



Sensor Voter untuk Triple Redundant Fault Tolerant Add On Module

Voter Sensor for Triple Redundant Fault Tolerant Add On Module

Faiz Ridha Syahputra¹, Widyawardana Adiprawita²

Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10, Bandung 40132, Indonesia
Email: ¹faiz_ridha@s.itb.ac.id, ²wadiprawita@gmail.com

Abstrak. *Triple Redundant Fault Tolerant Autopilot Add On Module* bekerja dengan prinsip *redundancy* yaitu TMR (*Triple Modular Redundancy*). TMR bekerja dengan memanfaatkan tiga buah modul dasar dan sebuah voter. Sistem ini memiliki *output* yang benar apabila minimal terdapat dua buah modul dasar yang bekerja secara normal. Sensor voter yang diimplementasikan akan digunakan sebagai modul voter sistem. Sensor voter akan menerima empat buah data internal masing-masing modul autopilot yang terdiri atas data *gyroscope*, *magnetometer*, barometer, dan *accelerometer*. Kemudian melakukan analisis terhadap keseluruhan data melalui algoritma pemrograman yang dibuat. Dan pada akhirnya akan menentukan modul autopilot mana yang akan digunakan. Diperoleh dari hasil pengujian bahwa rancangan sensor voter mampu menjalankan fungsinya pada produk *Triple Redundant Fault Tolerant Add On Module* dengan baik.

Kata Kunci: TMR, Sensor, Voter, Autopilot, Redundancy.

Abstract. *Triple Redundant Fault Tolerant Autopilot Add on Module* works with the redundancy principal which is the Triple Modular Redundancy (TMR). TMR works by using three basic modules and a voter. This system has the correct output if there are minimal two basic modules that work normally. The implemented voter sensor will be used as the voter system module. The voter sensor will receive four types of internal data from each of the autopilot module that are consisted of the gyroscope, magnetometer, barometer, and accelerometer data. Later on the overall data is analyzed through the programming algorithm. And in the end it will conclude which autopilot module that will be used. The experiment resulted showed that the sensor voter design was able to function well on the Triple Redundant Fault Tolerant Add on Module.

Keywords: TMR, Sensor, Voter, Autopilot, Redundancy.

1 Pendahuluan

Dewasa ini transportasi menjadi salah satu kebutuhan penting manusia. Hampir semua aspek kehidupan memerlukan sarana transportasi. Salah satu transportasi yang sering digunakan adalah transportasi udara. Transportasi udara menjadi sangat penting karena kemampuannya untuk menghubungkan antar pulau. Oleh karena itu berbagai teknologi terus dikembangkan guna meningkatkan fungsionalitas dari transportasi udara. Salah satu teknologi yang digunakan saat ini adalah autopilot. Autopilot memungkinkan pesawat beroperasi tanpa adanya kendali dari seorang pilot (tanpa campur tangan manusia) dalam menjalankan misi tertentu. Di Indonesia sendiri, teknologi autopilot bukan menjadi hal yang asing lagi. UAV (*Unmanned Aerial Vehicle*) sudah banyak digunakan dalam berbagai aspek kehidupan seperti peliputan video bencana, pencarian korban bencana, pengangkutan barang dan individu, dan lain-lain.

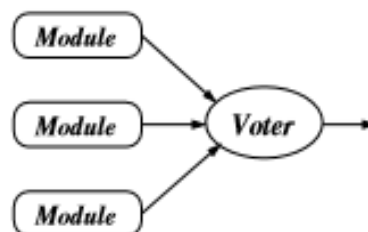
UAV digunakan secara luas dalam berbagai bidang, yang sebagian besar didominasi militer. Salah satu tujuannya adalah untuk memperkuat pertahanan dan keamanan. Hal ini ditandai dengan penggunaan UAV untuk keperluan militer seperti misi intelijen (*intelligence*),

pemantauan (*surveillance*), pengintaian (*reconnaissance*), bahkan pertempuran. Namun dalam penerapannya, seringkali autopilot mengalami kegagalan. Hal ini dikarenakan sistem-sistem yang sudah ada tidaklah tahan terhadap kerusakan (*fault*), padahal dalam menjalankan misi militer sebuah pesawat militer sangat rentan mengalami berbagai kegagalan sistem akibat pertempuran dan semacamnya. Tidak optimalnya kinerja dari autopilot juga akan memberikan dampak terhadap hasil dari misi yang dijalankan. Dalam bidang militer, hal ini tentunya akan berdampak terhadap melemahnya sistem pertahanan dan keamanan negara. Oleh karena itu diperlukan sebuah sistem autopilot yang memiliki kehandalan yang tinggi dan tahan terhadap *fault* sehingga pesawat dapat tetap beroperasi dengan baik meskipun terdapat gangguan-gangguan pada sistem baik internal maupun eksternal.

Oleh karena itu, dalam tugas akhir ini, akan dibuat *Triple Redundant Fault Tolerant Autopilot Add On Module* yang bekerja berdasarkan prinsip *redundancy* TMR (*Triple Modular Redundancy*). TMR bekerja dengan memanfaatkan tiga buah modul dasar dan sebuah *voter*. Sistem ini memiliki *output* yang benar apabila minimal terdapat dua buah modul dasar yang bekerja secara normal. Dalam desain yang dibuat, *Sensor Voter* akan berperan sebagai *voter* pada sistem. *Voter* berfungsi untuk menentukan satu/dua autopilot mana yang akan digunakan. *Sensor voter* dibuat menggunakan Arduino mega. Data yang digunakan dalam proses *voting* adalah data *gyroscope*, *magnetometer*, *barometer*, dan *accelerometer*. *Voter* akan melakukan analisis terhadap seluruh data yang diterima melalui algoritma pemrograman yang dibuat dan dihasilkan *output* autopilot yang digunakan kemudian mengirimkan sinyal digital sebagai sinyal pengaktif.

2 Triple Modular Redundancy, Arduino Mega, Pixhawk Autopilot, dan Komunikasi Serial

2.1 Triple Modular Redundancy



Gambar 1 Triple Modular Redundancy

Triple Modular Redundancy adalah sebuah sistem redundansi yang memanfaatkan tiga buah modul dasar dan sebuah *voter* seperti terlihat pada Gambar 1. Sistem ini memiliki *output* yang benar apabila minimal terdapat dua buah modul dasar yang bekerja secara normal. Apabila $R(t)$ adalah keandalan dari sebuah modul dan $R_{voter}(t)$ keandalan dari *voter*, maka keandalan sistem ini dapat dihitung dengan menggunakan formula berikut:

$$R_{TMR}(t) = R_{voter}(t)(3R^2(t) - 2R^3(t)) \quad (1)$$

Voter di sini berguna untuk menyeleksi kesalahan atau *fault* yang terjadi dalam masing-masing modul. Selanjutnya hasil benar pada *voter* dikeluarkan sebagai *output*. Dalam sistem tersebut, *output* akan dianggap benar apabila paling sedikit terdapat 1 modul yang tidak mengalami *fault*.

2.2 Arduino Mega

Arduino mega merupakan sebuah board mikrokontroller berbasis ATmega1280 yang diprogram menggunakan Bahasa pemrograman Arduino. Arduino mega dilengkapi dengan tiga port komunikasi serial (Rx,Tx), sehingga mampu menerima tiga data secara bersamaan. Selain itu kecepatan proses Arduino tergolong cepat yaitu dalam ukuran *microsecond*. Berikut merupakan spesifikasi dari Arduino Mega:

- *Flash memory* sebesar 128KB dengan 4KB digunakan sebagai *bootloader*, 8KB SRAM dan 4KB EEPROM (dapat dibaca dan ditulis dari EEPROM Library).
- Pin serial: 0 (RX) dan 1 (TX), Serial 1: 19 (RX) dan 18 (TX), Serial 2: 17 (RX) dan 16 (TX), Serial 3: 15 (RX) dan 14 (TX). Digunakan untuk menerima (RX) dan mentransmisi (TX) TTL serial data. Pins 0 dan 1 juga dapat disambungkan ke pin FTDI USB to TTL serial chip.
- *External Interrupts*: 2 (*interrupt* 0), 3 (*interrupt* 1), 18 (*interrupt* 5), 19 (*interrupt* 4), 20 (*interrupt* 3), and 21 (*interrupt* 2).
- PWM yaitu pin 2 sampai 13 dan 44 sampai 46, 8-bit PWM *output*.
- SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).
- LED: 13 serta I2C: 20 (SDA) dan 21 (SCL).
- 16 analog *inputs*, 10 *bits of resolution*

2.3 Pixhawk Autopilot

Pixhawk merupakan modul autopilot utama yang akan digunakan dalam sistem *Triple Redundant Fault Tolerant Autopilot Add On Module*. Pixhawk menggunakan 32 bit STM32F427 Cortex M4 core with FPU dengan 32 bit STM32F103 *failsafe co-processor*. Pixhawk juga dilengkapi dengan 256KB RAM, 2MB *Flash* Memori, dan 5 UART (*serial ports*). Pixhawk memiliki empat buah sensor yaitu:

2.3.1 ST Micro L3GD20H 16 bit gyroscope

Sensor ini merupakan *gyroscope* yang mampu mengukur tiga sumbu rotasional. Selain itu, L3GD20H juga dilengkapi dengan *interface* IC, sehingga memungkinkan pengiriman data melalui serial komunikasi I2C/SPI. Sensor ini memiliki ketelitian yang dapat ditentukan sesuai dengan keinginan pengguna yaitu $\pm 245/\pm 500/\pm 2000$ dps (*degree per second*) dan memiliki resolusi data 16 bit. Selain itu L3GD20H memiliki *buffer* terintegrasi 32-level *first in first out* (FIFO) yang memungkinkan pengguna untuk menyimpan data guna membatasi intervensi oleh prosesor *host* dan mampu beroperasi dalam rentang suhu -40° sampai 85° C.

2.3.2 ST Micro LSM303D 14 bit accelerometer / magnetometer

Selain berfungsi sebagai *magnetometer*, sensor ini juga berfungsi sebagai *accelerometer*. Sensor ini merupakan *magnetometer* yang mampu mengukur tiga sumbu axis (x,y,z). LSM303D mencakup antarmuka bus serial I2C yang mendukung mode standar dan cepat (100 kHz dan 400 kHz) dan antarmuka standar serial SPI. Sensor ini memiliki ketelitian yang dapat ditentukan sesuai dengan keinginan pengguna yaitu $\pm 2 / \pm 4 / \pm 8 / \pm 12$ gauss sebagai *magnetometer* dan $\pm 2g / \pm 4g / \pm 6g / \pm 8g / \pm 16g$ (g=gravitasi) sebagai *accelerometer* dan memiliki resolusi data 16 bit. LSM303D mampu beroperasi dalam rentang suhu -40° sampai 85° C.

2.3.3 Invensense MU 6000 3-axis *accelerometer/gyroscope*

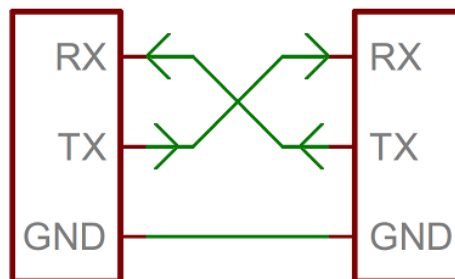
Selain berfungsi sebagai *accelerometer*, sensor ini juga berfungsi sebagai *gyroscope*. Sensor ini merupakan *accelerometer* dengan tiga sumbu axis (x,y,z) dan bekerja dengan menghitung percepatan *linear* pada masing-masing sumbu. Komunikasi MPU 6000 dapat dilakukan melalui antarmuka I2C (400kHz) atau serial SPI (1MHz). Sensor ini memiliki ketelitian yang dapat ditentukan sesuai dengan keinginan pengguna yaitu $\pm 2g$ / $\pm 4g$ / $\pm 6g$ / $\pm 8g$ / $\pm 16g$ (g=gravitasi) sebagai *accelerometer* dan ± 250 , ± 500 , ± 1000 , and ± 2000 dps (degree per second) sebagai *gyroscope* dan memiliki resolusi data 16 bit.

2.3.4 MEAS MS5611 barometer

Sensor ini akan mengukur besarnya tekanan dan mengkonversinya menjadi masing-masing 24 bit data *altitude pressure* (digital) dan temperatur. Sensor ini memiliki resolusi *temperature* 0,010C, resolusi tekanan 0,012 mbar, serta resolusi *altitude* 10 cm dan mampu beroperasi dalam rentang suhu -40° sampai 85° C.

2.4 Komunikasi Serial

Proses pengiriman dan penerimaan data pada sistem dilakukan melalui serial komunikasi. Dalam penerapannya, hal yang harus diperhatikan adalah baudrate. Baudrate menunjukkan seberapa cepat pengiriman data melalui jalur komunikasi serial. Baudrate memiliki satuan bps (*bit per second*). Terdapat beberapa jenis baudrate yang biasa digunakan, antara lain 1200, 2400, 4800, 9600, 19200, 38400, 57600, dan 115200. Proses komunikasi serial dilakukan dengan menghubungkan pin Rx (*receiver*) -> Tx (*transmitter*), Tx (*receiver*) -> Rx (*transmitter*), dan Gnd (*receiver*) -> Gnd (*transmitter*) sesuai dengan Gambar 2 di bawah ini:



Gambar 2 Komunikasi Serial

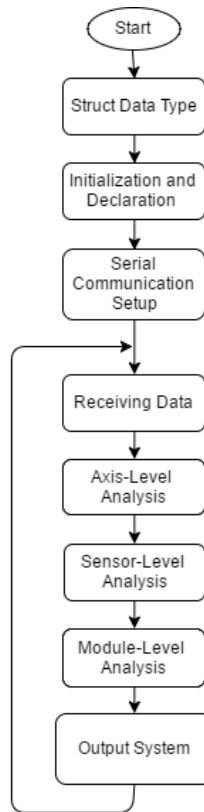
3 Implementasi Sensor Voter

Implementasi yang dibuat pada bagian ini dibagi atas beberapa tahapan proses yang dilakukan oleh sensor voter, sesuai dengan fungsi utama sensor voter yaitu melaksanakan proses *voting* pada data internal sistem.

3.1 Algoritma Umum Sensor Voter

Pada dasarnya, proses sensor voter dapat dibagi menjadi beberapa tahapan yaitu pemrosesan umum, penerimaan data, *axis-level* analisis, *sensor-level* analisis, dan *modul-level* analisis dan pada bagian *output system* dilakukan pengiriman sinyal *output* berupa sinyal digital sesuai dengan autopilot yang digunakan. Bagian pemrosesan umum terdiri atas *struct data type* yaitu membuat tipe data yang sesuai bagi masing-masing kategori sensor, kemudian proses inisialisasi dan deklarasi variabel-variabel yang akan digunakan untuk menyimpan data sesuai tipe data

yang dibutuhkan, dan terakhir melakukan *setup* komunikasi serial berupa penentuan baudrate sebesar 57600 bps (baudrate yang digunakan pixhawk). *Flowchart* algoritma umum dari sensor voter dapat dilihat pada Gambar 3 di bawah ini:



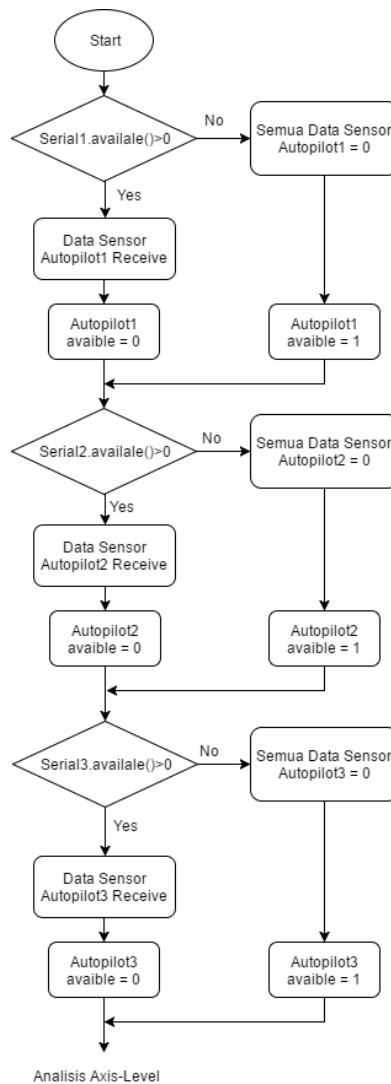
Gambar 3 *Flowchart* Algoritma Umum Sensor Voter

3.2 Algoritma Penerimaan Data

Pada bagian ini dilakukan pengecekan apakah ada data yang diterima atau tidak. Data yang dikirimkan oleh autopilot bertipe *string* dengan format data:

“gyro-x,gyro-y;gyro-z/(spasi)baro-alt,baro-temp/_mag-x,mag-y;/*acc-x,acc-y;acc-z/”

Tanda (spasi), (_), dan (*) berfungsi sebagai tanda pemisah data sensor yang berbeda. Sementara untuk membedakan sumbu axis digunakan karakter (,), (;), dan (/). Apabila ada data yang dikirim, maka data akan langsung disimpan ke dalam variabel yang telah dideklarasikan. Sementara jika tidak ada data yang diterima, maka data akan dianggap bernilai 0. *Flowchart* algoritma penerimaan data dapat dilihat pada Gambar 4 berikut ini:



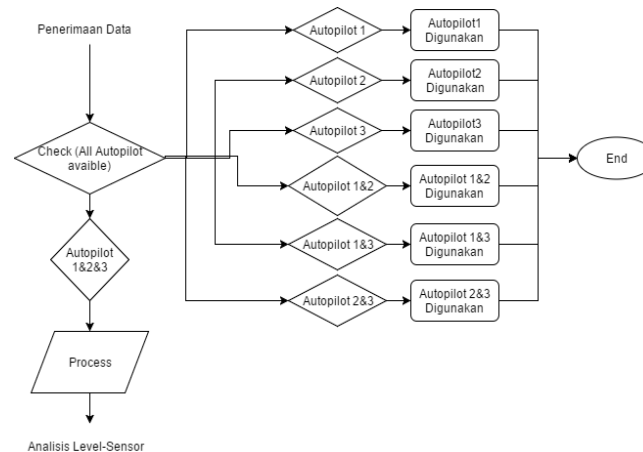
Gambar 4 Flowchart Algoritma Penerimaan Data

Kemudian terdapat variabel *available*, variabel ini akan bernilai “1” jika tidak ada data yang diterima dan bernilai “0” jika ada data yang diterima. Variabel ini akan digunakan dalam analisis axis-level.

3.3 Algoritma Analisis Axis-Level

Pada analisis axis-level, pertama-tama dilakukan pengecekan terhadap variabel *available* dari masing-masing autopilot. Apabila jika hanya satu autopilot maka autopilot tersebut yang digunakan dan program selesai. Apabila terdapat dua autopilot yang *available*, maka kedua autopilot tersebut yang digunakan dan program selesai. Sementara jika ketiga autopilot *available*, maka akan masuk ke dalam process. Pemrosesan dilakukan dengan mengurutkan data dari nilai terbesar hingga terkecil. Kemudian membandingkan masing-masing data dengan data terdekatnya. Digunakan *threshold* yang berasal dari data pengamatan sensor. Apabila selisih lebih kecil dari nilai *threshold* yang ditetapkan maka data dianggap akurat. Apabila semua data dianggap akurat maka dihasilkan *output string* “123” yang berarti diabaikan pada tahap analisis

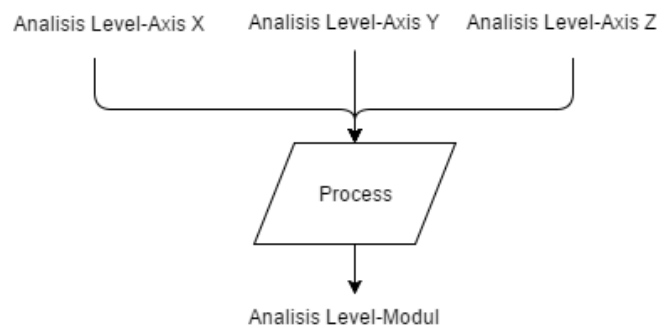
selanjutnya, jika tidak maka akan dihasilkan *output* sesuai dengan hasil pengolahan. Hal yang sama juga dilakukan pada sensor yang berbeda dan jenis axis (x,y,z) yang berbeda. Untuk *accelerometer* dan *gyroscope* digunakan 3 axis, untuk *magnetometer* digunakan 2 axis (x&y), dan untuk barometer hanya digunakan parameter *altitude pressure* (alt). *Flowchart* algoritma analisis axis-level dapat dilihat seperti pada Gambar 5 di bawah ini:



Gambar 5 Flowchart Algoritma Analisis Axis-Level

3.4 Algoritma Analisis Sensor-Level

Flowchart algoritma analisis level-sensor dapat dilihat pada Gambar 6 di bawah ini:

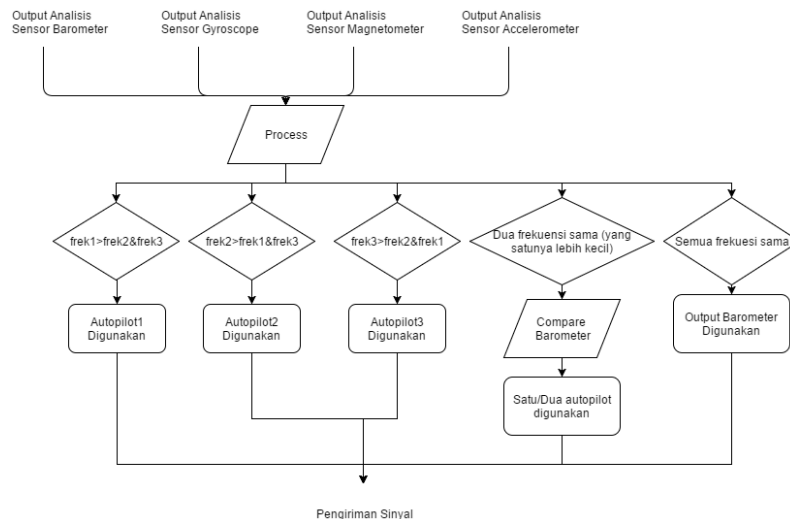


Gambar 6 Flowchart Algoritma Analisis Sensor-Level

Pada tahap ini, digunakan prinsip kesamaan *output*. *Output* dari masing-masing axis akan dibandingkan dan dilakukan mayoritas *voting*, artinya jika terdapat dua *output axis* yang sama, maka *output* tersebut yang digunakan pada tahap analisis level-modul. Namun, jika tidak ada yang sama maka *output* akan dihasilkan string “123” yang artinya diabaikan pada analisis level-modul. Untuk barometer, karena hanya digunakan satu parameter maka *output* analisis level-axis langsung diteruskan ke dalam analisis level-modul. Sementara untuk *magnetometer*, karena digunakan dua axis maka diambil autopilot irisan diantara kedua *output*.

3.5 Algoritma Analisis Module-Level

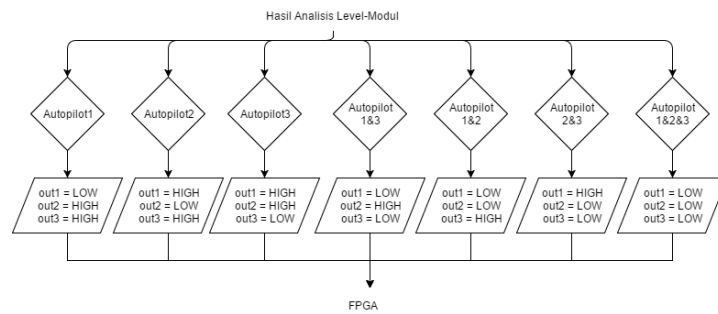
Tahapan ini merupakan tahapan analisis terakhir. Pada tahap ini *output* dari masing-masing sensor *gyroscope*, *barometer*, *magnetometer*, dan *accelerometer* sudah diketahui. Oleh karena itu analisis pada *level* modul dilakukan dengan cara menghitung frekuensi kemunculan dari masing-masing *output* sensor. *Output* dari masing-masing sensor berupa *string*. *String* “12” menandakan bahwa autopilot 1&2 yang akurat, “1” menandakan hanya autopilot 1 yang akurat, sementara “123” menandakan bahwa semua autopilot akurat. Sehingga dengan menghitung frekuensi kemunculannya maka dapat diperoleh *output* yang paling baik. Namun jika terjadi kasus dimana frekuensi kemunculan dua autopilot sama (dengan syarat frekuensi autopilot lainnya lebih kecil), maka akan dilakukan pengecekan pada *output* barometer. Hal ini dikarenakan barometer merupakan sensor paling vital pada modul autopilot. Apabila terdapat irisan (*output* barometer selalu dua autopilot), maka autopilot tersebutlah yang diambil. Jika *output* barometer berisi kedua autopilot dengan frekuensi kemunculan terbesar, maka *output* barometer akan menjadi *output* sistem. *Flowchart* analisis level-modul dapat dilihat pada Gambar 7 di bawah ini:



Gambar 7 Flowchart Algoritma Analisis Module-Level

3.6 Algoritma Pengiriman Sinyal Output

Setelah diketahui autopilot mana yang digunakan, maka dilakukan pengiriman sinyal digital ke FPGA yang berfungsi meneruskan sinyal PWM autopilot yang digunakan ke pesawat. *Flowchart* algoritma pengiriman sinyal dapat dilihat seperti pada Gambar 8 berikut ini:

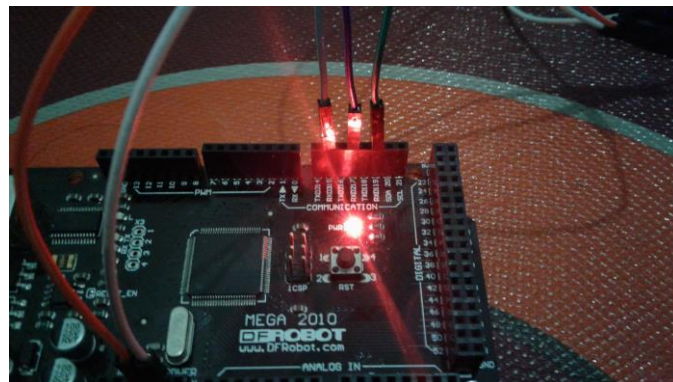


Gambar 8 Flowchart Algoritma Pengiriman Sinyal Digital

4 Pengujian Sensor Voter

4.1 Pengujian Penerimaan Data

Pengujian ini dilakukan untuk memeriksa apakah Sensor Voter berhasil menerima data dan melakukan analisis status available dari masing-masing autopilot. Pengujian ini dilakukan dengan tiga prosedur berbeda. Prosedur pertama dilakukan saat semua autopilot. Pengujian ini dilakukan dengan tiga prosedur berbeda. Prosedur pertama dilakukan saat semua autopilot menyala dan mengirimkan data. Berikut pada Gambar 9 dan Gambar 10 adalah hasil dari pengujian prosedur pertama:

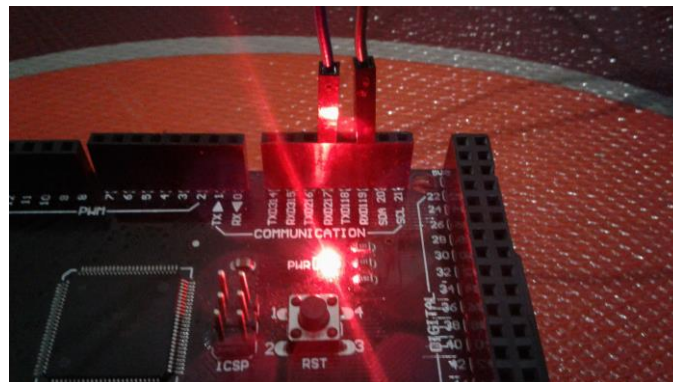


Gambar 9 Semua Autopilot Aktif dan Mengirimkan Data

Data sensor gyrometer 1 :	Data sensor gyrometer 2 :	Data sensor gyrometer 3 :
X :	X :	X :
0.0028	0.0014	-0.0019
Y :	Y :	Y :
0.0024	0.0012	-0.0010
Z :	Z :	Z :
-0.0055	-0.0031	0.0002
Data sensor barometer 1 :	Data sensor barometer 2 :	Data sensor barometer 3 :
ALT :	ALT :	ALT :
799.4868	804.7050	800.1165
TEMP :	TEMP :	TEMP :
42.6200	44.5100	44.7100
Data sensor magnetometer 1 :	Data sensor magnetometer 2 :	Data sensor magnetometer 3 :
X :	X :	X :
-0.3885	-0.3496	-0.1166
Y :	Y :	Y :
0.1698	0.3567	0.4040
Data sensor accelerometer 1 :	Data sensor accelerometer 2 :	Data sensor accelerometer 3 :
X :	X :	X :
-0.0448	0.0068	-0.0559
Y :	Y :	Y :
-0.1647	0.1877	-0.0926
Z :	Z :	Z :
-9.6484	-9.7026	-9.7303

Gambar 10 Hasil Penerimaan Data Saat Semua Autopilot Aktif dan Mengirim Data

Dari hasil yang diperoleh dapat terlihat bahwa semua data autopilot telah berhasil diterima dan tersimpan dalam variabel yang sesuai. Selanjutnya dilakukan prosedur kedua yaitu saat semua autopilot aktif tetapi salah satu kabel komunikasi serial dicabut (Rx3). Berikut pada Gambar 11 dan Gambar 12 hasil pengujian prosedur kedua:



Gambar 11 Semua Autopilot Aktif tetapi Salah Satu Tidak Mengirim Data

```

Data sensor gyrometer 1 :      Data sensor gyrometer 2 :      Data sensor gyrometer 3 :
X :                          X :                          X :
0.0028                        0.0016                        0
Y :                          Y :                          Y :
0.0024                        0.0002                        0
Z :                          Z :                          Z :
-0.0055                       -0.0022                       0
Data sensor barometer 1 :      Data sensor barometer 2 :      Data sensor barometer 3 :
ALT :                          ALT :                          ALT :
799.4868                       805.1549                       0
TEMP :                          TEMP :                          TEMP :
42.6200                         45.2700                         0
Data sensor magnetometer 1 :    Data sensor magnetometer 2 :    Data sensor magnetometer 3 :
X :                          X :                          X :
-0.3885                       -0.3373                       0
Y :                          Y :                          Y :
0.1698                         0.3582                         0
Data sensor accelerometer 1 :    Data sensor accelerometer 2 :    Data sensor accelerometer 3 :
X :                          X :                          X :
-0.0448                       0.0239                         0
Y :                          Y :                          Y :
-0.1647                       0.1872                         0
Z :                          Z :                          Z :
-9.6484                       -9.7165                         0
Data sensor gyrometer 2 :      Data sensor gyrometer 3 :      Data sensor gyrometer 1 :
X :                          X :                          X :
0.0016                         0                          0.0022
Y :                          Y :                          Y :
0                          0                          0.0002

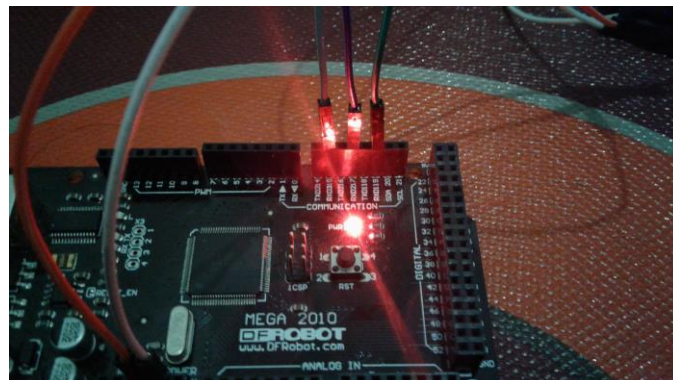
```

Gambar 12 Hasil Penerimaan Data Saat Semua Autopilot Aktif tetapi Salah Satu Tidak Mengirim Data

Hasil yang diperoleh sesuai dengan yang diharapkan. Dapat dilihat pada Gambar 12, semua data sensor autopilot3 bernilai 0. Berdasarkan dua hasil pengujian di atas dapat dikatakan bahwa penerimaan data berhasil dengan baik.

4.2 Pengujian Ketepatan Hasil Analisis

Untuk memverifikasi ketepatan hasil analisis dari sistem yang dibuat, maka akan dilakukan dua prosedur pengujian. Prosedur pertama adalah saat semua autopilot aktif dan mengirim data. Berikut pada Gambar 13 dan Gambar 14 adalah hasil pengujian pertama:



Gambar 13 Semua Autopilot Aktif dan Mengirimkan Data

Gyro X yang benar : 123	Gyro X yang benar : 123
Gyro Y yang benar : 123	Gyro Y yang benar : 123
Gyro Z yang benar : 123	Gyro Z yang benar : 123
Gyro yang benar : 123	Gyro yang benar : 123
Baro ALT yang benar : 13	Baro ALT yang benar : 13
Baro yang benar : 13	Baro yang benar : 13
Mag X yang benar : 12	Mag X yang benar : 12
Mag Y yang benar : 123	Mag Y yang benar : 123
Mag yang benar : 12	Mag yang benar : 12
Acc X yang benar : 123	Acc X yang benar : 123
Acc Y yang benar : 123	Acc Y yang benar : 123
Acc Z yang benar : 23	Acc Z yang benar : 123
Acc yang benar : 23	Acc yang benar : 123
Autopilot yang digunakan : 13	Autopilot yang digunakan : 1

Gambar 14 Output yang Dihasilkan Saat Semua Autopilot Aktif dan Mengirim Data

Dapat kita lihat pada Gambar 14, pada gambar sebelah kiri dihasilkan *output gyroscope* = “123”, barometer = “13”, *magnetometer* = “12”, dan *accelerometer* = “23”. Ketiga autopilot memiliki frekuensi kemunculan yang sama sebanyak tiga kali. Oleh karena itu kemudian sistem meninjau *output* barometer, sehingga dihasilkan autopilot 1 dan 3 yang digunakan. Pada gambar sebelah kanan terjadi perubahan pada *output accelerometer* menjadi “123”. Hal ini dikarenakan data yang dikirimkan berubah-ubah secara terus-menerus sehingga terjadi perubahan *output*. Hasil yang diperoleh yaitu autopilot 1 yang digunakan. Kemudian dilakukan pengujian kedua dimana salah satu autopilot yang benar (autopilot 1) dimatikan saat proses analisis. Hasil yang diperoleh seperti Gambar 15 di bawah ini:



Gambar 15 Autopilot yang Benar Dimatikan Saat Proses Analisis

Hasil yang diperoleh dapat dilihat pada gambar 16. Hasil yang diperoleh sudah sesuai dengan yang diharapkan. Hal ini dikarenakan jika hanya terdapat dua autopilot yang available, maka kedua autopilot itu digunakan (*mayoritas voting*).

```

Gyro X yang benar : 23
Gyro Y yang benar : 23
Gyro Z yang benar : 23
Gyro yang benar : 23
Baro ALT yang benar : 23
Baro yang benar : 23
Mag X yang benar : 23
Mag Y yang benar : 23
Mag yang benar : 23
Acc X yang benar : 23
Acc Y yang benar : 23
Acc Z yang benar : 23
Acc yang benar : 23
Autopilot yang digunakan : 23
Gyro X yang benar : 23
Gyro Y yang benar : 23
Gyro Z yang benar : 23
Gyro yang benar : 23
Baro ALT yang benar : 23
Baro yang benar : 23
Mag X yang benar : 23
Mag Y yang benar : 23
Mag yang benar : 23
Acc X yang benar : 23
Acc Y yang benar : 23
Acc Z yang benar : 23
Acc yang benar : 23
Autopilot yang digunakan : 23

```

Gambar 16 Hasil Analisis saat Autopilot yang Benar Dimatikan

5 Simpulan

Dari pembahasan diatas, dapat disimpulkan bahwa desain dari *Sensor Voter* telah berhasil diimplementasikan. Kedua fungsi utama dari *Sensor Voter* yaitu penerimaan data dan analisis data telah diimplementasikan dan diuji. Teramati bahwa *Sensor Voter* telah mampu untuk menerima data dengan baik. Selain itu algoritma analisis yang dibentuk juga sudah berhasil menentukan satu/dua autopilot yang akan digunakan (sesuai dengan karakteristik TMR).

Referensi

- [1] E. F. Hitt, D. Mulcare, *Fault-Tolerant Avionics*, CRC Press LLC, 2001
- [2] L. Meier, (2017, April 1), *Pixhawk Flight Controller Hardware Project*, ETH Computer Vision and Geometry Lab., <https://pixhawk.org/>
- [3] E. F. Hitt, *Digital Avionics Handbook, Second Edition*, Boca Raton: CRC Press, 2000
- [4] Jimbo, (2017, April 1), Serial Communication, <https://learn.sparkfun.com/tutorials/serial-communication>, Sparkfun LLC
- [5] I. Koren et.al., *Fault Tolerant Systems*, Oxford: Elsevier, 2007
- [6] (2017, April 1), Arduino Mega 2560, <https://www.arduino.c/en/Main/arduinoBoardMega>,
- [7] (2017, April 1), L3GD20 Low power 3-axis gyroscope, <http://www.st.com/en/mems-and-sensors/l3gd20.html>, ST Micro
- [8] (2017, April 1), Precision Micro Barometer Module MS5611, <http://www.amsys.info/products/ms5611.htm>, Amsys

- [9] (2017, April 1), *LSM303D e-Compass: 3D accelerometer, 3D magnetometer*, <http://www.st.com/en/mems-and-sensors/lsm303d.html>, ST Micro
- [10] (2017, April 1), *MPU-6000 and MPU-6050 Product Specification Revision 3.4*, <https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>, Invensens