* mulai dari akuisisi citra, proses inferensi, hingga keluaran deteksi agar kemampuan *real-time* dapat dinilai secara lebih objektif. Selain itu, sistem

dapat dikembangkan dengan integrasi koordinat lokasi target hasil deteksi untuk mendukung proses pencarian, penerapan pola pencarian yang lebih luas, serta pengujian pada lingkungan semi-kompleks yang mendekati kondisi pencarian dan penyelamatan sebenarnya.

Daftar Pustaka

- Ardupilot. (n.d.). *Measuring Vibration*.
Ardupilot. Retrieved May 6, 2026, from
<https://ardupilot.org/plane/docs/common-measuring-vibration.html>
- Harminto, T., Najib, D. F., Wicaksono, P., & Izdiyar, Z. N. (2025). Performance comparison of YOLOv8, YOLOv5 and YOLOv11 in nature image segmentation. *Journal of Image and Graphics*, 13(5), 515–520.
- Jeremia Zophie, H. H. T. (2022). Implementasi Algoritma You Only Look Once (YOLO) menggunakan Web Camera untuk Mendeteksi Objek Statis dan Dinamis. *Jurnal TNI Angkatan Udara*, 1(1), 98–109.
- Khairunnas, K., Yuniarno, E. M., & Zaini, A. (2021). Pembuatan modul deteksi objek manusia menggunakan metode yolo untuk mobile robot. *Jurnal Teknik ITS*, 10(1), A50–A55.
- Nurharlita, S., Widhiantoro, D., Multimedia, B., Elektro, T., Jakarta, P. N., & Depok, K. (2025). Analisa Komparatif Algoritma YOLO dan SSD untuk Sistem Deteksi Objek Abstrak. *SNIV: Seminar Nasional Inovasi Vokasi*, 4(1), 759–767.
- Pulungan, A. B., Putra, Z. Y., Sidiqi, A. R., Hamdani, & Parigalan, K. E. (2024). Drone Kit-Python for Autonomous Quadcopter Navigation. *International Journal on Informatics Visualization*, 8(3).
<https://doi.org/10.62527/joiv.8.3.2301>
- Rafiq, A. A., Riyanto, S. D., Aprilas, B. D., & Pratama, R. P. (2020). Image Processing untuk Deteksi Objek pada Daerah Bencana. *INVOTEK: Jurnal*

- Inovasi Vokasional Dan Teknologi*, 20(2), 9–18.
<https://doi.org/10.24036/invotek.v20i2.707>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, Real-Time Object Detection*.
- Sarosa, M., & Muna, N. (2021). Implementasi algoritma You Only Look Once (YOLO) untuk deteksi korban bencana alam. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 8(4), 787–792.
- Sugandi, A. N. (2022). Quadcopter v7: Kaji Pengolahan Citra untuk Misi Terbang Pendeteksian Manusia pada Daerah Bencana. *Politeknik Negeri Bandung, Tugas Akhir*.
- Sugandi, A. N., & Hartono, B. (2022). Implementasi Pengolahan Citra pada Quadcopter untuk Deteksi Manusia Menggunakan Algoritma YOLO. *Prosiding The 13th Industrial Research Workshop and National Seminar*, 183–188.
- Ultralytics. (2025). *YOLOv8 vs YOLOv11*. Ultralytics.
<https://docs.ultralytics.com/compare/yolo11-vs-yolov8/>
- Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, 2018(1), 7068349.