

SISTEM PEMBARUAN OTOMATIS PERANGKAT LUNAK PADA SISTEM KENDALI PESAWAT TERBANG, STUDI KASUS: MCAS BOEING 737 MAX

¹Muhammad Alfandy Pramana Lubis, ²Reo Yudhono, ³Muhammad Luqman Bukhori

^{1,2,3}*Teknik Dirgantara, STTKD Yogyakarta*

Abstrak

Maneuvering characteristic augmentation system adalah sistem komputer yang digunakan untuk mencegah terjadinya nose up berlebih pada pesawat Boeing 737 Max, semakin tinggi nose up dari pesawat akan menyebabkan pesawat kehilangan gaya angkat (stall). Sistem ini dibuat untuk mengkompensasi adanya pergeseran engine baru pada pesawat Boeing 737 Max yang cenderung kedepan, namun hal ini tidak diimbangi dengan perhitungan center of gravity yang baru. Sehingga pesawat memiliki kecenderungan untuk selalu nose up, sehingga dengan adanya sistem MCAS dapat menghindari pesawat cenderung nose up. Namun akibat sistem MCAS yang salah pembacaan terhadap posisi aktual pesawat maka MCAS akan mengendalikan elevator agar pesawat dapat nose down, karena pilot belum bisa mematikan MCAS maka pilot kesulitan untuk mengendalikan pesawat dan pesawat mengalami kecelakaan. Berdasarkan permasalahan tersebut peneliti membuat rancangan perangkat lunak pengganti MCAS yang memiliki kemampuan pembaruan dan kalibrasi secara otomatis untuk mendeteksi keadaan angle of attack dari pesawat aeromodelling. Software yang digunakan pada penelitian ini terintegrasi dengan indikator LED dan buzzer, selanjutnya dihubungkan dengan servo elevator. LED dan buzzer digunakan untuk memberitahu apabila terjadi kegagalan pembacaan angle of attack antara software dan posisi aktual pesawat. Hasil penelitian menunjukkan rancangan software dapat bekerja dengan baik dengan mengintegrasikan XAMPP, website server data, MySQL, dan Apache. Ketika terjadi kegagalan pembacaan sudut angle of attack maka LED dan buzzer akan menyala, kemudian menghubungkan server display ke XAMPP, Apache dan MySQL. Setelah semua terhubung maka kalibrasi akan dilakukan dan data akan dikirim ke sistem pesawat.

Kata kunci: MCAS, sudut, kegagalan, software, internet

Abstract

Maneuvering characteristic augmentation system is a computer system used to prevent excessive nose up on Boeing 737 Max aircraft, the higher the nose up of the aircraft will cause the aircraft to lose lift force (stall). The system was created to compensate for the new engine shifts on boeing 737 Max aircraft that are likely to be in the future, but this is not offset by the new center of gravity calculations. So that the aircraft has a tendency to always nose up, so that with the MCAS system, it can avoid aircraft tending to nose up. However, due to the MCAS system's misreading of the actual position of the aircraft, MCAS will control the elevator so that the aircraft can nose down, because the pilot has not been able to turn off MCAS, the pilot has difficulty controlling the aircraft and the aircraft has an accident. Based on these problems, researchers made a software design to replace MCAS that has the ability to update and calibrate automatically to detect the angle of attack state of the aeromodelling aircraft. The software used in this study is integrated with LED indicators and buzzers, then connected with servo elevators. LEDs and buzzers are used to notify in the event of a failure of the angle of attack reading between the software and the actual position of the aircraft. The results showed that the software design can work well by integrating XAMPP, data server websites, MySQL, and Apache. When there is a failure to read the angle of attack, the LED and buzzer will light up, then connect the display server to XAMPP, Apache and MySQL. Once all are connected then calibration will be carried out and data will be sent to the aircraft system.

Keywords: MCAS, angle, failure, software, internet

Pendahuluan

Pada saat ini banyak perusahaan manufaktur penerbangan yang saling berlomba-lomba untuk menciptakan inovasi, hal ini sesuai dengan revolusi industri 4.0. Ada 2 perusahaan pesawat di dunia yang saling bersaing yaitu Airbus dan Boeing. Berdasarkan pernyataan dari Coats (2019) kedua perusahaan tersebut saling bersaing dalam segi efisiensi *fuel* pesawat, sehingga dapat menarik perusahaan *airlines*. Ketika Airbus telah mengeluarkan pesawat terbarunya, maka Boeing langsung

¹Email Address: 180202064@students.sttkd.ac.id
Received 1 Juni 2022, Available Online 30 Juli 2022

berusaha keras agar dapat mengeluarkan pesawat baru juga. Agar cepat dalam mengeluarkan pesawat terbaru maka pihak Boeing merubah rancangan pesawat Boeing 737 dengan mengganti *engine* yang lebih besar dan memiliki efisien. Untuk mengatasi distribsui momen yang terjadi akibat dari *engine* yang lebih besar maka rancangan Boeing 737 memiliki *nose landing gear* yang lebih panjang dari sebelumnya sebesar 8 inch. Namun hal ini menyebabkan masalah, menurut Johnston dan Harris (2019) karena rancangan tersebut maka menyebabkan karakteristik *center of gravity* menjadi berubah dan menyebabkan pesawat memiliki kecenderungan untuk mendongak ke atas. Akibat hal tersebut tentunya dapat menyebabkan resiko terjadinya *stall*. Untuk mengatasi kemungkinan terjadinya *stall* maka Boeing memberikan sistem MCAS (*maneuvering characteristic augmentation system*). Dengan sistem tersebut maka dapat otomatis menyeimbangkan posisi pesawat yang cenderung mendongak ke atas, meskipun pesawat tidak dalam mode *autopilot* (Johnston dan Harris, 2019). Namun pada penerapannya dalam *cockpit* tidak dipasang *warning display* dan juga tidak memperbarui sistem *software* MCAS, selain itu tindakan yang harus dilakukan oleh pilot ketika sistem *error* belum tertulis di *manual checklist*. Berdasarkan uraian tersebut maka peneliti tertarik untuk membuat penelitian dengan membuat sistem pembaruan otomatis *software* pada sistem kendali pesawat terbang, dengan menggunakan referensi dari kasus MCAS Boeing 737 Max. Dengan adanya penelitian ini maka diharapkan dapat menjadi solusi terkait kegagalan sistem MCAS Boeing 737 Max.

Tinjauan Pustaka

MCAS

MCAS adalah sebuah sistem berbasis komputasi yang digunakan untuk mencegah terjadinya kecenderungan *nose up* yang berlebih pada pesawat Boeing 737 Max. Akibat dari *nose up* yang berlebih dapat menyebabkan pesawat kehilangan gaya angkat (*stall*). Pada kejadian jatuhnya pesawat Boeing 737 Max disebabkan adanya sistem MCAS yang membaca pesawat terlalu mendongak keatas, sehingga sistem akan memaksa pesawat untuk *nose down*. Namun pada kondisi aktual pesawat masih dalam kondisi normal, sehingga pilot akan berusaha untuk menaikkan pesawat, namun sistem lebih menguasai sistem kendali. Hal itulah yang menyebabkan pesawat Boeing 737 Max mengalami kecelakaan. MCAS digunakan untuk mengatasi perubahan karakteristik *center of gravity* dari pesawat keluarga 737 sebelumnya (Indhana, Sujanto, dan Setiyo 2019). Dengan demikian maka MCAS merupakan sistem untuk memberikan kendali cadangan ketika pesawat mengalami *nose up* yang berlebih.

Sensor IMU 10 DOF

Sensor jenis ini mampu mendeteksi orientasi, kecepatan dan gaya gravitasi sebuah objek. Sistem dari sensor ini menggunakan dasar akselerometer, *gyroscope*, dan magnetometer. Dengan menggunakan sensor ini maka dapat mendeteksi kemiringan atau kondisi dari objek yang dipasang sensor ini.

Gyroscope

Dasar dari sistem *gyro* menggunakan persamaan 1 dan 2.

$$\begin{bmatrix} z' \\ y' \\ x' \end{bmatrix} = \begin{bmatrix} 1 & \sin(x) \tan(y) & \cos(x) \tan(y) \\ 0 & \cos(z) & -\sin(z) \\ 0 & \sin z / \cos(y) & \cos(z) / \cos(y) \end{bmatrix} \begin{bmatrix} \omega z \\ \omega y \\ \omega x \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} z' \\ y' \\ x' \end{bmatrix} = \begin{bmatrix} 1 & \sin(x) \tan(y) & \cos(x) \tan(y) \\ 0 & \cos(z) & -\sin(z) \\ 0 & \sin z / \cos(y) & \cos(z) / \cos(y) \end{bmatrix} \begin{bmatrix} \omega z \\ \omega y \\ \omega x \end{bmatrix} \quad (2)$$

Accelerometer

Persamaan *accelerometer* menggunakan persamaan 3.

$$|a| = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (3)$$

Kemudian untuk *pitch* dan *yaw* menggunakan persamaan 4 dan 5.

$$\phi_a = \tan^{-1} \left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}} \right) \quad (4)$$

$$\phi_a = \tan^{-1} \left(\frac{a_x}{\sqrt{a_y^2 + a_z^2}} \right) \quad (5)$$

Magnetometer

Dasar dari prinsip kerja magnetometer menggunakan persamaan 6 dan 7.

$$Y = mx \times \cos(z) + my \times \sin(z) \quad (6)$$

$$X = mx \times \cos(y) + my \times \sin(z) \times \sin(y) - mz \times \cos(z) \times \sin(y) \quad (7)$$

$$Yaw, P = \arctan\left(\frac{Y}{X}\right) \quad (8)$$

Buzzer

Buzzer adalah komponen yang memiliki kemampuan mengubah sinyal listrik menjadi getaran suara. Pada umumnya *buzzer* digunakan untuk pembuatan alarm atau sebagai indikator suara pada suatu mekanisme. Cara kerja *buzzer* yaitu dengan memberikan tegangan listrik sebesar 3-12 V ke rangkaian yang terdapat *piezoelectric*. *Output* dari *piezo buzzer* dapat menghasilkan frekuensi sebesar 1-6 kHz dan sampai dengan 100 kHz.

Node MCU

Komponen ini berfungsi sebagai mikrokontroler dari sebuah rangkaian, untuk kemudahan komunikasi maka ditambahi dengan WiFi ESP8266. Mikrokontroler ini memiliki memori, port digital *input* dan *output*. Setiap port memiliki fungsi sebagai UART, SPI, I2C dan lain-lain.

XAMPP

XAMPP adalah *software* yang memberikan fasilitas *web hosting* secara *offline*, sehingga meskipun dalam keadaan *offline* maka tetap dapat mengirimkan data. Menurut Palit *et al.* (2015) XAMPP merupakan server yang dapat berdiri sendiri (*localhost*), untuk komponen yang ada pada XAMPP adalah apache, MySQL, PHP, HTTP Server dan Perl. Fungsi dari apache adalah sebagai web server yang bersifat *open source*. Selanjutnya fungsi dari MySQL adalah untuk mengelola database yang sudah terdaftar. PHP digunakan untuk melakukan *programming website* secara dinamis dibantu dengan Perl.

Pengiriman dan Penerimaan Data

Dalam pengiriman dan penerimaan data maka diperlukan beberapa komponen agar dapat saling terintegrasi. Menurut Handika dan Riadi (2014) komponen dalam sistem komunikasi data adalah sumber (*source*), pemancar (*transmitter*), sistem transmisi (*transmission system*), penerima (*receiver*), dan tujuan (*destination*).

Perhitungan CG

Dalam penempatan sensor untuk AoA pada pesawat maka diterapkan pada *center of gravity* dari model pesawat yang digunakan. Menurut Duncan (2018) persamaan mencari CG adalah sebagai berikut:

$$CG = \frac{\Delta M}{\Delta W} \quad (9)$$

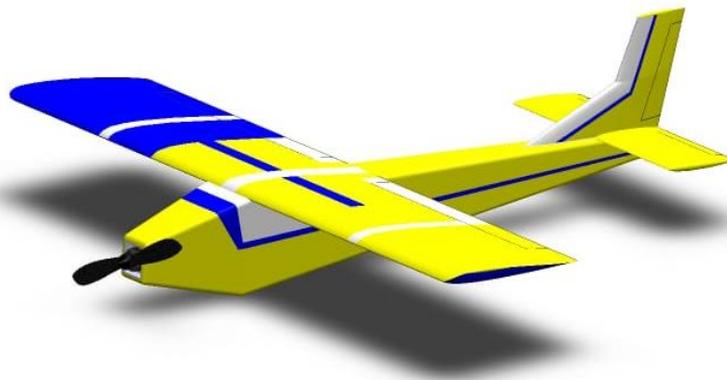
Metode Penelitian

Rancangan Penelitian

Penelitian dilakukan secara eksperimental dengan membuat model pesawat *aeromodelling* tipe cessna 172 yang terbuat dari *polyfoam*. Adapun untuk bahan yang digunakan pada penelitian terdapat pada Tabel 1.

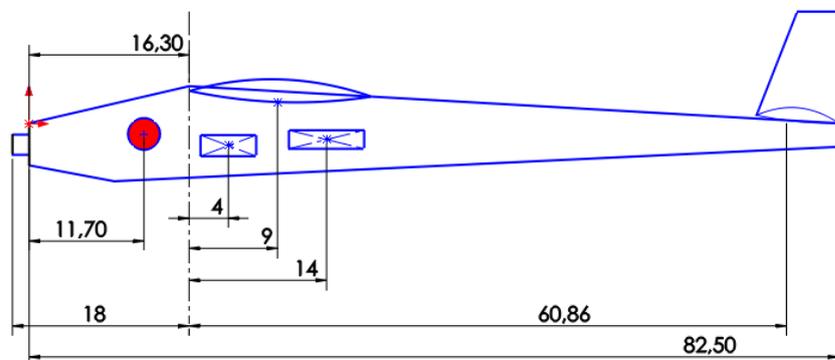
Tabel 1. Bahan penelitian

Bahan	Fungsi Bahan
Polyfoam	Bahan dasar pembuatan pesawat aeromodelling
Resin	Bahan pelapis dari polyfoam
Servo	Penggerak flight control
Motor	Engine pesawat
Transceiver	Pengirim dan penerima signal



Gambar 1. Model pesawat

Kemudian untuk menentukan peletakan sensor IMU maka perlu dihitung lokasi dari CG model pesawat. Lokasi CG pada model pesawat yang digunakan seperti pada Gambar 2.



Gambar 2. Perhitungan CG

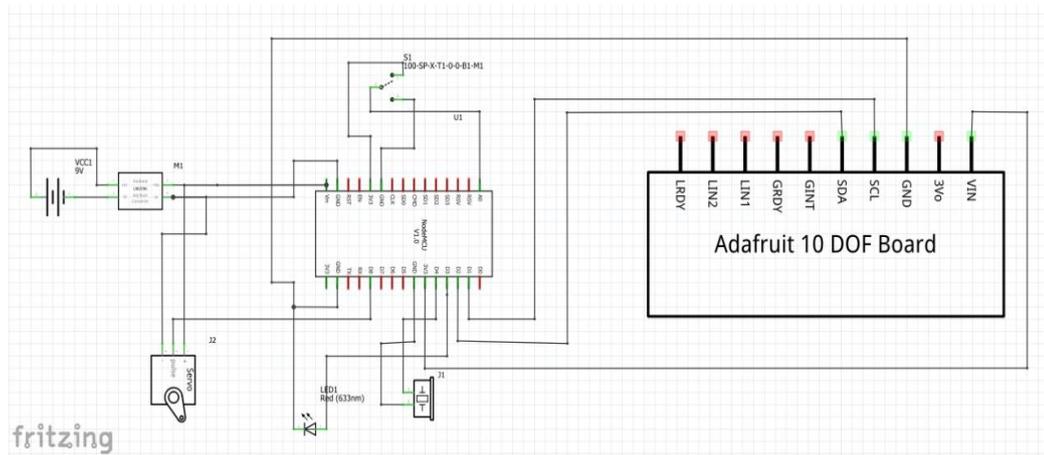
Untuk permodelan sistem MCAS maka menggunakan sensor IMU 10 DOF yang diletakkan pada CG model pesawat yang digunakan. Untuk sensor akan dipogram memiliki maksimal sudut 30° , dan apabila melebihi batas sudut yang telah ditentukan maka *buzzer* akan berbunyi. Untuk perancangan dari jaringan *localhost* maka menggunakan *internet wireless* yang digunakan untuk mencegah

terjadinya kegagalan *software* dalam memperbarui data secara otomatis dengan menghubungkan ke *internet server* dengan *client*. Adapun perancangan jaringan internet terdapat pada Tabel

Tabel 2. Perancangan jaringan internet

Alat	Fungsi
<i>Server display</i>	Sebagai server dari permintaan <i>client</i> dalam pembaruan <i>software</i> . <i>Server</i> ini akan terhubung dengan Xampp.
Xampp	Tempat penyimpanan data yang dihasilkan dari <i>server display</i> , dan akan diteruskan secara otomatis.
Node MCU	Sebagai <i>transceiver</i> dari sistem yang terhubung ke Xampp

Kemudian untuk *wiring diagram* dari sistem yang akan digunakan terdapat pada Gambar 3.



Gambar 3. Wiring diagram

Langkah-Langkah Penelitian

Dalam penelitian ini terdapat 2 langkah, yaitu untuk pengujian sensor dan *buzzer* serta *recovery* data pembaruan *software* pada pesawat.

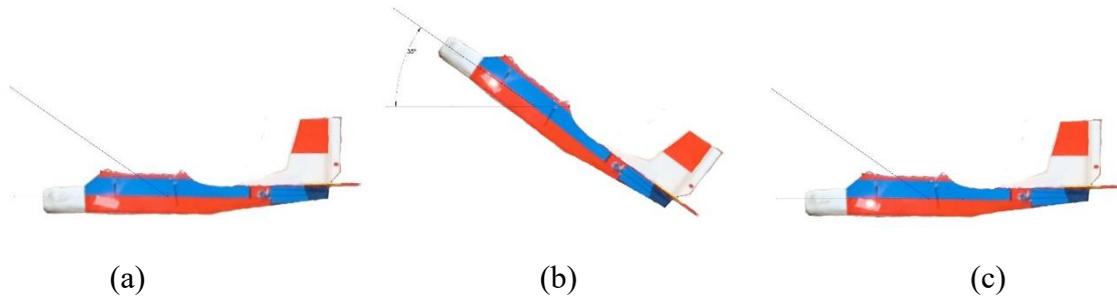
Pengujian sensor dilakukan dengan menjalankan animasi pergerakan dari pesawat yang melebihi *angle of attack* sebesar 30°, kemudian suara *buzzer* akan berbunyi dan memicu elevator agar bergerak untuk mengembalikan ke posisi 0°.

Untuk tahap *recovery data* dilakukan dengan menghubungkan *server display* ke *internet*, kemudian melakukan pengaturan pada XAMPP ke modul pesawat node MCU. Server akan melakukan pembaruan ke *server data calibration*. Setelah itu pesawat akan menerima pembaruan *software* terbaru yang diberikan *server*. Ketika terjadi data error pada pesawat, maka akan di feedback ke *cloud* untuk disimpan di *datatable*, selanjutnya user di server display akan mengkalkulasikan *datatable* yang diterima dan pembaruan *software*. Tahap *recovery* akan diimplementasikan pada *maintenance monthly check* oleh *engineer*.

Hasil dan Pembahasan

Perancangan Peringatan Kegagalan

Untuk indikator peringatan kegagalan maka peneliti menggunakan LED dan *buzzer*, prinsip kerjanya yaitu apabila pesawat dalam kondisi 0° maka LED dan *buzzer* akan mati. Kemudian jika pesawat melebihi batasan sudut 30° maka LED dan *buzzer* akan menyala. Selain itu LED dan *buzzer* akan menyala apabila terjadi pembacaan yang salah pada *software* terhadap posisi aktual.



Gambar 4. Posisi pesawat, (a) posisi AOA 0° , (b) posisi lebih dari 30° , (c) posisi aktual 0° namun pembacaan *software* berbeda

Kalibrasi Pitch pada IMU DOF 10

Untuk memastikan sensor yang digunakan berfungsi dengan baik maka melakukan kalibrasi, adapun kalibrasi dilakukan dengan menggunakan persamaan 10 (Islam et al., 2017).

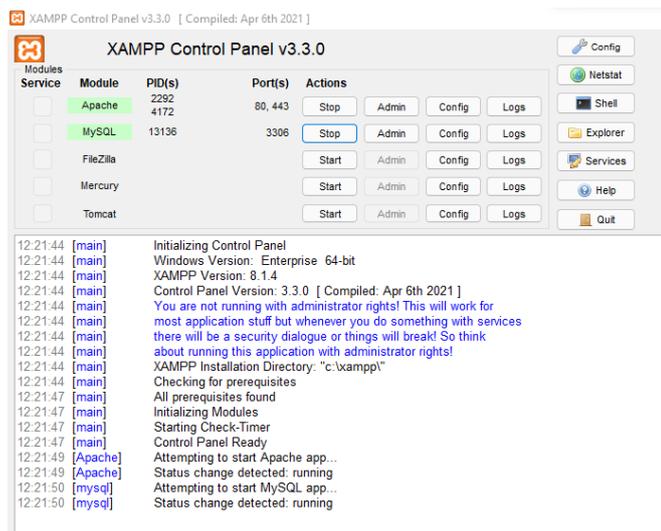
$$\phi_a = \tan^{-1}\left(\frac{a_y}{\sqrt{(a_{x^2}+a_{z^2})}}\right) \quad (10)$$

Atau dalam bahasa pemrograman dapat dikonversikan sebagai berikut.

$$(\text{atan2}(fXg, \text{sqrt}(fYg*fYg + fZg*fZg)) * 180.0) / M_PI;$$

Data Server

Proses kalibrasi *angle of attack* dengan menggunakan *software* XAMPP seperti tertampil pada Gambar 5.



Gambar 5. Server XAMPP

Website Server Data Calibration

Melakukan kalibrasi terhadap sumbu X, Y, dan Z pada pesawat untuk memastikan pembacaan dari *software* sudah akurat seperti pada Gambar 6.


```
COM5
.....
[INFO] Connected to STTKDedu
[INFO] IP address: 10.1.15.74
[INFO] MPU6050 Found!
[INFO] Accelerometer range set to: +-8G
[INFO] Gyro range set to: +- 500 deg/s
[INFO] Filter bandwidth set to: 21 Hz
```

Gambar 9. Terkoneksi jaringan internet

- b. Memulai *software* XAMPP, Apache, dan MySQL.
- c. Membuka website *localhost* yang sudah dibuat di browser untuk memastikan sudah terkoneksi, adapun linknya yaitu `10.1.11.107/ujicoba/dataread.php?id=1`.
- d. Jika sudah tertampil *Server Data Calibration* maka selanjutnya melakukan input data Y (*pitch*) menjadi $0,0^\circ$.
- e. Melakukan *turn off* dan *turn on switch* pada pesawat untuk melakukan proses kalibrasi data baru. Jika kalibrasi sudah selesai maka pada *software* akan tertampil seperti pada Gambar 10.

```
Update data Calibration
connecting to 10.1.11.107
Requesting URL: /ujicoba/dataread.php?id=1

closing connection

0.00
0.00
1.40

connecting to 10.1.11.107
Requesting URL: /ujicoba/dataread.php?id=1
```

Gambar 10. Hasil kalibrasi data

- f. Kemudian Y (*pitch*) akan kembali ke sudut 0° , menandakan bahwa data berhasil diperbarui.

```
roll: 0 pitch: 0 pos_servo: 90
```

Gambar 11. Pitch kembali ke 0° ,

Kesimpulan

Penelitian ini dilakukan dengan referensi dari kasus MCAS Boeing 737 Max, indikator yang ada pada pesawat tersebut hanya berupa *alert light* pada bagian SPEED TRIM FAIL yang berlokasi pada *overhead panel cockpit*. Penelitian ini mengembangkan sistem pembaruan *software* yang dapat mengupdate secara otomatis ketika terjadi kegagalan dalam pembacaan *angle of attack*.

Dari penelitian yang sudah dilakukan sistem yang sudah dibuat meliputi penggunaan XAMPP dapat bekerja dengan baik dalam memberikan kalibrasi *angle of attack* secara otomatis. Seperti contoh pesawat yang dalam posisi *pitch* aktual *pitch* (Y) 0° sedangkan dalam pembacaan *software* dalam posisi *pitch* (Y) 6° maka LED dan *buzzer* akan aktif. Kemudian *software* akan melakukan kalibrasi dengan menghubungkan ke *server display* dengan menggunakan XAMPP dengan menggunakan jaringan internet, lalu mengubah nilai menjadi *pitch* (Y) 0°.

Daftar Pustaka

- Coots, Madison. 2019. "Maneuvering Characteristics Control Systems using Discrete-Time MDPs."
- Duncan, Jhon s. 2018. *Aviation Maintenance Technician Handbook General*. Oklahoma: Federal Aviation Administration.
- Handika, dan Imam Riadi. 2014. "Media Pembelajaran Komunikasi Data Dan Jaringan." *Journal Sarjana Teknik Informatika* 2(3):10–20.
- Indhana, Ardi Putra, Sujanto, dan Setiyo. 2019. "Seminar nasional inovasi teknologi penerbangan (snitp) tahun 2018." *Seminar Nasional Inovasi Teknologi Penerbangan (SNITP)* 1–7.
- Islam, M. S., Shajid-Ul-Mahmud, M., Islam, T., Amin, M. S., & Hossam-E-Haider, M. (2017). A low cost MEMS and complementary filter based attitude heading reference system (AHRS) for low speed aircraft. *2016 3rd International Conference on Electrical Engineering and Information and Communication Technology, ICEEICT 2016, October*.
- Johnston, Phillip, dan Rozi Harris. 2019. "The Boeing 737 MAX Saga: Lessons for Software Organizations." *Software Quality Professional* 21(3):4–12.
- Palit, Randi V, Yaulie D. Y. Rindengan, dan Arie S. M. Lumenta. 2015. "Rancangan Sistem Informasi Keuangan Gereja Berbasis Web Di Jemaat GMIM Bukit Moria Malalayang." 4(7):1–7.