

IMPLEMENTASI 3D *GAME ENGINE* SEBAGAI VISUALISASI SISTEM NAVIGASI 6-DOF PADA ROKET KENDALI LAPAN

[3D GAME ENGINE IMPLEMENTATION AS 6-DOF NAVIGATION SYSTEM VISUALIZATION AT GUIDED ROCKET OF LAPAN]

Romi Wiryadinata^{*)}, Wildan Yusuf^{*)}, Ri Munarto^{*)}, Frandi A. Kaharjito^{*)}, Sri Kliwati^{**)}

^{*)} SINKEN Riset Group, Lab. Kendali, Teknik Elektro, Fak. Teknik, UNTIRTA

^{**)} Peneliti Bidang Kendali dan Telemetri, PUSTEKROKET, LAPAN

e-mail: romi@wiryadinata.web.id

ABSTRACT

Game engine, which are often used by game developers, has software framework that facilitates their work on important tasks such as graphic design or 3D and 2D graphics rendering. This research uses game engine called 3D state game engine for visualize 6-DOF navigation system. The software is developed for visualize range of motion of the object by reading and processing the streaming data from navigation sensor (accelerometers and rate-gyroscope). The test result show that developed software can visualize the motion of the objects in 3D animation. Software can accept and process the data as text file that is transmitted in real-time through serial port. IMU hardware with software can use through the calibration process first, and be given a filter, so that the better results can be obtained.

Key word: 3D Game Engine, Graphical User Interface, IMU, 6-DOF

ABSTRAK

Game engine, yang sering digunakan oleh para pengembang *game*, memiliki kerangka kerja perangkat lunak yang memfasilitasi pekerjaan mereka pada tugas-tugas penting seperti desain grafis atau 3D dan 2D *rendering* grafis. Penelitian ini menggunakan *game engine* yang disebut 3D *state game engine* untuk memvisualisasikan sistem navigasi 6-DOF. Perangkat lunak ini dikembangkan untuk memvisualisasikan rentang gerak obyek dengan membaca dan pengolahan data *streaming* dari sensor navigasi (*accelerometers* dan *rate-gyroscope*). Hasil pengujian menunjukkan bahwa mengembangkan perangkat lunak dapat memvisualisasikan gerakan obyek dalam animasi 3D. *Software* dapat menerima dan memproses data sebagai *file* teks yang ditransmisikan secara *real-time* melalui *port serial*. *Hardware* IMU dengan *software* dapat digunakan melalui proses kalibrasi terlebih dahulu, dan diberi *filter*, sehingga hasil yang lebih baik dapat diperoleh.

Kata kunci: 3D Game Engine, Graphical User Interface, IMU, 6-DOF

1 PENDAHULUAN

Pada perkembangan teknologi avionik, sistem *Attitude and Heading Reference System* (AHRS) memegang peranan yang sangat penting. AHRS dibutuhkan untuk memberikan informasi dan visualisasi yang akurat tentang perilaku pesawat (kecepatan, ketinggian, arah, dan sudut inklinasi) kepada pilot.

Tingkat akurasi yang diberikan oleh sebuah sistem AHRS memegang peranan penting dalam memberikan informasi yang valid tentang perilaku pesawat kepada pilot karena pada kondisi tertentu dalam penerbangan dapat terjadi salah persepsi tentang perilaku pesawat jika hanya mengandalkan indera perasa pilot untuk mendeteksi perilaku pesawat.

Kesalahan persepsi tentang perilaku pesawat tersebut dapat menyebabkan kesalahan pada jalur penerbangan yang ditempuh dan bahkan dapat menyebabkan kecelakaan seperti yang terjadi pada pesawat Adam Air penerbangan KI-574. Menurut laporan akhir hasil investigasi oleh KNKT, kecelakaan ini terjadi sebagai kombinasi beberapa faktor termasuk kegagalan kedua pilot dalam intensitas memonitor flight instrument khususnya dalam 2 menit terakhir penerbangan. Fokus konsentrasi pada malfungsinya IRS telah mengalihkan perhatian kedua pilot dari flight instrumen dan membuka peluang terjadinya *increasing decent* dan *bank angle* tidak teramati. Kedua pilot tidak mendeteksi dan menahan *decent* sesegera mungkin untuk mencegah kehilangan kendali (Kurniadi, 2008).

Pada 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. Sistem ini juga dapat dikembangkan lebih jauh untuk menyusun kecerdasan buatan sehingga obyek dapat bergerak secara *auto-pilot* (Adiprawita dkk., 2007).

Gambar beresolusi tinggi baik yang bergerak maupun tidak bergerak serta animasi 3D bukan merupakan hal yang mewah lagi, bahkan merupakan suatu keharusan dalam sebuah perangkat lunak komputer spesifikasi tinggi. Banyak sekali pengembang perangkat lunak yang menyediakan 3D *Game Engine* sebagai antar-muka pemrograman grafis. 3D *Game Engine* tersebut sangat mungkin untuk diimplementasikan pada pengembangan perangkat lunak yang dapat merepresentasikan sistem navigasi 6-DOF (*Six Degree of Freedom*), dengan menggunakan *engine* 3D tersebut, program dapat dengan mudah memvisualisasikan keadaan obyek yang sedang dipantau pada komputer secara *real-time*. Obyek yang

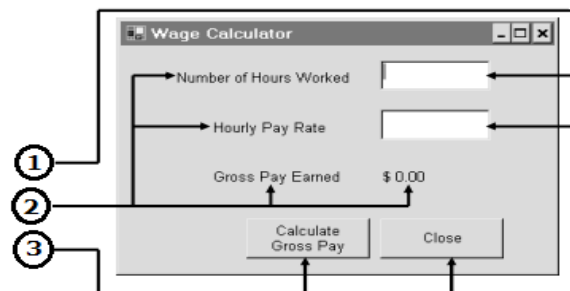
dipantau merupakan sebuah kendaraan yang dilengkapi dengan sensor-sensor navigasi.

2 LANDASAN TEORI

2.1 Pemrograman Visual

Pemrograman visual adalah metode pembuatan program dengan cara membuat koneksi antar obyek dengan cara menggambar, menunjuk dan mengklik pada diagram dan ikon dan dengan berinteraksi dengan diagram alur. Pemrograman visual memungkinkan *programmer* untuk merancang program dalam dua atau lebih dimensi perancangan.

Salah satu contoh obyek yang dapat dibuat menggunakan pemrograman visual adalah berupa *form* (Gambar 2-1). *Form* adalah sebuah obyek yang dapat memuat obyek lain seperti tombol, kotak teks atau label.



Gambar 2-1: Contoh obyek berupa *form*

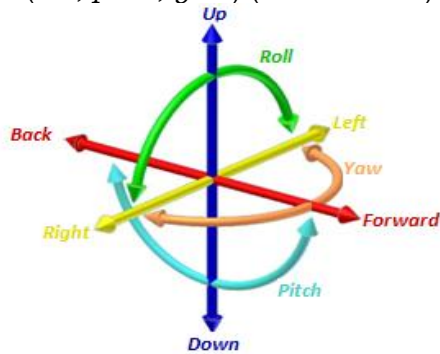
Form tersebut berisi obyek berupa data dan aksi. Obyek data seperti *hourly pay rate* merupakan properti berupa teks yang menentukan tampilan obyek *form* sedangkan obyek aksi seperti tombol *calculate gross pay* merupakan metode yang menentukan bagaimana sebuah obyek bereaksi terhadap interaksi pengguna.

Elemen-elemen yang terdapat di dalam sebuah *form* adalah obyek yang disebut *control*. *Form* yang ditampilkan pada gambar 2-1 memuat dua *control* kotak teks (1), empat *control* berupa label (2), dan dua *control* berupa tombol (3). Nilai yang ditampilkan oleh sebuah *control* adalah properti teks *control* tersebut dan setiap tombol memiliki

metode yang akan dijalankan apabila tombol tersebut ditekan.

2.2 Six Degree of Freedom

Six Degree of Freedom, atau 6-DOF adalah pergerakan sebuah benda tegar dalam ruang tiga dimensi, yaitu kemampuan untuk bergerak ke depan/belakang, atas/bawah, kanan/kiri (translasi pada tiga sumbu yang saling tegak lurus) yang dikombinasikan dengan rotasi pada tiga sumbu tegak lurus (*roll*, *pitch*, *yaw*) (Gambar 2-2).

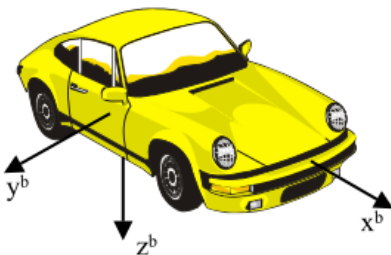


Gambar 2-2: *Six Degree of Freedom*

Sebuah sistem dikatakan memiliki 6-DOF apabila benda tersebut dapat melakukan gerakan translasi dan rotasi pada tiga sumbu yang saling tegak lurus secara terpisah.

2.3 Body Frame

Body frame (*b-frame*) adalah serangkaian sumbu ortogonal yang selaras dengan sumbu *roll* (x^b), *pitch* (y^b), dan *heading* (z^b) dari sebuah kendaraan (Gambar 2-3).



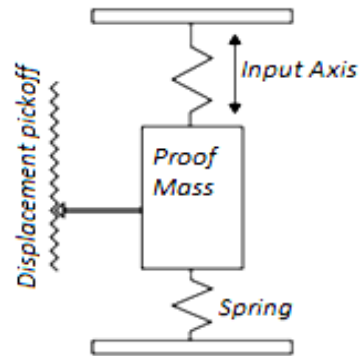
Gambar 2-3: *Body Frame* (Shin, 2001)

Body frame bersifat selalu mengikuti perubahan arah pergerakan kendaraan, baik translasi maupun rotasi pada ketiga sumbu tersebut. Sumbu x^b selalu mewakili arah depan, y^b mewakili

arah kanan, dan sumbu z^b mewakili arah bawah walaupun terjadi perubahan orientasi atau posisi pada kendaraan.

2.4 Akselerometer

Sebuah akselerometer secara garis besar dapat diklasifikasikan sebagai akselerometer mekanik atau elektronik. Sebuah akselerometer pada umumnya tersusun atas sebuah massa yang diapit oleh dua buah pegas dan selaras dengan sumbu masukannya (Gambar 2-4).



Gambar 2-4: Akselerometer Mekanik (Titterton and Weston, 2004)

Perpindahan massa diukur oleh *displacement pick-off*, menghasilkan sinyal yang sesuai dengan gaya F yang mempengaruhi massa pada arah yang sejajar dengan sumbu masukan. Hukum kedua Newton digunakan untuk menghitung akselerasi yang dialami oleh akselerometer.

Nilai yang dihasilkan oleh sebuah akselerometer merupakan percepatan dalam satuan m/s^2 tetapi kebanyakan produsen akselerometer mencantumkan nilai percepatan dalam satuan g pada *datasheet* yang disertakan. Nilai percepatan yang diukur oleh akselerometer dapat digunakan untuk menentukan kecepatan sebagai fungsi waktu melalui persamaan integral sederhana:

$$v(t) = \int a(t) dt \quad (2-1)$$

$$s(t) = \int v(t) dt \quad (2-2)$$

Keterangan:

$$s(t) = \text{Jarak (m)}$$

$$v(t) = \text{Kecepatan (m/s)}$$

$$a(t) = \text{Percepatan (m/s}^2\text{)}$$

2.5 Giroskop

Sebuah giroskop merupakan alat yang dirancang sedemikian rupa sehingga dapat mendeteksi perubahan orientasi yang dialami oleh giroskop tersebut. Kebanyakan giroskop yang digunakan merupakan sebuah *rate-gyro* yang dapat mendeteksi laju perubahan sudut dan menghasilkan keluaran berupa kecepatan sudut ω dalam satuan $^{\circ}/s$. Kecepatan sudut yang dideteksi oleh giroskop kemudian digunakan untuk memperoleh nilai perubahan orientasi obyek yang dipasang giroskop tersebut pada kerangka referensi *body frame* melalui persamaan integral sederhana:

$$\theta(t) = \int \omega(t)dt \quad (2-3)$$

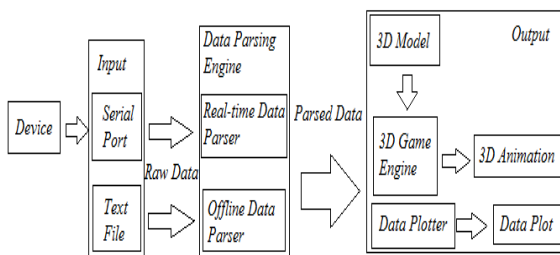
Keterangan:

$\theta(t)$ = Sudut rotasi ($^{\circ}$)

$\omega(t)$ = Kecepatan sudut ($^{\circ}/s$)

3 PERANCANGAN SOFTWARE

Cara kerja *software* secara garis besar dapat dilihat dalam diagram blok sistem yang ditunjukkan oleh Gambar 3-1.



Gambar 3-1: Diagram Blok Software

Aplikasi yang dikembangkan memiliki dua jenis sistem *input* untuk memperoleh informasi data sensor yaitu melalui *serial port* untuk memperoleh data secara *real-time* atau melalui *file* teks yang berisi informasi data sensor. Sistem *input* ini memberikan informasi data berupa gabungan data semua sensor (*raw data*) yang tersusun dalam satu baris kepada bagian program yang bernama *data parsing engine* untuk dikelompokkan menurut jenis sensor yang digunakan dan disusun agar menghasilkan data yang kontinu.

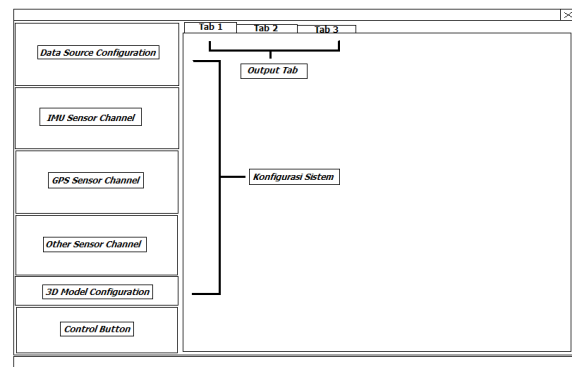
Input pada penelitian ini sama dengan *input* yang akan di olah untuk INS yaitu keluaran dari IMU yang berupa posisi ($S_x(t), S_y(t), dan S_z(t)$), rotasi ($\theta(t), \phi(t), dan \psi(t)$), dan GPS/INS akan (posisi, percepatan, dan *attitude*). (Wiryadinata, 2007)

Data yang telah diolah oleh *data parsing engine* dinamakan *parsed data* dan siap untuk divisualisasikan. Bagian implementasi *game engine* menerima data yang telah diolah tersebut untuk menghasilkan animasi dari model 3D yang telah dikonfigurasi sebelumnya, sedangkan sistem *plot* memvisualisasikan data tiap sensor dalam bentuk *plot* data.

Proses filterisasi dilakukan untuk meminimalisir kesalahan yang dihasilkan dari sensor. Pada penelitian ini dilakukan secara *hardware* menggunakan hasil penelitian yang ada (Wiryadinata, 2008) dan telah berhasil mengembangkan *prototype* IMU dengan menambahkan *high-pass filter* dan *low-pass filter* pada penelitiannya.

3.1 Perancangan GUI

Sistem yang dikembangkan pada penelitian ini menggunakan GUI (*Graphical User Interface*) sebagai antarmuka *software* untuk mempermudah pengoperasian (Gambar 3-2).



Gambar 3-2: Rancangan antar-muka software

Antar-muka *software* secara garis besar dapat dikelompokkan menjadi tiga bagian besar yaitu konfigurasi, *output tab*, dan *control button*.

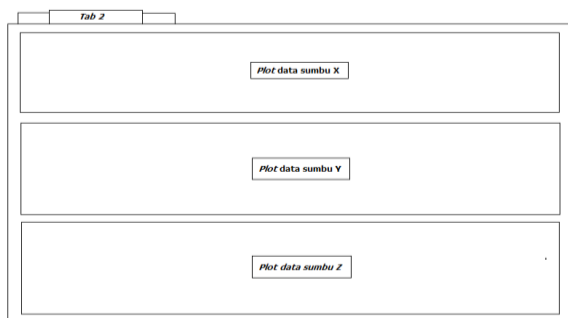
Bagian konfigurasi merupakan letak semua pengaturan yang diperlukan

agar *software* bekerja dengan baik. Terdapat pengaturan sumber data, *baud rate* dan alamat *port serial*, *file* teks yang akan dibaca, frekuensi *sampling*, pengaturan *header ID*, *channel* sensor dan jenis obyek 3D yang akan ditampilkan.

3.2 Perancangan Sistem Plot Data

Sistem *plot* data merupakan *output* yang berfungsi memberikan visualisasi data sensor akselerometer dan giroskop dalam bentuk *plot*. Fungsi utama sistem *plot* pada *output* adalah untuk melihat bentuk sinyal *input* yang diterima oleh *software*. Gambar 3-3 memperlihatkan rancangan visual sistem *plot* data.

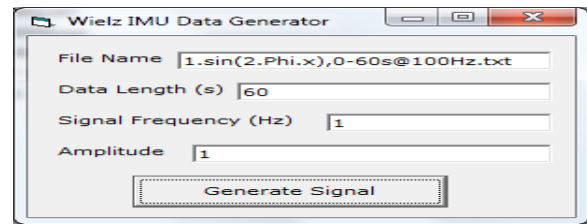
Terdapat tiga *plot* pada setiap *tab* yang masing-masing memvisualisasikan bentuk sinyal dari setiap sumbu sensor (*x*, *y*, dan *z*). Sistem *plot* untuk data sensor akselerometer terletak pada *tab* 2 sedangkan untuk sensor giroskop terletak pada *tab* 3.



Gambar 3-3: Plot Data

3.3 Data Simulasi

Sebuah program sederhana pembangkit sinyal *sinusoidal* yang dapat menghasilkan data sensor berupa *file* dengan format yang dikenali oleh sistem disusun menggunakan bahasa Visual Basic. Data yang dihasilkan oleh program tersebut digunakan sebagai *input* pada pengujian *software*. Program tersebut dapat menghasilkan data sensor berbentuk gelombang *sinusoidal* dan dapat diatur jumlah gelombang yang ingin dihasilkan dan frekuensi *sampling* data yang digunakan. Tampilan program pembangkit gelombang *sinusoidal* tersebut dapat dilihat pada Gambar 3-4.

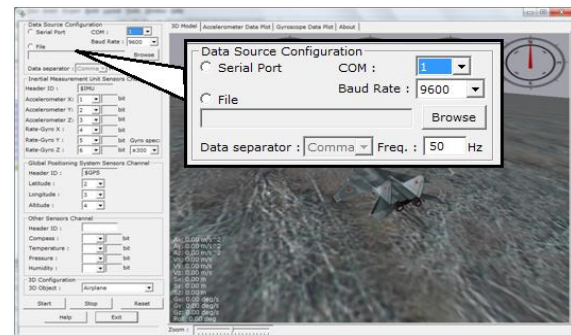


Gambar 3-4: Program sederhana pembangkit data berbentuk *sinusoidal*

Keseluruhan pengujian dilakukan menggunakan data yang berbentuk gelombang *sinusoidal* dengan frekuensi 1Hz, 10Hz dan 50Hz.

4 IMPLEMENTASI GUI DAN PEMBAHASAN

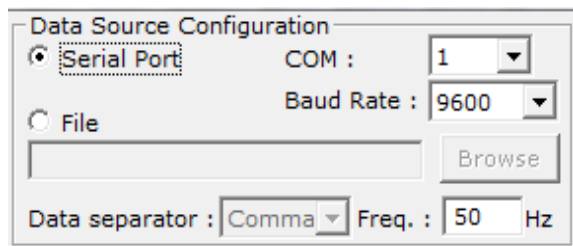
Data source configuration pada GUI merupakan bagian yang berfungsi untuk mengatur jalur sumber data yang akan dibaca dan kemudian diolah oleh *software* menjadi bentuk animasi 3D. Gambar 4-1 menunjukkan *data source configuration* pada GUI.



Gambar 4-1: Data Source Configuration

Terdapat dua pilihan berupa *radio button* yang merupakan pengaturan sumber data yang akan dibaca oleh *software*, yaitu melalui *port serial* atau melalui data yang telah direkam dalam bentuk *file* teks. Pengguna diwajibkan untuk memilih salah satu sumber data sebelum menekan tombol *start*. *Software* akan menampilkan pesan *error* berupa *pop-up* menu bertuliskan *Please specify data source* jika pengguna belum memilih sumber data *input*.

Kedua pilihan dirancang agar hanya dapat dipilih salah satu saja sehingga tidak memungkinkan pembacaan data secara bersamaan dari dua sumber data yang berbeda.

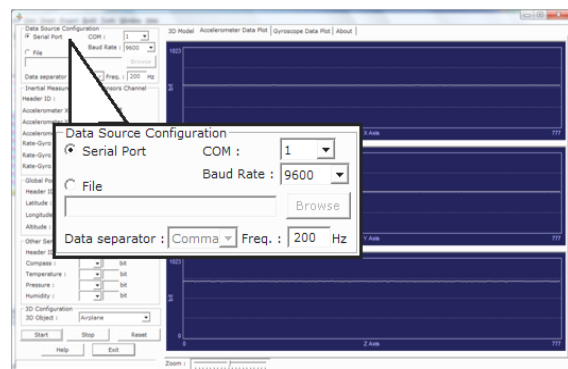


Gambar 4-2: Mode pembacaan melalui serial port

Hasil pengujian pada Gambar 4-2 menunjukkan bahwa ketika port serial dipilih sebagai sumber data yang akan dibaca, tombol browse pada pilihan mode file menjadi disable. Hal sebaliknya terjadi jika mode file dipilih, maka drop down list COM dan baud rate menjadi disable.

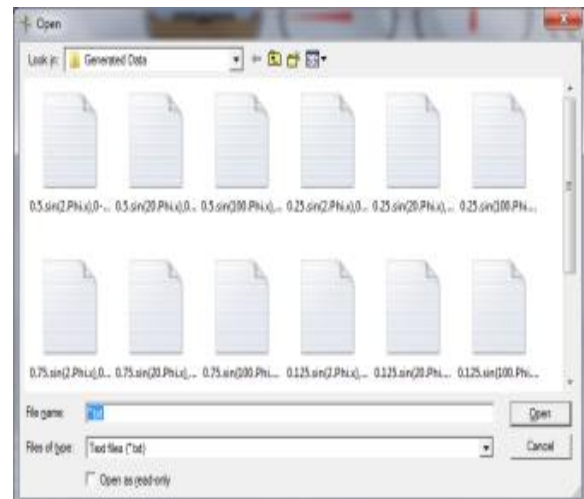
Terdapat pilihan pengaturan alamat COM dan baud rate pada mode serial port. Bagian tersebut harus dikonfigurasi dengan tepat agar software dapat membaca data melalui serial port dengan baik. Software akan menampilkan pesan error berupa pop-up menu bertuliskan Unable to open COM-port jika pengguna salah mengkonfigurasi port serial.

Gambar 4-3 menunjukkan bahwa aplikasi dapat membaca data melalui port serial pada alamat COM 1 dan baud-rate sebesar 9600 bps dengan baik.



Gambar 4-3: Aplikasi membaca data akselerometer melalui port serial

Pengguna diwajibkan untuk memilih file teks dahulu sebelum menekan tombol start pada mode file. Tombol browse yang terletak di samping radio button file berfungsi untuk menentukan file teks yang akan dibaca oleh software (Gambar 4-4).

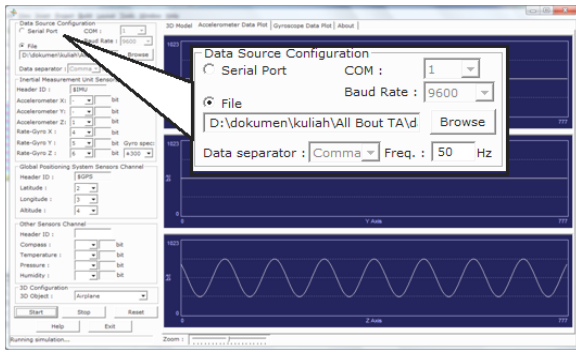


Gambar 4-4: Kotak dialog ketika tombol browse ditekan

Software akan menampilkan pesan error berupa pop-up menu bertuliskan Please specify file to read jika pengguna belum menentukan file teks yang akan digunakan/dibaca.

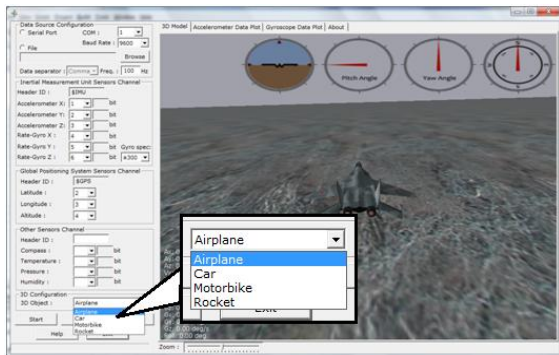
Sebuah drop down list bernama data separator yang terletak di bawah pengaturan mode file berfungsi untuk memilih karakter yang akan digunakan oleh software untuk memisahkan data setiap sensor yang tersusun dalam sebuah baris data. Kotak teks yang terletak di kanan pengaturan data separator berfungsi untuk mengkonfigurasi frekuensi sampling pembacaan data tiap baris data yang terdapat pada file teks. Software akan menampilkan pesan error berupa pop-up menu bertuliskan Please enter a positive number jika frekuensi sampling belum ditentukan.

Gambar 4-5 menunjukkan aplikasi dapat melakukan pembacaan data yang terdapat dalam sebuah file teks dengan baik. Gambar tersebut menunjukkan bentuk gelombang sinusoidal pada plot paling bawah yang merupakan data sensor yang terdapat di dalam file teks yang dibaca. Data yang dibaca oleh aplikasi menggunakan karakter koma sebagai karakter pemisah data antar sensor dan menggunakan frekuensi sampling sebesar 50Hz.



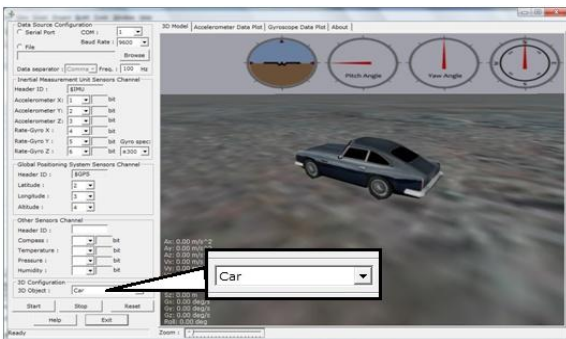
Gambar 4-5: Aplikasi membaca data pada sebuah file teks

Aplikasi yang dikembangkan mendukung penggunaan berbagai macam bentuk obyek 3D yang akan ditampilkan dalam animasi.



Gambar 4-6: Daftar obyek yang dapat digunakan pada animasi

Gambar 4-6 menunjukkan sebuah *drop-down list* yang berisi daftar obyek yang dapat ditampilkan dalam animasi. Obyek yang didukung oleh aplikasi adalah obyek pesawat, mobil, sepeda motor dan roket. Gambar 4-7 menunjukkan obyek animasi yang sedang digunakan berupa mobil.



Gambar 4-7: Obyek mobil digunakan

4.1 Pengujian Animasi Menggunakan Data Simulasi

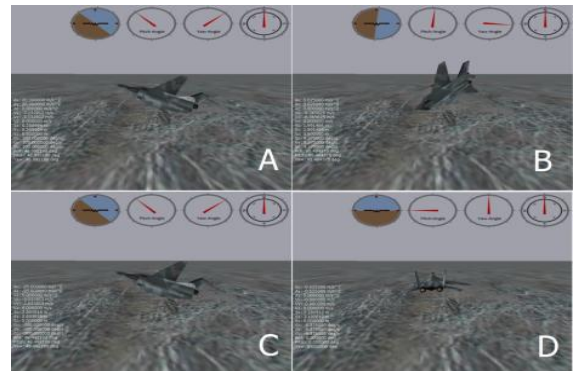
Pengujian dilakukan menggunakan simulasi data akselerometer dan giroskop

yang berbentuk gelombang sinusoidal dengan frekuensi 1Hz, 10Hz, dan 50Hz. Pemilihan frekuensi ini untuk melihat perubahan frekuensi secara ekstrim berdasarkan bentuk gelombang dan mengamati hasil pergerakan yang dihasilkan oleh animasinya.



Gambar 4-8: Bentuk plot grafik gelombang 1Hz

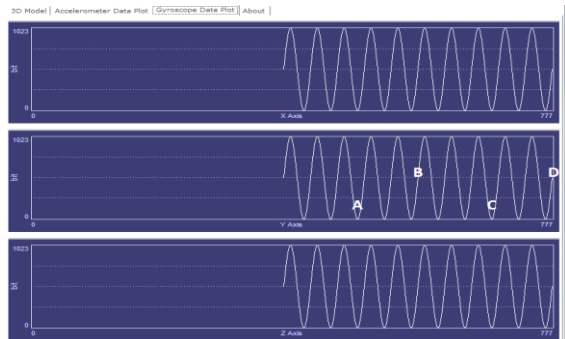
Gelombang *input* yang ditandai dengan huruf penanda seperti terlihat pada gambar 4-8 di ambil setiap $\frac{1}{4}$ detik. A menunjukkan data pada detik ke 0,25, B menunjukkan data pada detik ke 0,5, C menunjukkan data pada detik ke 0.75, dan D menunjukkan data pada 1 detik. Animasi yang dihasilkan dari masing-masing penanda terlihat pada Gambar 4-9.



Gambar 4-9: Hasil animasi menggunakan data sinusoidal dengan frekuensi 1Hz

Terlihat bahwa animasi menunjukkan pergerakan yang sesuai dengan bentuk gelombang sinusoidal yang dibentuk. Gambar 4-9(A) menampilkan sudut obyek yang berubah dan terus meningkat hingga pada sudut terjauhnya pada 0,5s (Gambar 4-9(B)) dan kemudian bergerak kembali menuju posisi semula seperti yang ditampilkan pada Gambar 4-9(D). Titik sudut terjauh obyek terjadi pada detik ke-0,5s seperti yang terlihat pada Gambar 4-9(C).

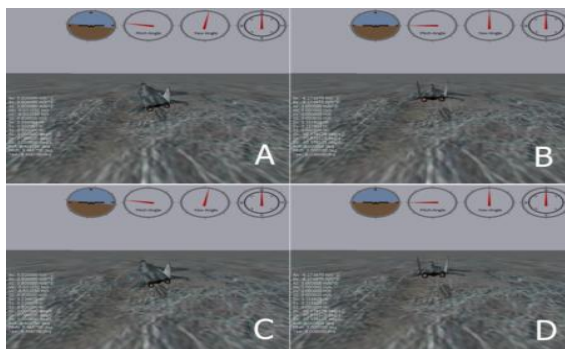
Berikut ini (Gambar 4-10) percobaan dengan gelombang sinusoidal 10Hz, terlihat bahwa untuk setiap $\frac{1}{4}$ detik menghasilkan 2,5 gelombang yang ditandai dengan huruf penanda A, B, C, dan D.



Gambar 4-10: Bentuk plot grafik gelombang 10Hz

Bentuk gelombang sinusoidal dari Gambar 4-10 menghasilkan animasi 3D seperti yang terdapat pada Gambar 4-11. Bentuk animasi Gambar 4-11(A) akan sama dengan Gambar 4-11(C) karena bentuk gelombang sinusoidalnya juga berada pada pola yang sama. Gambar 4-11(B) terlihat bahwa obyek animasi kembali ke posisi semula, hal tersebut sesuai dengan bentuk gelombang 10Hz yang pada detik ke-0,5 tepat merupakan akhir dari gelombang kelima (Gambar 4-10(B)).

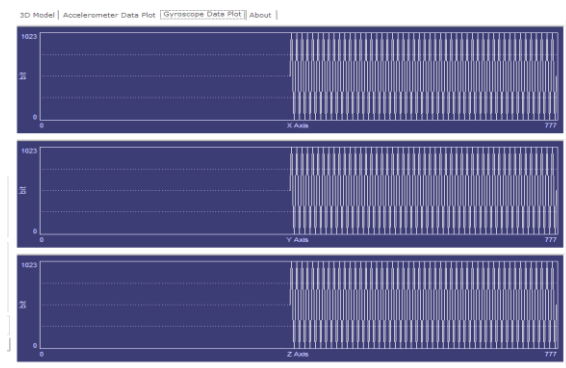
Pada percobaan gelombang sinusoidal 10Hz perubahan/pergerakan obyek semakin sedikit terlihat perbedaannya disebabkan oleh kerapatan frekuensi yang dihasilkan.



Gambar 4-11: Hasil animasi menggunakan data sinusoidal dengan frekuensi 10Hz

Percobaan berikut ini membangkitkan gelombang sinusoidal sebesar 50Hz. Bentuk grafik data *input*

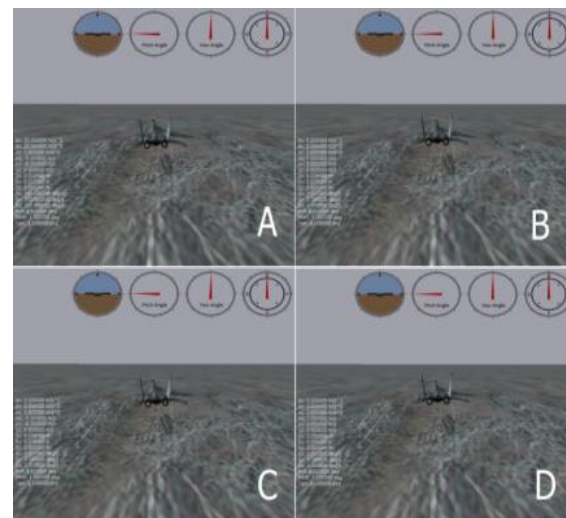
nya dapat dilihat pada gambar 4-12 berikut ini.



Gambar 4-12: Bentuk plot grafik gelombang 50Hz

Gambar yang ditampilkan oleh Gambar 4-12 di atas menunjukkan bahwa pada frekuensi *input* 50Hz, sudut rotasi obyek tidak sebesar sudut yang dihasilkan pada pengujian menggunakan frekuensi *input* 1Hz dan 10Hz. Hal tersebut menunjukkan bahwa semakin tinggi frekuensi sinyal *input*, maka semakin kecil pengaruhnya terhadap keadaan obyek pada amplitudo gelombang yang sama.

Animasi 3D yang dihasilkan saat di cuplik persekian detik juga menghasilkan data yang sama (Gambar 4-13), hanya saja pada saat menggunakan data gelombang sinusoidal 50Hz animasi menunjukkan bahwa obyek melakukan perubahan dan pergerakan yang cepat seperti *noise* bahkan tidak tampak adanya pergerakan dan hanya terlihat berupa getaran pada obyek 3D.



Gambar 4-13: Hasil animasi menggunakan data sinusoidal dengan frekuensi 50Hz

4.2 Pengujian Animasi Menggunakan Data Hardware IMU

Pengujian ini dilakukan menggunakan data sensor yang berasal dari sebuah perangkat IMU. Sensor inersia yang digunakan pada perangkat tersebut adalah sebuah akselerometer tiga sumbu tipe KXM52-1050 yang mendeteksi percepatan dengan jangkauan deteksi percepatan translasi sebesar $\pm 2g$ dan tiga buah giroskop tipe ENC-03JA dengan jangkauan deteksi kecepatan sudut sebesar $\pm 300^\circ/s$ yang disusun secara ortogonal pada sumbu x , y , dan z .

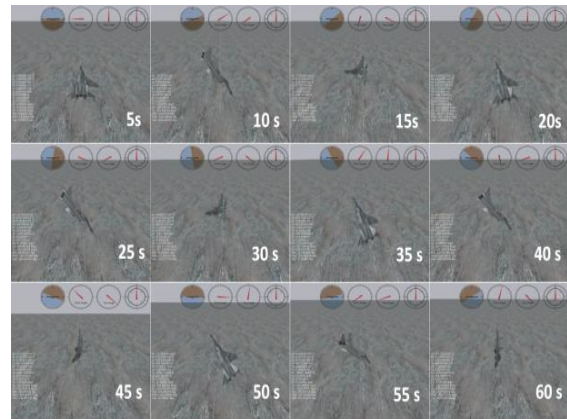
Output yang dihasilkan oleh perangkat IMU tersebut merupakan data sensor dalam satuan bit yang tersusun atas tiga nilai percepatan dan kecepatan sudut rotasi masing-masing pada sumbu x , y , dan z dengan resolusi 10bit. *Output* yang dihasilkan oleh perangkat tersebut disalurkan melalui *port serial*.

Perangkat IMU (Gambar 4-14) yang digunakan merupakan perangkat yang masih dalam tahap pengembangan yang dilakukan oleh Wiryadinata pada tahun 2008 dan belum menjalani proses kalibrasi yang sempurna sehingga data yang dihasilkan belum bisa dikatakan akurat. Pengujian ini dilakukan untuk menguji konektivitas antara *software* dengan perangkat IMU dan beserta animasi yang dihasilkan.



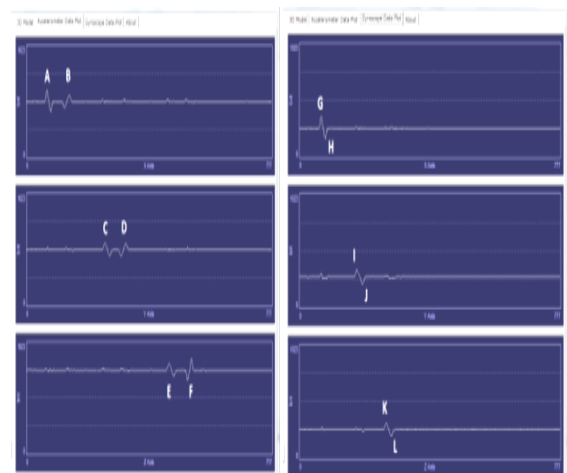
Gambar 4-14: Perangkat IMU yang digunakan

Hasil pengujian pada Gambar 4-15 merupakan tampilan animasi yang diambil setiap lima detik sekali sejak dimulainya pembacaan data perangkat IMU oleh *software*.



Gambar 4-15: Animasi yang dihasilkan

Tampilan grafik plot data yang berasal dari sensor akselerometer dan giroskop dapat dilihat pada Gambar 4-16.

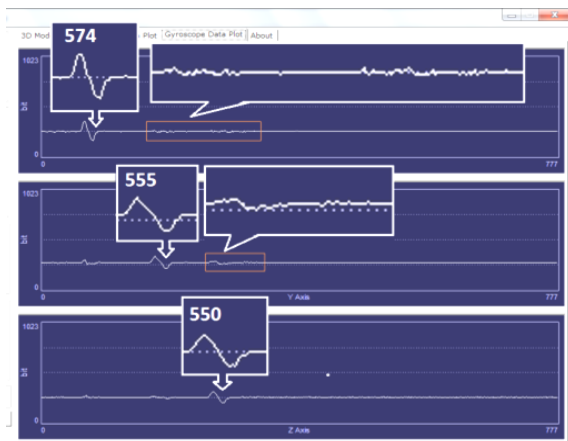


Gambar 4-16: Grafik data akselerometer dan giroskop

Tiga buah grafik plot pada sebelah kiri menampilkan data akselerometer yang mewakili percepatan yang dialami oleh perangkat IMU. Pengujian dilakukan dengan mentranslasikan perangkat IMU sejauh 5 cm pada sumbu x positif (a), sumbu x negatif (b), pada sumbu y positif (c), sumbu y negatif (d), dan pada sumbu z positif (e), dan sumbu z negatif (f).

Data yang berasal dari giroskop ditampilkan oleh tiga buah grafik plot sebelah kanan pada Gambar 4-16. Grafik tersebut mewakili rotasi pada perangkat IMU. Pengujian dilakukan dengan merotasikan perangkat IMU sebesar 90° pada sumbu x searah jarum jam (g), berlawanan arah jarum jam (h), pada sumbu y searah jarum jam (i), berlawanan arah jarum jam (j), dan pada sumbu z searah jarum jam (k), berlawanan arah jarum jam (l).

Gambar 4-17 menampilkan perbedaan nilai puncak gelombang yang dihasilkan oleh giroskop di tiap sumbu ortogonal pada sudut rotasi yang sama. Hal tersebut disebabkan oleh perbedaan faktor kalibrasi tiap giroskop dan diperlukan penerapan *gain* yang sesuai agar menghasilkan *output* giroskop yang seragam. Gambar 4-17 tersebut juga menunjukkan bahwa jika rotasi diberikan pada salah satu sumbu obyek, giroskop pada sumbu yang lain pun terpengaruh. Hal tersebut disebabkan oleh pemasangan giroskop yang tidak tepat ortogonal pada perangkat IMU yang digunakan untuk pengujian.



Gambar 4-17: Grafik plot data giroskop diperbesar

5 KESIMPULAN

Berikut ini adalah beberapa poin yang merupakan indikator kualitas *software* yang dikembangkan berdasarkan pada serangkaian tahap pengujian yang telah dilakukan.

1. Sistem *software* yang dikembangkan mampu mengolah informasi sensor dalam bentuk teks menjadi gambar obyek bergerak berbentuk animasi 3D. Animasi yang mampu ditampilkan meliputi gerakan translasi dan rotasi obyek dalam sumbu x , y dan z pada kerangka acuan *body*,
2. *Software* yang dikembangkan mampu menerima dan mengolah data yang berasal dari *file* maupun *port serial*,
3. Semakin tinggi frekuensi gelombang sinyal *input* maka semakin kecil pergerakan obyek animasi dibanding-

kan dengan frekuensi gelombang *input* yang lebih rendah pada amplitudo yang sama,

4. *Hardware* IMU harus dikalibrasi dan diberi *filter* terlebih dahulu jika ingin digunakan bersama *software* ini agar hasil yang diperoleh lebih maksimal.

Saran yang akan diajukan sebagai topik penelitian berikutnya, *software* dikembangkan agar dapat menampilkan posisi obyek berdasarkan data yang diterima dari sensor GPS dan optimalisasi *source code* agar pengalokasian *memory* dan waktu eksekusi lebih cepat sehingga meningkatkan tingkat akurasi *software*.

DAFTAR RUJUKAN

- Adiprawita, W., Ahmad, A.S., Sembiring, J., 2007. *Development of AHRS (Attitude and Heading Reference System) for Autonomous UAV (Unmanned Aerial Vehicle)*, Bandung Institute of Technology, Indonesia.
- KNKT, 2008. *Final Report Kecelakaan Pesawat Adam Air PK-KKW Nomor Penerbangan DHI 574*, Dephub KNKT, Jakarta, Indonesia. Dapat diakses pada http://www.dephub.go.id/knkt/ntsc_home/Media_Release/IMR_Adam_Air_PK-KKW.pdf (Januari 2012).
- Shin, E., 2001. *Accuracy Improvement of Low Cost INS/GPS for Land Applications*, UCGE Reports, Calgary.
- Titterton, D., H., Weston., J., L., 2004. *Strapdown Inertial Navigation System-2nd Edition*, The Institution of Electrical Engineer, USA.
- Wiradinata, R., dan Widada, W., 2007. *Development of Inertial Navigation System for Guided Rocket Flight Test*, Proceeding of Seminar Nasional IPTEK Dirgantara XI (SIPTEKGAN), Serpong, Indonesia.
- Wiradinata, R., dan Widada, W., 2008. *Prototype of A Low-Cost Inertial Measurement Unit for Guided Rocket Flight Test*, Proceeding of SEMNAS UTY 4, jogjakarta, Indonesia.

