

VISUALISASI 3D EDR (*EVENT DATA RECORDER*) PADA SEPEDA MOTOR

Nadia Ukhti Dzulhasni^{1*}, Mohamad Safrodin¹, Artiarini Kusuma N¹

¹Teknologi Game, Politeknik Elektronika Negeri Surabaya

Jalan Raya ITS-Kampus PENS Sukolilo, Surabaya 60111

*E-mail : nadiadz109@gmail.com

ABSTRAK

Indonesia memiliki angka kecelakaan lalu lintas yang tinggi, pada 2013 terdapat 100.106 kasus kecelakaan dan 36% nya didominasi oleh pengendara sepeda motor. EDR (*Event Data Recorder*) sebagai alat perekam kejadian atau yang biasa dikenal dengan black box ini, belum menjadi standar keamanan kendaraan bermotor di Indonesia. Pemvisualisasian EDR terutama pada sepeda motor sebagai alat bantu reka ulang kejadian masih sebatas gambar grafik dan data berupa angka. Maka dari itu penelitian ini mengajukan solusi baru untuk visualisasi 3D EDR pada sepeda motor yang bisa membantu mereka ulang kejadian sesuai dengan data dan kondisi yang terjadi untuk membantu pembaca memahami kejadian yang sebenarnya. Data yang digunakan untuk membantu pemvisualisasian bisa dari data EDR ataupun dari data sensor IMU (*Inertial Measurement Unit*) *smartphone* yang terekam. Visualisasi 3D EDR ini bisa diakses secara offline dan dinamis, karena pengguna cukup memasukkan data yang mereka dapatkan sendiri kedalam Visualisasi 3D EDR tanpa harus menganimasikannya secara manual. Gambar visualisasi lingkungan berasal dari Google Maps untuk merealisasikan tempat kejadian. Pengguna juga dapat memilih tipe sepeda motor yang ingin divisualisasikan. Pemvisualisaian ini nantinya bisa membantu kepolisian dalam menangani kasus kecelakaan pada sepeda motor, peningkatan standar keamanan, dan penelitian untuk perkembangan produk motor selanjutnya yang ada di Indonesia.

Kata Kunci: Visualisai 3D, EDR, *smartphone*, IMU

ABSTRACT

Indonesia has a high number of traffic accidents, in 2013 there were 100,106 case and 36% of the precentage are dominated by motorcyclists. EDR (Event Data Recorder) as an event recorder or commonly known as black box, has not become the standard of motor vehicle safety in Indonesia. Visualization of EDR mainly on motorcycles as a tool for recreating events is still limited in graphic images and mostly in numerical data. Therefore this research proposes a new solution for 3D EDR Visualization on motorcycles that can help them re-occurrence events according to the data and conditions that help the reader to understand the actual events. The data used to assist the visualization can be from EDR data or IMU (Inertial Measurement Unit) sensor on smartphone. Visualization 3D EDR can be accessed offline and dynamic, because users simply enter their own data into 3D visualization EDR without have to animate it manually. The environmental visualization image comes from Google Maps to realize the scene. Users can also choose the type of motorcycle that wants to be visualized. This visualization will be able to assist the police in handling accident cases on motorcycles, improving safety standards, and research for the further development of motor products in Indonesia.

Keywords: 3D visualization, EDR, *smartphone*, IMU

PENDAHULUAN

Indonesia dengan populasi penduduk terpadat keempat didunia menjadi sasaran empuk untuk bisnis penjualan transportasi pribadi. Pada tahun 2013 jumlah pengguna kendaraan bermotor mencapai 104 juta unit kendaraan. Angka tersebut didominasi oleh pengguna sepeda motor yang mencapai 84 juta unit. Jumlah pengguna kendaraan bermotor yang tinggi ini tidak lepas dari angka kecelakaan lalu lintas yang ada di Indonesia yang cukup tinggi pula. Dari data yang dikeluarkan *World Health Organization* (WHO) menunjukkan bahwa 36% dari kecelakaan lalu lintas yang menyebabkan kematian di Indonesia didominasi oleh pengendara sepeda motor.

EDR (*Event Data Recorder*) sebagai *black box* yang sudah familiar untuk mereka ulang kejadian kecelakaan, banyak digunakan sebagai alat bukti kejadian. Di negara maju pemasangan EDR sudah merupakan standar keamanan kendaraan bermotor terutama kendaraan bermobil. Namun di Indonesia, alat perekam pada kendaraan bermotor belum menjadi standar keamanan kendaraan, terutama pada sepeda motor yang mempunyai jumlah dominan di masyarakat. Sehingga di zaman serba teknologi ini, perlu adanya penunjang yang bisa membantu masalah standar keamanan dan visualisasi EDR. Maka dari itu visualisasi pembacaan *black box* perlu dibuat dan dikembangkan untuk mempermudah pengembang kendaraan bermotor untuk memberikan produk yang lebih bagus, mempermudah pengguna mengetahui keadaan ketika dia berkendara, dan mempermudah mereka ulang kejadian dengan bantuan visualisasi tersebut.

Beberapa penelitian tentang visualisasi kejadian bermotor atau rekaman kejadian kendaraan sudah sangat banyak. Seperti dalam sebuah studi yang dilakukan oleh Luke Kingsley Bell dan rekan-rekannya dari *University of Cape Town*, Departemen Ilmu Komputer, *Cape Town*, Afrika Selatan yang membuat aplikasi MOVER (*Mobile On-board Vehicle Event Recorder*) yang khusus ditujukan untuk merekam kejadian pada kecelakaan mobil, dengan bantuan data ambang batas dari akselerometer dengan akurasi tinggi. MOVER juga dapat mengulang kejadian sebelumnya ataupun saat ini untuk meningkatkan keselamatan di jalan raya. Studi

kedua adalah penelitian yang berjudul *Context-Aware Big Data Analytics and Visualization for City-Wide Traffic Accidents* yang dilakukan oleh Xiaoliang Fan dan teman-temannya dari Sekolah Ilmu Pengetahuan Alam dan Teknik, *Lanzhou University*, Lanzhou, China. Visualisasi kecelakaan ilustratif ini menganalisis *big data* untuk kecelakaan lalu lintas di seluruh kota, mereka mengusulkan metode analisis kecelakaan dan visual kecelakaan yang bisa memprediksi tempat dan waktu kejadian kecelakaan dengan jenis kecelakaan yang sering terjadi dalam di tempat dan waktu tersebut, diharapkan dengan penelitian mereka ini bisa membantu polisi dalam penyebaran pengamanan lalu lintas dan menginformasikan kepada pengendara mengenai tempat dan waktu yang rawan terhadap kecelakaan. Dan juga penelitian yang dilakukan oleh Taratia Pangayuh K. dari Politeknik Politeknik Negeri Surabaya, Surabaya, Indonesia mengenai rekonstruksi 3D EDR pada sepeda motor telah menggambarkan kejadian sebenarnya namun dalam penelitian ini, tidak semua orang dapat memasukkan data mereka sendiri untuk reka ulang kejadian, penelitian ini ditujukan untuk bukti forensik.

Berdasarkan penelitian yang telah ada sebelumnya, kami menyimpulkan bahwa visualisasi yang jelas sangat membantu pembaca untuk memahami kejadian yang tercatat dalam alat perekam data. Oleh karena itu perlu adanya visualisasi EDR dalam tampilan yang lebih baik seperti visualisasi 3D. Data yang dibutuhkan untuk pemvisualisasian adalah kecepatan, posisi lintang, posisi bujur, dan ketinggian. Data ini adalah serangkaian angka yang mewakili kejadian nyata di kendaraan bermotor. Namun, karena kurangnya aplikasi visualisasi 3D yang bisa menggambarkan kejadian pada EDR, pembaca hanya bisa membayangkan pergerakan yang terjadi tersebut dengan melihat data grafis dari data mentah yang terekam. Sensor yang melekat pada EDR sangat berpengaruh pada detail data yang diambil untuk visualisasi. Berbagai data yang diambil dari pemasangan sensor di setiap bagian kendaraan bisa berdampak pada hasil visualisasi.

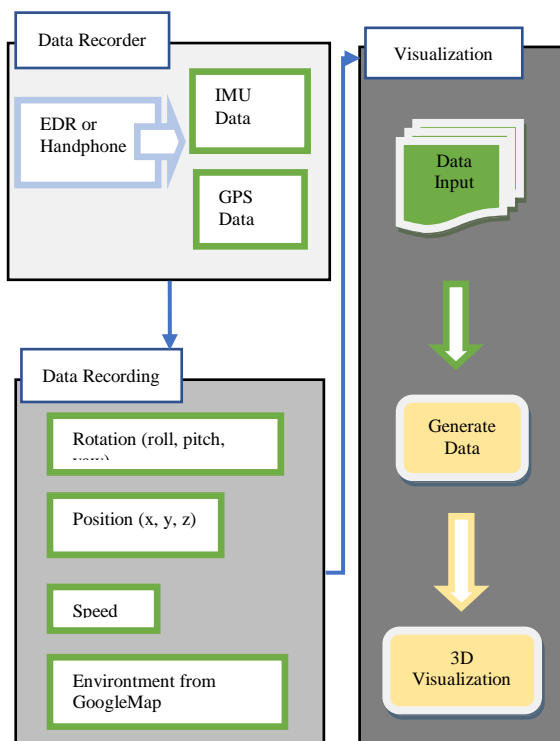
Ide yang kami usulkan untuk masalah visualisasi EDR tersebut adalah membuat pendekatan baru untuk visualisasi EDR secara 3D dengan cakupan kendaraan sepeda motor sebagai objek penelitiannya yang merupakan

kendaraan paling umum di Indonesia. Visualisasi ini bisa diakses secara offline dan mengambil data map secara online. Pengguna juga dapat memasukkan data mereka sendiri untuk memahami kejadian tersebut.

METODE

Metode penelitian ini menggunakan kausal komparatif dengan membandingkan hasil uji coba pada kejadian sebenarnya untuk perekaman data dengan visualisasi yang dianimasikan oleh program visualisasi 3D. Diharapkan dengan membandingkan hasil rekam data mentah dengan hasil visualisasi yang akan dianimasikan sesuai pergerakannya dengan kejadian sebenarnya.

Perancangan sistem visualisasi 3D ini mempunyai beberapa tahapan seperti yang terlihat pada Gambar 1 dimana setiap tahapannya akan dijelaskan seperti berikut:



Gambar 1. Rancangan Sistem

1. Perekam Data

Perekam data berasal dari EDR atau *smartphone* yang memiliki GPS (*Global Positioning System*) dan sensor IMU (*Inertial Measurement Unit*) yang mampu mendeteksi guncangan pada *smartphone* pengguna. Informasi yang bisa tertangkap dari alat-alat tersebut berupa posisi, *roll*, *pitch*, *yaw*,

kecepatan, serta data gambar lokasi yang bisa diambil dari *Google Maps* atau foto tempat kejadian.

2. IMU (*Inertial Measurement Unit*)

IMU atau *Inertial Measurement Unit* adalah alat yang dapat mengukur gaya tertentu seperti percepatan, kecepatan sudut, dan biasanya juga ada yang bisa menghitung medan gaya yang ada disekitar alat tersebut. IMU mengkombinasikan sensor *accelerometer* dan *gyroscope*, biasanya juga ditambah *magnetometers* pada alat yang mempunyai ketelitian lebih baik. IMU data yang diambil untuk visualisasi ini berasal dari *handphone* yang memiliki sensor IMU dengan bantuan aplikasi *Sensorstream IMU+GPS*. Dalam penelitian ini, IMU data berperan sebagai *input* data kecepatan dan rotasi dari pergerakan motor. Kecepatan akan diambil dari sensor *accelerometer* dan rotasi akan diambil dari sensor *gyroscope* dalam *axis x, y, dan z*.

- Aplikasi *Sensorstream IMU + GPS*

Sensorstream IMU+GPS adalah aplikasi yang dibuat oleh *Axel Lorenz* yang menyediakan fasilitas pembacaan sensor IMU dan GPS. Aplikasi yang digunakan untuk membaca sensor tersebut bisa menunjukkan angka pada sensor *accelerometer*, sensor *gyroscope*, sensor medan gaya, GPS, orientasi, percepatan linear, gravitasi, kecepatan rotasi, tekanan, yang semuanya terekam dalam *axis x, y, dan z*, serta temperatur baterai. Aplikasi ini merupakan aplikasi gratis yang bisa diunduh melalui *Play Store* untuk *Android* versi 2.3.3 keatas dan mempunyai fasilitas IMU didalamnya.

3. GPS Data

GPS atau *Global Positioning System* berperan sebagai *input* data posisi yang diambil dari *longitude* dan *latitude* yang terekam. GPS data digunakan untuk memberikan data posisi yang lebih akurat dibanding data *accelerometer*.

4. Proses Visualisasi

Proses Visualisasi ini mempunyai beberapa data yang dibutuhkan seperti:

- Posisi Motor

Data posisi motor yang dimaksud dalam *input data* merupakan posisi motor ketika berpindah dari satu titik ke titik yang lain. Data ini diperoleh dari *latitude* dan *longitude* yang diambil dari GPS.

Pada data yang telah terekam, yang digunakan untuk menggerakkan motor hanya *longitude* dan *latitude*, untuk *altitude* tidak digunakan, karena pergerakan naik turun motor pada visualisasi animasi 3D ini memanfaatkan *pitch* yang direkam oleh *gyroscope*.

- Rotasi Motor (*Roll, Pitch, Yaw*)

Rotasi motor dalam kasus ini adalah guncangan yang diterima motor ketika bergerak. Guncangan pada motor akan ditangkap oleh giroskop dalam bentuk *roll*, *pitch*, dan *yaw*. Pengertian *roll* adalah kemiringan sepeda motor saat motor berputar, *pitch* adalah pergerakan motor atas dan bawah saat bertabrakan, sedangkan *yaw* adalah pergerakan motor yang menekuk ke kanan atau ke kiri saat berbelok.

- Kecepatan (*Speed*)

Kecepatan pada animasi motor bisa didapat dari data EDR maupun GPS. Pada penelitian ini kecepatan motor yang didapat berasal dari sensor akselerometer, sehingga data yang didapat masih berupa percepatan. Percepatan ini harus diubah kedalam bentuk kecepatan dengan rumus sebagai berikut:

$$V_t = V_0 + at \quad (1)$$

Dimana:

V_t = Kecepatan saat waktu tertentu

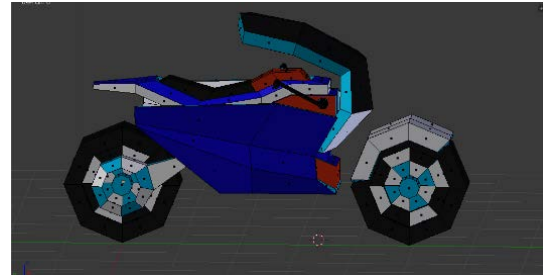
V_0 = Kecepatan awal

a = percepatan

t = waktu

- Visualisasi Motor

Visualisasi motor dari penelitian ini dibuat dari aplikasi *Blender* dengan konsep motor sport 3D. Untuk membuat asset motor seperti Gambar 2 di bawah yang dilakukan pertama adalah *modelling* 3D. Pada tahap *modelling*, motor masih diberi detail permukaan agar menyerupai bentuk sepeda motor yang diinginkan. Pemberian warna dilakukan setelah pemodelan selesai. Warna yang diberikan menggunakan fasilitas material yang ada pada *software Blender* tanpa menggunakan *UV Mapping* karena cukup menggunakan warna dasar sederhana tanpa gradasi.



Gambar 2. Visualisasi Motor

- Lingkungan Visualisasi

Pengambilan gambar lokasi pada visualisasi 3D EDR ini didapat dari gambar secara *online* melalui *Google Maps static*. Gambar yang tampil merupakan gambar tipe *Road Map*. Map yang dihasilkan adalah map statis yang hanya bisa mengambil lokasi dalam satu plane yang besar jangkauannya tergantung besar *zoom*, resolusi atau *size*, *center*, dan *path* yang dipilih. Karena penempatan *center* dan *path* yang dimasukkan kedalam penampilan map sangat berpengaruh pada cakupan luas peta, *zoom* berpengaruh pada kejelasan map, semakin besar *zoom* semakin dekat (terlihat lebih besar) jangkauan yang tampil dan *size* berpengaruh pada kejelasan map yang tampil pada *plane*, sehingga jangkauan map bisa berubah sesuai empat faktor tersebut.

5. Spesifikasi Lokasi Map

Pada map statis yang ada dalam *Unity* spesifikasi lokasi tengah terdapat pada posisi alamat *longitude* dan *latitude* yang dimasukkan pada fungsinya. Besaran *longitude* dan *latitude* yang bisa digunakan terbatas yaitu pada *longitude* besaran yang digunakan dari -180 sampai 180 derajat, sedangkan besaran pada *latitude* -90 sampai 90 derajat. Map yang ada pada *Google Map static* mempunyai *zoom level* yang menentukan resolusi gambar yang ditampilkan. *Zoom level* berkisar 0 (tingkatan terendah *zoom* yang menampilkan seluruh permukaan bumi) sampai +21 (yang menampilkan jalan dan bangunan) yang bisa dilihat jika menggunakan *roadmap view*. Gambaran bangunan akan tampak pada *zoom level* +17. Parameter ukuran gambar *view* dalam hal ini berkaitan dengan letak *center* yang menentukan keseluruhan area dari *map*. Hal ini juga menentukan hasil ukuran *map* dalam *pixel*, ketika dikalikan dengan *scale* dimana *defaultscale* adalah 1. Parameter ukuran dari *Google Static Map API* ditentukan dalam *pixel*. Jika ukuran *mapsize*=200x 200 akan diartikan sebagai 200 *pixel* x 200 *pixel*. Pada monitor LCD komputer, dimana tipe

tampilan 100 *pixel* per inch (ppi), 200x200 pada map akan menjadi sekitar 2 inci setiap dimensi.

Jadi luasan daerah yang digambarkan dalam satu area tergantung pada ukuran map yang diinginkan (*size* dalam *pixel*), *zoom level* (Osampai +21), dan DPI monitor LCD komputer (100x100 *pixel* sama dengan 1 inci setiap dimensi).

- Tampilan Perspektif Visualisasi

Tampilan visualisasi akan dibagi menjadi dua sudut pandang yang menampilkan gambar tampak belakang atas dan gambar samping. Pengambilan gambar kamera yang berbeda bisa memberikan prespektif yang berbeda terhadap pengguna. Penggunaan kamera pada aplikasi ini hanya menggunakan satu kamera yang bisa berpindah tempat kebelakang motor dan samping motor. Pergerakan kamera menggunakan program agar kamera bisa bergerak ketempat yang diinginkan.

```
transform.position = Vector3.Lerp(transform.position, currentMount.position, speedFactor);
transform.rotation = Quaternion.Slerp(transform.rotation, currentMount.rotation, speedFactor);
```

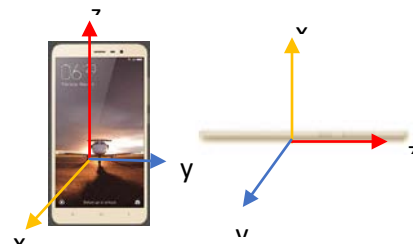
Program diatas merupakan pemanfaatan perpindahan posisi dan rotasi yang bisa digunakan pada kamera. Kamera yang berpindah ini bisa digunakan asalkan ada titik tujuan sebagai tempat berpindah kamera.

- Parsing Data

Penerapan parsing data pada penelitian ini menggunakan data dari aplikasi *Sensor Stream IMU+GPS* sebagai data masukan untuk visualisasi motor. Data yang diambil berupa file .csv yang bisa dibaca di Unity. Pergerakan yang dihasilkan dari data yang terekam akan menghasilkan gerakan yang terlihat seperti animasi perpindahan suatu objek. Besarnya perpindahan benda dari angka data awal menuju data yang terakhir ditentukan dengan besar skala *plane* dan resolusi map yang digunakan. Pada penelitian kali ini menggunakan map dengan resolusi 1000x1000 pixel dengan besar plane berskala 50.

Pembacaan data pada unity bisa menggunakan sistem library yang sudah ada dalam unity yang berupa pembacaan input dan output yaitu:

```
using System.IO;
```



Library unity diatas memberikan fasilitas kepada programmer untuk bisa membaca data dari luar unity untuk bisa dibaca sebagai input dan menulis data dari unity untuk disimpan kedalam bentuk format yang lain. Pembacaan data pada unity salah satunya bisa menggunakan:

```
filelines = File.ReadAllLines (filepath);
```

Setelah pembacaan dilakukan perlu adanya pemecahan data sebagai pembeda angka satu dengan yang lain dengan menggunakan:

```
parts = filelines[i].Split ("","[0]);
```

Setelah data dipecah berdasarkan simbol kode yang telah didefinisikan, data tersebut di kelompokkan dan diubah dari bentuk *string* kedalam bentuk *float* agar bisa dibaca oleh fungsi selanjutnya sebagai informasi posisi. Pengelompokan data ini dibutuhkan untuk mengetahui fungsi data yang tercatat dengan kebutuhan visualisasi. Untuk data yang tidak diperlukan maka data hanya perlu dipecah saja tanpa dimasukkan kedalam fungsi selanjutnya.

Tabel 1. Tabel Kode

Kode	Arti
1	GPS
8	GPS time in Epoch
81	Derajat Orientasi
82	Linear Acceleration
83	Gravitation
84	Rotation Vector
85	Preasure
86	Battery temperature

Tabel 1 merupakan simbol kode yang digunakan untuk memisah data yang terekam untuk bisa dibaca dalam program. Pembacaan parsing selesai jika data yang dipecah sudah sesuai dengan panjang data yang terekam. Panjang data tiap barisnya memiliki jumlah yang berbeda.

HASIL DAN PEMBAHASAN

Mengingat bahwa sumbu yang ada pada *smartphone* berbeda dengan sumbu yang ada

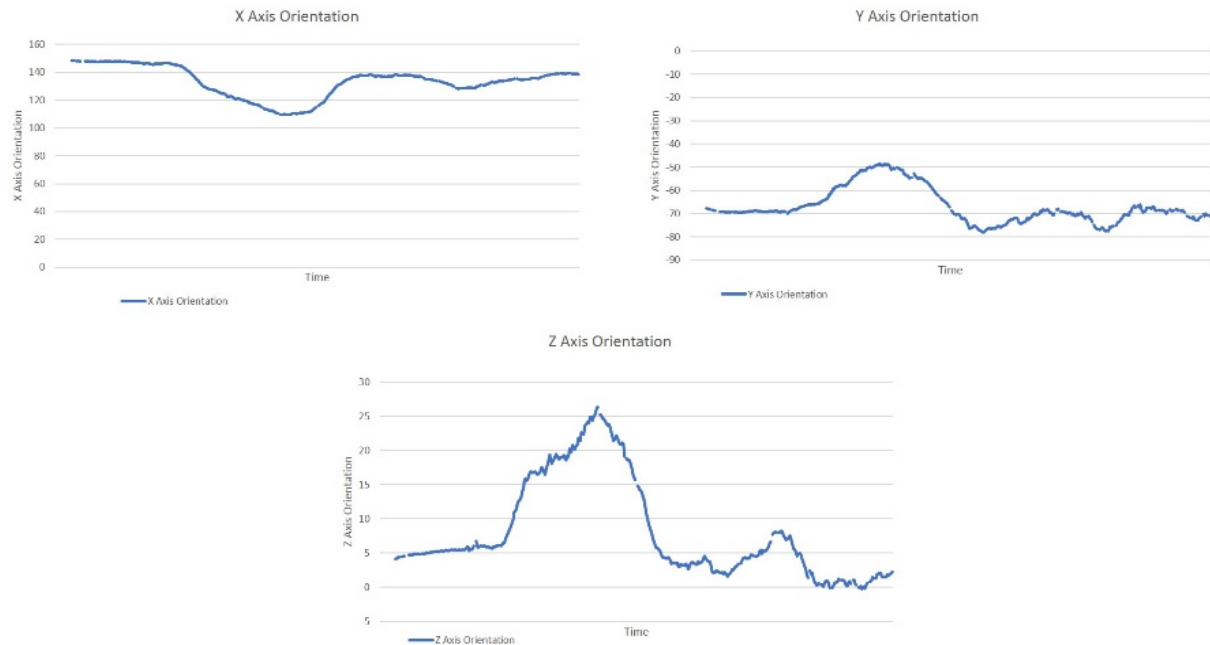
di *Unity*, maka penyesuaian data yang dilihat juga berbeda.

Seperti yang terlihat pada Gambar 3. mengenai sumbu sensor pada *smartphone* sebagai perekam data. Tampak bahwa sumbu x pada sensor menuju ke arah layar depan, sumbu y mengarah ke arah samping, dan sumbu z mengarah ke arah atas *smartphone*. Penentuan sumbu pada *smartphone* ini dilakukan percobaan berulang-ulang untuk memastikan perubahan data mana yang berubah ketika *smartphone* dilajukan kedepan-belakang, samping, dan atas-bawah.

Gambar 3. Sumbu Sensor *Smartphone*

1. Data Orientasi

Pengujian perekaman data derajat orientasi ini memperhatikan sudut awal sebagai penentu sudut nol nya. Sumbu yang ada pada derajat orientasi ini memiliki sudut maksimum dan minimum yang berbeda. Untuk menentukan sudut maksimum dan minimum



Gambar 4. Grafik Data derajat Orientasi

2. Pembacaan Parsing Data

Pengujian pembacaan data csv ini memperhatikan dan memastikan bahwa data yang ingin di visualisasi masuk kedalam *Unity*. Ketika data sudah dipastikan masuk, monitoring selanjutnya adalah pembacaan data menggunakan *parsing* data sudah bisa berjalan sesuai program yang dituliskan. Selanjutnya ketika data sudah di *parsing*, data yang dikelompokkan dipastikan untuk bisa masuk

dilakukan beberapa percobaan seperti memiringkan *smartphone* dari prespektif x, y, dan z. Sumbu x memiliki derajat maksimum 360° dan minimum -360° . Sumbu y memiliki derajat maksimum 180° dan minimum -180° . Sumbu z memiliki derajat maksimum 90° dan minimum -90° .

Pengujian yang dilakukan seperti gambar 4. merupakan data sepuluh detik pertama. Dari data yang terekam bisa dilihat bahwa sensor orientasi ini bisa terekam juga bisa tidak terekam, sehingga ada sepersekian *nanosecond* data derajat orientasi yang hilang dari rekam data. Untuk data setiap sumbu memiliki arti sendiri yaitu sumbu x sebagai *yaw*, sumbu y sebagai *pitch*, dan sumbu z sebagai *roll*. Dari data di bawah, angka derajat orientasi awal akan dijadikan titik nol perotasian, sehingga tidak membuat visualisasi sepeda motor bergerak dengan rotasi yang amat besar.

ke dalam fungsi selanjutnya untuk dimasukkan dan dihitung kedalam bagian sesuai fungsi datanya.

Pengujian ini mengandalkan fungsi log yang ada pada *Unity*. Seperti pada Gambar 5 data yang telah didapat sudah bisa masuk ke dalam fungsi selanjutnya dan sudah bisa menunjukkan data yang terbaca sudah dipecah sesuai fungsinya.


```

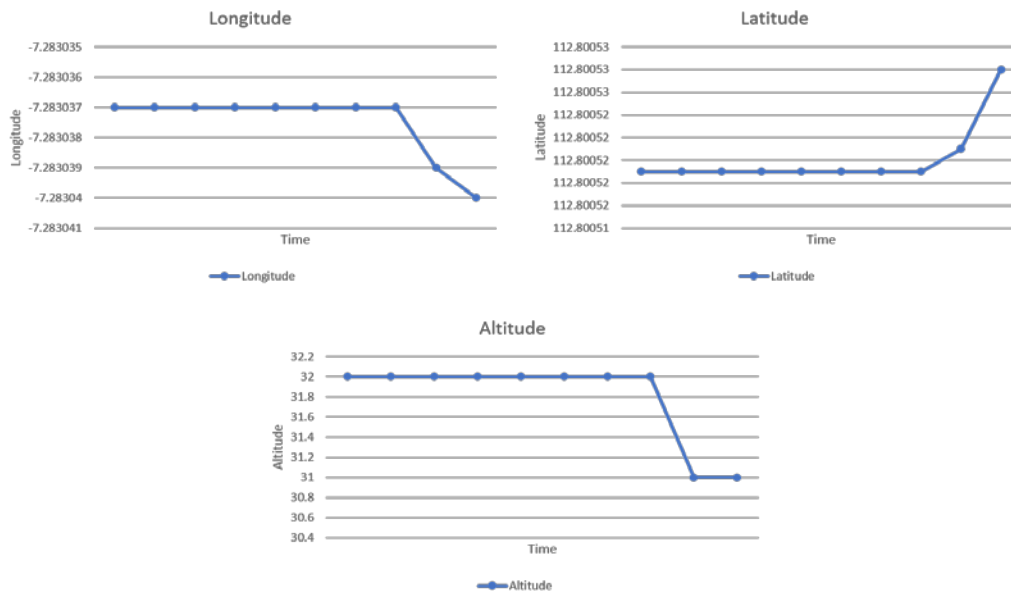
Console
Clear Collapse Clear on Play Error Pause
! Lokasi x= 0.2839999 y= 0 z= 9.711
UnityEngine.MonoBehaviour:print(Object)
! Rotasi x= 4.384003 y= 3.31 z= 0.187
UnityEngine.MonoBehaviour:print(Object)
! Lokasi x= -0.07000005 y= 0 z= 9.692
UnityEngine.MonoBehaviour:print(Object)
! Rotasi x= 4.352997 y= 3.29 z= 0.113
UnityEngine.MonoBehaviour:print(Object)
! Lokasi x= -0.645 y= 0 z= 9.615
UnityEngine.MonoBehaviour:print(Object)
! Rotasi x= 4.257996 y= 3.259 z= -0.004
UnityEngine.MonoBehaviour:print(Object)
! Lokasi x= -1.631 y= 0 z= 9.682
UnityEngine.MonoBehaviour:print(Object)
! Rotasi x= 3.858994 y= 3.234 z= -0.417
UnityEngine.MonoBehaviour:print(Object)
! Lokasi x= -2.905 y= 0 z= 9.529
UnityEngine.MonoBehaviour:print(Object)
! Rotasi x= 3.664993 y= 3.222 z= -0.74
UnityEngine.MonoBehaviour:print(Object)

```

Gambar 5. Log Program

3. Data Posisi

Pengujian ini memperhatikan seberapa sering data GPS yang tercatat dalam aplikasi mengingat bahwa ada dua pilihan pencatatan yaitu bisa menggunakan UDP dan atau hanya *stream* data pada *SD Card*. Pencatatan posisi



Gambar 6. Grafik Data Posisi

4. Visualisasi Gerakan

Pengujian visualisasi ini memperhatikan ketepatan visualisasi terutama pada gerak *roll*, *pitch*, dan *yaw* yang paling terlihat pada kasus kecelakaan. Pengujian ini menggunakan kendaraan sepeda untuk uji kejadiannya dikarenakan lebih ringan untuk digunakan reka adegan seperti kecelakaan yang sebenarnya. Monitoring yang dilakukan adalah dengan melihat video kejadian sebenarnya dengan visualisasi 3D yang bergerak sesuai data masukan, dimana video kejadian tersebut akan

ini diperlukan untuk menampilkan map sebagai tempat kejadian. Awal monitoring GPS dilakukan dengan menunggu GPS untuk bisa menemukan lokasi pertama ketika memulai aplikasi, selanjutnya jika GPS sudah terintegrasi dengan aplikasi maka pengujian sudah siap dilakukan.

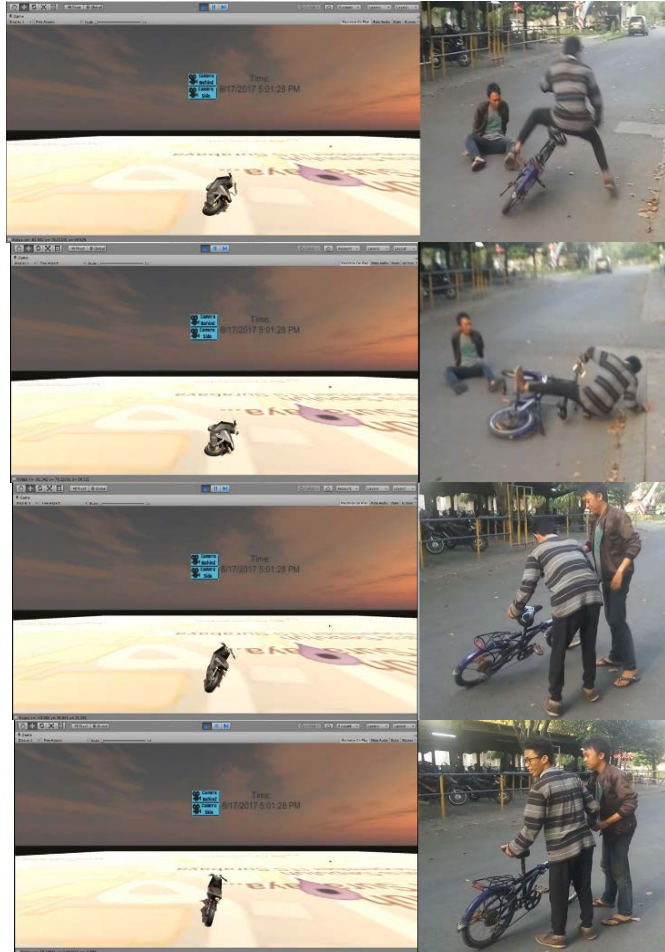
Pengujian ini dilakukan disekitar titik pusat map yang telah di-*setting* pada *latitude* -7.2828213 dan *longitude* 112.800457 dengan alamat Jalan Teknik Komputer IV, Sukolilo, Surabaya. Sesuai Gambar 6 yang menunjukkan *longitude* dan *latitude* posisi bergantinya sepeda motor sangat kecil sekali. Dengan melihat hasil data yang terekam, data GPS terekam setiap detik. Namun ketika sudah masuk pada menit kedua data GPS yang terekam tidak se-stabil seperti menit pertama. Data dibawah diambil pada sepuluh detik pertama, sehingga perubahan posisi sangat amat kecil yaitu 10^{-5} derajat dari posisi awal.

disimpulkan dengan mengambil 3 sampai 5 *frame* setiap pengambilan kejadian.

Untuk pengujian *roll*, *pitch*, dan *yaw* yang terekam, sudut nol pergerakan merupakan sudut awal yang terekam pada data detik pertama, sehingga perlu adanya pengecekan *raw* data untuk mengoptimalkan hasil yang diinginkan. Perlu juga filter antara baris data tanpa GPS dengan baris data yang ada data GPS, karena data GPS terekam setiap detik maka hal ini memudahkan pemvisualisasian dan perhitungan.

Gambar 6 merupakan pengujian kecelakaan yang berakibat pada *roll* kekanan dimana pengujian ini mengambil 4 frame posisi sebagai alat bukti kejadian riil dengan visualisasi 3D. Frame pertama menunjukkan sepeda mulai goyah dan condong kekanan, frame kedua merupakan jatuhnya pengendara

akibat tumbukan dengan korban, frame ketiga dan keempat sepeda mulai diangkat untuk dikembalikan kekeadaan semula. Dari visualisasi 3D dan rekaman data riil yang diuji menunjukkan bahwa pergerakan *roll* kekanan berhasil.



Gambar 6. Pengujian Roll Kanan

Data yang terekam pada pengujian ini bisa dilihat pada Gambar 7 terlihat bahwa perbedaan rotasi yang terjadi pada raw data dan visualisasi di *Unity*. Selisih ini sebesar

$$\text{rotasi } x_{Unity} = \text{derajat}_{orientasi } y_0 - \text{derajat}_{orientasi } y_{(n-1)} \quad (2)$$

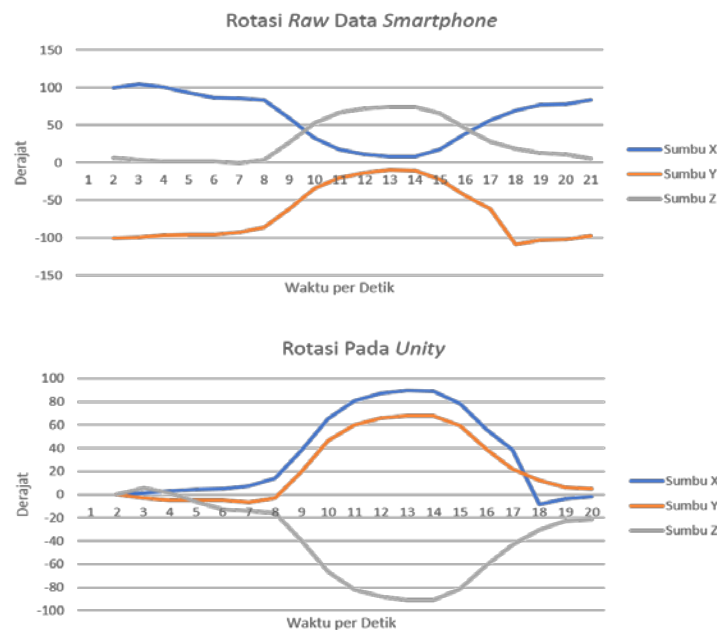
$$\text{rotasi } y_{Unity} = \text{derajat}_{orientasi } z_0 - \text{derajat}_{orientasi } z_{(n-1)} \quad (3)$$

$$\text{rotasi } z_{Unity} = \text{derajat}_{orientasi } x_0 - \text{derajat}_{orientasi } x_{(n-1)} \quad (4)$$

Rumus diatas digunakan pada semua rotasi yang digunakan untuk visualisasi pada *Unity*. Dan perlu diingat bahwa sumbu x pada *Unity* adalah sumbu y pada *smartphone*, sumbu y pada *Unity* adalah sumbu z pada

nilai data pertama yang terekam pada *Unity*. Berikut merupakan perhitungan untuk mencari rotasi pada *Unity*:

smartphone, dan sumbu z pada *Unity* adalah sumbu x pada *smartphone*. Maka dari itu untuk visualisasi 3D ini perlu adanya pengecekan, pengelompokan, dan *filtering* dari raw data yang didapat dari aplikasi sensor.

Gambar 7. Data Selisih *Roll* Kanan Raw Data dan Visualisasi

SIMPULAN DAN SARAN

Indonesia tercatat memiliki 84 juta unit pengguna sepeda motor pada tahun 2013 dan terus meningkat pada tahun 2017. EDR sangat akrab dengan alat perekam data kecelakaan yang banyak digunakan sebagai bukti kejadian kecelakaan. Namun di Indonesia, perekam kendaraan bermotor belum menjadi standar keselamatan kendaraan, terutama pada sepeda motor. Jadi dalam perkembangan teknologi yang pesat ini, perlu ada dukungan yang bisa membantu masalah keamanan standar dan visualisasi EDR. Oleh karena itu, visualisasi EDR harus dikembangkan agar lebih mudah bagi semua masyarakat.

Visualisasi yang jelas sangat membantu pembaca untuk memahami kejadian yang tercatat dalam alat perekam data. Oleh karena itu perlu untuk memvisualisasikan EDR dalam tampilan yang lebih baik seperti visualisasi 3D. Visualisasi EDR khususnya untuk sepeda motor yang merupakan kendaraan paling umum di Indonesia merupakan hal yang penting. Data visualisasi membutuhkan data kecepatan, garis lintang, garis bujur, dan rotasi pengendara (*roll, pitch, yaw*).

Aplikasi dalam penelitian ini menggunakan dua metode via offline dan online. Pengguna juga bisa memasukkan data mereka sendiri untuk memahami kejadian yang akan dipaparkan dalam model 3D. Dengan bantuan aplikasi Sensorstream IMU +

GPS pada *smartphone* yang bisa digunakan untuk merekam kejadian saat berkendara sebagai ganti alat EDR. Pengguna bisa memasukkan data tersebut ke aplikasi visualisasi 3D untuk sepeda motor dan data tersebut akan terbaca menjadi input animasi pergerakan sepeda motor.

Kesimpulan dari penelitian ini yang pertama adalah hasil data yang didapat sangat berpengaruh dengan jalannya visualisasi. Sehingga jika data yang didapat kurang sesuai dengan yang dibutuhkan dan satuan yang didapat juga kurang diketahui maka perlu adanya analisa lebih untuk data yang tercatat. Yang kedua, penentuan seberapa sering data yang tercatat menurut satuan waktu juga berpengaruh pada banyaknya data yang tercatat. Ketiga, sumbu rotasi sensor pada *smartphone* berbeda dengan sumbu rotasi pada *Unity*. Pada *smartphone* sumbu x merupakan sumbu z pada *Unity*, sumbu y merupakan sumbu x pada *Unity*, dan sumbu z merupakan sumbu y pada *Unity*. Meskipun berbeda namun tetap menunjukkan visualisasi yang sama dengan memasukkan *raw* data yang ada kedalam sumbu rotasi *Unity* yang merepresentasikan rotasi yang sama dengan kejadian riil. Keempat, peletakan *smartphone* saat pengujian sangat berpengaruh terhadap data sudut orientasi mana yang akan dimasukkan kedalam sumbu rotasi pada *Unity*.

Saran untuk penelitian selanjutnya adalah yang pertama perlu adanya penelitian

untuk perekaman data menggunakan *smartphone* yang bisa digunakan oleh segala jenis sensor yang tertanam dalam semua jenis *smartphone*. Selanjutnya, perlu adanya penelitian pembuatan *terrain* sesuai *Google Map* secara gratis yang bisa digunakan oleh banyak masyarakat. Yang terakhir adalah perlu adanya pembuatan plugin tambahan fungsi pengubahan titik pusat objek 3D yang dibuat dari *software* luar yang dimasukkan dalam projek *Unity*.

DAFTAR PUSTAKA

Agus, IG K. W. 2015 Sistem Informasi Trafik Lalu Lintas Cerdas di Bali, Denpasar: Universitas Udayana Denpasar.

Fan, X., Baoqin He, Patrick Brezillon. 2017. *Context-Aware Big Data Analytics and Visualization for City-Wide Traffic Accidents*. Lanzhou University. China.

Hynd, D., Mike McCarthy, 2014. *Study on The Benefits Resulting from The Installation of Event Data Recorders*. Project Report PPR707. European Union.

Kantor Kepolisian Republik Indonesia
Kingsley, L., Bell, Francois Mouton, Tommie Meyer. 2017. *Mobile on-board vehicle event recorder: MOVER*, Cape Town, South Africa.

Pangayuh, T.K. 2016. *3D Data Reconstruction of Motorcycle's EDR (Event Data Recorder)*. Politeknik Elektronika Negeri Surabaya, Surabaya.

World Health Organization. 2010. Indonesia