

Desain Awal Flight Control Panel dari Pesawat LSA-02

¹Adi Wirawan*), ¹Eries Bagita Jayanti

¹Pusat Teknologi Penerbangan - LAPAN

*adi.wirawan@lapan.go.id**)

Abstrak

Pada artikel ini penulis akan membahas mengenai pengembangan Flight Control Panel (FCP) pada pesawat teknologi demonstrator, LSA-02. FCP merupakan interface antara pilot dengan *electronic flight control system* (EFCS) pada pesawat teknologi demonstrator, LSA-02. FCP berfungsi sebagai alat bagi pilot untuk memberikan perintah kepada EFCS dan juga untuk menampilkan informasi terkait dengan kondisi EFCS. Perangkat keras FCP terdiri dari *embedded computer*, LCD, dan *keyboard*. *Keyboard* pada FCP merupakan input dari FCP, terdiri dari tombol dan saklar putar yang diatur oleh sebuah *controller*. FCP dilengkapi dengan perangkat lunak yang dikembangkan untuk mengakomodir tampilan data dari EFCS. FCP menggunakan jalur komunikasi CAN bus untuk berkomunikasi dengan EFCS. **Kata Kunci:** *electronic flight control system, interface, flight control panel.*

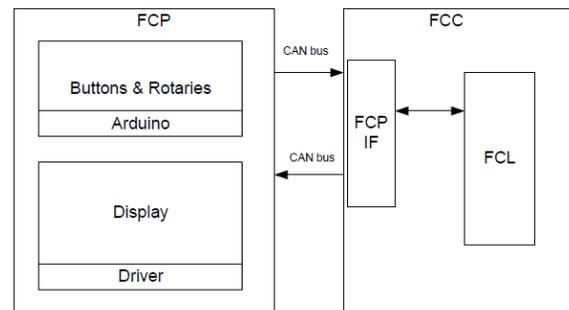
1 Pendahuluan

Light Surveillance Aircraft 02 (LSA-02) merupakan pesawat teknologi demonstrator yang digunakan untuk investigasi dan demonstrasi teknologi *unmanned aerial vehicle* (UAV). LSA-02 merupakan hasil modifikasi dari *basic aircraft* Stemme-15. Pesawat ini memiliki kapasitas muatan yang cukup untuk diintegrasikan dengan *electronic flight control system* (EFCS), instrumen uji terbang dan tambahan muatan uji coba [1]. Pesawat dioperasikan sebagai pesawat berawak oleh *safety pilot*, untuk melakukan percobaan dan memonitor perintah bidang kendali selama percobaan berlangsung. Sedangkan *Flight Control Panel* (FCP) adalah antarmuka antara *safety pilot* dengan *electronic flight control system* (EFCS). EFCS adalah sistem kendali elektronik yang dipasang pada sistem dasar kendali mekanik pesawat. Sistem ini dikembangkan untuk melakukan uji coba pada teknologi demonstrator UAV [2]. FCP adalah *human machine interface* (HMI). FCP diletakkan pada panel avionik di depan pilot. FCP memiliki tiga fungsi utama yaitu: untuk memuat dan mengirimkan data inisialisasi ke FCC untuk dibaca oleh FCL, untuk memungkinkan pilot memasukkan data atau memberi perintah secara manual ke EFCS, dan untuk menampilkan informasi ke pilot [3].

Pada penelitian ini akan dijelaskan terkait pembuatan desain awal FCP. Desain mencakup desain perangkat keras dan perangkat lunak FCP. Perangkat keras sebagai media untuk *safety pilot* dalam memberikan masukan kepada EFCS. Perangkat keras terdiri dari *embedded computer*, layar tampilan FCP dan *mikrokontroler* untuk tombol dan saklar putar [4]. Sedangkan perangkat lunak berfungsi untuk mengakomodir tampilan data dari EFCS. Perangkat lunak ini terdiri dari komunikasi data serial, komunikasi data CAN dan *Graphic Interface unit* (GUI) [4].

2 Diskusi

Secara umum posisi FCP terhadap FCC dijelaskan sebagai berikut [4]:

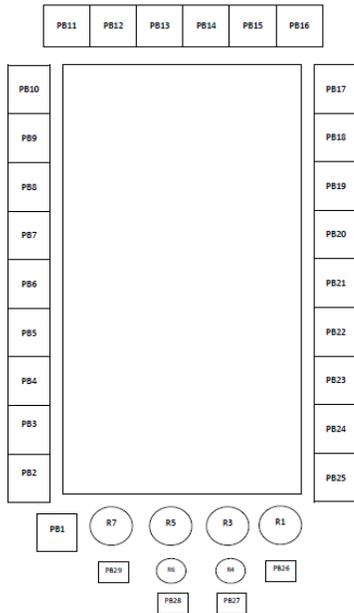


Gambar 1. Posisi FCP terhadap FCC [4]

Gambar (1) menunjukkan konsep dasar interaksi antara FCP dengan FCC. Data masukan dari pilot masuk melalui tombol dan saklar putar dengan Arduino sebagai kontrollernya. Selanjutnya data dikirimkan menuju FCC melalui CAN bus. FCP IF adalah antarmuka antara FCP dan FCL (*Flight Control Law*). Kemudian data masukan tersebut dikirimkan menuju *display* untuk ditampilkan.

Desain konseptual FCP akan ditunjukkan pada gambar (2). Gambar tersebut menunjukkan nama-nama dari tombol dan rotari. Penamaan tombol diberikan "PB" yang berarti push button yang selanjutnya diikuti dengan penomoran. Hal tersebut menunjukkan nomor dari tombol (push button). Sedangkan penamaan saklar putar diberikan "R" yang berarti saklar putar (rotary) dan selanjutnya

diikuti nomor saklar putar. Perangkat keras terdiri dari 29 tombol dan 8 saklar putar. Namun hanya 6 saklar putar yang digunakan [4].

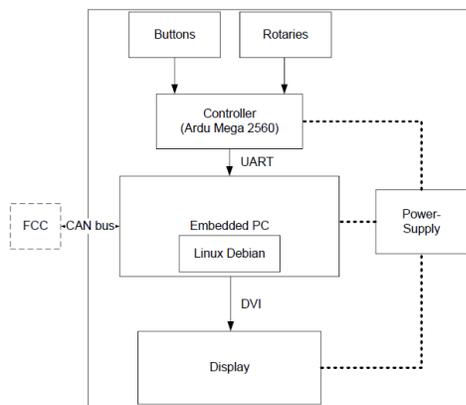


Gambar 2. Desain Konseptual FCP

2.1 Konsep Desain

2.1.1 Perangkat Keras

Terdapat empat komponen utama dalam perangkat keras FCP yaitu *embedded computer*, *LCD display*, tombol dan saklar putar dengan *controller*. Perangkat keras FCP juga terhubung dengan *power supply* yang menggunakan power konverter DC. Berikut merupakan diagram blok desain konseptual dari FCP [4]:



Gambar 3. Diagram Blok Desain Konseptual FCP [4]

1. Sistem Embedded

Sistem embedded adalah desain sistem elektronik/ elektro mekanikal untuk menunjukkan fungsi spesifik dan kombinasi dari perangkat keras dan perangkat lunak. Sistem embedded menjadi bagian penting dari peralatan rumah tangga, telekomunikasi, peralatan kesehatan, industri dll. Sistem embedded didesain untuk suatu tujuan atau kombinasi dengan tugas tugas seperti pengumpulan data/penyimpanan/representasi, komunikasi data, pemrosesan data sinyal, monitoring, kendali dan spesifik aplikasi untuk antarmuka pengguna [5].

Embedded computer yang digunakan pada FCP merupakan produk dari Jantzec Company.



Gambar 4. CPU Intel Atom E3825 [6]

CPU Intel atom E3825 dengan RAM 2GB, port LAN 2 buah, port RS232 sebanyak 1 buah, port CAN 1 buah, port USB 2.0 3 buah, port DVI sebanyak 1 buah, memori onboard Emmc 4GB, dan memori CFast 32 GB. Sistem operasi yang telah diinstall adalah Linux Debian.

2. Layar Tampilan FCP

Layar tampilan FCP merupakan produk dari Litemax, dengan tingkat kecerahan sebesar 1000 nits, resolusi 1024x600 dan memiliki port koneksi DVI.

3. Mikrokontroler Arduino Mega 2560

Arduino adalah sebuah platform perangkat keras *open source* yang memiliki masukan/keluaran yang sederhana [7]. Arduino memberikan masukan/keluaran yang sudah fix dan bisa digunakan dengan mudah. Arduino dapat digabungkan dengan modul elektro yang lain sehingga proses perakitan jauh lebih efisien [8]. Arduino mega 2560 adalah papan mikrokontroler berdasar pada ATmega2560. Arduino mega memiliki 54 digital pin masukan/keluaran (14 diantaranya digunakan untuk keluaran PWM), 16 masukan analog, 4 UARTS (port serial perangkat

keras), 16 MHz oscillator Kristal, penghubung USB, sebuah powerjack, header ICSP, dan tombol reset. Terdiri dari apapun yang dibutuhkan untuk mendukung mikrokontroler, terhubung sederhana ke komputer dengan kabel USB atau power dengan adaptor AC ke DC atau baterai untuk memulainya [9]. Arduino merupakan mikrokontroler untuk tombol dan saklar putar pada FCP.



Gambar 5. Arduino Mega 2560 [9]



Gambar 6. Tombol [10]

4. DC to DC converter

DC to DC converter yang digunakan adalah 24 V dan 12 V, dengan kriteria 9-38 V / 24 V 10 W dan 9-38 V / 12 V 10 W.

2.1.2 Perangkat Lunak

Pada bagian ini akan dijelaskan *requirement* secara umum perangkat lunak dan *requirement* berdasarkan fungsional. Requirement fungsional menjelaskan tentang input pada perangkat lunak, *output* dan detail hubungan antara *input* dan *output* [4].

1. Requirement secara umum

FCP terdiri dari tombol dan saklar putar yang diatur oleh mikrokontroler. FCP juga memiliki sistem *embedded* yang memiliki fungsi untuk mengatur komunikasi dari controller ke FCC. Komunikasi dari controller ke sistem *embedded* menggunakan data UART. Sedangkan komunikasi dari sistem *embedded* ke FCC menggunakan CAN bus. Perangkat lunak FCP menerima masukan tombol dan saklar putar melalui data UART dan meneruskan data ini ke FCC menggunakan CAN Bus. Perangkat lunak FCP juga menerima data CAN dari FCC lalu ditampilkan ke layar FCP. Perangkat

keras FCP menggunakan CAN Bus sebagai komunikasi antarmuka FCP karena data rate yang tinggi dan lebih dapat diandalkan dibandingkan UART.

Controller Area Network (CAN) yang selanjutnya disebut CAN Bus adalah bus yang dirancang untuk memungkinkan unit kontrol elektronik dan perangkat untuk berkomunikasi antar satu sama lain dalam aplikasi tanpa komputer host. CAN Bus memungkinkan berbagai jenis komponen elektronik (seperti: unit kontrol elektronik, mikrokontroler, sensor, aktuator dan lainnya) untuk berkomunikasi pada single atau dual data bus hingga 1 Mbit/s [11].

2. Requirement fungsional.

a. Komunikasi data serial

Fungsinya adalah untuk menangani data serial dari port USB dan menginisiasi koneksi port serial.

b. Komunikasi data CAN

Fungsinya adalah untuk menangani data CAN, menginisiasi CAN *interface*, serta mengirim dan menerima CAN message.

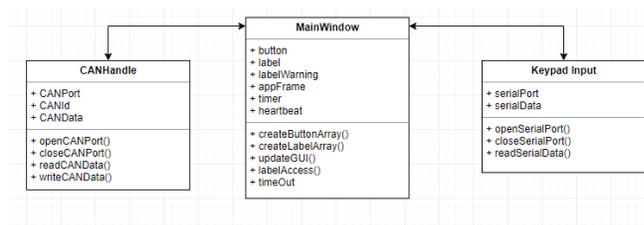
c. Graphical user interface (GUI)

Fungsinya adalah untuk menangani *layout* GUI pada FCP.

Arsitektur perangkat lunak ditampilkan dalam bentuk format diagram UML dan menggunakan pemrograman C++ pada Qt.

Pemilihan Qt sebagai platform pemrograman karena Qt mendukung pengembangan lintas platform aplikasi GUI. Dapat menggunakan sistem operasi Windows untuk pengembangan PC dan selanjutnya mengcompile sumber pada sistem *embedded* Linux. Alasan penting lainnya adalah dengan menggunakan pemrograman Qt mendukung untuk library perangkat lunak yang dibutuhkan, seperti contoh library untuk serial bus dan CAN bus [12].

Berikut merupakan diagram UML dari perangkat lunak FCP:



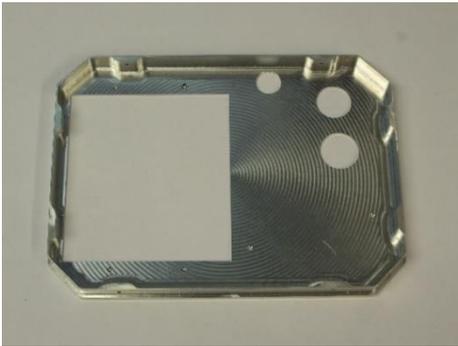
Gambar 7. Diagram UML Perangkat Lunak FCP

Pada diagram tersebut, dijelaskan bahwa perangkat lunak FCP memiliki kelas-kelas. *mainWindow* yang menangani proses utama untuk semua fungsi yang terdiri dari tombol, label, *labelWarning*, *frame*, *timer* dan *heartbeat*. *canHandle* menangani komunikasi CAN bus yang terdiri dari CAN Port, CAN Id, CAN data. *Keypad input* menangani komunikasi data UART yang terdiri dari port serial dan data serial.

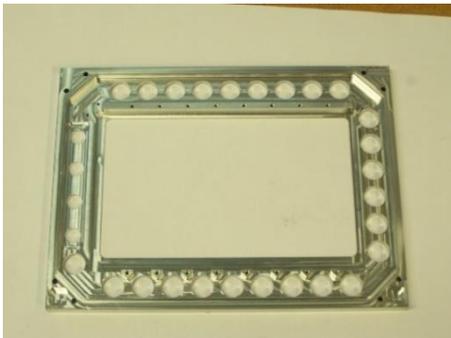
2.2 Implementasi Desain

2.2.1 Perakitan Perangkat Keras

Perakitan dimulai dengan menyiapkan *plate-plate*. Beberapa seperti pada gambar berikut:



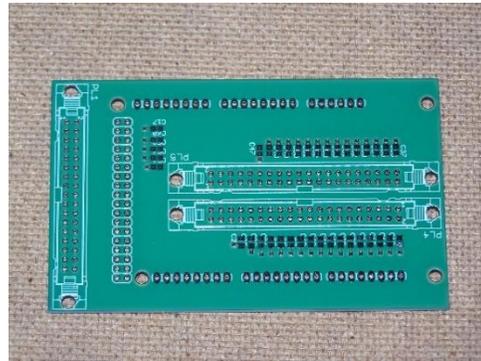
Gambar 8. Plate FCP bagian dalam



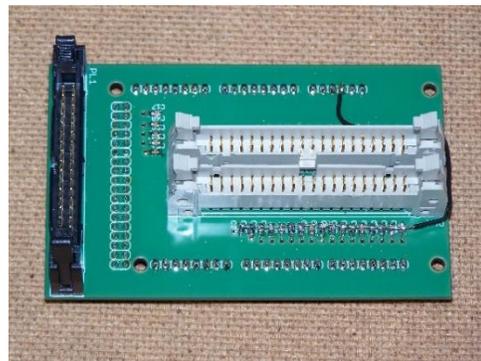
Gambar 9. Tempat Tombol dan Saklar Putar

Kemudian melakukan pemasangan tombol pada *frontplate*, menempatkan kaca tampilan dan tempatnya pada bingkai yang telah tersedia, dan memfiksasi layar tampilan tersebut dengan *display holder*.

Selanjutnya adalah memasang saklar putar ke *frontplate*. Memasang tombol pada kabel yang longgar. Lalu menghubungkan tombol tersebut ke konektor. Langkah selanjutnya adalah pemasangan board pada *plate boardholder*. Memasang koneksi USB SD CardReader dan memasang SD Cardreader pada *frontplate*.



Gambar 10. Plateboard

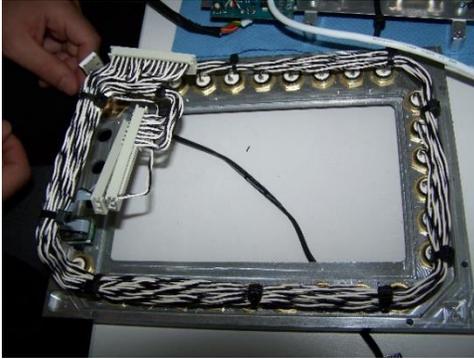


Gambar 11. Plateboard

Langkah berikutnya yaitu melakukan pengkabelan tombol dan saklar putar, menyesuaikan kabel DVI dan konektor power untuk *display control board*, menyesuaikan Arduino mega dan *shield connector* dan membuat Arduino Shield untuk tombol input. Kemudian menempatkan *board* dan *plate board*, menghubungkan kabel dari *plate board* ke bagian belakang layar tampilan, melakukan interkoneksi *board* ke *plate board* dan melakukan pemasangan *plate board* ke *holder* layar tampilan dan *frontplate* ke *frame*.



Gambar 12. Pemasangan Kabel



Gambar 13. Kabel telah terpasang

Perakitan dilanjutkan dengan memasang komputer ke *backplate*, membangun konektor ke *backplate*, menyesuaikan konektor pada *backplate*, menempatkan dan mengkoneksikan DC to DC converter dan terakhir yaitu pengkabelan *backplate* dan *frontplate*.

Berikut merupakan tampilan dari perangkat keras FCP:



Gambar 14. Perangkat Keras FCP

2.2.2 Implementasi perangkat lunak

Berdasarkan spesifikasi *requirement* dan desain perangkat lunak [4]:

1. Modul 1: Komunikasi data UART

Modul ini menggunakan *library serialhandles header, keyInput header, QSerialPort (QSerialPort dan QSerialPortInfo)*. Modul menyediakan antarmuka untuk perangkat lunak ke perangkat keras serial port. Sedangkan file *Keyinput header* adalah antarmuka untuk menerjemahkan serial data dari FCP keyboard.

Modul komunikasi data UART memiliki lima fungsi yaitu:

- *checkSerialPort* untuk mendapatkan informasi terkait kesediaan port serial

- *openSerialPort* untuk mengatur nama port, baud rate, bit data, parity, bit stop, dan kontrol flow dari port yang telah terplih serta menunjukkan status koneksi.
- *closeSerialPort* memiliki fungsi untuk menutup serial port.
- *readSerialPort* memiliki fungsi untuk membaca data yang masuk.
- *writeSerialPort* memilik fungsi untuk menulis data ke serial port.

2. Modul 2: Komunikasi data CAN

Modul ini menggunakan *library QSerialBus (QCanBusDeviceInfo dan QCanBusFrame)* dan *QCanBus*. Modul ini memiliki lima fungsi yaitu:

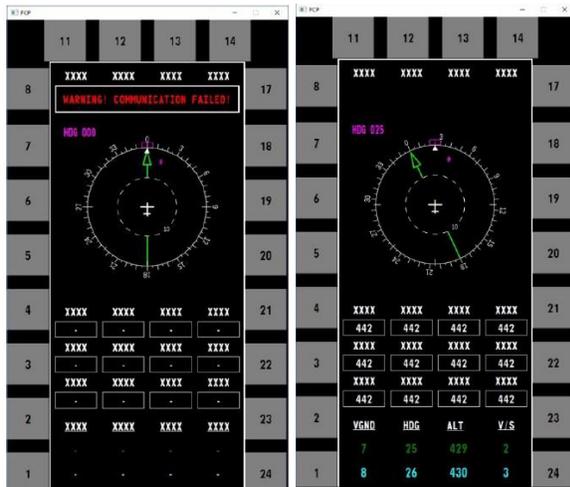
- *CheckCANPort* untuk mengecek kesediaan dari port CAN
- *OpenCANPort* untuk membuka port CAN
- *CloseCANPort* untuk menutup koneksi port CAN.
- *readCANData* untuk mengecek jika *frame* CAN telah tersedia lalu membaca *frame* tersebut
- *writeCANData* untuk mengatur ID CAN, mengisi muatan CAN lalu menuliskan *frame* dari CAN tersebut ke CAN Bus.

3. Modul 3: GUI

Modul ini memiliki tujuh fungsi yaitu:

- *MainWindow* sebagai inialisasi terjadinya fungsi lain.
- *timeOut* untuk mengecek status dari pesan *heartbeat* dari FCC.
- *createButtonArray* untuk menciptakan array dari tombol.
- *createLabelArray* untuk menciptakan label array.
- *createLabelAcess* untuk memberikan kemungkinan kepada pengguna dalam mengatur pengaturan dari tombol dan label.
- *showData* untuk membaca data UART yang masuk.
- *showDataCAN* untuk memanggil ketika *frame* CAN baru telah tersedia, memanggil *frame* dari CAN dan menyimpan data.

Berikut merupakan tampilan perangkat lunak dari FCP:



Gambar 15. Tampilan perangkat lunak FCP

Pada gambar (15) terdapat dua tampilan perangkat lunak yaitu ketika tidak ada data dan ada data. Data-data yang ditampilkan pada FCP versi awal ini adalah terkait *vertical groundspeed*, *heading*, *altitude* dan *vertical airspeed*. Tulisan peringatan berwarna merah akan muncul jika koneksi antara FCP dan FCC gagal.

Masing-masing perangkat keras dan lunak kemudian diintegrasikan sehingga menjadi seperti tampilan berikut ini:



Gambar 16. Integrasi Perangkat Keras dan Lunak FCP

3 Kesimpulan

Berdasarkan pengerjaan yang telah dilakukan, kesimpulannya adalah:

Pembuatan desain awal perangkat keras dan perangkat lunak FCP sudah dilakukan.

Perangkat keras FCP yang dibuat dapat menerima *input* dari pilot berdasarkan komponen yang telah disusun.

Perangkat lunak FCP yang dibuat dapat menampilkan data dari EFCS sesuai dengan modul yang telah disusun.

4 Daftar Pustaka

- [1] S. Bahri, "Spesification and Concept of LSA-02 (Technology Demonstrator)," Berlin, 2017.
- [2] T. I. Hakim, "EFCS Requirements and Concept," Berlin, 2016.
- [3] T. I. Hakim, "BEFCS Definition, Requirement, and concept," Berlin, 2016.
- [4] A. Wirawan, "Development and Test of The Software for The Flight Control Panel of LSA-02 Technology Demonstrator Aircraft," Institut Teknologi Bandung, 2018.
- [5] Shibu K V. "Intro to Embedded Systems," IE. Tata McGraw Hill. New Delhi, 2009.
- [6] Janztec, "emPC-X: Embedded Computing System based on Intel Atom," Janztec, 9 12 2018.[Online].<https://www.janztec.com/en/devices/embedded-computer/empc-x/>. [Diakses 09 12 2018].
- [7] Banzi, "Getting started with Arduino," 2011.
- [8] Y. M. Dinata, "Buku Arduino itu pintar", Elex Media Komputindo, Indonesia, 2016.
- [9] Arduino, "Arduino Mega 2560 Datasheet".
- [10] Pushbutton Switches, "Series 59", [Online] <http://www.itswitches.com/S59.shtml>. [Diakses 09 12 2018]
- [11] Quora, "What is a CAN bus?." Quora, 24 9 2017. [Online]. <https://www.quora.com/What-is-a-CAN-bus>. [Diakses 09 12 2018].
- [12] QT, "QT Whitepaper," QT, 24 3 2016. [Online]. <https://wiki.qt.io/QtWhitepaper>. [Diakses 09 12 2018].