

Nama : Abrar Abhirama Widyadhana
Nim : 13523038

semua jawaban code: <https://github.com/Abrar-Abhirama/TugasCaAksan>

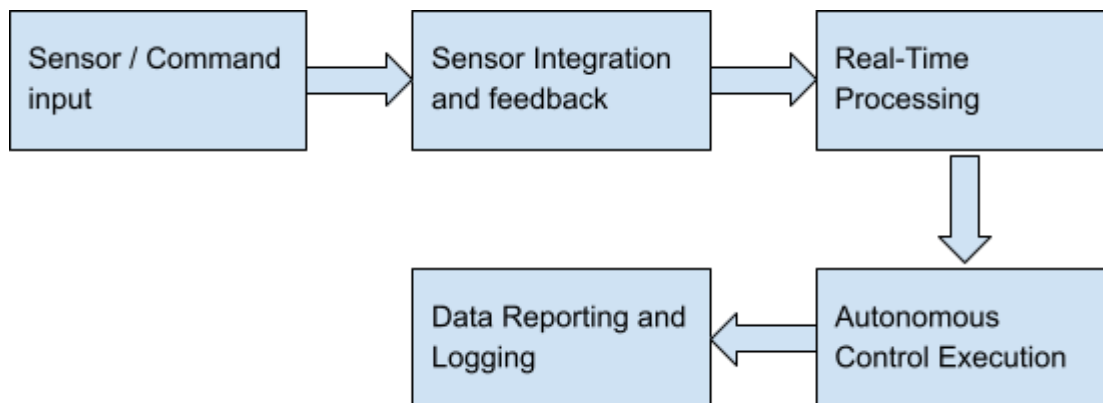
SOAL WAJIB

1. Bagaimana program RSC drone menganalisis lingkungan secara real-time dan mengambil keputusan berdasarkan data dari lingkungan? Buatlah Diagram! (5 poin).

Karena konteks RSC tidak dijelaskan secara spesifik maka asumsi

Robotic Software Control (RSC)

- Sensor / Command input, drone sudah dilengkapi dengan sensor - sensor dan sudah di program untuk navigasi, pemantauan dan pengumpulan data
- Sensor Integration and feedback, drone akan mengumpulkan data dan memberikannya kembali ke system untuk di update
- Real-Time Processing, perangkat akan memproses data dan tindakan yang akan dilakukan sesuai dengan apa yang telah di program berdasarkan data yang diterima secara langsung
- Autonomous Control Execution, Drone akan melakukan tindakan seperti mengubah arah gerakan dan lain lain
- Data Reporting and Logging, kemudian dengan menggunakan ROS, setiap gerakan, tindakan, dan data yang diterima, akan dikirimkan ke operator melalui system dan segala tindakan terbaca dan terdata



2. Jelaskan apa itu Object Oriented Programming (OOP)! Jelaskan juga apa keuntungan OOP dibandingkan Prosedural Programming! (5 poin)

OOP adalah suatu paradigma pemrograman yang mengorganisasikan kode berdasarkan objek-objek nyata yang mewakili entitas dunia atau konsep abstrak, jadi ini adalah suatu pendekatan yang kuat untuk pemrograman yang memungkinkan kita untuk menciptakan program yang lebih terstruktur dan mudah dipelihara. Konsep OOP

- Encapsulation, menyembunyikan detail implementasi dan hanya menampilkan interface secukupnya
- Inheritance, pewarisan atribut dan prosedur dari class ke subclass

- Polymorphism, kemampuan objek untuk memiliki banyak bentuk seperti prosedur yang memiliki nama sama namun fungsinya berbeda
- Abstraction, menyederhanakan representasi objek

OOP fokusnya ke objek dan kelas, jadi programnya dibangun dari bagian-bagian kecil yang saling terpisah dan bisa dipakai ulang. Sebaliknya, Procedural Programming lebih berpusat pada urutan langkah-langkah atau fungsi, OOP unggul soal penggunaan ulang kode. Misalnya, dengan pewarisan, kita bisa membuat kelas baru dari kelas yang sudah ada tanpa perlu penulisan ulang semuanya. Sementara itu, di Procedural Programming, biasanya cuma fungsi yang bisa dipakai ulang, dan itu pun sering kurang fleksibel. Dalam hal pemeliharaan, OOP lebih mudah, apalagi untuk proyek besar, karena modularitasnya bikin perubahan jadi lebih aman dan nggak bikin kacau bagian lain. Kalau di Procedural Programming, makin besar programnya, makin ribet juga untuk dikelola, karena fungsi-fungsinya sering saling bergantung.

Soal keamanan data, OOP juga lebih unggul. Data di dalam objek bisa dilindungi dengan enkapsulasi, jadi tidak sembarangan diakses atau diubah dari luar. Di Procedural Programming, datanya lebih terbuka, jadi rawan jika ada kesalahan. Intinya, OOP cocok untuk proyek besar yang butuh struktur rapi, fleksibel, dan bisa berkembang di masa depan. Sementara Procedural Programming lebih cocok untuk program kecil yang sederhana dan cepat selesai.

3. Dalam drone digunakan berbagai sensor untuk otomatisasi pengoperasian atau untuk meringankan beban pilot. Sebutkan minimal 3 sensor yang digunakan dalam drone dan jelaskan fungsinya serta kapan harus kita pake sensor tersebut! (7 poin)
 1. GPS, untuk menentukan posisi drone secara akurat, GPS digunakan jika sedang melakukan uji coba, ketika melakukan penerbangan ke suatu tempat tertentu.
 2. Inertial Measurement Unit, untuk mengukur akselerasi, kecepatan sudut, dan orientasi drone melalui akselerometer dan giroskop. Data ini digunakan untuk menjaga kestabilan drone, memastikan pergerakan yang halus, digunakan ketika drone harus stabil, membutuhkan manuver yang cepat.
 3. Kamera, untuk mengambil gambar atau video serta mendeteksi objek di sekitar drone, digunakan jika membutuhkan pemetaan visual, atau pencarian objek
4. Di era modern, kecepatan dan efisiensi komputasi sangatlah penting. Bayangkan Anda sedang membuat aplikasi yang harus memproses ribuan gambar untuk mendeteksi objek di dalamnya. Jika proses ini dilakukan secara linear, tentu akan memakan waktu lama. Oleh karena itu, developer sering menggunakan teknik seperti multithreading dan parallel processing untuk mempercepat pekerjaan program. Jelaskan perbedaan antara keduanya! (5 poin).

Multithreading adalah teknik di mana beberapa thread dalam satu proses berjalan secara bersamaan untuk menyelesaikan tugas yang berbeda atau bagian yang sama dari tugas tertentu. Semua thread berbagi memori yang sama, sehingga cocok untuk pekerjaan yang saling terkait, seperti mengelola antarmuka pengguna sambil memproses data di latar belakang. Sementara itu, parallel processing adalah teknik yang menjalankan beberapa tugas atau proses secara bersamaan di banyak inti prosesor atau bahkan banyak mesin. Teknik ini ideal untuk pekerjaan besar yang dapat dipecah menjadi bagian independen, seperti pemrosesan ribuan gambar sekaligus. Multithreading lebih fokus pada kerja bersama dalam satu proses, sedangkan parallel processing memungkinkan tugas-tugas berjalan secara terpisah di berbagai inti prosesor.

5. Melanjut soal sebelumnya, terapkan program dengan salah satu metode, disarankan pembuatan timer, tetapi bebas sesuai kreativitas masing masing. (bonus 10 poin)

MultiThreading :

```
#include <iostream>
#include <thread>
#include <chrono>
#include <atomic>

class Timer {
private:
    int hours, minutes, seconds;
    std::atomic<bool> running;

    void decrement() {
        while (running && (hours > 0 || minutes > 0 || seconds > 0)) {
            std::this_thread::sleep_for(std::chrono::seconds(1));
            if (seconds > 0) {
                --seconds;
            } else {
                if (minutes > 0) {
                    --minutes;
                    seconds = 59;
                } else if (hours > 0) {
                    --hours;
                    minutes = 59;
                    seconds = 59;
                }
            }
        }
        displayTime();
    }
};
```

```

        if (running) {
            std::cout << "\nTimer Finished!\n";
        }
    }

    void displayTime() {
        std::cout << "\r" << (hours < 10 ? "0" : "") << hours <<
":"
            << (minutes < 10 ? "0" : "") << minutes << ":"
            << (seconds < 10 ? "0" : "") << seconds << " "
<< std::flush;
    }

public:
    Timer(int h = 0, int m = 0, int s = 0)
        : hours(h), minutes(m), seconds(s), running(false) {}
    void start() {
        running = true;
        std::thread timerThread(&Timer::decrement, this); //
Thread
        timerThread.detach(); // Jalankan thread
    }
    void stop() {
        running = false;
    }
};

int main() {
    int hours, minutes, seconds;
    std::cout << "Enter timer duration (hours minutes seconds):
";
    std::cin >> hours >> minutes >> seconds;

    Timer myTimer(hours, minutes, seconds);
    myTimer.start();
    std::cout << "Press Enter to stop the timer.\n";
    std::cin.ignore();
    std::cin.get();

    myTimer.stop();
    std::cout << "Timer Stopped.\n";

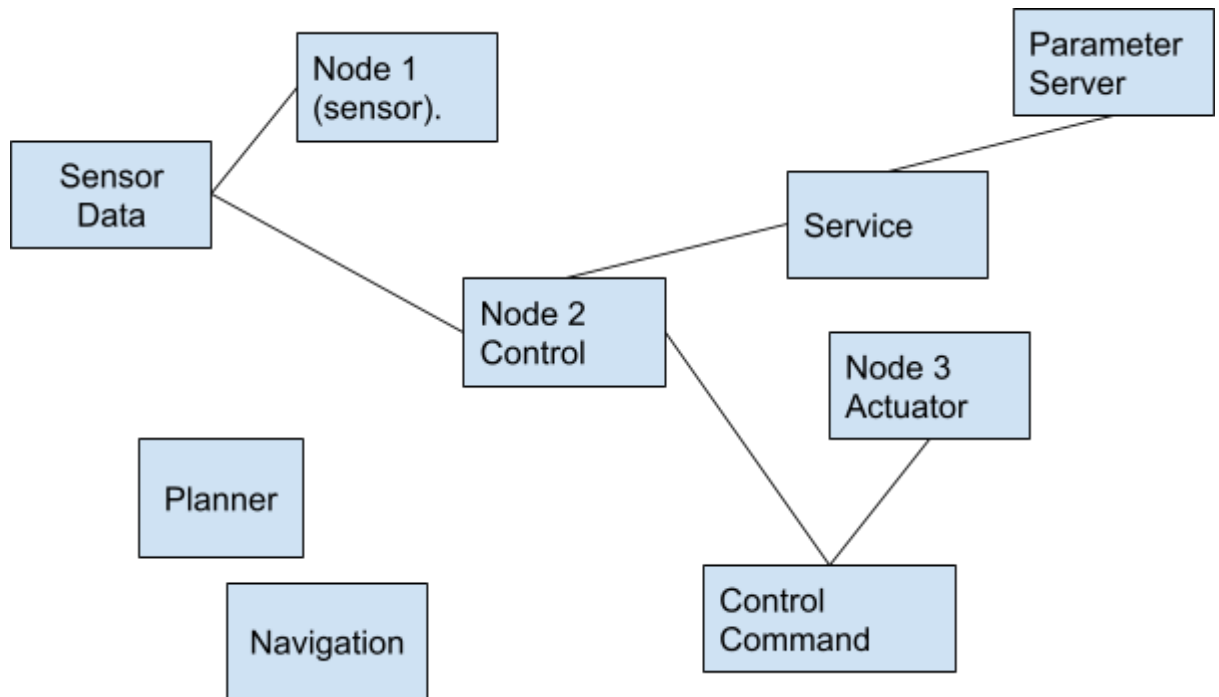
    return 0;
}

```

```
}
```

Soal ROS/OpenCV

1. Jelaskan apa itu Node, Message, Topic, Pub/Sub, Service, Parameter, dan Action dalam ROS! (5 poin)
 - a. Node : suatu file executable yang ada di dalam package ROS, dan Node bisa saling bekerja sama antara satu sama lain untuk melakukan tugas spesifik
 - b. Message : biasanya ini dalam bentuk file .msg, isinya adalah tipe tipe pesan yang akan digunakan untuk melakukan komunikasi antar Node
 - c. Topic : Topic ini sebagai perantara antar publish dengan subscribe yang akan menerima dan memberi pesan
 - d. Pub/Sub : Publish adalah untuk mengirim pesan dan Subscriber sebagai penerima pesan
 - e. Service : Suatu node yang berperan sebagai request dan respons
 - f. Parameter : Nilai yang akan disimpan dan kemudian menjadi konfigurasi suatu ROS
 - g. Action : Mekanisme yang mengatur untuk permulaan tugas, monitoring progress dan bisa membatalkan tugas.
2. Bagaimana hubungan dan mekanisme kerja mereka dalam satu sistem robot yang menggunakan ROS? Serta buatlah diagram! (7 poin)



Sistem robot ROS terdiri dari komponen yang saling berhubungan melalui mekanisme kerja yang terstruktur. Node sensor (Node 1) membaca data lingkungan dan mengirimkannya melalui Topic 1 (Data Sensor). Node kontrol (Node 2) menerima data dari Topic 1, memprosesnya, dan menghasilkan perintah kontrol (Control Command), yang kemudian diterima oleh Node aktuator (Node 3) untuk menjalankan. Topic juga menyediakan mekanisme komunikasi antar node dengan model publish/subscribe, yang memungkinkan komunikasi bersifat asinkron. Dalam sistem ini, Action Client, mirip dengan planner, mengirimkan tujuan navigasi ke Action Server untuk menjalankan tugas navigasi, memberikan umpan balik berkala, dan mengirimkan hasil akhir setelah tugas selesai. Dengan kombinasi komponen ini, sistem robot dapat bekerja secara dinamis dan efisien

3. Buat contoh implementasi topic (lengkap dengan pub/sub node), service(sertakan node client dan node server), dan action. Gunakan custom interface. Implementasi dibebaskan untuk apa saja, akan tetapi penilaian didasarkan pada:
 - ROS best practice (40%)
 - b. Keunikan (10%)
 - c. Kedekatan implementasi untuk memecahkan masalah praktikal (50%).
 Sertakan juga screenshot RQT graph Anda. (15 poin)

<https://github.com/Abrar-Abhirama/TugasCaAksan>

4. Jelaskan apa itu ORB Feature Extraction dan bagaimana cara bekerja dari ORB Feature Extraction! (5 poin)

ORB (Oriented FAST and Rotated BRIEF) adalah algoritma yang dikembangkan sebagai alternatif yang lebih cepat dan bebas royalti dari SIFT dan SURF untuk mendeteksi dan mendeskripsikan fitur dalam gambar. ORB menggabungkan algoritma BRIEF (Binary Robust Independent Elementary Features) untuk mendeteksi titik kunci dan membuat deskriptor fitur

dalam bentuk vektor biner. ORB menggunakan metode Intensity Centroid untuk menentukan orientasi tombol untuk memastikan rotasi-invariansi. Metode ini melihat distribusi intensitas pixel di sekitar tombol. Metode ini memungkinkan ORB untuk tetap konsisten meskipun gambar diputar.

Selain itu, piramida multi-skala ORB bersifat skala-invarian, yang memungkinkan deteksi fitur pada berbagai ukuran gambar. Setelah deskriptor dihitung, metrik jarak Hamming digunakan untuk mencocokkan dua gambar. Metode ini sangat cepat dan efektif untuk deskriptor biner. ORB sering digunakan dalam bidang seperti robotika, pengenalan objek, dan AR karena ringan, cepat, dan cocok untuk aplikasi real-time.

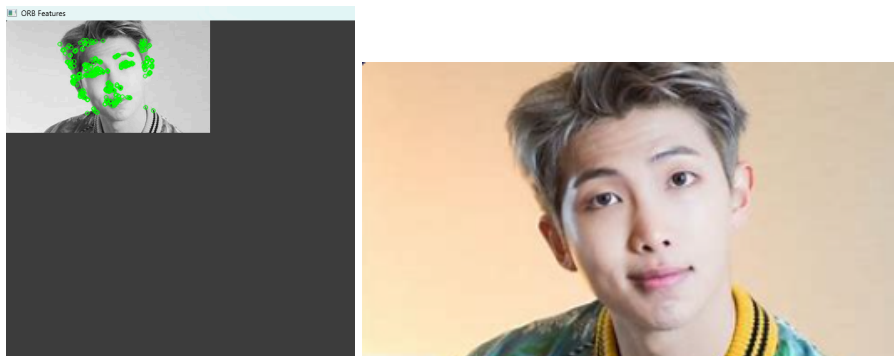
5. Implementasikan ORB Feature Extraction menggunakan OpenCV! (10 poin)

```
import cv2

img = cv2.imread('image.png', 0)
orb = cv2.ORB_create()
keypoints, descriptors = orb.detectAndCompute(img, None)

# Draw keypoints on the image
output_img = cv2.drawKeypoints(img, keypoints, None, color=(0,
255, 0))

cv2.imshow('ORB Features', output_img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



6. Implementasi dari scratch tanpa menggunakan ORB dari library OpenCV! Contoh Bonus 2 (10 poin) Sertakan juga screenshot RQT graph Anda. (15 poin)

```
import cv2
import numpy as np
```

```

import matplotlib.pyplot as plt

def harris_corner_detection(image):
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    gray = np.float32(gray)

    dst = cv2.cornerHarris(gray, 2, 3, 0.04)

    dst = cv2.dilate(dst, None)
    image[dst > 0.01 * dst.max()] = [0, 0, 255]

    return image, dst

def feature_matching(image1, image2):
    img1, corners1 = harris_corner_detection(image1.copy())
    img2, corners2 = harris_corner_detection(image2.copy())

    plt.subplot(1, 2, 1), plt.imshow(img1)
    plt.title('Pic 1'), plt.xticks([]), plt.yticks([])
    plt.subplot(1, 2, 2), plt.imshow(img2)
    plt.title('Pic 2'), plt.xticks([]), plt.yticks([])
    plt.show()

    points1 = np.argwhere(corners1 > 0.01 * corners1.max())
    points2 = np.argwhere(corners2 > 0.01 * corners2.max())

    matches = []
    for p1 in points1:
        min_distance = float('inf')
        best_match = None
        for p2 in points2:
            dist = np.linalg.norm(p1 - p2)
            if dist < min_distance:
                min_distance = dist
                best_match = p2
        if best_match is not None:
            matches.append((tuple(p1), tuple(best_match)))

    return matches

image1 = cv2.imread('image.png')
image2 = cv2.imread('image2.png')

```



```

matches = feature_matching(image1, image2)

print(f'Jumlah pasangan fitur yang ditemukan: {len(matches)}')

for match in matches:
    pt1 = match[0]
    pt2 = match[1]
    cv2.line(image1, tuple(pt1), tuple(pt2), (255, 0, 0), 2)

cv2.imshow('Pencocokan Fitur', image1)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

Pic 1



Pic 2



SOAL FIRMWARE

1. Jelaskan pengertian dari firmware dan berikan contoh penerapan firmware di dunia nyata! (3 poin)

Perangkat lunak yang tertanam langsung ke dalam hardware disebut firmware. Dia bertanggung jawab untuk mengawasi dan mengatur cara perangkat beroperasi, mulai dari memulai sistem hingga melakukan fungsi dasar. Biasanya, firmware disimpan dalam memori non-volatile, seperti memori flash atau ROM. Ini memastikan bahwa firmware tetap ada meskipun perangkat dimatikan. Contohnya adalah smartphone, di smartphone kita bisa mengatur sendiri cara smartphone itu bekerja sesuai kemauan kita.

2. Jelaskan masing-masing protokol komunikasi (UART, I2C, SPI) yang dapat digunakan oleh Arduino. Jelaskan bagaimana cara mereka bekerja! (5 poin)

Protokol komunikasi serial UART mengirim data satu bit per waktu melalui dua pin: TX (Transmit) dan RX (Receive). Ini bekerja secara asynchronous, jadi tidak perlu clock eksternal untuk sinkronisasi. Digunakan untuk berkomunikasi dengan Arduino atau dengan modul GPS dan Bluetooth.

Sementara SPI menggunakan empat pin: MOSI, MISO, SCK, dan SS untuk komunikasi yang lebih cepat dan stabil, dengan satu master mengontrol beberapa slave, protokol I2C menggunakan dua jalur: SDA (Data) dan SCL (Jam). Protokol ini memungkinkan banyak perangkat berbagi jalur yang sama, cocok untuk menghubungkan berbagai sensor dengan hanya dua pin.

3. Jelaskan jenis komunikasi apa yang cocok untuk digunakan pada aplikasi tertentu! (Misalnya, jelaskan kapan sebaiknya menggunakan I2C dibandingkan dengan SPI atau sebaliknya) (5 poin)

I2C atau SPI adalah dua jenis komunikasi. Yang mana yang digunakan bergantung pada kebutuhan aplikasi. I2C lebih baik untuk sistem dengan banyak perangkat karena hanya membutuhkan dua jalur (SDA dan SCL) untuk berkomunikasi. Ini menghemat pin dan membuat koneksi lebih mudah. Namun, I2C memiliki kecepatan transfer yang lebih lambat, yang membuatnya baik untuk penggunaan yang tidak membutuhkan banyak bandwidth, seperti monitor yang mengirim data sesekali. SPI, di sisi lain, memiliki kecepatan transfer data yang lebih cepat, yang membuatnya sempurna untuk situasi di mana Anda perlu berbicara dengan cepat, seperti saat Anda perlu menghubungkan mikrokontroler ke perangkat yang perlu mengirim banyak data. Meskipun SPI membutuhkan lebih banyak pin, ia dapat menangani gangguan dengan lebih baik dan berkomunikasi lebih baik dalam jarak yang sedikit lebih jauh. Hasilnya, SPI bekerja lebih baik untuk situasi di mana kecepatan dan kontak antara beberapa perangkat penting, sementara I2C bekerja lebih baik untuk sistem yang membutuhkan banyak perangkat yang menggunakan sedikit daya dan memiliki koneksi yang lebih kecil.

4. Buatlah kode singkat untuk menghubungkan Arduino dengan dua perangkat menggunakan I2C dan SPI dan jelaskan juga bagaimana Anda akan mengkonfigurasi perangkat tersebut agar dapat berkomunikasi satu sama lain! Anda dapat menggunakan referensi berikut (10 poin)

<https://github.com/Abrar-Abhirama/TugasCaAksan>

5. Jelaskan perbedaan utama antara PWM (Pulse Width Modulation) dan PPM (Pulse Position Modulation). Lalu tuliskan program sederhana pada Arduino untuk menghasilkan sinyal PWM dengan duty cycle 50%! (10 poin) Note: tidak boleh menggunakan fungsi analogwrite()!

<https://github.com/Abrar-Abhirama/TugasCaAksan>

6. Jelaskan apa itu PID Controller ! Jelaskan juga mengapa harus menggunakan PID Controller pada drone! (5 poin)

PID Controller adalah sistem kontrol umpan balik yang terdiri dari tiga komponen utama, yaitu Proportional, Integral, dan Derivative yang membantu memodifikasi kesalahan antara nilai yang diinginkan dan nilai aktual suatu sistem, membentuk pengontrol PID, suatu sistem kontrol umpan balik. Derivative memperkirakan laju perubahan kesalahan untuk meminimalkan overshoot dan meningkatkan stabilitas;; komponen Proportional mengelola kesalahan secara proporsional, Integral menggabungkan kesalahan selama periode panjang untuk menghilangkan offset; pengontrol PID diperlukan dalam drone untuk memodifikasi posisi, orientasi, dan kecepatan secara tepat, sehingga menjaga stabilitas penerbangan. Agar tetap pada jalur yang dituju, ia juga memungkinkan drone untuk bereaksi terhadap gangguan luar seperti angin, memodifikasi keseimbangan dalam situasi terbang yang berbeda, dan menghentikan overshoot. Pengontrol PID menjamin fleksibilitas dan efisiensi yang hebat dengan memungkinkan penyesuaian parameter kontrol dinamis, oleh karena itu memastikan drone stabil, responsif, dan akurat dalam menghadapi kondisi atau beban yang berubah selama penerbangan.

7. Secara garis besar, jelaskan proses penggunaan firmware pada drone mulai dari persiapan awal kode firmware hingga drone bisa diterbangkan! Gunakan bahasamu sendiri jika kamu mengambil informasi dari banyak sumber! (5 poin)

Jadi pertama tama, saat set up drone, yaitu saat persiapan hardware, pastikan bahwa drone sudah dilengkapi dengan hardware yang memadai, lalu kita pilih platform firmware yang sesuai dengan drone, setelah sudah dipilih maka kita tulis dan unggah kode firmware ke flight yang pastinya berhubungan dengan IDE. Setelah di unggah, harus ada kalibrasi dan pengujian awal setelah firmware diunggah, dengan memasukkan komponen sensor dll bekerja dengan baik dan akurat. Setelah kalibrasi, parameter penerbangan seperti kecepatan maksimum, ketinggian, batasan GPS, dan mode penerbangan, setelah semua selesai di kalibrasi maka akan langsung ada pengujian di lapangan, saat di uji coba ini, akan ada penyesuaian dan jika sudah berhasil semua maka drone siap untuk dijalankan dari jarak jauh.

8. Jelaskan konsep Hardware-in-the-Loop (HITL) secara umum! Kemudian sebutkan juga keuntungan menggunakan metode HITL dalam pengembangan sistem UAV! (5 poin)

Untuk menguji sistem secara lebih realistis tanpa harus mengujinya di lapangan, hardware-in loop (HITL) merupakan teknik pengujian yang menggabungkan perangkat keras fisik dengan model simulasi perangkat lunak dalam satu loop kontrol. Karena perangkat keras dan perangkat lunak HITL dievaluasi dalam simulasi yang mendekati kenyataan, memanfaatkannya dalam pengembangan sistem UAV memiliki keuntungan termasuk pengujian yang lebih realistis. Lebih jauh, HITL mempercepat siklus pengembangan sehingga meningkatkan efisiensi pengujian dan menurunkan kemungkinan kegagalan perangkat keras karena pengujian pertama kali dilakukan dalam lingkungan simulasi. Pendekatan ini juga memungkinkan validasi algoritma kontrol waktu nyata dan validasi perangkat keras aktual, sehingga menurunkan biaya pengujian lapangan.

Bonus Meme

Me Looking for ROS 2 Documentations

