

Pengendali Terbang Drone FPV Berbasis Betaflight via ExpressLRS HappyModel EP-1

1st Ridho Ramadani Saputra
Fakultas Ilmu Terapan
Telkom University
Bandung, Indonesia

ridhoxic@student.telkomuniversity.ac.id

2nd Nina Hendrarini
Fakultas Ilmu Terapan
Telkom University
Bandung, Indonesia

ninahendrarini@tass.telkomuniversity.ac.id

3rd Ema
Fakultas Ilmu Terapan
Telkom University
Bandung, Indonesia

emacdef@telkomuniversity.ac.id

Abstrak

Penelitian ini membahas perancangan dan implementasi sistem kendali terbang pada drone FPV (*First person view*) melalui integrasi modul penerima *expresslrs happymodel ep-1*. Tujuan dari penelitian ini adalah merancang sistem kendali yang secara relatif memiliki karakteristik relatif efisien, ringan, dan andal dalam performa transmisi sinyal pada frekuensi 2.4 GHz. Pengembangan sistem ini ditujukan sebagai solusi alternatif bagi drone yang tidak mendukung sistem *plug and play*, dengan konfigurasi dilakukan menggunakan *betaflight configurator*. Selain itu, komunikasi antara *flight controller* dan penerima dilakukan melalui jalur transmisi *serial TX-RX (transmit-receive)*. Hasil pengujian menunjukkan bahwa sistem mampu memberikan respons kendali yang presisi dan responsif dalam berbagai skenario penerbangan FPV. Diharapkan, penyusunan ini dapat mendukung pengembangan teknologi berbasis *open-source* yang lebih terjangkau dan mudah diakses, serta memberikan kontribusi positif terhadap kemajuan drone FPV, khususnya dalam lingkup RA ATAP.

Kata kunci: Drone FPV, Betaflight, Expresslrs

I. PENDAHULUAN

Perkembangan drone FPV di Indonesia di dominasi oleh sistem *plug and play*, yang mempunyai keterbatasan dalam aspek modifikasi[1], [2]. Kebutuhan akan sistem kendali yang efisien dan fleksibel dalam komunikasi menjadi krusial, terutama untuk kalangan riset ataupun penggemar FPV. *Expresslrs* ini sebagai alternatif dengan memiliki keandalan mengenai latensi, namun belum banyak didokumentasikan integrasinya secara teknis di lingkungan akademik dan riset.

Sistem kendali terbang (*flight controller*) merupakan komponen utama dan memiliki kapabilitas berbeda, tergantung pada arsitektur perangkat keras serta jenis *firmware* yang digunakan[3]. *IFlight Beast H7 55A aio* adalah salah satu *flight controller* yang digunakan, mempunyai prosesor STM32H743 sehingga dapat memberikan performa tinggi, terutama dalam respons kendali dan kestabilan yang dibutuhkan pada drone FPV[4]. Namun, pengaplikasiannya ditemukan permasalahan berupa keterbatasan kompatibilitas terhadap sistem komunikasi dalam pengintegrasian modul penerima sinyal *expresslrs happymodel ep-1*. Keterbatasan ini mengharuskan adanya konfigurasi manual melalui perangkat lunak serta mengharuskan penyesuaian pada jalur komunikasi *serial*, yang dimana ketika tidak dilakukan akan menghambat kinerja sistem secara keseluruhan. Oleh karena itu, penelitian ini bertujuan mengembangkan sistem kendali FPV berbasis *firmware open-source betaflight* dan modul *expresslrs happymodel ep-1*, sebagai solusi drone FPV yang lebih teknis dan dapat dikustomisasi.

Sebagai solusi atas keterbatasan *plug and play* pada beberapa unit drone, penelitian mengusulkan konfigurasi manual menggunakan *firmware open-source betaflight*, yang dipilih karena menyediakan fleksibilitas dalam pengaturan parameter penerbangan serta mendukung integrasi dengan protokol komunikasi eksternal. Dalam implementasinya, protokol *expresslrs* yang dikenal memiliki latensi rendah dan efisiensi tinggi, serta modul penerima *happymodel ep-1* yang terhubung melalui komunikasi *serial TX-RX*[5]. Modul *ep-1* memungkinkan pembaruan *firmware* melalui WiFi dan mendukung *refresh rate* hingga 500Hz, menjadikannya cocok untuk aplikasi FPV racing atau long-range[6]. Pendekatan ini memungkinkan integrasi antara *flight controller* dan sistem komunikasi secara mandiri, tanpa bergantung pada kompatibilitas otomatis. Dengan demikian, sistem kendali yang dibangun tidak hanya mampu mengatasi hambatan konektivitas, tetapi juga memberikan ruang untuk modifikasi dan pengembangan lebih lanjut sesuai kebutuhan pengguna FPV.

II. KAJIAN TEORI

2.1 Drone FPV

Drone FPV (*first person view*) adalah jenis drone yang memungkinkan pilot untuk mengendalikan pesawat secara *real-time* melalui *radio controller* dan tampilan langsung dari kamera yang dipasang pada drone. Teknologi FPV ini memberikan pengalaman visual yang seolah-olah berada pada kokpit drone, yang dimana akan sangat berguna dalam kompetisi racing, eksplorasi wilayah sulit hingga pemetaan udara presisi tinggi. Sistem FPV terdiri dari kamera, *transmitter* video, *receiver* video, dan kaca mata FPV atau monitor.

2.2 Flight Controller

Komponen inti yang bertindak sebagai otak pada setiap drone, memungkinkan beroperasi dengan stabil, aman, dan bahkan secara otonom. Dengan menerima data dari sensor (seperti *gyroscope*, *accelerometer*, dan *GPS*), kemudian memproses melalui mikrokontroler yang menjalankan algoritma kontrol, *flight controller* secara otomatis mengirimkan perintah ke motor atau aktuator lain untuk menyesuaikan posisi drone. Kemampuan ini tidak hanya memastikan drone tetap seimbang saat di udara meskipun ada gangguan eksternal, tetapi juga memungkinkan fungsi-fungsi lainnya untuk mempertahankan posisi pada saat di udara.

2.3 Betaflight Configurator

Perangkat lunak antarmuka penting *betaflight configurator* memungkinkan pengguna mengkonfigurasi dan mengatur *flight controller* yang menjalankan *firmware*

betaflight, khususnya untuk *drone FPV*. Berfungsi sebagai pusat kendali untuk mengkalibrasi sensor hingga melakukan *tuning PID* untuk dapat mengoptimalkan kinerja penerapan. Konfigurator ini juga sangat memungkinkan sebagai pengaturan mode penerbangan yang menjadikan perangkat lunak ini penting bagi pilot untuk menyesuaikan setiap aspek karakteristik *drone* diberbagai kebutuhan.

2.4 ExpressLRS

Expresslrs atau banyak menyebutnya *ELRS* adalah sistem *radio control open-source* yang telah digunakan untuk merubah komunikasi *drone FPV*. Dengan memiliki *refresh rate* yang sangat tinggi (hingga 1000Hz) dan latensi ekstrim rendah. Keunggulannya ini ditandai dengan menggunakan teknologi *RF (radio frequency)* berbasis *LoRa* dan modulasi pita sempit yang sangat efektif, menghasilkan respons kontrol yang hampir instan dan jangkauan sinyal yang jauh lebih baik bahkan dalam kondisi penuh dengan gangguan.

2.5 Plug and Play

Pada *drone* mengacu dengan pendekatan perakitan dan konfigurasi yang dirancang untuk meminimalkan kebutuhan akan penyolderan atau pemrograman yang rumit bagi pengguna. Konsep ini diwujudkan melalui berbagai komponen *drone* agar mudah dihubungkan. Serta sering kali melalui integrasi beberapa fungsi ke dalam satu papan, yang mempunyai tujuan untuk menyederhanakan proses pembangunan *drone* agar lebih mudah di akses oleh pemula sekaligus mempercepat waktu pembuatan bagi para profesional.

2.6 Modul Receiver HappyModel EP-1

Unit yang dirancang dengan implementasi spesifik dari teknologi *expresslrs* yang sangat ideal untuk *drone FPV* berukuran kecil hingga menengah. Alat ini terintegrasi dengan *flight controller* melalui koneksi *serial UART* dan bertanggung jawab menerima sinyal kontrol frekuensi radio dari pemancar *expresslrs* yang dioperasikan pilot.

III. METODE

Pendekatan sistematis merupakan bagian yang akan diuraikan dalam penelitian ini, dimulai dengan perancangan sistem hingga pengujian kinerja. Pendekatan ini dirancang agar mengatasi tantangan kompatibilitas yang sering terjadi pada integrasi sistem kendali *drone* komersial, khususnya dalam komunikasi nirkabel dan konfigurasi *firmware open source*.

3.1 Gambaran sistem saat ini



GAMBAR 1
Sistem saat ini[7]

Sebagian besar penggunaan *drone FPV* di Indonesia masih menggunakan kontrol *drone FPV* yang komersial dengan dilengkapi *firmware* bawaan pabrik saja[8]. Sistem ini biasanya menggunakan *SBUS* atau *FrSky* sebagai protokol komunikasi. Namun memiliki keterbatasan kustomisasi atau pengembangan lebih lanjut dan membuatnya tidak cocok untuk eksplorasi teknis yang lebih mendalam. Gambaran sistem saat ini ditunjukkan pada gambar 1 yang menggambarkan alur sistem *drone* komersial yang bersifat *plug and play* dengan *firmware* bawaan pabrik.

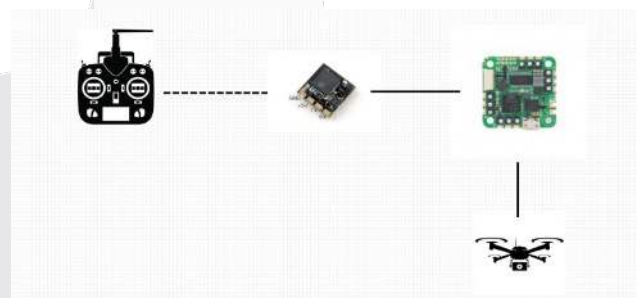
TABEL 1
Spesifikasi prosesor *flight controller*[9]

	F1	F3	F4	F7	H7
Speed	72MHz	72MHz	168MHz	216MHz	480MHz
Memory	128KB	256KB	512Kb/1MB	512Kb/1MB	1MB/2MB

Pada table 1 menunjukan kapasitas dari *drone* bawaan. Dengan memiliki keterbatasan yang berbeda beda dalam hal pemrosesan dan jumlah fitur yang didukung. Namun meskipun *flight controller* dengan *CPU F4* dan *F7* menawarkan performa lebih baik, banyak pengguna Indonesia masih menggunakan *CPU F1* atau *F3* karena faktor ketersediaan dan harga relatif terjangkau[10]. Hal ini menyebabkan keterbatasan dalam pengembangan sistem kendali *drone*.

3.2 Gambaran Pengembangan Sistem

Gambar 2 merupakan pengembangan alternatif yang akan mengatasi keterbatasan terhadap sistem *drone plug and play*. Dengan menambahkan perangkat keras *expresslrs happymodel ep-1* dan menggunakan perangkat lunak *betaflight configurator*. Maka sistem ini dibuat agar keterbatasan kustomisasi pada *drone* komersial tergantikan dengan penggunaan protokol komunikasi *expresslrs*.

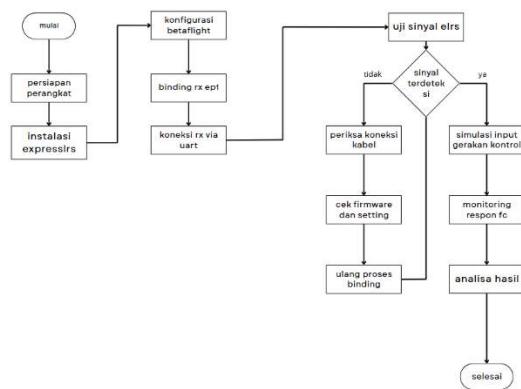


GAMBAR 1
Desain sistem

Dengan menerapkan sistem berbasis modul *expresslrs*, desain ini memungkinkan koneksi langsung antara *transmitter* dan *receiver* dengan jangkauan lebih luas. Selain itu, penggunaan *firmware* seperti *betaflight* memungkinkan konfigurasi yang lebih kompleks dan penyesuaian terhadap kebutuhan penggunaan *FPV* yang lebih berpengalaman[11], [12]. Hal ini memberikan keleluasan dalam menyesuaikan jalur komunikasi, pemetaan *channel*, hingga pengaturan *failsafe*, menjadikan sistem lebih adaptif dan mendukung pengembangan *drone* yang lebih profesional dan teknis.

3.3 Tahapan Pengembangan Sistem

1. Flowchart



GAMBAR 2
Flowchart pengembangan sistem

Proses diawali pada tahap persiapan dan instalasi, yang mencakup kegiatan awal seperti memulai sistem, menyiapkan seluruh perangkat yang diperlukan, dan menginstal *firmware expresslrs* pada perangkat *transmitter* dan *receiver*. Tahapan konfigurasi dimulai dengan penggunaan aplikasi *betafight* untuk mengatur *port UART* dan protokol *receiver* agar sesuai dengan sistem *expresslrs*. Proses *binding* dilakukan antara *receiver ep-1* dan *transmitter* melalui frekuensi 2.4GHz, kemudian *receiver* dihubungkan ke *flight controller* melalui jalur *UART* untuk mentransmisikan sinyal kendali[13]. Tahap pengujian dan *troubleshooting* dimulai dengan menguji sinyal *expresslrs* untuk memastikan koneksi berhasil. Jika sinyal terdeteksi, simulasi gerakan kontrol dilakukan untuk menguji respon *flight controller*. Jika tidak ada sinyal, sistem akan memeriksa koneksi kabel, *firmware*, dan mengulangi *binding*. Sistem ini menggantikan solusi *plug and play* dengan alternatif yang lebih fleksibel dan dapat dikembangkan. Dengan menggunakan *betafight*, sistem mengoptimalkan komunikasi antara *receiver happymodel ep-1* dan *flight controller iFlight Beast H7 55A aio*. Sinyal *transmitter expresslrs* diterima oleh *receiver*, diteruskan ke *flight controller*, dan kemudian disalurkan ke *ESC (Elektronik Speed Control)* untuk mengatur kecepatan motor *drone*.

2. Blok diagram

Sistem pengendalian *drone FPV* diawali dengan perintah yang diberikan oleh pengguna melalui *transmitter*, dimana setiap intruksi gerak seperti *throttle*, *pitch*, *roll*, dan *yaw* dikodekan menjadi sinyal digital menggunakan protokol *expresslrs* pada frekuensi 2.4GHz. Sinyal tersebut dipancarkan dan diterima oleh modul *happymodel ep-1* yang terpasang pada unit *drone*[14]. Kemudian *receiver* menerjemahkan data yang diterima dalam bentuk sinyal komunikasi *serial UART* dan meneruskannya ke *flight controller iFlight Beast H7 55A aio*, yang

telah di program menggunakan *firmware betafight* untuk mengelola interpretasi perintah kontrol yang dapat dilihat pada gambar 4.

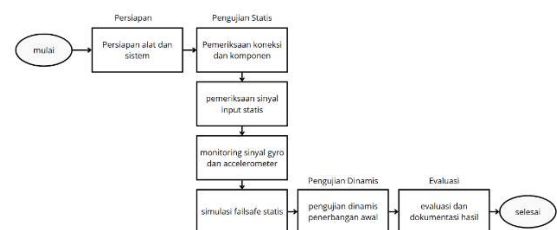


GAMBAR 3
Blok diagram

Setelah data diterima, *flight controller* melakukan proses komputasi *real-time* untuk menentukan respon motoric yang optimal berdasarkan konfigurasi yang telah dilakukan. *Output* dari *flight controller* selanjutnya dikirimkan ke *ESC* yang bertugas untuk mengatur kecepatan motor yang akhirnya menuju masing-masing motor. Dengan arsitektur ini, jalur komunikasi yang terjaga latensinya dan pemrosesan cepat oleh *flight controller* memastikan bahwa *drone* dapat merespon *input* pengguna secara akurat.

3.4 Metode Pengujian

Metode pengujian yang digunakan adalah pengujian statis dan dinamis pada sistem kendali *drone FPV* yang telah dikembangkan. Fokus pengujian adalah memastikan bahwa koneksi antara komponen (*transmitter*, *receiver*, *flight controller*) berfungsi dengan baik, *gyro* dan *accelerometer* beroperasi stabil, serta fitur *failsafe* berjalan sesuai konfigurasi. Gambar 5 merupakan flowchart metode pengujian yang dilakukan pada penelitian ini.



GAMBAR 4
Flowchart metode pengujian

Pengujian dilakukan secara bertahap melalui penerbangan awal untuk memsadikan performa sistem dalam berbagai kondisi operasional.

1. Pengujian statis

Pemeriksaan koneksi antara komponen melibatkan hubungan *transmitter* dan *receiver* untuk memastikan sinnyal dapat dikirim dan diterima dengan baik. Selanjutnya, peeriksaan konektivitas penerima ke kontrol *drone* melalui *port UART*, dilakukan bersamaan dengan penyesuaian konfigurasi pada *betafight* untuk memastikan komunikasi antar perangkat berjalan lancar. Selain itu, koneksi antara sensor (*gyro* dan *accelerometer*) dengan *flight controller* juga diperiksa. *Betafight* bertugas untuk mengawasi *channel input*, untuk memastikan bahwa sistem dapat membaca dan menanggapi setiap perintah dengan responsif[15].

Dan yang terakhir monitoring stabilitas sensor dilakukan dengan memeriksa sensor *gyeo* dan *accelerometer* secara *real-time* saat *drone* dalam kondisi tidak melakukan penerbangan maupun saat digerakan secara perlahan.

2. Pengujian dinamis

Selama penerbangan, *drone* diterbangkan secara manual untuk mengamati respons kendali dan kestabilan sistem terhadap perintah *transmitter* serta pengaruh kondisi lingkungan sekitar. Dalam skenario *failsafe*, *transmitter* sengaja dimatikan saat *drone* diterbangkan di ketinggian yang aman untuk mengamati sistem merespons dengan memutus motor atau menjalankan protokol *failsafe* yang telah dikonfigurasi. Sebagai dasar untuk menilai dan memperbaiki konfigurasi sistem, semua hasil pengujian dicatat, termasuk informasi visual dan data dari *betafight configurator*.

IV. HASIL DAN PEMBAHASAN

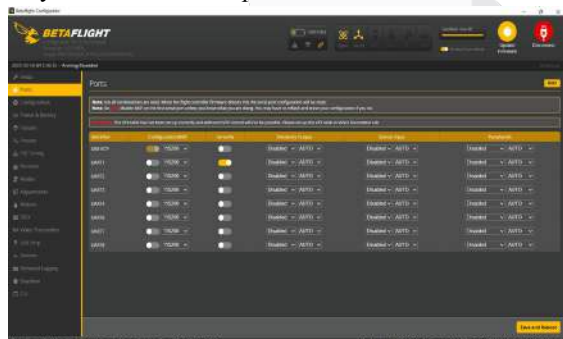
Bab ini berisi penjelasan mengenai hasil pengembangan sistem. Yang bertujuan untuk menjawab permasalahan yang dihadapi, khususnya dalam proses pengembangan *drone FPV*.

4.1 Implementasi Sistem

Sistem berhasil diimplemetasikan dan semua konfigurasi pada *betafight* dapat dilakukan dengan baik. Modul *happymodel ep-1* dapat menerima sinyal dan mengirimkannya ke *flight controller* melalui *port UART*.

1. Tampilan konfigurasi UART

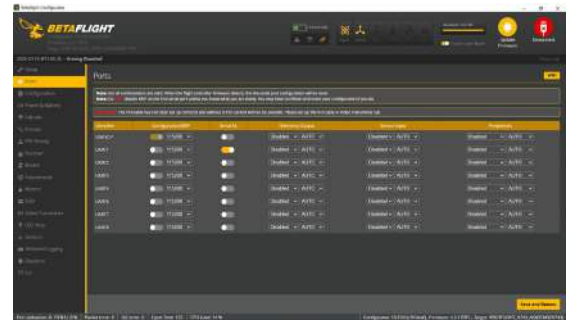
Gambar 6 memperlihatkan antarmuka *betafight configurator* yang menunjukkan bahwa *port UART* yang digunakan telah diaktifkan untuk *serial RX*, memastikan bahwa *flight controller* dapat menerima sinyal dari *receiver expresslrs*.



GAMBAR 6
Tampilan konfigurasi UART

2. Konfigurasi gyro dan accelerometer

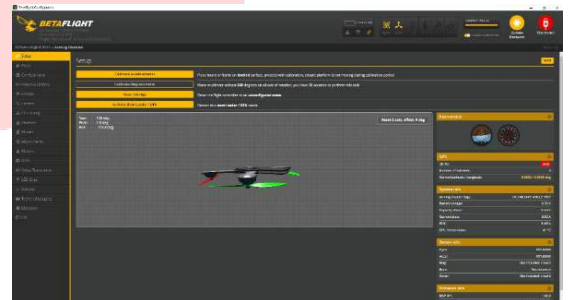
Aktifasi *gyro* dan *accelerometer* dilakukan secara manual karena *InvenSense MPU6000 IMU* sangat penting untuk menjaga stabilitas dan presisi pada saat penerbangan, serta kalibrasi yang baik memastikan bahwa pembacaan akselerasi dan rotasi dapat digunakan.



GAMBAR 5
Konfigurasi port gyro dan accelerometer

3. Kalibrasi accelerometer

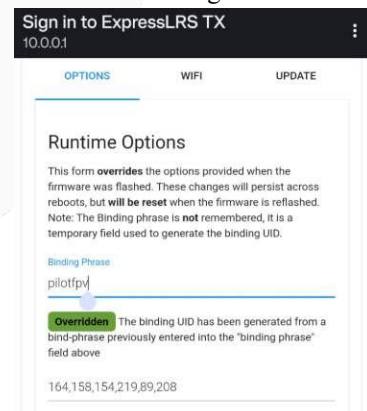
Langkah ini dilakukan dengan cara meletakkan *drone* dipermukaan yang rata untuk menunjang arah ketika melakukan penerbangan.



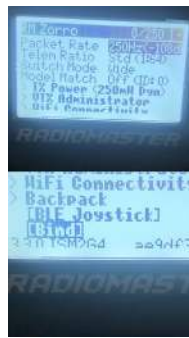
GAMBAR 6
Kalibrasi accelerometer

4. Binding phrase dan Binding RX-TX

Pada tahap ini, dilakukan konfigurasi *binding phrase* melalui web dengan ip 10.0.0.1 pada bagian *runtime options*. Ini dilakukan berfungsi sebagai kunci penghubung antara modul *TX* dan *RX* dapat terhubung secara otomatis tanpa melakukan proses *binding* manual setiap kali dinyalakan. Bisa dilihat pada gambar 9 dan 10 sebagai berikut.



GAMBAR 7
Binding phrase



GAMBAR 8

Binding via RC (radio controller)

4.2 Konfigurasi *acro*, *arming*, dan *GPS Rescue*

Konfigurasi dilakukan agar *drone* dapat dikendalikan dengan aman sesuai kebutuhan pilot. Ketiga bagian penting ini merupakan elemen dasar penerbangan, peranan penting tiga bagian ini untuk bagaimana *drone* merespons perintah *transmitter*. Selain itu konfigurasi ini menjadi bagian dari standar prosedur keselamatan dan efisiensi operasional dalam penggunaan *drone FPV*.



GAMBAR 9

Konfigurasi *aux*

Pengaturan utama dengan menghubungkan fungsi-fungsi vital ke *switch* pada *radio controller*. *Arming drone* dikendalikan oleh *switch E (aux 1)*, yang mengaktifkan motor ketika rentang nilai 1800-2000, memastikan keamanan saat tidak diterbangkan. Mode penerbangan diatur pada *switch F (aux 2)* dengan tiga pilihan yaitu, *horizon* pada rentang nilai 900-1200, *angle* 1300-1700, dan *acro* pada rentang nilai 1800-2000 untuk kendali penuh bagi pilot berpengalaman. Terakhir, fitur *GPS Rescue* diaktifkan melalui *switch B (aux 3)* pada rentang nilai 1800-2000, fitur ini memungkinkan *drone* secara otomatis Kembali ke titik lepas landas GPS jika terjadi kehilangan sinyal atau situasi darurat (*red zone*), asalkan GPS berhasil mengunci titik awal.

TABEL 1
Rentang *channel aux*

Mode	Channel	Rentang Aktif
ARM	AUX 1	1800 – 2100
ACRO	AUX 2	1800 – 2100
ANGLE	AUX 2	1300 – 1700
HORIZON	AUX 2	900 – 1200
GPS RESCUE	AUX 3	1800 – 2100

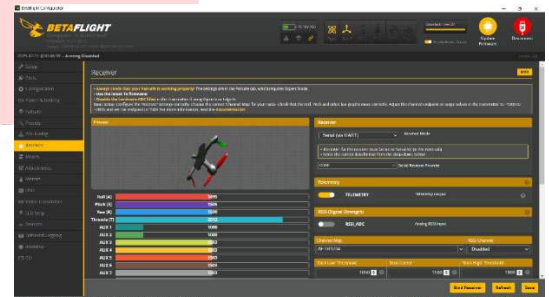
4.3 Pengujian Luaran

Pengujian luaran dilakukan untuk memverifikasi bahwa sistem kendali *drone FPV* yang telah dikembangkan mampu bekerja sesuai dengan konfigurasi yang telah dilakukan sebelumnya. Dibagi menjadi dua yaitu verifikasi respons dan verifikasi performa.

1. Verifikasi respons

- Sinyal transmitter diterima dengan baik oleh *flight controller* dan diterjemahkan dengan 4 gerakan yaitu *throttle*, *pitch*, *roll*, dan *yaw*.
- Respon *throttle*

Memperlihatkan perubahan nilai saat *throttle* pada *transmitter* digerakkan. Ini membuktikan bahwa sinyal daya dorong dari *transmitter* diterima oleh *flight controller* untuk mengatur ketinggian *drone*.

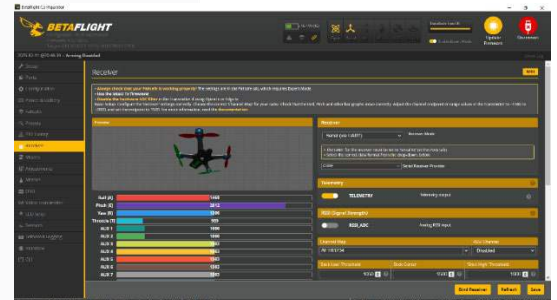


GAMBAR 10

Respons *throttle*

Respon *pitch*

Respon *pitch* bergerak maju dan mundur mengikuti *input* dari *stick pitch* pada *transmitter*. ini menandakan bahwa kontrol gerakan *vertical* ke depan dan belakang.



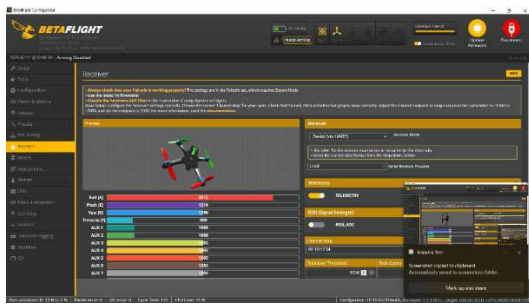
GAMBAR 11

Respons *pitch*

diterima dengan benar oleh *flight controller*, yang akan mengatur kemiringan *drone*.

- Respon *roll*

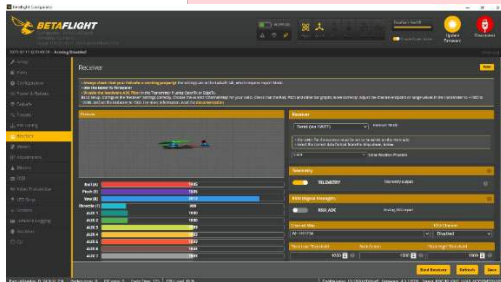
Menunjukkan pergerakan ke kiri dan kanan saat *stick roll* pada *transmitter* digerakkan. Perubahan nilai ini mengidentifikasi bahwa *input* kontrol *lateral* telah diterima oleh *flight controller* dan dapat menggerakkan *drone* untuk kemiringan ke kiri atau kanan.



GAMBAR 12
Repos roll

- Respon yaw

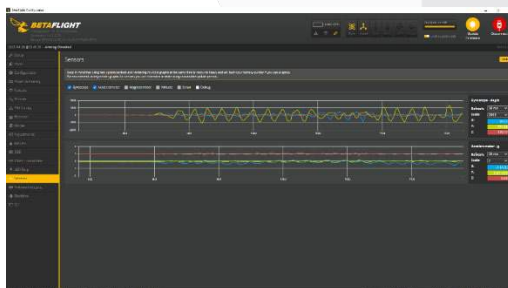
Gerakan *yaw* merespons putaran *stick* ke kiri dan kekanan, menunjukan bahwa perintah rotasi *horizontal* (memutar arah hadap *drone*).



GAMBAR 13
Repos yaw

Hasil bahwa semua respon *channel* berada pada rentang normal (1000-2000) dan bekerja sesuai perintah tanpa adanya gangguan sinyal. Hal ini menunjukkan *transmitter*, *receiver* dan *flight controller* berjalan optimal.

- b) Stabilitas *gyro* dan *accelerometer* yang ditunjukkan gambar 16 dengan cara menggerakkan dan menggoyangkan *drone* secara manual, guna memverifikasi apakah sensor dapat membaca perubahan secara akurat dan *real-time*.



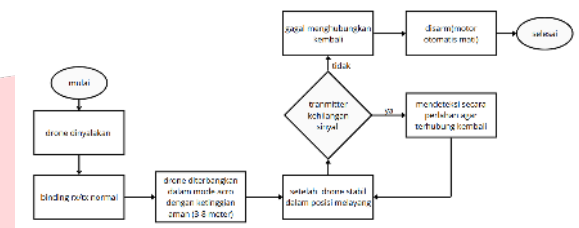
GAMBAR 14
Grafik *gyro* dan *accelerometer*

Setelah pengujian statis menunjukkan bahwa sistem beroperasi secara normal dan stabil dalam kondisi diam, pengujian dinamis menguji sistem kendali *drone* kondisi penerbangan langsung. Tujuan dari pengujian dinamis adalah untuk mengevaluasi kinerja sistem secara keseluruhan saat *drone* menerima *input* kendali dan menghadapi kondisi lingkungan secara langsung. Berikut skenario beserta uji coba.

2. Skenario *Failsafe*

Failsafe merupakan salah satu aspek yang harus diperhatikan dalam *flight controller drone FPV*, keamanan saat terjadi kehilangan sinyal kendali, baik akibat gangguan sinyal maupun kondisi yang tak terduga pada saat melakukan penerbangan. Untuk itu, dilakukan skenario uji fitur *failsafe*, guna memastikan sistem dapat merespon.

Gambar 17 adalah skenario pengujian *failsafe*: sta



GAMBAR 15
Skenario *failsafe*

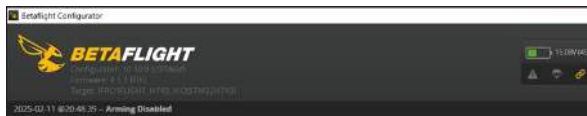
System info	
Arming Disable Flags:	RX_FAILSAFE MSP
Battery voltage:	0.46 V
Capacity drawn:	0 mAh
Current draw:	0.00 A
RSSI:	10 dBm
CPU Temperature:	42 °C

GAMBAR 16
Informasi *failsafe*

Pada gambar 18 terdapat indikasi *arming disable flags* yang berarti sistem *flight controller* mendeteksi hilangnya sinyal dari *receiver*(RX) atau belum menerima *input* valid dari *transmitter*. Kondisi *rx-failsafe* ini otomatis mencegah *drone* untuk melakukan *arming* sebagai langkah pengamanan, karena tanpa kendali dari pilot, *drone* berpotensi bergerak tanpa kontrol. Tahapan skenario *failsafe* aktif ketika respons sinyal terganggu atau hilang yang mengakibatkan *transmitter* dan *receiver* mengalami pencarian sinyal (*proses binding*). Untuk mengatasi *rx-failsafe*, perlu dilakukan pengecekan pada koneksi *receiver*, memastikan *transmitter* dalam keadaan menyala dan terhubung dengan benar, serta melakukan *binding* ulang apabila diperlukan. Mode *failsafe* ini penting untuk menjaga keselamatan *drone* dan lingkungan sekitar jika terjadi kegagalan komunikasi.

3. Kestabilan dan keamanan *power supply*

Gambar 19 menampilkan menunjukkan tampilan atas dari *betaflight configurator* versi 10.10.0 yang terhubung dengan *flight controller*. Pada sisi kanan atas, terdapat indikator baterai yang menunjukkan tegangan sebesar 15.08V, yang berarti sistem telah terhubung ke sumber daya eksternal, baterai yang digunakan adalah *LiPo 4S (4 sel)*, karena tegangan nominal untuk 4S adalah sekitar 14.8V hingga 16.8V saat penuh.

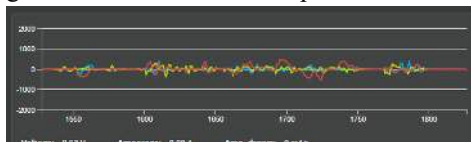


GAMBAR 17
Daya baterai

Flight controller membutuhkan suplai tegangan yang stabil untuk mengoperasikan sensor, *receiver* dan *output ESC*. Nilai 15.08V menandakan baterai dalam kondisi yang baik dan cukup penuh.

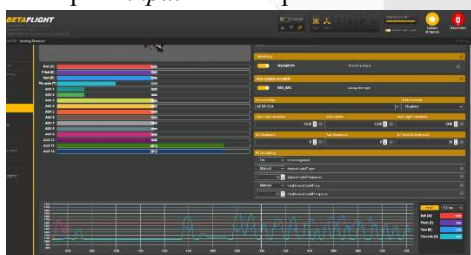
4. Verifikasi performa

1. Responsi *arming* yaitu ketika motor dijalankan dengan menggunakan *radio controller (transmitter)*, berfungsi dengan baik, dimana ketika perintah remot pada bagian *stick* kiri kondisi motor stabil dan menunjukkan grafik pada gambar 20, bahwa *drone* dalam kondisi aman untuk di-*arming* dengan data *gyro* yang stabil dan tanpa adanya *input* yang tidak diinginkan. Indikasi ini menandakan bahwa sensor bekerja dengan normal dan tidak terdapat getaran berlebih atau *error* pada saat *idle*.



GAMBAR 18
Grafik *arming*

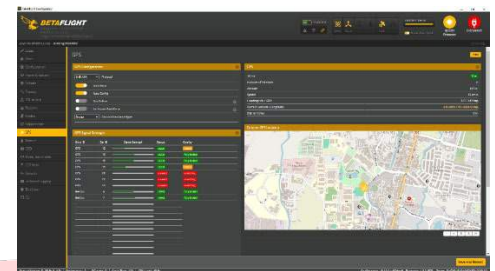
2. Responsi *acro* pada grafik dalam gambar 21 menunjukkan bahwa sistem kendali *drone* merespons *input* dari *transmitter* saat berada pada model penerbangan manual tanpa stabilisasi otomatis. Grafik menampilkan nilai dari kanal utama seperti *throttle*, *pitch*, *roll*, dan *yaw* dalam bentuk fluktuasi sinyal terhadap waktu. Data grafik menunjukkan bahwa mode *acro* telah dikonfigurasi dengan baik, dengan semua sumbu kontrol merespons *input* secara tepat.



GAMBAR 19 Grafik dan respons *acro*

3. Respon GPS, pada konfigurasi ini mengacu pada kesiapan sistem navigasi satelit untuk mendukung fitur *GPS Rescue*. Dalam gambar 22 menampilkan status *GPS lock* menunjukkan bahwasannya *drone* berhasil

mengunci posisi dengan akurasi yang cukup untuk digunakan dalam manuver otomatis seperti kembali ke titik awal. Sementara itu, *tracking map* memberikan gambaran visual posisi *drone* terhadap koordinat yang cukup akurat.



GAMBAR 20
GPS respons

4.4 Analisa hasil pengujian

Pengujian sistem kendali *drone FPV* menunjukkan kinerja yang sesuai. Semua kanal kontrol (*throttle*, *pitch*, *roll*, dan *yaw*) memberikan respons linier dalam rentang kerja normal, yang menandakan komunikasi berbasis *betafight* via protokol komunikasi *expresslrs* bekerja pada *iFlight Beast H7* ini. Sensor *gyro* dan *accelerometer* teruji dengan baik juga, dengan menunjukkan pola pembacaan yang halus. Fitur *failsafe* berfungsi sesuai standar keamanan, secara otomatis menonaktifkan mode *arming* dan menampilkan indikator sinyal yang terputus. Daya baterai *LiPo 4S* (15.08V) cukup untuk mendukung operasional sistem, dan fitur *GPS rescue* menunjukkan koneksi serta penguncian satelit, memungkinkan *drone* mengakses koordinat dan mengenali titik awal penerbangan.

4.5 Keterbatasan sistem

Selama proses perancangan dan implementasi sistem kendali *drone FPV* berbasis *betafight* dan *expresslrs happymodel ep-1*, terdapat beberapa keterbatasan teknis dan non-teknis yang berhasil diidentifikasi melalui serangkaian pengujian. Keterbatasan ini menjadi evaluasi penting terhadap sistem yang dikembangkan.

1. Ruang lingkup uji terbatas, sistem hanya dilakukan dalam area laboratorium RA ATAP Universitas Telkom dan sekitarnya.
2. penggunaan *GPS rescue* telah berhasil dikonfigurasi dan diaktifkan, fungsi ini hanya diuji pada skenario dasar.
3. Pengujian belum mencakup berbagai kondisi lingkungan.

V. KESIMPULAN

Penelitian ini berhasil merancang dan mengimplementasikan sistem kendali *drone FPV* berbasis *firmware open-source betafight* dan protokol komunikasi *expresslrs* yang menggunakan modul *receiver happymodel ep-1*. Sistem ini dikembangkan sebagai solusi terhadap keterbatasan sistem *plug and play* pada *drone* komersial, sekaligus mendukung kebutuhan riset yang menuntut fleksibilitas dalam konfigurasi dan integrasi perangkat.

Hasil pengujian menunjukkan bahwa seluruh kanal kendali berfungsi dengan akurat, respons kontrol bersifat *real-time* pada *radio controller*, dan fitur keselamatan seperti *failsafe* dan *GPS rescue* bekerja optimal, walaupun pengujian bersifat statis. Sistem mampu mempertahankan kestabilan daya dan komunikasi, serta menunjukan performa yang relatif responsif saat diuji di berbagai skenario statis dan dinamis. Selain itu, hambatan teknis seperti kesalahan *binding* dan deteksi *UART* berhasil diatasi melalui pembaruan *firmware* dan rekonstruksi jalur koneksi.

REFERENSI

- [1] G. Gamin, "IMPLEMENTASI KEBIJAKAN PENGGUNAAN DRONE PADA KEMENTERIAN LINGKUNGAN HIDUP DAN KEHUTANAN," *Jurnal Analisis Kebijakan Kehutanan*, vol. 18, no. 2, pp. 125–143, Nov. 2021, doi: 10.20886/jakk.2021.18.2.125-143.
- [2] J. Peksa and D. Mamchur, "A Review on the State of the Art in Copter Drones and Flight Control Systems," Jun. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/s24113349.
- [3] H. Y. Irwanto *et al.*, "Development Steps of Avionics and Flight Control System of Flight Vehicle," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 4, pp. 791–796, Aug. 2023, doi: 10.29207/resti.v7i4.4953.
- [4] ArduPilot, "iFlight Beast H7 55A AIO Specifications," 2024.
- [5] O. Nahorniuk, "METHOD OF RECOGNITION OF FPV-UAV RADIO SIGNALS FORMED ACCORDING TO CROSSFIRE AND EXPRESSLRS STANDARDS," *Наукові праці Державного науково-дослідного інституту випробувань і сертифікації озброєння та військової техніки*, vol. 20, no. 3, pp. 90–98, Oct. 2024, doi: 10.37701/dndivsovt.21.2024.11.
- [6] "A Complete Guide to Flashing and Setting Up ExpressLRS A step-by-step guide to setting up ExpressLRS on your OpenTX/EdgeTX radio and Betaflight, including flashing the latest firmware," 2024. [Online]. Available: <https://oscarliang.com/product-j7yn>
- [7] P. Li, D. Liu, and S. Baldi, "Plug-and-play adaptation in autopilot architectures for unmanned aerial vehicles," in *IECON Proceedings (Industrial Electronics Conference)*, IEEE Computer Society, Oct. 2021. doi: 10.1109/IECON48115.2021.9589106.
- [8] "Firmware Options - ExpressLRS," no. firmware checking, Jan. 2024.
- [9] "Understanding the Different STM32 Processors in Flight Controllers." [Online]. Available: https://oscarliang.com/fl-f3-f4-flight-controller/?utm_source=chatgpt.com
- [10] "Understanding the Different STM32 Processors in Flight Controllers." [Online]. Available: https://oscarliang.com/fl-f3-f4-flight-controller/?utm_source=chatgpt.com
- [11] ot0tot, AghaSaad04, and deadbytefpv, "Receiver Wiring - ExpressLRS," Feb. 2024.
- [12] U. Kristen and S. Wacana, "Analisis Kualitas Signal Wireless Berdasarkan Received Signal Strength Indicator (RSSI) pada."
- [13] S. Prakosa, A. Nugraha, L. Sunuharjo, and M. ' Atiq, "Komunikasi Arduino I2C, SPI dan UART," no. 4, pp. 80–85, 2024, doi: 10.62951/switch.v2i4.18.
- [14] deadbytefpv, ot0tot, maybenikhil, and AghaSaad04, "Happymodel EP - ExpressLRS," Aug. 2024.
- [15] NUR ALAMSYAH, ROSMIATI, and HARIANI, "PERANCANGAN PROTOTYPE UNMANNED AERIAL VEHICLE (UAV) BERBASIS QUADCOPTER SEBAGAI SISTEM KEAMANAN GEDUNG," Apr. 2022.