IMPLEMENTASI SENSOR *ACCELEROMETER, GYROSCOPE DAN MAGNETOMETER* BERBASIS MIKROKONTROLER UNTUK MENAMPILKAN POSISI BENDA MENGGUNAKAN *INERTIAL NAVIGATION SYSTEM (INS)*

Yuga Aditya Pramana
Program Studi Teknik Elektro, Universitas Komputer Indonesia
Yuga.AdityaPramana@gmail.com

ABSTRAKSI

Sensor adalah jenis tranduser yang digunakan untuk mengubah besaran mekanis, magnetis, panas, sinar, dan kimia menjadi tegangan dan arus listrik. Perkembangan teknologi MEMS (*Micro Electro Mechanical System*) sensor dapat digunakan dalam *Inertial Navigation System* (INS). INS adalah sebuah sistem navigasi yang berbasis komputer dan beberapa keluaran dari sensor *Inertial Measurement Unit* (IMU). Sistem navigasi ini digunakan dalam *Attitude and Heading Reference System* (AHRS) untuk mengetahui informasi dan visualisasi seperti kecepatan, ketinggian, arah dan sudut, contohnya dalam teknologi avionik dan kendaraan tanpa awak. Informasi dan visualisasi tersebut sangatlah penting karena ada kondisi-kondisi tertentu saat indera perasa manusia mengalami salah persepsi dalam penerbangan sehingga sangat tidak mungkin untuk mengandalkan indera perasa manusia sebagai alat bantu navigasi penerbangan. Untuk dapat membantu hal tersebut, terdapat beberapa sensor yang dapat digunakan dalam sistem navigasi diantaranya yaitu sensor *accelerometer*, *gyroscope* dan *magnetometer*. Sensor tersebut dapat mengukur percepatan, kecepatan sudut, dan kekuatan atau arah medan magnet. Pada penelitian ini sensor IMU yang digunakan adalah Razor-IMU 9DoF, dimana sensor tesebut memiliki 9 derajat kebebasan. Data keluaran dari sensor tersebut akan menghasilkan sudut *roll*, *pitch* dan *yaw* berdasarkan dari komputasi algortima *Direct Cosine Matrix*.

ABSTRACT

Sensor is a type of transducer that is used to change the amount of mechanical, magnetic, heat, light, and chemicals into the voltage and amperage. The development of MEMS technology (Micro Electro Mechanical System) sensors can be used in Inertial Navigation System (INS). INS is a computer-based navigation systems and the output of the Inertial Measurement Unit (IMU) sensor. The navigation system used in the Attitude and Heading Reference System (AHRS) to find out information and visualizations such as speed, altitude, direction and angle, for example in avionics technology and unmanned vehicles. Information and visualization is very important because there are certain conditions when humans experience any sense of taste perception in flight so it is not possible to rely on the human sense of taste as a navigation aid flight. To be able to help it, there are several sensors that can be used in navigation systems among which sensor accelerometer, gyroscope and magnetometer. The sensors can measure acceleration, angular velocity, and the strength or direction of the magnetic field. In this study IMU sensor used a Razor-IMU 9DoF, where the proficiency level sensor has 9 degrees of freedom. Data output from the sensor will produce angles roll, pitch and yaw based on the Direct Cosine Matrix computation algorithms.

Kata kunci: *Inertial Navigation System* (INS), *Inertial Measurement Unit* (IMU), *Direct Cosine Matrix*, *Razor-IMU 9DoF*

1. PENDAHULUAN

1.1 Latar Belakang

Sensor adalah jenis tranduser yang digunakan untuk mengubah besaran mekanis, magnetis, panas, sinar, dan kimia menjadi tegangan dan arus listrik. Sensor sering digunakan untuk pendeteksian pada saat melakukan pengukuran atau pengendalian. Perkembangan teknologi MEMS (*Micro Electro Mechanical System*) menyebabkan sensor mempunyai ukuran kecil dan mempunyai kehandalan tinggi sehingga dapat dimanfaatkan di berbagai aplikasi, salah satunya dalam *Inertial Navigation System* (INS).

INS adalah sebuah sistem navigasi yang berbasis komputer dan beberapa keluaran dari sensor *Inertial Measurement Unit* (IMU). Sistem navigasi ini digunakan dalam *Attitude and Heading Reference System* (AHRS) untuk mengetahui informasi dan visualisasi yang akurat seperti kecepatan, ketinggian, arah dan sudut, contohnya dalam teknologi avionik dan kendaraan tanpa awak.

Informasi dan visualisasi tersebut sangatlah penting karena ada kondisi-kondisi tertentu saat indera perasa manusia mengalami salah persepsi dalam penerbangan sehingga sangat tidak mungkin untuk mengandalkan indera perasa manusia sebagai alat bantu navigasi penerbangan. Sedangkan pada konteks kendaraan tanpa awak, peranan AHRS dibutuhkan untuk memberikan informasi dan visualisasi perilaku obyek kepada pemantau yang berada pada jarak yang tidak memungkinkan untuk melihat obyek secara langsung. Oleh karena itu diperlukan sensor IMU yang dapat membantu dalam sistem navigasi untuk memberikan informasi dan dapat mengirimkan data ke tempat pengamatan secara kontinyu.

Untuk dapat melakukan hal tersebut, terdapat beberapa sensor yang dapat digunakan dalam sistem navigasi. Sensor yang dapat digunakan dalam sistem navigasi diantaranya sensor *accelerometer*, *gyroscope* dan *magnetometer*. Sensor tersebut dapat mengukur percepatan, kecepatan sudut, dan kekuatan atau arah medan magnet. Implementasi dari sensor ini dapat diterapkan dalam menentukan posisi dan arah suatu obyek yang bergerak. Sehingga berdasarkan data *attitude* yang diterima oleh operator, dapat dibentuk informasi dan visualisasi dalam aplikasi yang dibuat di tempat pengamatan dari obyek yang diamati.

1.2 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut.

1. Dapat mengimplementasikan sensor IMU untuk menampilkan posisi dan arah suatu benda dengan metode *Inertial Navigation System*.
2. Dapat mengimplementasikan algoritma *Direct Cosine Matrix*, sehingga dapat mengetahui percepatan, kecepatan sudut, dan arah pada suatu obyek yang bergerak.
3. Dapat ditampilkan dalam bentuk visualisasi grafis di aplikasi *ground station*.

1.3 Batasan Masalah

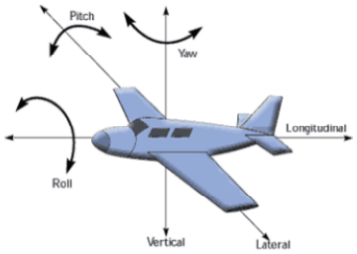
Dalam penelitian ini diperlukan suatu batasan masalah agar tidak terlalu luas pembahasannya. Adapun batasan masalahnya adalah sebagai berikut.

1. Modul IMU yang digunakan adalah Razor-IMU 9DoF.
2. Algoritma IMU *sensor fusion* yang digunakan adalah algoritma orientasi dengan representasi *Direct Cosine Matrix*.
3. Aplikasi dikembangkan menggunakan bahasa pemograman *C/C++* yang dikompilasi dengan perangkat lunak *Processing* sebagai pengolahan data dan tampilan yang dilakukan di personal komputer.
4. Transmisi data dikirim melalui *port USB*.

2. DASAR TEORI

2.1 *Attitude and Heading Reference System* (AHRS)

AHRS digunakan pada aplikasi penerbangan pada umumnya untuk sistem kendali perilaku dan arah pesawat. Informasi yang diberikan oleh sistem AHRS pada kebanyakan sistem penerbangan pada umumnya diperoleh dari sistem INS yang terpasang pada pesawat. Sistem INS ini sendiri terdiri dari sensor-sensor inersia yang memanfaatkan perubahan inersia untuk memberikan informasi tentang keadaan dan perilaku pesawat.



Gambar 2. 1 *Roll*, *Pitch* dan *Yaw*

2.2 *Inertial Navigation System* (INS)

INS atau ada juga yang menyebutnya dengan INU *(Inertial Navigation Unit)* adalah sebuah sistem navigasi yang berbasis computer dan beberapa keluaran sensor dari IMU *(Inertial Measurement Unit)* yang secara kontinyu mengetahui posisi, kecepatan dan *attitude* dari suatu benda.

2.3 *Inertial Measurement Unit* (IMU)

IMU adalah instrumentasi elektronik yang digunakan untuk mengukur kecepatan, orientasi, dan gaya gravitasi dengan menggunakan *accelerometer* dan *gyroscope*. Namun seiring dengan perkembangan zaman sensor IMU didukung dengan sensor yang dapat mengukur kuat medan magnet, tekanan, dan beberapa fungsi lainnya.

IMU biasa digunakan untuk keperluan pesawat terbang, pesawat penjelajah angkasa, pesawat tanpa awak dan satelit. IMU adalah komponen penting dalam INS dan digunakan untuk mendeteksi lintasan dengan *dead reckoning*. Pengembangan dan penelitian sensor IMU sampai saat ini masih terus dilakukan dengan tujuan untuk mendapatkan sensor IMU yang ideal, akurat, bentuk minimalis, pembacaan cepat, tahan derau dan hemat dalam penggunaan energi. Metode ini merupakan metode yang digunakan dalam proses integrasi data percepatan hingga menjadi posisi.

2.4 Rotasi Matrik dan Sudut *Euler*

Sudut *Euler* adalah 3 sudut yang membentuk rotasi sebuah objek dalam ruang (R^3). Sehingga dibutuhkan 3 parameter untuk merepresentasikan orientasi sebuah objek pada sistem 3 dimensi sudut-sudut itu antara lain:

- Rotasi melalui sumbu x, disebut dengan *roll* (φ).
- Rotasi melalui sumbu y, disebut dengan *pitch* (θ).
- Rotasi melalui sumbu z, disebut dengan *yaw* (ψ).

2.5 *Direct Cosine Matrix*

Matrix C_n^b atau sering juga disebut DCM *(Direction Cosine Matrix)* adalah *matrix* 3x3 yang merpresentasikan rotasi sekuensial dari *roll*, *pitch* dan *yaw*. Representasi ini tidak mengalami masalah singularitas, tetapi tidak intuitif dan menggunakan sembilan nilai untuk menggambarkan *attitude* (Adiprawita dkk.,2007).

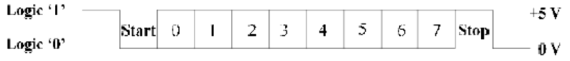
DCM ditulis dalam bentuk *matrix* rotasi yang mendeskripsikan orientasi dari kerangka kordinat *b (body)* terhadap kerangka navigasi *n*. Orde rotasi adalah *zz*, *yy* dan kemudian *xx* yang dalam sudut *Euler* adalah *yaw (ψ), pitch (θ), roll (ϕ)*. *Matrix* rotasi dapat dinyatakan sebagai (Adiprawita dkk., 2007, Liu dkk., 2008 dan Kang dkk., 2009).

$$C_n^b = \begin{bmatrix} c\theta c\psi - c\phi s\psi s\theta c\psi + s\phi s\psi & s\phi s\theta c\psi + c\phi s\psi & -s\theta c\psi \\ c\theta s\psi + c\phi c\psi s\theta s\psi - s\phi c\psi & s\phi s\theta s\psi + c\phi c\psi & -s\theta s\psi \\ s\phi c\theta & c\phi s\theta & c\theta c\psi \end{bmatrix}$$

2.6 Komunikasi Serial

Komunikasi data serial sangat berbeda dengan format pemindahan data pararel. Disini, pengiriman bit-bit tidak dilakukan sekaligus melalui saluran pararel, tetapi setiap bit dikirimkan satu persatu melalui saluran tunggal.

Dalam pengiriman data secara serial harus ada sinkronisasi atau penyesuaian antara pengirim dan penerima agar data yang dikirimkan dapat diterima dengan tepat dan benar oleh penerima. Salah satu mode transmisi dalam komunikasi serial adalah mode *asynchronous*.

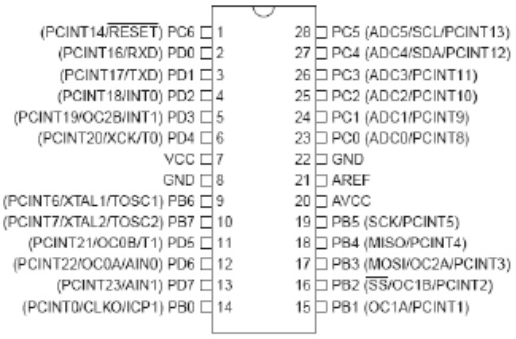


Gambar 2. 6 Format Pengiriman Data Serial

2.7 Mikrokontroler ATmega328

ATmega328 adalah mikrokontroller keluaran dari atmel yang mempunyai arsitektur RISC (*Reduce Instruction Set Computer*) yang dimana setiap proses eksekusi data lebih cepat dari pada arsitektur CISC (*Completed Instruction Set Computer*). Mikrokontroller ini memiliki beberapa fitur antara lain :

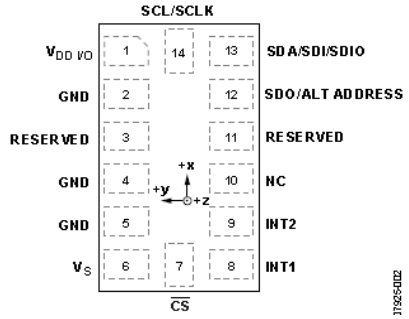
- 130 macam instruksi yang hampir semuanya dieksekusi dalam satu siklus *clock*.
- 32 x 8-bit register serba guna.
- Kecepatan mencapai 16 MIPS dengan clock 16 MHz.
- 32 KB *Flash memory* dan pada arduino memiliki *bootloader* yang menggunakan 2 KB dari flash memori sebagai *bootloader*.
- Memiliki *EEPROM* (*Electrically Erasable Programmable Read Only Memory*) sebesar 1KB sebagai tempat penyimpanan data semi permanent karena *EEPROM* tetap dapat menyimpan data meskipun catu daya dimatikan.
- Memiliki *SRAM* (*Static Random Access Memory*) sebesar 2KB.



Gambar 2. 7 Konfigurasi Pin ATmega328

2.8 Sensor *Accelerometer* (percepatan)

Percepatan merupakan suatu keadaan dimana terjadi suatu perubahan bertambahnya kecepatan terhadap waktu. Sedangkan apabila terjadi perubahan berkurangnya kecepatan terhadap waktu pada keadaan tertentu disebut perlambatan. Sedangkan sensor percepatan sendiri merupakan suatu piranti elektronik yang digunakan untuk mengukur percepatan yang terjadi pada keadaan tertentu. Sensor percepatan dapat digunakan untuk mendapatkan posisi dari suatu benda dengan melakukan integral percepatan itu sendiri sebanyak dua kali terhadap waktu. Salah satu contoh dari sensor percepatan yang dapat digunakan adalah ADXL345 *(triple-axis accelerometer)*, untuk konfigurasi pin dan blok diagram sensor ini dapat dilihat di Gambar 2. 9. Beberapa fitur dari sensor ini diantaranya adalah sebagai berikut.

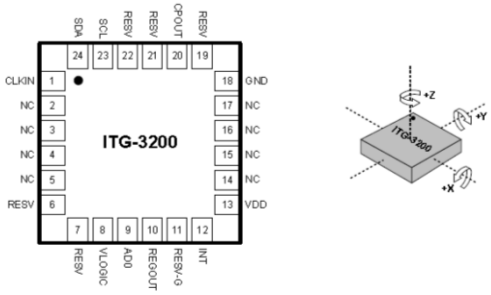


Gambar 2. 8 Konfigurasi Pin ADXL345

2.9 Sensor *Gyroscope*

Gyroscope merupakan suatu alat elektronik yang berfungsi untuk mengukur kecepatan sudut dengan satuan (°/s) yang dialami oleh suatu benda pada *pitch*, *roll* dan *yaw*. Sehingga dengan memanfaatkan data

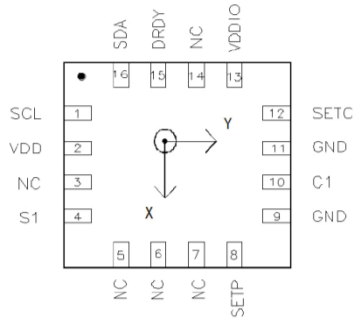
kecepatan sudut tersebut dapat diketahui sudut kemiringan suatu benda.



Gambar 2. 9 Konfigurasi Pin Sensor ITG-3200

2.10 *Sensor Magnetometer*

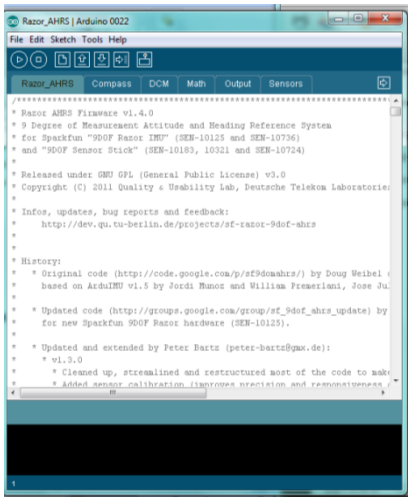
Sensor IMU umumnya terdiri dari kombinasi sensor percepatan *(accelerometer)*, sensor angular *(gyroscope)* dan sebagian ada yang dilengkapi dengan sensor penentuan medan magnet *(magnetometer)* untuk menjejaki keberadaan dan pergerakan suatu benda. *Magnetometer* adalah instrumen ilmiah yang digunakan untuk mengukur kekuatan atau arah medan magnet di sekitar alat tersebut. Salah satu contoh sensor *magnetometer* yang dapat digunakan adalah sensor HMC58883L *(triple-axis magnetometer)*. untuk konfigurasi pin sensor ini dapat dilihat di Gambar 2. 12, sedangkan beberapa fitur dari sensor ini diantaranya adalah sebagai berikut.



Gambar 2.10 Konfigurasi Pin Sensor HMC5883L

2.11 Bahasa Pemograman *Arduino*

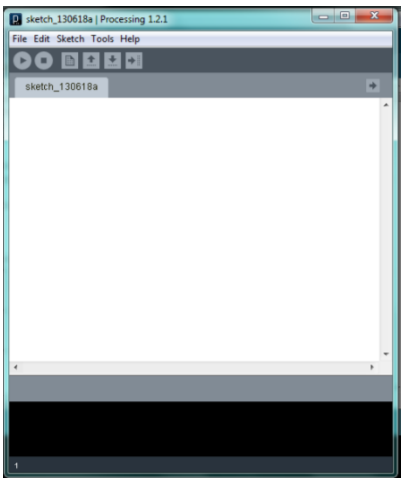
Arduino Uno adalah salah satu jenis dari versi keluarga lainnya, *board* ini berbasis mikrokontroler pada ATmega328 , memiliki 14 digital *input*/*output* pin (dimana 6 pin dapat digunakan sebagai *output PWM*), 6 input *analog*, 16 MHz *osilator* kristal, konektor *USB*, konektor catu daya dan tombol reset. Pin-pin ini berisi semua yang diperlukan untuk mendukung mikrokontroler untuk terhubung ke komputer dengan kabel *USB* atau sumber tegangan bisa didapat dari adaptor AC-DC atau baterai untuk menggunakannya. Arduino menggunakan pemograman dengan bahasa C. Berikut tampilan dari *board Arduino Uno*.



Gambar 2. 11 Tampilan *Framework Arduino IDE*

2.12 Bahsa Pemograman *Processing*

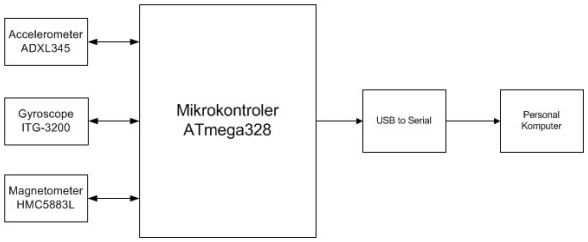
Processing adalah bahasa pemrograman dan lingkungan pemrograman (*development environment*) *open source* untuk memprogram gambar, animasi dan interaksi. Digunakan oleh pelajar, seniman, desainer, peneliti, dan hobbyist untuk belajar, membuat *prototype* dan produksi. *Processing* digunakan untuk mengajarkan dasar-dasar pemrograman komputer dalam konteks rupa dan berfungsi sebagai buku sketsa perangkat lunak (*software*) dan *tool* produksi profesional. *Processing* bebas untuk di *download* dan tersedia untuk *GNU/Linux*, *Mac OS* dan *Windows*.



Gambar 2. 12 Tampilan Perangkat Lunak *Processing*

3. PERANCANGAN ALAT

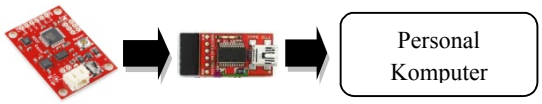
Pada bab ini akan membahas sistem IMU yang dapat diamati pada Gambar 3. 1, dimana pengiriman data dari *accelerometer* untuk pengukuran percepatan, *gyroscope* untuk pengukuran kecepatan sudut dan *magnetometer* untuk pengukuran medan magnet akan diolah dalam mikrokontroler menggunakan metode algoritma *Direct Cosine Matrix* dengan keluaran berupa sudut *roll*, *pitch* dan *yaw*. Setelah mendapatkan sudut orientasi sistem IMU akan mengirim data melalui *USB to Serial* ke personal komputer.



Gambar 3. 1 Diagram Blok Sistem IMU

3.1 Konfigurasi Sistem IMU

Dalam tugas akhir ini akan digunakan sensor IMU jenis Razor IMU 9DoF, sensor IMU jenis ini menggabungkan tiga sensor yaitu ITG-3200 *(triple-axis gyro)*, ADXL345 *(triple-axis accelerometer)* dan HMC5883L *(triple-axis magnetometer)* yang sudah terintegrasi dan berfungsi memberikan sembilan derajat pengukuran inersia. *Output* dari semua sensor diproses oleh *on-board* ATmega328 dan di *output*kan kembali melalui *port USB* ke personal komputer. konfigurasi sistem dapat diamati pada Gambar 3. 2.



Gambar 3. 2 Konfigurasi Sistem IMU

3.2 Perangkat Lunak Razor IMU

Pada bagian ini akan dijelaskan program *firmware* Razor AHRS yang terdiri dari dua bagian yaitu dari aplikasi *software* Arduino dan Processing. Pada bagian aplikasi *software* Arduino akan menampilkan data melalui serial monitor dan pada bagian aplikasi *software* Processing akan menampilkan simulasi pergerakan obyek yaitu modul Razor itu sendiri dengan visualisasi grafis.

3.2.1 Program Arduino

Pada bagian aplikasi *software* Arduino data keluarannya ditampilkan di dalam serial monitor. Didalam program tersebut 5 subprogram yaitu program *Compass*, *DCM*, *Mathematic*, *Output* dan *Sensors*. Fungsi dari masing-masing subprogram adalah sebagai berikut.

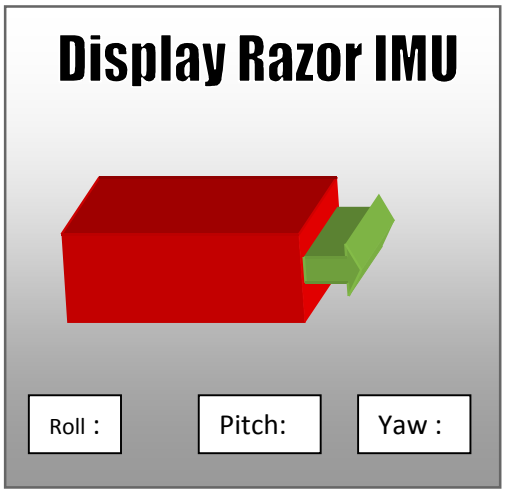
1. *Sensors :* Mendeklarasikan variabel untuk inisialisasi I2C dari setiap sensor.
2. *Output :* Mendeklarasikan mode *output* seperti *ouput angles*, *output calibration* dan *output sensors*.
3. *Mathematic :* Mendeklarasikan varibel yang berfungsi sebagai komputasi *dot* dan *cross product* serta inisialisasi rotasi matrik.
4. *DCM :* Mendeklarasikan varibel yang berfungsi sebagai algoritma *Direct Cosine Matrix* yang terdiri dari

Normalize, *Drift correction*, *Matrix update* dan *Euler_Angles*.

5. *Compass :* Mendeklarasikan variabel *Compass Heading* untuk kompensasi kemiringan medan magnet di sumbu X dan Y dan sebagai penunjuk arah medan magnet.

3.2.2 Program Processing

Processing adalah bahasa pemrograman dan lingkungan pemrograman (*development environment*) *open source* untuk memprogram gambar, animasi dan interaksi. Digunakan oleh pelajar, seniman, desainer, peneliti, dan hobbyist untuk belajar, membuat *prototype* dan produksi. Oleh karena itu, aplikasi *software* Processing akan digunakan untuk menampilkan visualisasi grafis pergerakan dari modul Razor IMU. Tampilan pada program Processing dan flowchartnya dapat dilihat di Gambar 3. 3.



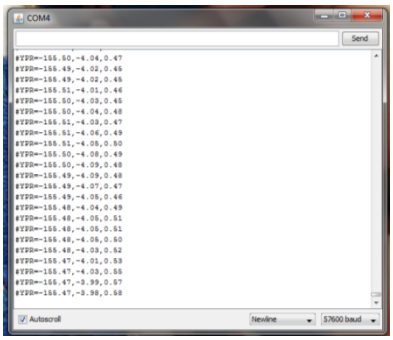
Gambar 3. 3 Tampilan Aplikasi *Software* Processing

4. PENGUJIAN DAN ANALISA

Pada bab ini akan membahas pengujian dan analisa dari sistem yang digunakan. Pengujian ini terbagi menjadi beberapa bagian seperti : pengujian sistem komunikasi serial, pengujian kalibrasi sensor *accelerometer*, *gyroscope* dan *magnetometer*, pengujian kirim *command* dan pengujian program Processing.

4.1 Pengujian Sistem Komunikasi Serial

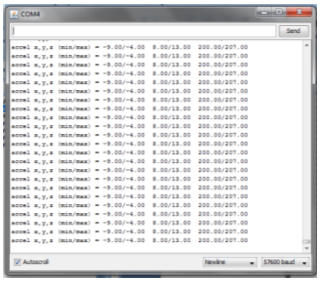
Untuk pengujian sistem komunikasi antara modul Razor IMU dengan *port* serial komputer dapat dilakukan dengan memanfaatkan aplikasi *software* Arduino dengan cara memilih menu "Tools" lalu "Serial Monitor" dengan men-*set* baudrate 57600. Pada pengujian ini diberikan *command* #ot untuk menampilkan sudut *output* dalam format teks. Berikut tampilan dari pengujian serial monitor.



Gambar 4. 1 Pengujian Sistem Komunikasi

4.2 Pengujian Kalibrasi Sensor

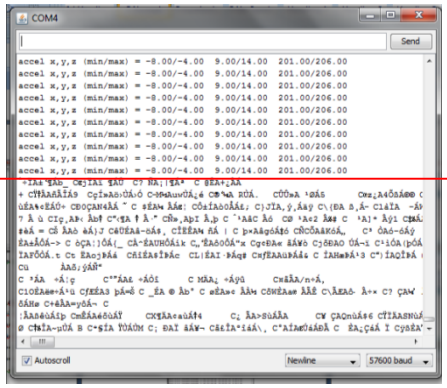
Untuk mendapatkan nilai minimum dan maksimum agar bisa menggunakan modul sensor ini maka dilakukan proses kalibrasi dengan cara mengirim *command* #on untuk kalibrasi setiap sensor. Berikut ini salah satu langkah kalibrasi sensor pada *accelerometer*.



Gambar 4. 2 Pengujian Kalibrasi Sensor *Accelerometer*

4.3 Pengujian Kirim *Command*

Pada bagian ini langkah pengujian yang akan dilakukan adalah dengan cara mengirimkan setiap *command* yang terdapat dalam program Arduino. Berikut ini salah satu fungsi pengujian kirim command #ob untuk menampilkan *output* dengan format *binary*.

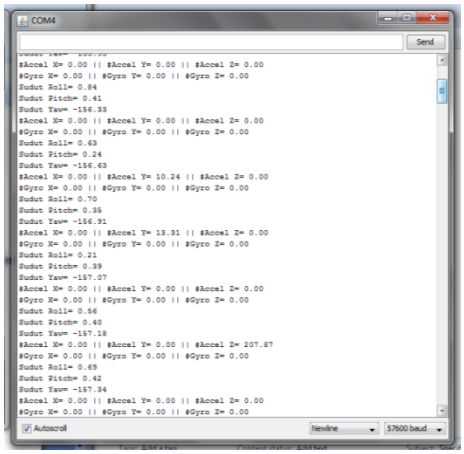


Gambar 4.3 Tampilan Serial Monitor saat *Command* #ob

Untuk mngetahui nilai keluaran dari percepatan, kecepatan sudut , dan posisi dalam orientasi sudut *roll*, *pitch* dan *yaw* dapat dilihat nilai perubahannya dengan menggunakan *command* #os yang berfungsi untuk menampilkan mode *output* sensor secara format teks.

#acc adalah nilai keluaran dari sensor *accelerometer*
#gyr adalah nilai keluaran dari sensor *gyroscope*
#mag adalah nilai keluaran dari sensor *magnetometer*

Agar dapat melihat adanya nilai perubahan dari setiap sensor dengan tidak secara berlanjut yaitu dengan menggunakan *command* #o0, selanjutnya dengan kirim #f agar dapat melihat nilai perubahan dari setiap sensor satu persatu.



Gambar 4.4 Tampilan Mode Ouput Sensor

Dalam keadaan diam terlihat nilai rata-rata dari sensor accelerometer dan gyroscope adalah sebagai berikut.

#Accel X= 0.00 || #Accel Y= 0.00 || #Accel Z= 0.00
#Gyro X= 0.00 || #Gyro Y= 0.00 || #Gyro Z= 0.00
Sudut Roll= 0.84
Sudut Pitch= 0.41
Sudut Yaw= -156.33
#Accel X= 0.00 || #Accel Y= 0.00 || #Accel Z= 0.00
#Gyro X= 0.00 || #Gyro Y= 0.00 || #Gyro Z= 0.00
Sudut Roll= 0.63
Sudut Pitch= 0.24
Sudut Yaw= -156.63
#Accel X= 0.00 || #Accel Y= 10.24 || #Accel Z= 0.00
#Gyro X= 0.00 || #Gyro Y= 0.00 || #Gyro Z= 0.00
Sudut Roll= 0.70
Sudut Pitch= 0.35
Sudut Yaw= -156.91
#Accel X= 0.00 || #Accel Y= 13.31 || #Accel Z= 0.00
#Gyro X= 0.00 || #Gyro Y= 0.00 || #Gyro Z= 0.00
Sudut Roll= 0.21
Sudut Pitch= 0.39
Sudut Yaw= -157.07
#Accel X= 0.00 || #Accel Y= 0.00 || #Accel Z= 0.00
#Gyro X= 0.00 || #Gyro Y= 0.00 || #Gyro Z= 0.00
Sudut Roll= 0.56
Sudut Pitch= 0.40
Sudut Yaw= -157.18

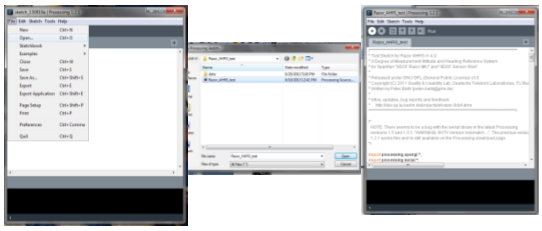
Berdasarkan dari proses tersebut untuk melihat adanya perubahan besar nilai keluarannya yaitu dengan cara menggerakkan modul Razor terhadap sumbu x, y dan z. Nilai-nilai tersebut didapat berdasarkan dari fungsi variabel proses kompensasi *error* sensor terhadap

pembacaan nilai awal dari setiap sensor. Berikut ini adalah varibel yang digunakan dalam proses kompensasi *error* sensor pada program Arduino.

- Kompensasi *Error Aceelerometer*
  accel[0] = (accel[0] − accel x offset) ∗ accel x scale;
  accel[1] = (accel[1] − accel y offset) ∗ accel y scale;
  accel[2] = (accel[2] − accel z offset) ∗ accel z scale;
- Kompensasi *Error Magnetometer*
  magnetom[0] = (magnetom[0] − magn x offset) ∗ magn x scale;
  magnetom[1] = (magnetom[1] − magn y offset) ∗ magn y scale;
  magnetom[2] = (magnetom[2] − magn z offset) ∗ magn z scale;
- Kompensasi *Error Gyroscope*
  gyro[0] −= gyro average offset x ;
  gyro[1] −= gyro average offset x ;
  gyro[2] −= gyro average offset x ;

4.4 Pengujian Program Processing

Tujuan dari pengujian program Processing adalah untuk menampilkan visualisasi grafis pergerakan dari modul Razor IMU. Langkah-langkahnya adalah sebagai berikut. Menjalankan aplikasi *software* Processing dan membuka *file* "Razor_AHRS_test.pde" di *folder* "Processing" dalam program *firmware* Razor AHRS.



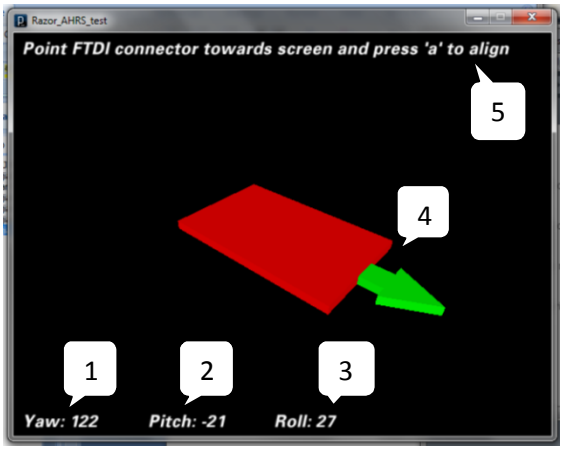
Gambar 4. 13 Tampilan Awal Aplikasi Software Processing

Proses selanjutnya adalah menghubungkan modul Razor IMU ke *port* USB komputer. Ketika program tersebut di *run* maka ada proses untuk sinkronisasi *serial port*, jika sesuai maka tampilan programnya akan seperti Gambar 4. 14.



Gambar 4. 14 Proses Sinkronisasi modul Razor dengan Program Processing

Proses selanjutnya akan menampilkan visualisasi grafis yang bergerak sesuai dengan gerak nyata modul Razor IMU.



Gambar 4. 15 Tampilan Visualisasi Grafis Program Processing

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melakukan pengujian terhadap device yang digunakan, penulis dapat mengambil beberapa kesimpulan, diantaranya sebagai berikut.

1. Modul sensor Razor IMU dapat diimplementasikan sebagai *sensor Inertial Measurement Unit* dengan metode *Inertial Navigation System*.
2. Berdasarkan dari tujuan tugas akhir ini, untuk mengetahui nilai percepatan dan kecepatan sudut tidak dengan cara komputasi algoritma *Direct Cosine Matrix* yang terdapat dalam program Arduino, karena algoritma ini bertujuan untuk dapat mengetahui representasi orientasi berupa sudut *roll*, *pitch* dan *yaw* pada suatu obyek yang bergerak.
3. Aplikasi *software* Processing dapat menampilkan pergerakan obyek dalam bentuk visual grafis.

5.2 Saran

Pada bab ini penulis ingin memberikan saran yang nantinya dapat dijadikan pengembangan selanjutnya. Saran tersebut diantaranya adalah.

1. Modul Razor IMU dapat diaplikasikan dalam sistem gerak quadqopter atau aplikasi yang membutuhkan sistem navigasi.
2. Untuk aplikasi tampilan bisa diaplikasikan ke dalam aplikasi *software* lain sepeti *Visual Basic* atau *Labview*.

DAFTAR PUSTAKA

Woodman, Oliver J. (2007), *An introduction to inertial navigation, University Of Cambridge Computer Laboratory.*

W. Premerlani and Paul Bizar. Direction Cosine Matrix IMU : Theory.

Seifert, Kurt., Camacho Oscar. (2007), *Implementing Positioning Algorithms Using Accelerometers*, freescale Semiconductor, Rev 0.

9 Degrees of Freedom-Razor IMU-Manual Book.

Winoto, Ardi. (2010). Mikrokontroler AVR ATmega8/32/18/8535 dan Pemrogramannya dengan Bahasa C pada WinAVR. Penerbit : Informatika. Bandung.



Nama : Yuga Aditya Pramana
TTL : Sumedang
13 Januari 1989

Program Studi
Teknik Elektro UNIKOM