

APLIKASI PENGIDENTIFIKASIAN KECELAKAAN KENDERAAN BERMOTOR MENGGUNAKAN INTERNET OF THINGS (IoT)

Holder Simorangkir¹, Malabay², Andrew Yehezkiel Patasik³

^{1,2,3}) Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Esa Unggul

Jalan Arjuna Utara No.9, Kebon Jeruk, Jakarta Barat

Surel : holder@esaunggul.ac.id¹; malabay@esaunggul.ac.id²,
andrew.yehezkiel17@gmail.com

Abstrak

Tingkat pertambahan kendaraan bermotor dengan pertambahan jalan tidak berimbang yang mengakibatkan sering terjadi kemacetan dan bahkan kecelakaan. Faktor penyebab kecelakaan tersebut adalah para pengendara merasa lelah selama bekerja, kebutuhan waktu yang mendesak atau pengendara yang lain tidak sadar dalam mengendarai kendaraannya yang mengakibatkan orang lain berdampak. Berdasarkan informasi tersebut perlu dirancang sebuah sistem yang berbasis Internet Of Things (IoT), serta komponen-komponen yang diperlukan dalam merancang Prototype tersebut. Dengan percobaan pengujian sistem pintar ini memberikan akurasi 73% dengan kecepatan memberikan respon ke pos penanggulangan kecelakaan adalah lebih kurang 6 detik dan tempat kejadian dapat dengan segera diketahui, agar pengendara dapat segera tertolong dari kondisi tersebut.

Kata Kunci : *Prototype, Kecelakaan, Sistem Pintar, Pengujian, Internet of Things.*

1. PENDAHULUAN

Tingkat pertumbuhan kendaraan bermotor dengan jalan raya adalah tidak sebanding, ini dipicu dengan mudahnya mendapat pinjaman untuk angsuran kendaraan bermotor tersebut. Pertumbuhan jalan raya hanya 1% sementara kendaraan bermotor antara 10% sampai 15% setiap tahunnya. [1] Inilah faktor yang memberi dampak kemacetan dan setiap harinya pelaku kegiatan membutuhkan waktu yang tepat sampai ditujukan dengan intensitas konsentrasi yang penuh mengakibatkan terjadinya kelelahan dan berdampak kepada kecelakaan. Pengendara lupa bahwa kecelakaan mengintai setiap saat, yang dapat mengakibatkan kerugian harta benda bahkan juga meninggal dunia.

Pada Undang-Undang RI Nomor 22 Tahun 2009, Pasal 1, Tentang Lalu Lintas Dan Angkutan Jalan, tertulis bahwa Kecelakaan Lalu Lintas (LAKA) adalah “ Suatu peristiwa di jalan yang tidak diduga dan tidak disengaja, melibatkan kendaraan dengan atau tanpa pengguna jalan lain, yang

mengakibatkan korban manusia dan atau kerugian harta benda”[2].

Dari informasi LAKA yang terjadi, ada rata-rata 3 orang meninggal setiap jam akibat LAKA jalan raya. Jumlah korban jiwa ini adalah keterlambatan memberikan pertolongan pertama dengan cepat. Dalam meminimalisir resiko dari dampak kecelakaan tersebut, perlu dirancang sebuah sistem yang berbasis *Internet Of Things* (IoT). Sistem ini nantinya dapat menginformasikan terjadi kecelakaan pada Pusat LAKA secara respon time yang cepat serta memberitahu tempat kejadian LAKA dengan menggunakan *Global Positioning System* (GPS) [3].

Penelitian sebagai acuannya adalah “ *Car Accident Detection and Notification System Using Smartphone*” [4], dan “ *Rancang Bangun Smart Vehicle untuk Mendeteksi Dini Kecelakaan dan Keadaan Darurat*”[5]. Dalam penelitian tersebut, meringankan resiko yang disebabkan kecelakaan lalu lintas, memanfaatkan komponen-komponen elektronika seperti microcontroller, sensor accelerometer dan juga fitur pada *smartphone*.

Kekurangan dari penelitian tersebut adalah insiden yang terjadi pada berhenti mendadak akan memberikan informasi ke Pusat LAKA secara otomatis dan menunjukkan tempat LAKA

2. METODE PENELITIAN

Revolusi Industri 4.0 ini adalah Teknologi Informasi dan Komunikasi yang dimanfaatkan sepenuhnya guna mencapai efektif dan efisiensi dalam pemecahan masalah dan menghasilkan model bisnis yang berbasis digital.

2.1 Faktor Pendukung

Konsep pengujian benturan yang dilakukan adalah pengidentifikasi berdasarkan perubahan pada percepatan objek yang dilakukan secara tiba-tiba dalam ukuran g-force yang besar, umumnya 4G [6]. Nilai akselerasi ini digunakan sebagai dasar untuk mengidentifikasi kejadian tabrakan, perekam data pada sumbu x dan y dari sensor akselerometer [7]. Parameter yang dibutuhkan adalah penentuan titik lokasi kejadian yang dapat diketahui melalui implementasi GPS dengan mengekstrak variabel *latitude* dan *longitude*.

2.2 Metode Prototype

Pada metode *prototype* ini adalah salah satu metode Siklus Hidup Sistem yang didasarkan pada konsep model kerja (*working model*) yang dikembangkan menjadi sistem final. [8]

Tahapan Metode Prototype

Metode *Prototype* sering disebut *Rapid Application Development* (RAD) karena praktis dan dapat dilakukan untuk desain sistem.



Gambar 1, Tahapan Metode Prototype

Algoritma Tahapan Prototype

Tahap 1, Pengumpulan Kebutuhan

Tahap 2, Membangun Prototyping

Tahap 3, Evaluasi Prototyping . Dilakukan evaluasi ditahap 1 dan 2, bila memenuhi kriteria maka dilanjut ke

tahap 4, bila belum dilakukan perbaikan pada tahap sebelumnya.

Tahap 4, Pengkodean Sistem

Tahap 5, Pengujian Sistem . Pengujian ini menggunakan *White Box* dan *Black Box*.

Tahap 6, Evaluasi Sistem

Dilakukan pengevaluasian sistem, apakah hasil yang diberikan sistem sesuai dengan kebutuhan?, bila belum, dilakukan perbaikan atau penyempurnaan di tahap ke-4 dan 5, agar aplikasi ini dapat menyelesaikan masalah yang dihadapi secara berkelanjutan.

Tahap 7, Penggunaan Sistem.

Pada tahapan ini sistem telah dapat difungsikan.

3. HASIL DAN PENGUJIAN

Pada tahap ini dilakukan pengumpulan data yang sesuai dengan tujuan dalam pembuatan sistem, seperti perangkat lunak dan perangkat keras yang diperlukan.

Tahap 1, Pengumpulan Kebutuhan

Alat pengidentifikasi tabrakan kendaraan bermotor ini dibangun berbasis IoT dan diperlukan karakteristik dari komponen perangkat keras dan perangkat lunaknya

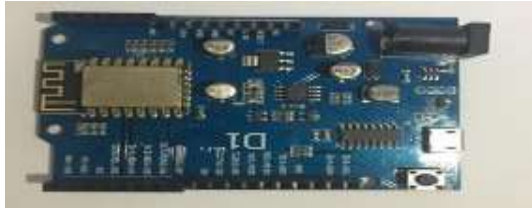
Tabel 1, Komponen Hardware

Karakteristik	
Hardware	Microcontroller : Wemos D1 R1 Mini
	Sensor Modul : MPU-6500 Sensor Accelerometer Ublox Neo-6M GPS Sensor
	Perangkat Mobil : Smartphone Android
	Kapasitas Penyimpanan : > 100MB free
	Komponen Pendukung : 30 Cm, Breadboard, Kabel Micro USB
	Sumber Daya Listrik : Power Bank, Laptop

Penjelasan komponen Hardware

1. Microcontroller : Wemos D1R1 Mini

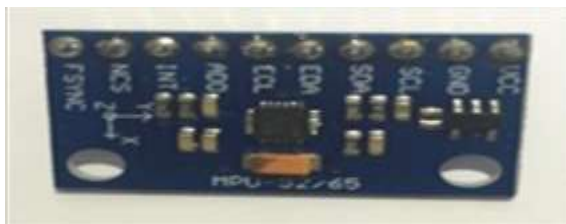
Microcontroller ini berbasis ESP8266, memiliki kelebihan yang didukung oleh fitur konektivitas jaringan *wireless* berbasis IoT dan microcontroller ini memiliki processor 32-bit.



Gambar 2, Wemos D1 R1 Mini

2. Sensor Modul MPU-6500, Sensor Accelerometer

Fungsi sensor accelerometer ini memantau gerak 6 sumbu, yaitu 3-sumbu *gyroscope* dan 3-sumbu accelerometer dan dilengkapi dengan interface jenis I2C yang fungsinya adalah untuk menerima langsung input eksternal I2C driver. Fitur sensor accelerometer ini diprogram secara bebas dalam jangkauan yang dimulai dari $\pm 2g$, $\pm 4g$, $\pm 8g$ dan, $\pm 16g$ [9].



Gambar 3, Sensor Accelerometer

3. Ublox Neo-6M GPS Modul

Komponen ini adalah modul GPS dari Ublox, berfungsi mengetahui posisi (koordinat) yang dibantu dengan satelit GPS. Modul GPS ini untuk koneksi *serial TTL* sebagai *output* dengan 4 buah pin yaitu TX, RX, VCC, dan GND. Chip ini dapat menerima data lokasi dari 22 satelit dengan 50 *channel* yang tingkat *sensitifitas* sekitar -161 dB.



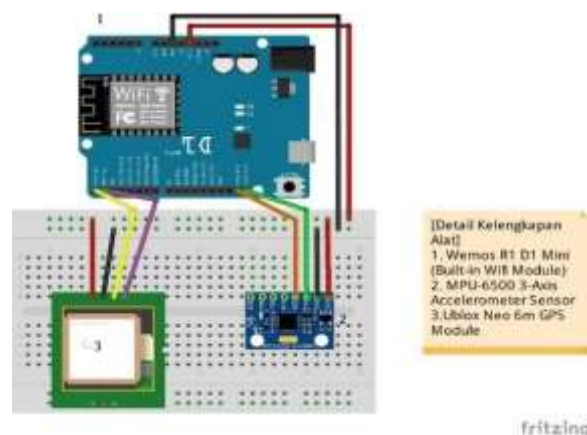
Gambar 4, Ublox Neo-6M GPS Modul

Tabel 2, Komponen Perangkat Lunak

Karakteristik	
Perangkat lunak	Sistem Operasi Perangkat Mobile : Android API 4.4 (Jelly Bean)
	Server : Eclipse Mosquito MQTT Broker 64 bit
	Database: SQLite Database, DB Browser For SQLite V. 3.11.2

Tahap 2, Membangun Prototyping

Tahap ini membangun prototype, fokus pada masalah yang dihadapi, seperti jenis *input*, bentuk *proses* dan jenis *output* yang dibutuhkan. Output yang dibutuhkan dalam masalah ini adalah diagram rangkaian circuit yang diintegrasikan dengan MPU-6500 dan Ublox Neo-6M serta microcontroller Wemos D1 R1 Mini.



Gambar 5, Rancangan Circuit Prototype

Nilai accelerometer dibaca melalui kabel serial yang terhubung pada port SCL dan SDA. Nilai GPS dibaca melalui port TX dan RX pada microcontroller. Bentuk Prototype yang dihasilkan adalah sebagai berikut :



Gambar 6, Prototype Obyek

Tahap 3, Evaluasi Prototyping

Untuk dapat dilanjut ke tahap 4, perlu diperiksa kondisi prototype yang dibuat, bila belum memenuhi kriteria dapat kembali ke tahap-1 dan 2, karena menjadi penentu ke proses selanjutnya, memberikan hasil yang sesuai dengan yang diinginkan secara berkelanjutan.

Tahap 4, Pengkodingan Sistem

Untuk dapat dilakukan pengkodingan diperlukan flowchart. Microcontroller menerima input dari GPS dan accelerometer. Bila nilai accelerometer di atas threshold merupakan indikator dan akan memberikan signal untuk notifikasi ke smartphone user.

Langkah-langkah dari algoritma di atas adalah :

Langkah 1 : Akses ke Hotspot atau wifi

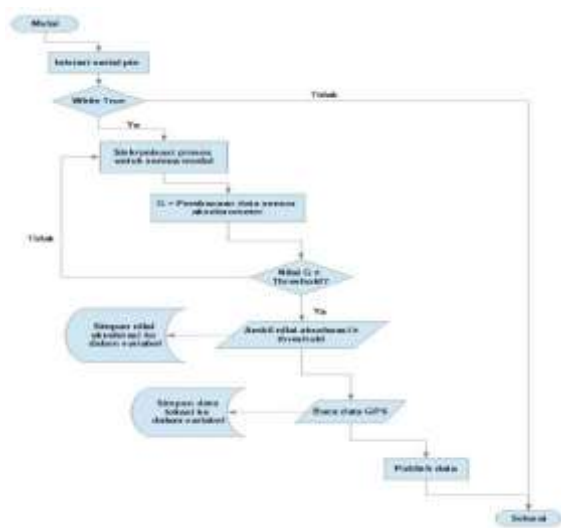
Langkah 2 : Mengkoneksi ke Server

Langkah 3 : Membaca accelerometer untuk sumbu-x dan sumbu-y

Langkah 4 : Membaca posisi latitude dan Longitude

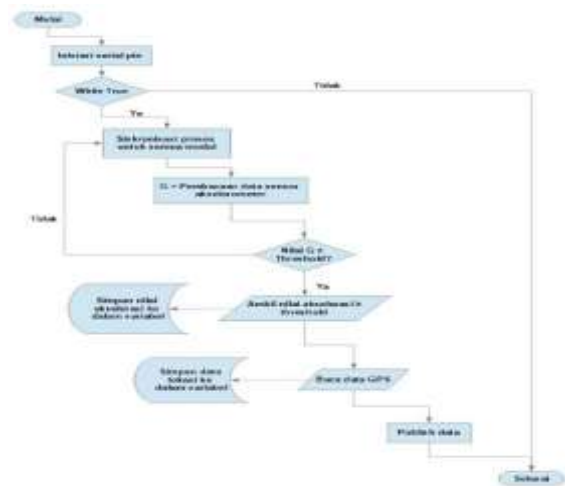
Langkah 5 : Bila kondisi accelerometer $\geq$ threshold, maka simpan nilai accelerometer dan g

Langkah 6 : Mengirimkan data ke Server, kembali ke tahap 2



Gambar 7, Flowchart Diagram Perangkat

Sedangkan untuk aplikasi adalah seperti pada flowchart tersebut.



Gambar 8, Flowchart Diagram Aplikasi

Algoritma dari sistem aplikasi ini, adalah :

Langkah 1: Dapat mengkoneksi ke wifi atau hotspot

Langkah 2: Terhubung dan subscribe ke MQTT Broker.

Langkah 3: Bila data diterima, menampilkan data accelerometer yang sesuai dengan format dan Google Maps menampilkan lokasi LAKA dan langsung disimpan di database, kemudian menampilkan notifikasi darurat dan kembali ke langkah 2.

Tahap 5, Pengujian Sistem

Ditahap ini dilakukan simulasi seperti kejadian yang sebenarnya, menggunakan mobil-mobilan yang memiliki remote control dan mampu membawa peralatan sesuai dengan yang direncanakan, dapat dijalankan menggunakan remote control.

Benda yang ditabrak adalah sebuah obyek yang cukup berat dan dapat berhenti bila terjadi tabrakan seperti sebenarnya.



Gambar 9, Simulasi Tabrakan/Benturan

Simulasi tabrakan, Prototype bergerak dari titik A ke titik B sejauh sekitar 3 meter, dengan rata-rata kecepatan = 1,083 m/s.

Output yang dihasilkan dari respon sensor tertangkap pada baud rate 9600 bps.

Tabel 3, Parameter Pengujian

	Kondisi	Nilai Sensor	Metode Pengujian	Jumlah Pengambilan Data
Data Sample Threshold	Berjalan	• X-Axis Nilai Positif • Y-Axis Nilai Positif	• Pengamatan Langsung • Membaca output Serial Monitor	10x
	Berhenti Mendadak	• X-Axis Nilai Negatif • Y-Axis Nilai Negatif		25x
	Tabrakan			30x
	Total			60x
Data Uji / Data Latih	Berjalan	• X-Axis Nilai Positif • Y-Axis Nilai Positif	• Pengamatan Langsung • Membaca output Serial Monitor	15x
	Berhenti Mendadak	• X-Axis Nilai Negatif • Y-Axis Nilai Negatif		15x
	Tabrakan			15x
	Total			45x

Pengambilan data pertama dipastikan dalam rentang nilai threshold pada setiap kondisi obyek. Pengambilan data kedua dipastikan untuk menguji keakurasian pada nilai threshold yang dihasilkan.

A. Pengujian Tanggap Accelerometer Untuk Kondisi Berjalan



Gambar 10, Hasil Sensor Accelerometer dari Obyek Waktu Jalan

Dari 10 kali pengujian pada kondisi berjalan dan pada accelerometer diberi rentang grafik untuk sumbu-X : $-2,16G < X < 0,96G$, hasil uji ada di nilai $-0,03G$, sedangkan pada sumbu-Y: $-1,21G < Y < 1,8G$, Hasil uji ada di nilai $0,04G$. Informasi ditunjukkan pada kedua sumbu di bawah nilai threshold dengan skala 4G.

B. Pengujian Tanggap Accelerometer Pada Mendadak Berhenti



Gambar 11, Hasil Sensor Accelerometer dari Obyek Mendadak Berhenti

Dari 25 kali pengujian pada kondisi mendadak berhenti, accelerometer diberi rentang grafik untuk sumbu-X : $-3,29G < X < 1,25G$, hasil uji ada di nilai $0,11G$, sedangkan pada sumbu-Y : $-1,32G < Y < 1,45G$, hasil uji ada di nilai $-0,04G$. Informasi ditunjukkan pada kedua sumbu di bawah nilai threshold dengan skala 4G. Grafik yang dihasilkan, bentuk palung untuk beberapa saat di detik ke-3 dan ke-4 dan setelah itu kembali ke posisi normal. Informasi ada mendadak berhenti yang digabungkan dengan nilai peak di setiap sumbu pada sensor accelerometer. Data yang dihasilkan dalam keadaan mendadak berhenti, dapat dilihat rentang nilai peak pada sumbu-X: $-3,78G < X < 3,64G$ dan sumbu-Y: $-2,79G < Y < 2,75G$. Kondisi nilai threshold untuk sistem diinformasikan terjadi mendadak berhenti.

C. Pengujian Respon Accelerometer pada Kondisi Tabrakan.



Gambar 12, Hasil Sensor Accelerometer Saat Kondisi Tabrakan

Dari 30 kali pengujian pada kondisi tabrakan, diperoleh informasi, pada sumbu-Y menerima input terjadi perlambatan dari $1,04G$ ke $4G$ di saat t adalah 5 detik. Ini menunjukkan terjadi tabrakan karena pada sumbu-Y berada di titik threshold yaitu $-4G$ dan sumbu-X merespon nilai rata-rata di $-0,03G$ dengan kondisi $-2,23G < X < 2,15G$. Hasil yang ditunjukkan di bawah nilai threshold tabrakan dengan skala 4G dan

sumbu-Y menampilkan rata-rata responnya ada di -0,02G.

Kondisi rentang terjadi tabrakan adalah $-4,00G < \text{Tabrakan} < 1,79G$. Hasil yang ditunjukkan menyentuh nilai threshold tabrakan dengan skala 4G. Ini menunjukkan hasil pengujian terjadi tabrakan yang digabungkan dengan nilai peak dari setiap sumbu pada sensor accelerometer.

Data yang dihasilkan pada saat terjadi tabrakan, nilai peak pada sumbu-X : $-4,00G < \text{Tabrakan}$ di sumbu-X $< 4,00G$, dan sumbu-Y: $-4,00G < \text{Tabrakan}$ di sumbu-Y $< 4,00G$. Nilai ini menjadi Nilai Threshold pada sistem untuk mengetahui kondisi tabrakan.

A. Pengujian Obyek Berjalan

Dari 15 kali pengujian waktu obyek berjalan diperoleh informasi ada 3 output yang anomali, yaitu 2 dalam kondisi berhenti mendadak dan 1 kondisi tabrakan. Untuk mengetahui data tersebut dalam ketelitiannya dapat dilihat dari table Confusion.

Tabel 4, Kondisi Obyek Berjalan

Tabel Confusion	Prediksi	Kelas		
		Berjalan	Berhenti Mendadak	Tabrakan
	Berjalan	12	0	0
	Berhenti Mendadak	0	1	0
	Tabrakan	0	1	1
Tingkat Akurasi=		93%		

Dari model klasifikasi Naïve-Bayes ini memberikan tingkat ketelitian pada nilai threshold, menunjukkan kondisi obyek (mobil Remote Control) berjalan sebesar 93%.

B. Pengujian Obyek Mendadak Berhenti

Dari 15 kali pengujian obyek mendadak berhenti, diperoleh informasi ada 5 output yang anomali yaitu: 4 kondisi berjalan dan 1 kondisi tabrakan. Untuk mengetahui data tersebut dalam ketelitiannya dapat dilihat dari table Confusion.

Tabel 5, Kondisi Obyek Mendadak Berhenti

Tabel Confusion	Prediksi	Kelas		
		Berjalan	Berhenti Mendadak	Tabrakan
	Berjalan	0	0	0
	Berhenti Mendadak	4	10	0
	Tabrakan	0	0	1
Tingkat Akurasi=		73%		

Dari model klasifikasi Naïve-Bayes ini memberikan tingkat ketelitian pada nilai threshold, menunjukkan kondisi obyek (mobil RC) mendadak berhenti sebesar 73%.

C. Pengujian Obyek Tabrakan

Dari 15 kali pengujian obyek tabrakan, diperoleh informasi ada 3 output yang anomali yaitu 2 kondisi mendadak berhenti dan 1 kondisi tabrakan. Untuk mengetahui data tersebut dalam ketelitiannya dapat dilihat dari tabel Confusion.

Tabel 6, Kondisi Obyek Tabrakan
Dari model klasifikasi Naïve-Bayes ini memberikan tingkat ketelitian pada nilai

Tabel Confusion	Prediksi	Kelas		
		Berjalan	Berhenti Mendadak	Tabrakan
	Berjalan	0	0	0
	Berhenti Mendadak	0	0	0
	Tabrakan	1	3	11
Tingkat Akurasi=		73%		

threshold, menunjukkan kondisi obyek (mobil RC) tabrakan sebesar 73%.

Tahap 6, Evaluasi Sistem

A. Evaluasi Sistem

Pada simulasi pengujian dilakukan, obyek berjalan secara garis lurus, dan hasil yang diperoleh pada tahap simulasi yang dibuat dalam 3 kondisi, yaitu :

- Kondisi obyek berjalan,
- Kondisi mendadak berhenti dan
- Kondisi tabrakan.

Dari pengujian ini, kondisi obyek berjalan di garis lurus diperoleh hasil pengujiannya sebagai berikut

- Untuk kondisi obyek berjalan adalah 93% artinya tidak ada terjadi mendadak berhenti ataupun tabrakan.
- Untuk kondisi obyek yang mendadak berhenti adalah 73%, artinya ada terjadi benturan terhadap obyek tersebut, sedangkan

c. Untuk kondisi obyek yang tabrakan 73%, artinya ada terjadi benturan pada obyek. Dari hasil evaluasi ini, dapat dikatakan cukup ketelitiannya untuk dibuatkan menjadi produk dalam mengidentifikasi terjadinya tabrakan. Prototype ini dapat ditempatkan pada posisi di mana bisa membaca g-force seperti di ruang mesin ataupun di balik dashboard sebuah kendaraan.

B. Evaluasi Pengujian Respon Ublox Neo-6M GPS

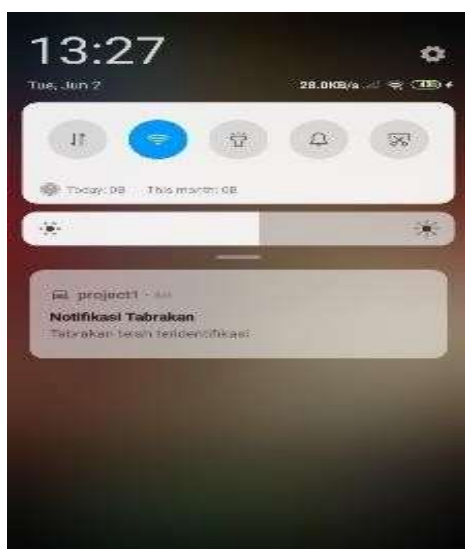
Microcontroller yang dihubungkan dengan GPS dengan *Serial Port1* Input, dan GPS menerima data mentah yang memiliki format NMEA, terdiri dari 6 baris pesan, terdiri dari VTG, RMC, GGA, GLL, GSA dan GSV. Nilai latitude dan longitude dapat dikonversi dari format pesan NMEA dengan *Library TinyGPS++*.

Tahap inisialisasi modul GPS diperlukan waktu ± 3 menit dalam mengirimkan pesan NMEA yang utuh dari satelit.

C. Evaluasi Pengujian Respon Time

Pengujian kecepatan respon time dalam mengirimkan informasi tabrakan melalui aplikasi Android dapat diterima dalam waktu 6 detik ketika microcontroller mengidentifikasi tabrakan

Bentuk screenshot dari notifikasi tabrakan ke Pusat LAKA



Gambar 13 Tampilan di Pusat LAKA

Lokasi tabrakan yang terjadi dikirim melalui GPS

Gambar 14 User Interface Lokasi LAKA



Tahap 7, Penggunaan Sistem

Dari hasil evaluasi sistem di tahap 6 ini, maka model prototype tersebut layak untuk diproduksi dan diaplikasikan pada kendaraan bermotor untuk persiapan kemungkinan terjadi kecelakaan, agar pertolongan pertama pada korban dapat diberikan segera

4. KESIMPULAN

Dari hasil simulasi yang dilakukan, diperoleh : Sensor Accelerometer MPU-6500 menunjukkan kondisi Obyek jalan 93%, mendadak berhenti 73% dan kondisi tabrakan 73%. Secara Real-time di UI Pusat LAKA dapat menerima informasi dalam waktu 6 detik setelah terjadi LAKA. Dalam penerimaan data dari modul GPS secara respon time adalah ± 3 menit dan kondisi tabrakan pada aplikasi Android ± 6 menit.

DAFTAR PUSTAKA

- [1] HA, "Bambang Susantono Gulirkan Revolusi Transportasi," *Berita Satu*, 2014. [Online]. Available: <https://www.beritasatu.com/nasional/207907/bambang-susantono-gulirkan-revolusi-transportasi>.
- [2] *Undang-Undang Republik Indonesia Nomer 22 Tentang Lalu Lintas Dan Angkutan Laut*. 2009.
- [3] K. Amiruddin, "Penanganan Korban Akibat Kecelakaan Lalu Lintas," *Gadar Evakuasi Ditjen Bina Yanmed Gakce P2TM Ditjen PP&PL*, pp. 1-27, 2010.

- [4] H. M. Ali and Z. S. Alwan, "Car Accident Detection and Notification System Using Smartphone," *Int. J. Comput. Sci. Mob. Comput.*, vol. 4, no. 4, pp. 620–635, 2015.
- [5] N. Fathurrahman, A. Hendriawan, and S. Wasista, "Rancang Bangun Smart Vehicle untuk Mendeteksi Dini Kecelakaan dan Keadaan Darurat," *Politek. Elektron. Negeri Surabaya*, pp. 1–9, 2011.
- [6] K. A. Khaliq, S. M. Raza, O. Chughtai, A. Qayyum, and J. Pannek, "Experimental Validation of an Accident Detection and Management Application In Vehicular Environment," *Comput. Electr. Eng.*, pp. 137–150, 2018.
- [7] D. S. Dima and D. Covaciu, "Solutions for acceleration measurement in vehicle crash tests," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 252, no. 1, 2017.
- [8] R. S. Pressman and B. Maxim, *Software Engineering: A Practitioner's Approach*, Eight. McGraw-Hill Education, 2015
- [9] <https://shopee.co.id/MPU-6500-GY-6500-6-Axis-Accelerometer-Gyrometer-Sensor-Module-i.40647041.6806522656>