

Technical Journal

Aurelia Jennifer Gunawan
13524089

Last Updated: 2026-01-30

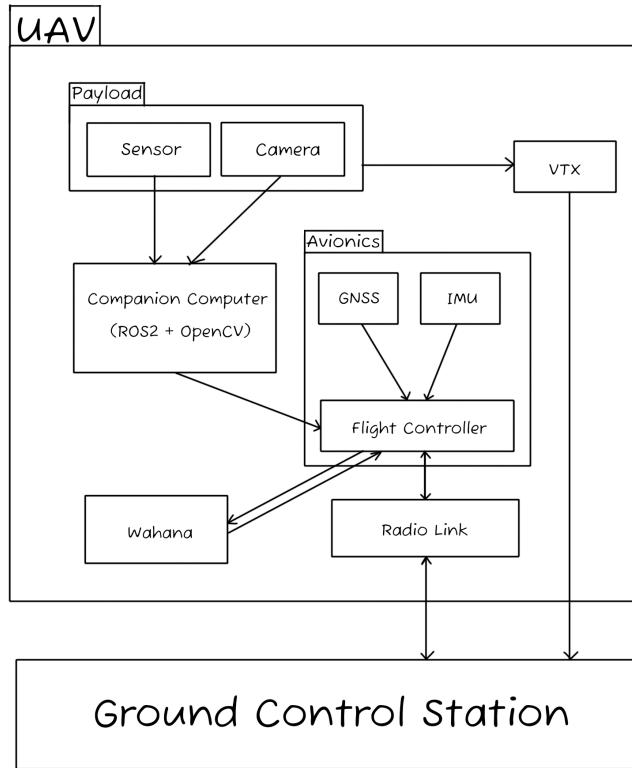
Contents

1	Catatan Pendidikan	2
1.1	Arsitektur Sistem UAV	2
1.1.1	Komponen” UAV	2
1.1.2	Komunikasi	4
1.1.3	Logika Operasi	6
1.2	Dasar Pemrograman	7
1.2.1	Konsep Dasar OOP	7
1.2.2	Design Pattern	7
1.2.3	Error Handling	8
1.2.4	Virtual Function	8
1.2.5	Pengenalan Pemrograman Konkuren	8
2	Hands-on/THT	8
2.1	Kendala	8
2.2	Upaya Penyelesaian	8
3	Referensi	9

1 Catatan Pendidikan

1.1 Arsitektur Sistem UAV

1.1.1 Komponen” UAV



Wahana terdiri atas airframe dan sistem propulsi. Airframe pada pesawat terdiri atas fuselage, sayap, dan ekor. Airframe pada drone terdiri atas frame atau center plate, dan arm. Sistem propulsi terdiri atas tiga komponen:

- Propeller
Menghasilkan gaya angkat/dorong melalui rotasi.
- Motor
Mengubah energi listrik menjadi energi kinetik rotasi untuk memutar *propeller*
- Electronic Speed Controller (ESC)
Mengatur kecepatan dan arah putaran motor sesuai perintah dari FC
- Baterai Lithium Polymer (LiPo)
Sumber energi utama

Flight controller berperan sebagai sistem saraf motorik dari UAV yang memproses dan menyatukan data dari sensor serta menjalankan algoritma kontrol untuk mengendalikan penerbangan, baik berdasarkan perintah pilot (dari GCS) maupun secara otomotif (dari companion computer). FC terdiri atas beberapa *chip* atau unit:

- Microcontroller Unit (MCU) Prosesor utama yang menjalankan *firmware* dan mengolah semua data dan logika kontrol penerbangan
- Barometer
Mengukur tekanan atmosfer untuk mengestimasi ketinggian wahana
- Inertial Measurement Unit (IMU) Mengintegrasikan *accelerometer* dan *gyroscope* yang mengukur percepatan, kecepatan sudut, dan orientasi UAV untuk mengetahui pergerakannya.
- Blackbox Mencatat log keseluruhan UAV sejak arm hingga disarm, untuk analisis pasca penerbangan.
- Modul GNSS Menerima sinyal dari sistem navihasi satelit global, seperti GPS, GLONASS, Galileo, dan BeiDou untuk menentukan posisi, kecepatan, dan waktu UAV.
- Airspeed sensor Mengukur kecepatan relatif wahana terhadap udara menggunakan tabung pitot
- Magnetometer (kompasnya UAV) Mendeteksi medan magnet bumi untuk menentukan arah mata angin yang dituju wahana. Biasanya terintegrasi dalam modul GNSS sebagai *chip* kompas.

Payload dapat berupa:

- Kamera FPV Menyediakan video real time ke pilot/GCS
- Kamera Depth Menghasilkan peta kedalaman/*depth map* dan titik 3D secara real-time
- Kamera Tracking Mengestimasi posisi dan orientasi UAV secara real time tanpa GPS
- LiDAR Sensor pemetaan 3D high precision untuk misi survei topografi dan kehutanan
- Sensor Partikel Sensor untuk memantau kualitas udara
- and more...

Ground control station (GCS) adalah sistem di darat untuk memantau, mengendalikan, dan mengelola misi uAV lewat gelombang radio. GCS terdiri atas remote controller, laptop dan aplikasi GCS, modul radio telemetry, dan antena yagi. UAV dan GCS melakukan komunikasi dengan **gelombang radio** pada spektrum frekuensi tertentu. Jalur komunikasi fungsional yang terbentuk dari pertukaran data lewat frekuensi ini disebut dengan **link**. Terdapat dua jenis link utama dalam sistem UAV:

1. C2/Data/Radio/Non-payload Link Saluran utama yang bertanggung jawab atas pengendalian UAV. Secara fisik hubungan ini dijalin antara *radio transceiver* di GCS dan di UAV. Terdiri atas:
 - Uplink (*telecommand*) mengirimkan perintah kendali dari GCS/pilot ke UAV

- Downlink (*telemetry*) mengirimkan data status UAV (attitude, baterai, GPS, ketinggian, dsb.) ke GCS
2. Payload Link Saluarn khusus untuk mengirimkan data dari payload ke GCS secara real-time. Untuk kamera, link ini dijalin secara fisik antara VTX (video transmitter) dan VRX (video receiver)

Companion computer adalah "otak" yang melakukan komputasi tingkat tinggi, yang melengkapi FC. CC disambungkan ke FC secara serial dan berkomunikasi dengan protokol MAVLINK. Payload dan sensor memberikan data ke CC untuk mengambil keputusan, lalu CC mengirimkan perintah ke FC untuk dieksekusi.

1.1.2 Komunikasi

Terdapat dua cara mengirim data:

1. Wired
 - Digunakan untuk komunikasi jarak pendek antarkomponen
 - Media fisik berupa kabel tembaga
 - Laju transfer sangat tinggi, latensi rendah, dan lebih baik dari segi keamanan, tapi jangkauannya terbatas
2. Wireless
 - Digunakan untuk komunikasi jarak jauh, misalnya antara GCS dan UAV
 - Media fisik berupa gelombang radio dari transceiver
 - Mobilitas lebih tinggi, tapi rentan interferensi dan bandwidth, serta latensi terbatas

Proses pengiriman data diatur oleh **protokol**. Protokol mengatur:

1. sintaks: cara menyusun struktur data
2. semantik: makna dari setiap bit data
3. sinkronisasi: keselarasan kecepatan pengiriman data

Wired:

1. UART (Universal Asynchronous Receiver-Transmitter)
 - Bersifat **asinkron**. Pengirim dan penerima menyepakati laju pengiriman data (baud rate, bit/s), panjang data, dan bit (0 atau 1) untuk menandakan awal dan akhir satu pesan.
 - Mempunyai dua pin, yaitu **RX** dan **TX**. RX untuk menerima data dari perangkat lain. TX untuk mengirim data ke perangkat lain.
 - Digunakan untuk menghubungkan FC ke modul telemtri, GPS, dan CC.
2. SPI (Serial Peripheral Interface)
 - Bersifat **sinkron**.

- Mempunyai empat pin, yaitu **SCLK**, **MOSI**, **MISO**, dan **SS**. SCLK mirip SCL di I2C. MOSI adalah jalur data dari master ke slave. MISO jalur data dari slave ke master. SS adalah pin untuk memilih slave mana yang diajak bicara.
- Digunakan untuk menghubungkan FC (master) dengan komponen kritis (slave) dengan data padat seperti IMU dan blackbox

3. I2C (Inter-Integrated Circuit)

- Bersifat **sinkron**.
- Mempunyai dua pin, yaitu SDA dan SCL. SDA adalah jalur dua arah untuk mengirim dan menerima data. SCL membawa sinyal detak dari master untuk sinkronisasi.
- Digunakan untuk menghubungkan FC ke sensor berkecepatan rendah seperti airspeed sensor dan barometer

Wireless:

1. TCP (Transmission Control Protocol)

- Sebelum mengirim data, pengirim dan penerima melakukan **handshake** untuk memastikan kesiapan pengiriman data.
- Kalau ada data yang hilang/rusak di tengah jalan, pengirim berhenti dan mengirim ulang data tsb sampai diterima secara sempurna.
- Digunakan untuk **upload waypoints** atau **mengubah parameter**, karena merupakan informasi krusial yang tidak boleh salah satu bit pun (dapat berakibat fatal), dan reliability diprioritaskan.

2. UDP (User Datagram Protocol)

- Data dikirim terus menerus tanpa memedulikan apakah penerima sudah siap atau data sampai secara sempurna.
- Pengiriman data sangat cepat dengan latensi rendah, tapi tidak reliable
- Digunakan untuk **streaming video dan data telemtry secara real time**

Setiap data yang dikirim umumnya dikemas berdasarkan protokol perangkat lunak untuk UAV, seperti MAVLink (Micro Air Vehicle Link). MAVLink adalah protokol komunikasi pesan *open source* yang sangat ringan dan penting karena jaringan nirkabel memiliki *bandwidth* yang terbatas.

- Setiap wahana UAV memiliki system ID (1-255) dan setiap komponen dalam wahana memiliki component ID.
- UAV mengirimkan pesan HEARTBEAT secara rutin untuk memberi tahu GCS bahwa koneksi masih aktif dan sistem dalam keadaan normal.
- MAVLink lebih sering menggunakan UDP (kalau pake TCP akan ada delay yang berpotensi membuat wahana *crash*. MAVLink memiliki *checksum*, kode verifikasi yang dihasilkan secara matematis dari isi pesan itu sendiri sehingga ketidakcocokan *checksum* akan mengakibatkan paket data terbuang.)

1.1.3 Logika Operasi

System state dan flight mode menentukan seberapa besar kontrol pilot dibandingkan dengan kontrol otomatis oleh FC dan apa yang dilakukan wahana dalam suatu kondisi/mode.

- **System State** Sebelum terbang, wahana harus melewati tahapan status berikut:
 1. **Initialization:** FC melakukan kalibrasi sensor internal tepat setelah baterai dihubungkan.
 2. **Pre-arm Check:** Diagnosa otomatis yang dilakukan wahana untuk memastikan kondisinya layak terbang.
 3. **Disarm:** Kondisi standar saat di darat. Motor terkunci total dan tidak akan berputar.
 4. **Arm:** Kondisi aktif, motor mulai berputar pada kecepatan rendah dan wahana siap menerima input *throttle* untuk lepas landas.
- **Flight Mode** Berikut adalah beberapa contoh mode terbang untuk ArduPilot:
 1. **Acro:** Kendali penuh tanpa stabilisasi otomatis.
 2. **Stabilize:** Wahana otomatis kembali datar jika stick dilepas.
 3. **Alt-Hold:** Menjaga ketinggian otomatis, posisi horizontal tetap diatur manual.
 4. **Loiter:** Menjaga koordinat dan ketinggian secara statis di udara.
 5. **RTL/RTH (Return To Land/Home):** Kembali ke titik landas/home.
 6. **Auto:** Menjalankan misi berdasarkan waypoints yang telah ditentukan.
 7. **Guided:** Bergerak secara interaktif berdasarkan input pada peta di GCS.

Mekanisme Failsafe adalah prosedur otomatis yang dilakukan UAV ketika terjadi kegagalan sistem untuk mencegah *crash* atau kehilangan wahana. Beberapa pemicu failsafe meliputi:

- **RC Link Lost:** sinyal antara *remote control* dan *receiver* di UAV terputus
- **GCS Link Lost:** koneksi telemetri antara GCS dan UAV terputus
- **Low Battery**
- **EKF Check/GPS Glitch**
- **Fence Breach:** wahana keluar dari batas wilayah geografis yang sudah ditentukan (geofence).

Beberapa tindakan yang diambil FC dalam kondisi failsafe (di-configure sebelumnya) adalah:

- RTL
- Land
- Brake
- and more...

1.2 Dasar Pemrograman

1.2.1 Konsep Dasar OOP

Why OOP?

- Fokus pada objek yang memodelkan benda nyata (data + perilaku)
- Modularitas tinggi, kode bisa dipecah menjadi bagian” (class)
- Rapi, terstruktur, dan mudah di-maintenance Class = skematik/datasheet komponen (rancangan) Object = Komponen fisik yang sudah dirakit Sebuah class memiliki atribut (variabel yang menyimpan data), dan metode (fungsi/prosedur yang mendefinisikan tindakan atau perilaku yang bisa dilakukan oleh class tersebut).

Encapsulation :

1. Membungkus data sensitif agar tidak bisa diakses sembarangan dari luar. Akses untuk membaca dan mengubah data dilakukan dengan *getter* dan *setter*.
2. Menentukan siapa yang bisa mengakses suatu atribut/method dari suatu kelas dengan access modifier:

Modifier	Own Class	Derived Class	Main()
Public	Yes	Yes	Yes
Protected	Yes	Yes	No
Private	Yes	No	No

Abstraction bertujuan untuk menyembunyikan proses/cara kerja di belakang layar.

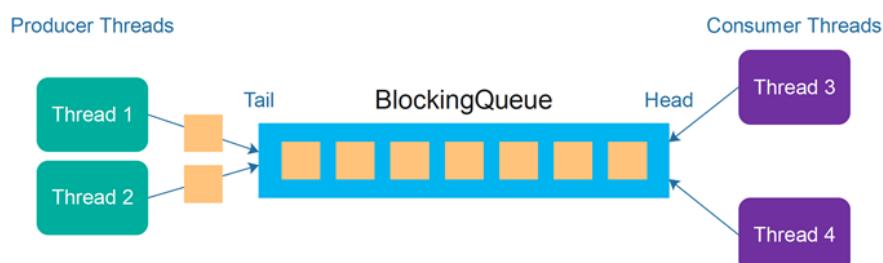
Inheritance (based on prinsip Don't Repeat Yourself/DRY). Parent class/base class adalah class yang mewariskan sifat. Child class/derived class adalah class yang mewarisi sifat dari parent.

Polymorphism adalah prinsip yang memungkinkan respon yang berbeda-beda berdasarkan perintah yang sama, tergantung object.

1.2.2 Design Pattern

Pola umum yang digunakan untuk mempermudah pengembangan aplikasi.

Producer-Consumer Pattern adalah pola desain konkuren, di mana satu atau lebih thread producer menghasilkan objek yang diantreankan, lalu dikonsumsi oleh satu atau lebih consumer. Objek yang diantreankan umumnya mewakili beberapa pekerjaan yang perlu dilakukan.



1.2.3 Error Handling

Mekanisme Try-Catch:

- Blok try berisi kode yang dicoba untuk dijalankan
- Kalau ada error pindah ke block catch dengan Exception yang sesuai
- better than if-else karena dapat lebih mengcover error tak terduga

1.2.4 Virtual Function

Virtual function adalah fungsi parent class/base class yang dideklarasikan dengan kata kunci virtual, dan di-override dalam child class/derived class. Fungsi virtual memungkinkan *runtime polymorphism*, dimana fungsi yang tepat dipanggil melalui base class pointer atau reference.

1.2.5 Pengenalan Pemrograman Konkuren

Multithreading

- Thread adalah unit terkecil dari proses
- Multithreading adalah teknik memecah proses menjadi task kecil yang dijalankan oleh thread di dalamnya sehingga bisa berjalan secara konkuren/bersama.
- Problems? Race condition, dan deadlock!

Race condition:

- thread saling mengakses resources yang sama, maka dapat terjadi rebutan resources antar thread.
- solusi: use **mutex/lock** sehingga hanya satu thread yang bisa mengakses suatu resource di suatu waktu.

Deadlock:

- dua atau lebih proses/pihak dapat saling menunggu resources yang dipegang oleh pihak lain
- solusi:
 1. deadlock prevention: menghilangkan salah satu syarat kondisi terjadinya deadlock, yaitu mutual exclusion, hold and wait, no preemption, dan circular wait.
 2. deadlock avoidance dengan banker's algorithm (memeriksa secara dinamis status alokasi resource untuk memastikan sistem tidak memasuki unsafe state)
 3. deadlock detection and recovery dengan timeout dan deteksi cycle.

2 Hands-on/THT

2.1 Kendala

2.2 Upaya Penyelesaian

3 Referensi

- Design Patterns
- The Producer Consumer Pattern
- PPT materi saat day pendidikan