

## RANCANG BANGUN DETEKSI KECELAKAAN PADA MOBIL LISTRIK MENGGUNAKAN *FLEXIBLE SENSOR* DENGAN *EXPONENTIAL SMOOTHING*

Ridlo Alfatoriq Putro <sup>1)</sup> Harianto Harianto <sup>2)</sup> Yosefine Triwidyastuti <sup>3)</sup> Pauladie Susanto <sup>4)</sup>

Program Studi/Jurusan Teknik Komputer  
Universitas Dinamika.

Jl. Raya Kedung Baruk 98 Surabaya, 60298

Email: 1) [18410200033@dinamika.ac.id](mailto:18410200033@dinamika.ac.id) 2) [hari@dinamika.ac.id](mailto:hari@dinamika.ac.id) 3) [yosefine@dinamika.ac.id](mailto:yosefine@dinamika.ac.id) 4) [pauladie@dinamika.ac.id](mailto:pauladie@dinamika.ac.id)

**Abstrak:** Mobil listrik merupakan transportasi yang populer dikarenakan memiliki keunggulan nol polusi udara, tanpa bahan bakar fosil, serta memiliki fitur seperti *Lane departure alert* dan *Crash detection*. Pentingnya fitur *Crash detection* untuk menghindari keterlambatan penanganan korban, sehingga tidak berakibat fatal. Namun, pada kenyataannya kecelakaan masih terhambat oleh proses pelaporan serta penggunaan sensor yang tidak tepat. Oleh karena itu, dibuatlah sebuah alat untuk mendeteksi kecelakaan menggunakan *Exponential Smoothing*. Beberapa komponen pada alat adalah sensor MPU6050, *flexible sensor*, dan GPS. Adapun metode *Exponential Smoothing* dapat mengurangi *noise* sinyal *output* dari *flexible sensor* dengan cara melakukan trial error untuk menentukan nilai permulasan alpha. Hasil pengujian pada *flexible sensor* menggunakan metode *Exponential Smoothing* mampu mengurangi *noise* tanpa menghilangkan data asli dengan nilai permulasan atau alpha 0.3 serta berkurang sebesar 10.4853% dari data asli. Sensor MPU6050 memiliki nilai *error* sebesar 0.8164% pada sudut *pitch* dan 1.6140% pada sudut *roll* dibandingkan dengan busur, sedangkan pengujian GPS Neo6M mampu mendeteksi lokasi dibandingkan dengan handphone dengan nilai selisih 168.8000 meter, serta dibandingkan dengan Garmin GPS nilai selisih sebesar 8.8150 meter. Selanjutnya uji coba pengiriman data GPS ke Telegram Bot menghasilkan nilai rata-rata pengiriman sebesar 4.1490 detik.

**Kata Kunci:** *Exponential Smoothing*, *Flexibel Sensor*, Sensor MPU6050, GPS Neo6M, Telegram Bot.

Transportasi merupakan alat yang sangat penting untuk menunjang pembangunan, perekonomian dan kemakmuran daerah. Pemerintah berencana untuk mempercepat transportasi yang diatur pemerintah tentang program mobil baterai atau kendaraan listrik. Kendaraan listrik lokal memang sudah ada di Indonesia, namun fitur yang bermanfaat bagi pengguna tidak diterapkan pada kendaraan listrik. Salah satu fungsi *Lane departure alert* adalah untuk mencegah terjadinya kecelakaan dengan cara mengerem dan memutar kemudi. Kendaraan tidak lepas dari kecelakaan human error yang salah satunya bisa berakibat fatal. Selain itu, kurangnya kecepatan penanganan saat terjadi kecelakaan juga dapat menimbulkan kesalahan fatal yang dapat mengakibatkan korban menjadi meninggal. Kecelakaan juga menimbulkan kemacetan apabila tidak segera ditangani. Hingga tahun 2017, laporan data kecelakaan mobil Surabaya tercatat 1.338

didominasi oleh mobil pribadi seperti sepeda motor dan mobil. Jumlah tersangka kecelakaan lalu lintas terbanyak adalah luka ringan yaitu 473 (70%), sedangkan terendah 83 (12%). Dari 1.338 kecelakaan mobil, 461 tewas dan 47 luka berat (Agustin et al., 2020).

Berdasarkan informasi di atas, penulis terdorong untuk membuat alat pendeteksi kecelakaan dengan mengirimkan informasi data GPS dari sebuah alat yang diterapkan pada kendaraan listrik menggunakan aplikasi Telegram. Alat ini dapat mengirimkan koordinat GPS (*Global Positioning System*) melalui koneksi internet jika terjadi kecelakaan. Deteksi kecelakaan menggunakan sensor MPU6050 dan *fleksibel sensor* sebagai data *input* untuk pendeteksian kecelakaan. Alat ini mampu mengetahui menentukan lokasi kecelakaan, sehingga dapat mempercepat penanganan korban, dan menghindari kemacetan.

## LANDASAN TEORI

### Flexible sensor

*Flexible Sensor* adalah sensor yang dapat mendeteksi pembengkokan akibat perubahan resistansi. Sensor beroperasi pada tegangan 5 volt dengan daya puncak 1 watt. Rentang resistansi *flexibel sensor* adalah 45K hingga 125K ohm, dan toleransi kesalahan sekitar 30% (Componen101, 2018). Resistansi memiliki nilai yang berbeda untuk setiap derajat dan bertambah seiring dengan pembengkokan yang terjadi. Rumus pembagi tegangan pada rangkaian *flexibel sensor* adalah:

$$V_o = V_{cc} \left( \frac{R_x}{R_1 + R_x} \right) \quad (1)$$

Keterangan:

Rx: resistansi *flexible Sensor*

Vo: *voltage output*

Vcc: tegangan 5 volt

R1: resistor 170 kilo ohm



Gambar 1. Rangkaian *flexible Sensor*

### NodeMCU ESP32

NodeMCU ESP32 adalah salah satu keluarga mikrokontroler yang paling populer. Diperkenalkan dan dikembangkan oleh *Espressive Systems*. Chip tersebut sudah memiliki modul WiFi (*Wireless Fidelity*) untuk mendukung pembangunan sistem aplikasi *Internet of Things* ESP32. NodeMCU ESP32 memiliki dual core 32-bit yang lebih unggul dari seri ESP8266 sebelumnya dan dilengkapi dengan *Bluetooth Low Energy (BLE)*. Akibatnya, NodeMCU ESP32 mengkonsumsi daya yang sangat kecil. Sekitar 1/100 konsumsi daya *Bluetooth (BLE)* (tergantung pada kasus penggunaan) (Ramadhana & Putra, 2020).



Gambar 2. NodeMCU ESP32

### Global Position System Neo6M (GPS)

GPS yang berarti *Global Positioning System* adalah sistem navigasi yang menyediakan informasi dari perangkat yang terhubung dengan satelit. Data informasi yang diperoleh berupa posisi

lintang dan bujur. Informasi lintang dan bujur ini memungkinkan untuk menemukan objek dari satelit (Desnanjaya et al., 2021). Modul ini sesuai dengan APM2 dan APM2.5 dengan EEPROM on board (memori read-only yang dapat dihapus secara elektrik) dapat digunakan untuk menyimpan data konfigurasi. Antarmuka komunikasi serial TTL (*Transistor-Transistor Logic*) (RX / TX) dapat diakses dari mikrokontroler dengan fungsi UART (*Universal Asynchronous Receiver Transmitter*) atau emulasi TTL serial.



Gambar 3. GPS Neo 6M

### Sensor MPU 6050

Sensor MPU 6050 adalah modul sensor yang memiliki dua fungsi. dua fungsi tersebut adalah *accelerometer* yang dilengkapi dengan sistem mikro elektromekanis dan *Gyroscope* dalam sebuah otak *chip* (Suprayogi et al., 2019). Dengan 16-bit yang dilakukan konversi untuk menentukan sumbu agar sensor ini dapat beroperasi secara optimal. Nilai sumbu x, sumbu y, dan sumbu z dari sensor ini dapat diambil secara bersamaan.



Gambar 4. Gyroscope MPU 6050

### Chat Telegram Bot

Telegram Bot API (*Application Programming Interface*) adalah teknologi *open source* yang dikembangkan dan disediakan oleh Telegram untuk membangun aplikasi Telegram Bot untuk pengembang. Telegram menyediakan kode sumber yang saat ini digunakan. Jenis API kedua adalah API Telegram Bot memungkinkan siapa saja untuk membuat bot yang merespon semua pengguna ketika mengirimkan pesan berupa perintah yang dapat diterima oleh bot tersebut (Cokrojoyo et al., 2017).

### Exponential Smoothing

Metode *Exponential Smoothing* adalah metode prediksi yang umum digunakan untuk analisis *time series*, karena prosedur prediksi

otomatisnya sederhana dan mudah (Maricar, 2019). Metode ini mengarah pada perbaikan dengan meredam data masa lalu dan mendapatkan prediksi tentang data masa depan. Data yang diredam biasanya berisi data yang tidak stabil atau acak. *Exponential Smoothing* adalah metode prediksi yang paling banyak digunakan, karena metode ini membutuhkan komputasi yang lebih sedikit. Prosedur metode ini adalah terus mengulang perhitungan dengan data terbaru. Metode *Exponential Smoothing* dapat dituliskan secara sistematis sebagai berikut (Firdaus & Sani, 2021).

$$F_t = \alpha X_t + (1 - \alpha)F_{t-1} \quad (2)$$

Keterangan:

$F_t$ : nilai *output* filter sekarang

$\alpha$ : nilai konstanta

$X_t$ : nilai aktual sensor

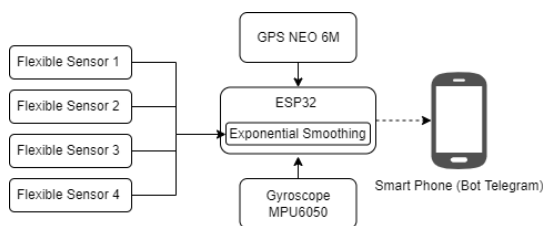
$F_{t-1}$ : nilai *output* filter sebelumnya

Bobot disimbolkan dengan  $\alpha$  (alpha) yang dapat disebutkan dengan nilai permulusan. Nilai konstanta ini dapat dipilih antara nilai 0 dan 1, karena berlaku  $0 < \alpha < 1$ . Nilai  $\alpha$  (alpha) menentukan berapa banyak data sinyal yang harus diperhitungkan. Nilai  $\alpha$  (alpha) ditentukan berdasarkan pengguna. Pembobotan  $\alpha$  (alpha) berpengaruh terhadap sinyal jika nilai  $\alpha$  (alpha) rendah, maka menerima banyak data sinyal yang diperhitungkan dan semakin lambat respon perubahan sinyal, sedangkan jika nilai  $\alpha$  (alpha) tinggi, maka menerima sedikit data sinyal yang diperhitungkan dan semakin cepat respon perubahan sinyal (Firdaus & Sani, 2021).

## METODE PENELITIAN

### Blok Diagram

Diagram skema sistem dari deteksi kecelakaan mobil listrik dapat dilihat pada Gambar 5 Blok Diagram berikut. Diagram tersebut terdiri dari beberapa komponen *input*, proses dan *output* yang memiliki fungsi masing-masing.



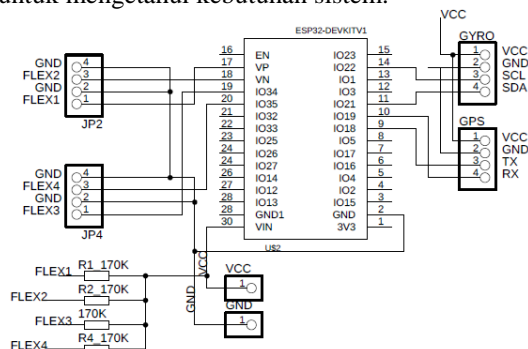
Gambar 5. Blok diagram

Pada gambar blok diagram diatas, masing-masing komponen memiliki peran masing diantaranya adalah *flexible sensor* berfungsi untuk mengambil data yang berasal dari lengkungan yang diakibatkan tabrakan bodi depan mobil listrik. Sensor MPU 6050 berfungsi mengambil data sudut *pitch* dan *roll* ketika mobil mengalami kecelakaan terguling. GPS Neo 6M berfungsi mengambil data koordinat *latitude* dan *longitude* untuk mengetahui lokasi mobil korban kecelakaan. NodeMCU ESP32 komponen utama yang berfungsi untuk mengontrol *input* dan *output*. *Exponential Smoothing* digunakan untuk menghaluskan *output* dari 4 buah *flexible sensor* agar data yang diperoleh menjadi tidak *noise* dengan menggunakan  $\alpha$  (alpha) yang ditentukan dengan cara *Trial and Error*. Telegram Bot merupakan aplikasi untuk menerima data GPS yang disebabkan oleh kecelakaan.

## Rancangan Elektronika

### Rangkaian Skematik Keseluruhan

Pada gambar 6 merupakan rangkaian skematik keseluruhan yang dibuat pada *software Eagle*. Skematik tersebut dicetak dalam PCB sebagai *shield* pendukung NodeMCU ESP32 dan sensor lainnya. Penggunaan pin juga dapat dilihat untuk mengetahui kebutuhan sistem.

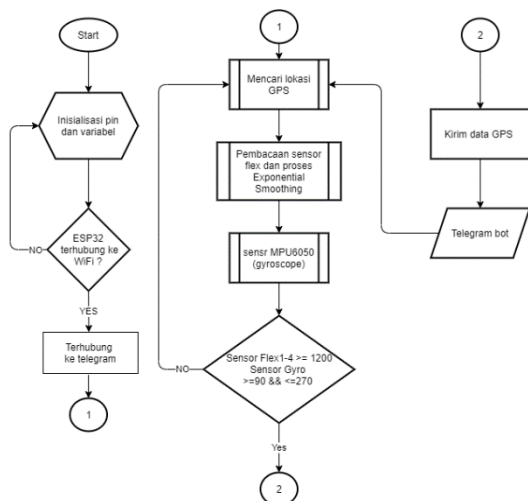


Gambar 6. Rangkaian skematik keseluruhan

### Flowchart Sistem

Algoritma diawali dengan inialisasi masing-masing sensor dan variabel untuk menentukan port, SSID dan *Password* WiFi. Kemudian melakukan koneksi ke WiFi dengan pengecekan NodeMCU ESP32. Berikutnya proses koneksi telegram dengan NodeMCU ESP32 dengan melakukan *test connection*. Selanjutnya GPS Neo6M mencari lokasi *latitude* dan *longitude* mobil secara *real time* kemudian disimpan dalam variabel *string*. Proses pembacaan data dilakukan oleh *flexible sensor* dan sensor MPU 6050 untuk mengetahui nilai dari masing-masing sensor. *Threshold* dari *flexible sensor* untuk mendeteksi

lengkungan menggunakan metode *Exponential Smoothing* yang diakibatkan kecelakaan mobil adalah nilai ADC 1200 dari setiap sensor, sedangkan pada sensor MPU 6050 untuk mendeteksi kemiringan pada saat mobil terguling dengan *Threshold* sudut *roll* sebelah kanan yaitu diatas 90 derajat, kemudian untuk sebelah kiri dibawah 270 derajat dan *Threshold* sudut *pitch* ke arah depan yaitu diatas 90 derajat kemudian untuk belakang dibawah 270. Jika Sensor Flex1-4  $\geq 1200$  dan Sensor Gyro  $\geq 90$  &  $\leq 270$  (Yes), maka data informasi lokasi pada GPS Neo6M yang disimpan pada variabel *string* kemudian dikirimkan ke telegram dengan ID telegram yang sudah dibuat. Jika tidak (No) melebihi *Threshold*, maka kembali melakukan pembacaan GPS Neo 6M.



Gambar 7. Flowchart sistem

### Rancangan Mekanik

Bagian ini dijelaskan tentang rancangan pemasangan komponen yang terdiri dari *flexible sensor*, serta penempatan *mikrokontroler* sensor MPU6050, GPS Neo6m agar dapat berjalan sesuai yang diinginkan. Pada gambar 8 merupakan tempat untuk pemasangan *flexible sensor* dengan dibuatkan *mounting*, kemudian dipasangkan ke lubang baut yang telah disediakan. Selain itu komponen tersebut dijadikan satu pada wadah *blackbox* untuk memudahkan ketika pemeliharaan komponen dan diletakkan di bawah *dashboard* mobil listrik.



Gambar 8. Tempat komponen

## HASIL DAN PEMBAHASAN

### Hasil Pengujian Flexible Sensor

Pengujian *flexible sensor* ini merupakan pengambilan data dengan hasil nilai ADC. Pengujian ini dilakukan pada 4 buah *flexible sensor* dimana masing-masing sensor ditabrakkan pada sebuah benda dengan derajat yang telah ditentukan. Menggunakan resistor yang sama yaitu 170k ohm dengan rangkaian pembagi tegangan untuk menghasilkan nilai ADC yang mudah dibaca.

Tabel 1. *Flexible sensor* tanpa metode.

Data	Busur (derajat)	Pengujian Tanpa Metode (Kelengkungan)			
		Flex1	Flex2	Flex3	Flex4
1	20	554	757	882	940
2		989	1,104	897	951
3		561	823	866	1,021
4		1,102	875	837	1,012
5		592	1,013	1,150	1,110
1	30	590	897	860	896
2		623	827	829	897
3		695	1,241	1,638	942
4		1,472	813	810	1,311
5		699	1,285	1,491	1,017
1	40	1,424	802	885	869
2		1,095	944	858	918
3		1,210	1,479	842	880
4		1,274	1,187	1,190	1,363
5		1,444	1,838	790	1,040

Pengujian dilakukan menabrakkan sensor pada sebuah benda dengan sudut 20, 30, dan 40 derajat. Ini adalah simulasi tabrakan dengan benda didepan, sehingga dapat menghasilkan nilai keluaran berupa adc. Dapat dilihat bahwa nilai tersebut tidak konsisten pada 20 dan 30 derajat, sehingga dapat dikatakan banyak *noise* saat pengambilan data tanpa metode.

## Hasil Pengujian *Flexible Sensor* menggunakan *Exponential Smoothing*

Pengujian *flexible sensor* ini dilakukan pada 4 buah *flexible sensor* dimana masing-masing sensor ditabrakkan pada sebuah benda dengan derajat yang telah ditentukan. Menggunakan rangkaian pembagi tegangan untuk menghasilkan nilai ADC yang mudah dibaca. Pengujian ini menggunakan metode *Exponential Smoothing* yang menentukan nilai dengan cara *trial error*.

Tabel 2. *Flexible sensor* pengujian menggunakan metode

Data	Busur (derajat)	Pengujian Menggunakan Metode (Kelengkungan)			
		Flex1	Flex2	Flex3	Flex4
1	20	657.1310	837.5780	880.9240	900.0200
2		789.8790	917.5050	885.7470	915.3140
3		698.3270	889.1530	879.8230	947.0200
4		859.7960	884.9070	866.9760	966.5140
5		752.6780	923.3350	951.8830	1,009.5600
1	30	650.0090	786.9600	859.1940	894.7750
2		792.3980	798.9720	850.1360	895.4430
3		838.5430	931.5800	1,086.4950	909.4100
4		1,091.9260	896.0060	1,003.5470	1,029.8870
5		934.7560	1,012.7040	1,149.7830	1,026.0210
1	40	1,223.4720	759.2780	859.2270	887.3000
2		1,172.0830	814.6950	858.8590	896.5100
3		1,187.2500	1,013.9860	853.8010	891.5570
4		1,221.9500	1,065.8900	954.6610	1,032.9900
5		1,310.7700	1,297.5230	905.2630	1,035.0930

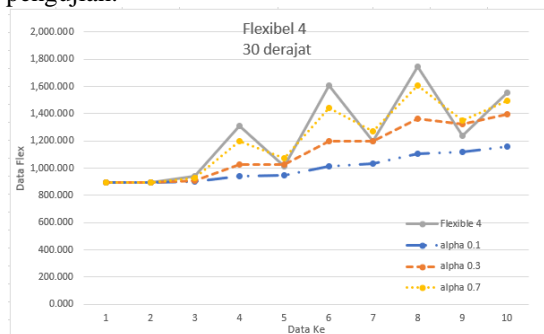
Pada tabel 2 tersebut dapat dilihat bahwa nilai data ADC *flexible sensor* mengalami peredaman *noise*. Meskipun peredaman tidak terlalu mulus tetapi nilai tersebut mendekati nilai dari keluaran *flexible sensor*. Nilai alpha tersebut adalah 0.3 pada masing-masing *flexible sensor* dengan melakukan simulasi tabrakan ke benda dengan sudut 20, 30, 40 derajat. Dapat dilihat bahwa nilai tersebut konsisten pada 30 dan 40 derajat jika dibandingkan dengan pengambilan data tanpa metode. Nilai *flexible sensor* meningkat mengikuti lengkungan pada setiap derajat pada sensor. Kelengkungan ini mencapai 1.234.195 pada sudut 30 derajat dimana nilai tersebut dapat dijadikan *Threshold* untuk menentukan nilai ketika terjadi tabrakan.

Tabel 3. Pengujian GPS

No	Nama Lokasi	Data GPS			Selisih handphone sensor (KM)	Selisih Garmin sensor (KM)
		Garmin	Handphone	Sensor		
1	Rungkut madya v2	-7.33167, 112.78342	-7.331693, 112.783465	-7.3317470, 112.7834010	0.009267	0.008815
2	Ir Soekarno	-7.32946, 112.78116	-7.329483, 112.78119	-7.3294250, 112.7810440	0.01735	0.01337
3	Rungkut madya	-7.33153, 112.77936	-7.331536, 112.779345	-7.3317470, 112.7791210	0.03407	0.03573
4	Rungkut asri tengah	-7.32801, 112.77566	-7.32806, 112.775644	-7.3280580, 112.7756960	0.005739	0.006652
5	Rungkut harapan	-7.32275, 112.77823	-7.322704, 112.778313	-7.3227110, 112.7782750	0.004263	0.006591
6	Kertajaya	-7.27946, 112.76388	-7.279449, 112.763869	-7.2794550, 112.7638090	0.006651	0.007851
7	Pucang Anom timur	-7.28501, 112.75574	-7.285002, 112.755799	-7.2847800, 112.7559970	0.03296	0.03818
8	Ngagel jaya Selatan	-7.29340, 112.75378	-7.293303, 112.753728	-7.2932880, 112.7537540	0.003317	0.01278
9	Dinoyo	-7.28427, 112.74605	-7.284282, 112.746043	-7.2842660, 112.7460250	0.002666	0.002793

## Hasil Pengujian nilai Alpha pada Metode *Exponential Smoothing*

Pada pengujian ini bertujuan untuk mengetahui berapa nilai permulasan (alpha) yang cocok pada *flexible sensor*. Pengambilan data sensor ini dilakukan secara bersamaan dengan tiga permulasan yaitu 0.1, 0.3, dan 0.7. Masing-masing dilakukan pengujian dengan menabrakkan suatu benda ke sensor. Berikut merupakan pengambilan data *flexible sensor* 4 dapat dilihat pada grafik pengujian:



Gambar 9. Grafik *flexible sensor* 4 buah

Pengujian ini dilakukan dengan nilai alpha yang dijalankan secara bersamaan dengan menggunakan metode dan tidak menggunakan metode. Hasil grafik menunjukkan peningkatan mencapai nilai 1200 pada kelengkungan 30 derajat. Pengujian tersebut dapat disimpulkan bahwa grafik yang menggunakan metode lebih teredam daripada tanpa metode, tetapi nilai alpha juga mempengaruhi tingkat *noise* dari sensor tersebut. Dari grafik diatas dapat disimpulkan pergerakan Nilai permulasan tersebut juga berkurang sebesar 10.4853% pada alpha 0.3.

## Hasil Pengujian GPS Neo6M

Pengujian ini untuk membandingkan antara GPS *handphone* dan GPS sensor, sehingga dapat mengetahui jarak masing-masing.

No	Nama Lokasi	Data GPS			Selisih handphone sensor (KM)	Selisih Garmin sensor (KM)
		Garmin	Handphone	Sensor		
10	Soetomo	-7.28311, 112.73591	-7.28313, 112.735929	-7.2832100, 112.7359540	0.009313	0.01213
11	Ciliwung	-7.29371, 112.73507	-7.293744, 112.735058	-7.2937220, 112.7350160	0.005239	0.006104
12	Diponegoro	-7.28780, 112.73486	-7.287799, 112.734852	-7.2876320, 112.7350690	0.03029	0.02967
13	Adityawarman	-7.29358, 112.73058	-7.293518, 112.730567	-7.2940280, 112.7303540	0.06138	0.05570
14	Jl Nusa indah	-7.33206, 112.79189	-7.33054, 112.791879	-7.3320570, 112.7919240	0.168800	0.003765
15	dukuh pakis	-7.28859, 112.71382	-7.288606, 112.713852	-7.2886970, 112.7139660	0.01614	0.02002
<b>Rata-rata</b>					<b>0.168800</b>	<b>0.0088</b>

Tabel 3 menunjukkan bahwa setiap lokasi mempunyai jarak yang berbeda-beda antara GPS *handphone* dan GPS Neo6m. Pengambilan data setiap titik memakan waktu sekitar 5-7 menit agar mendapatkan data yang maksimal. Dari tabel tersebut dapat disimpulkan bahwa rata-rata dari *handphone* dan GPS yaitu sebesar 0.168800 Km atau 168.8000 meter. Kemudian rata-rata dari Garmin dan GPS 0.008815 Km atau 8.8150 meter.

### Hasil Pengujian Sensor MPU 6050

Pengujian sensor MPU 6050 berfungsi untuk mengetahui berapa besar *error* yang didapatkan pada sensor MPU 6050. Pengujian dilakukan dengan beberapa derajat kemudian dibandingkan dengan busur. Pengambilan sudut sensor MPU 6050 adalah sudut *pitch* (y) dan *roll* (x).

Tabel 4. Pengujian sudut *pitch*

Pengujian sudut pitch depan					Pengujian sudut pitch belakang			
No	Busur	Hasil Pengujian (°)	Selisih (°)	Error (%)	Busur	Hasil Pengujian (°)	Selisih (°)	Error (%)
1		29.3080	0.6920	2.3070		329.5200	0.4800	0.1450
2		29.6540	0.3460	1.1530		329.2080	0.7920	0.2400
.	30	.	.	.	330	.	.	.
.		.	.	.		.	.	.
9		28.7040	1.2960	4.3200		328.8470	1.1530	0.3490
10		28.6330	1.3670	4.5570		329.2950	0.7050	0.2140
<b>Rata-rata</b>			<b>0.8650</b>	<b>2.8830</b>	<b>Rata-rata</b>		<b>0.7530</b>	<b>0.2280</b>
11		59.7710	0.2290	0.3820		298.1810	1.8190	0.6060
12		59.9130	0.0870	0.1450		298.8890	1.1110	0.3700
.	60	.	.	.	300	.	.	.
.		.	.	.		.	.	.
19		59.3140	0.6860	1.1430		298.8710	1.1290	0.3760
20		59.9720	0.0280	0.0470		298.6830	1.3170	0.4390
<b>Rata-rata</b>			<b>0.2520</b>	<b>0.4200</b>	<b>Rata-rata</b>		<b>1.3020</b>	<b>0.4340</b>
21		89.6810	0.3190	0.3540		269.2720	0.7280	0.2700
22		89.5310	0.4690	0.5210		269.3630	0.6370	0.2360
.	90	.	.	.	270	.	.	.
.		.	.	.		.	.	.
29		89.1430	0.8570	0.9520		268.1950	1.8050	0.6690
30		89.2250	0.7750	0.8610		269.0500	0.9500	0.3520
<b>Rata-rata</b>			<b>0.4970</b>	<b>0.5520</b>	<b>Rata-rata</b>		<b>1.0290</b>	<b>0.3810</b>
<b>Rata-rata error keseluruhan</b>								<b>0.8160</b>

Tabel 5. Pengujian sudut *roll*

Pengujian sudut Roll kanan					Pengujian sudut Roll kiri			
No	Busur	Hasil Pengujian (°)	Selisih (°)	Error (%)	Busur	Hasil Pengujian (°)	Selisih (°)	Error (%)
1		28.2470	1.7530	5.8430		326.5130	3.4870	1.0570
2		28.6520	1.3480	4.4930		326.1650	3.8350	1.1620
.	30	.	.	.	330	.	.	.
.		.	.	.		.	.	.
9		27.9590	2.0410	6.8030		326.3520	3.6480	1.1050
10		28.6480	1.3520	4.5070		326.9700	3.0300	0.9180
<b>Rata-rata</b>			<b>1.5670</b>	<b>5.2240</b>	<b>Rata-rata</b>		<b>3.6297</b>	<b>1.1000</b>
11		58.4150	1.5850	2.6420		298.0390	1.9610	0.6540
12		58.2940	1.7060	2.8430		297.6080	2.3920	0.7970
.	60	.	.	.	300	.	.	.
.		.	.	.		.	.	.
19		58.5930	1.4070	2.3450		298.3110	1.6890	0.5630
20		59.8200	0.1800	0.3000		297.5260	2.4740	0.8250
<b>Rata-rata</b>			<b>1.0460</b>	<b>1.7430</b>	<b>Rata-rata</b>		<b>2.2330</b>	<b>0.7440</b>
21		89.5350	0.4650	0.5170		268.5920	1.4080	0.5210
22		89.7760	0.2240	0.2490		268.5940	1.4060	0.5210
.	90	.	.	.	270	.	.	.
.		.	.	.		.	.	.
29		89.7070	0.2930	0.3260		268.6720	1.3280	0.4920
30		89.5900	0.4100	0.4560		269.0370	0.9630	0.3570
<b>Rata-rata</b>			<b>0.4050</b>	<b>0.4500</b>	<b>Rata-rata</b>		<b>1.1440</b>	<b>0.4240</b>
<b>Rata-rata error keseluruhan</b>								<b>1.6140</b>

Pengujian sensor MPU 6050 pada sudut *pitch* (y) ke depan dan belakang pada sudut yang telah ditentukan menggunakan busur. Pengujian ke depan dapat dilihat nilai rata-rata *error* pada masing-masing sudut yaitu 2.8830%, 0.4200%, dan 0.5521%. Kemudian sudut *pitch* (y) ke belakang nilai rata-rata *error* sebesar 0.2281%, 0.4341%, 0.3811% dan total *error* keseluruhan sudut *pitch* (y) adalah 0.8164 %. Hasil tersebut menunjukkan nilai pada sudut *pitch* (y) memiliki *error* yang kecil.

Pengujian selanjutnya yaitu pada sudut *roll* (x) ke kanan menunjukkan nilai rata-rata *error* 5.2240%, 1.7427%, dan 0.4496%. Sudut *roll* (x) ke kiri nilai rata-rata *error* sebesar 1.0999%, 0.7443%, 0.4237%. Rata-rata *error* keseluruhan pada sudut *roll* (x) adalah 1.6140%.

### Hasil Pengiriman Data GPS ke Telegram

Pada pengujian ini untuk mengetahui berapa waktu yang diperlukan pengiriman data GPS dari NodeMCU ESP32 ke aplikasi telegram bot. Pengambilan data ini diambil pada saat pengujian GPS Neo6m menentukan titik lokasi.

Tabel 6. Pengiriman data GPS ke telegram

Data	Data GPS	Terkirim di telegram (detik)
1	-7,3317470, 112,7834010	4.5570
2	-7,3294250, 112,7810440	4.1790
3	-7,3317470, 112,7791210	4.3830
4	-7,3280580, 112,7756960	4.3220
5	-7,3227110, 112,7782750	4.0120
6	-7,2794550, 112,7638090	4.1200
7	-7,2847800, 112,7559970	4.0710
8	-7,2932880, 112,7537540	3.9620
9	-7,2842660, 112,7460250	3.7210
10	-7,2832100, 112,7359540	4.1970
11	-7,2937220, 112,7350160	4.0070
12	-7,2876320, 112,7350690	3.4630
13	-7,2940280, 112,7303540	4.5900
14	-7,3320570, 112,7919240	4.2240
15	-7,2886970, 112,7139660	4.4300
<b>Rata Rata</b>		<b>4.1492</b>

### Hasil Pengujian Keseluruhan Sistem

Pengujian ini dilakukan pada setiap lokasi jalan raya umum yang berbeda-beda. Pengujian ini dilakukan pada kedua sensor agar mendapatkan lokasi kecelakaan. *Flexible sensor* dilengkungkan sebagai simulasi kecelakaan untuk mendapatkan lokasi GPS. Kemudian sensor MPU6050 dimiringkan ke kanan dan kiri, serta diarahkan kedepan dan belakang agar mendapatkan titik lokasi kecelakaan.

Tabel 7. Pengujian keseluruhan sistem

No	Nama Lokasi	Sensor GPS Neo6M	Status Kecelakaan	Status Keberhasila	Terkirin di telegra (detik)
1	Jl. Rungkut Madya 149-167,	-	Kecelakaan	Berhasil	4.557
		7.3317470			
2	Jl. Dr. Ir. H. Soekarno, No. J51,	112.78340	Kecelakaan	Berhasil	4.179
		10			
		7.3294250			
3	Rungkut Asri Tengah I No.2,	112.78104	Kecelakaan	Berhasil	4.322
		40			
		7.3280580			
4	Jl. Manyar Kertoarjo 31,	112.77569	Kecelakaan	Berhasil	4.12
		60			
		7.2794550			
5	Jl. Pucang Anom Tim.,	112.76380	Kecelakaan	Berhasil	4.071
		90			
		7.2847800			
6	Jl. Krukah Ps., Ngagelrejo,	112.75599	Kecelakaan	Berhasil	3.962
		70			
		7.2932880			
7	Jl. Dr. Soetomo 94-84	112.75375	Kecelakaan	Berhasil	4.197
		40			
		7.2832100			
8	Jl. Diponegoro 64-56,	112.73595	Kecelakaan	Berhasil	3.463
		40			
		7.2876320			
9	Jl. Adityawarman No.55,	112.73506	Kecelakaan	Berhasil	4.197
		90			
		7.2940280			
10	Jl. Raya Dukuh Kupang,	112.73035	Kecelakaan	Berhasil	4.43
		40			
		7.2886970			
		112.71396			
		60			
<b>Rata-rata</b>					<b>4.1498</b>

Dari tabel pengujian diatas merupakan seluruh rangkaian pengujian kecelakaan. Setiap sensor sudah dimasukkan *Threshold* masing-masing, diantaranya 4 *flexible sensor* adalah 1200 ADC, dan sensor MPU6050 masing-masing sudut diatas 90 dan dibawah 270 derajat. *Flexible sensor* menggunakan metode permlusan (alpha) 0.3 agar *noise* teredam dan tidak menghilangkan nilai aslinya. Pada saat pengujian, beberapa sensor diuji dengan simulasi tabrakan kecelakaan kemudian pada saat bersamaan klik tombol start pada stopwatch. Saat notifikasi berupa data koordinat muncul pada *handphone*, maka klik tombol stop pada stopwatch untuk melihat berapa lama pengiriman data koordinat ke telegram bot. selama pengujian untuk mengambil data satu titik koordinat memerlukan waktu sekitar 7-10 menit agar koordinat GPS sesuai dengan lokasinya.



## KESIMPULAN

Pada penelitian ini, terdapat beberapa tahapan dimulai dari perancangan, hasil dan pembahasan. Dari tahapan tersebut tentang penelitian Rancang Bangun Deteksi Kecelakaan Pada Mobil Listrik Dengan *Exponential Smoothing* dapat disimpulkan sebagai berikut:

1. Berdasarkan pengujian *flexibel sensor* menggunakan metode *Exponential Smoothing* yang dapat dilihat pada grafik, mampu mengurangi *noise* tanpa menghilangkan data asli dengan nilai permulasan atau  $\alpha$  0.3 dan nilai berkurang sebesar 10.4853% dibandingkan dengan data asli.
2. Sensor MPU6050 memiliki nilai *error* sebesar 0.8164% pada sudut *pitch* dan 1.6140% pada sudut *roll* dibandingkan dengan busur.
3. GPS Neo6M mampu mendeteksi lokasi dibandingkan dengan *handphone* dengan nilai selisih 168.8000 meter, serta dibandingkan dengan Garmin GPS nilai selisih sebesar 8.8150 meter.
4. Uji coba pengiriman data GPS ke Telegram Bot menghasilkan nilai rata-rata pengiriman sebesar 4.1490 detik.

## Saran

Pada perancangan hasil dan pembahasan yang telah dilakukan oleh penulis, maka terdapat poin-poin yang memungkinkan melakukan pengembangan untuk lebih meningkatkan hasil yang baik, diantaranya:

1. Pada tahap pengujian sebaiknya dilakukan pada mobil listrik untuk mendapatkan nilai yang aktual.
2. Apabila diterapkan pada mobil listrik sebaiknya memakai *accu* mobil sebagai sumber tegangan. Dikarenakan saat pada penelitian ini belum ada *stepdown* yang dapat menurunkan tegangan sampai 5 volt 2 ampere.

## DAFTAR PUSTAKA

- Agustin, I. W., Meidiana, C., & Muljaningsih, S. 2020. Studi Simulasi Model Kecelakaan Pengendara Mobil untuk Meningkatkan Keselamatan Lalu Lintas di Daerah Perkotaan. *Warta Penelitian Perhubungan*. Malang. 32(2), 93–102. <https://doi.org/10.25104/warlit.v32i2.1513>
- Cokrojoyo, A., Andjarwirawan, J., & Noertjahyana, A. 2017. Pembuatan Bot Telegram Untuk Mengambil Informasi dan

Jadwal Film Menggunakan PHP. *Jurnal Infra*, 5(1), 224–227, Program Studi Teknik Informatika Fakultas.

<http://studentjournal.petra.ac.id/index.php/teknik-informatika/article/view/5163>

- Componen101. 2018. *Fitur Sensor Fleksibel, Bekerja, Sirkuit & Lembar Data*. (Online). (<https://components101.com/sensors/flex-sensor-working-circuit-datasheet>, diakses 31 Mei 2022).
- Desnanjaya, I. G. M. N., Nugraha, I. M. A., & Hadi, S. 2021. Sistem Pendeteksi Keberadaan Nelayan Menggunakan GPS Berbasis Arduino. *Jurnal Sumberdaya Akuatik Indopasifik*, 5(2), 157–168. <https://ejournalfpikunipa.ac.id/index.php/JS-AI/article/view/143>
- Firdaus, F., & Sani, A. 2021. Stasiun Pemantau Cuaca Berbasis IoT (Internet of Things) dengan Metode Exponential Smoothing. *Journal of Applied Electrical Engineering*, 5(2), 50–54. <https://doi.org/10.30871/jaee.v5i2.3172>
- Maricar, M. A. 2019. Analisa Perbandingan Nilai Akurasi Moving Average Dan Exponential Smoothing Untuk Sistem Peramalan Pendapatan Pada Perusahaan XYZ. *Jurnal Sistem Dan Informatika*, 13(2), 36–45.
- Ramadhana, M., & Putra, S. 2020. *Sistem Kontrol Suhu Dan Kelembapan Pada Budidaya Jamur Tiram Berbasis Iot*. Politeknik Menufaktur Negeri Bangka Belitung.
- Suprayogi, A., Fitriyah, H., & Tibyani. 2019. Sistem Pendeteksi Kecelakaan Pada Sepeda Motor Berdasarkan Kemiringan Menggunakan Sensor Gyroscope Berbasis Arduino. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(3), 3079–3085.