

## AUGMENTED REALITY UNTUK PEMANTAUAN KONDISI PADA KENDARAAN INJEKSI BERBASIS ANDROID

Muhammad Hamzah Robbani<sup>1</sup>, Mat Syai'in<sup>2</sup>, Didik Sukoco<sup>3</sup>

<sup>1,2,3</sup>Program Studi Teknik Otomasi, Jurusan Teknik Kelistrikan Kapal

Politeknik Perkapalan Negeri Surabaya

Jalan Teknik Kimia, Kampus ITS Sukolilo, Surabaya 60111

E-mail : mhamzahr@gmail.com

### Abstrak

Kendaraan modern saat ini menggunakan sistem injeksi yang terintegrasi dengan sistem *OBD* (*On Board Diagnostics*). *OBD* berisi informasi mengenai data kondisi kendaraan. Data pada *OBD* dapat dibaca dengan menggunakan alat bernama *scanner OBD* dan aplikasi pendukungnya. Saat ini, di *playstore* tersedia beberapa aplikasi untuk membaca data dari *scanner OBD*. Aplikasi *scanner OBD* yang tersedia saat ini menampilkan data sensor dalam bentuk tampilan angka, grafik, dan *gauge*. Tampilan seperti ini kurang informatif dalam menjelaskan kondisi kendaraan bagi beberapa pengguna yang awam di bidang kendaraan. Berdasarkan permasalahan tersebut, penelitian ini membahas mengenai pembuatan aplikasi sistem pemantauan kondisi pada kendaraan injeksi yang diintegrasikan dengan *Augmented Reality* (*AR*). Sistem pemantauan kondisi ini dapat menampilkan informasi dari *OBD* dan beberapa informasi tambahan mengenai kegagalan untuk mempermudah proses *maintenance*. *AR* membantu untuk memberikan tampilan yang lebih informatif dalam menjelaskan kondisi dari kendaraan dibandingkan tampilan *aplikasi scanner OBD* yang tersedia saat ini. Hasil dari penelitian ini untuk sistem *AR* menghasilkan kesalahan 27% dari 78 sampel ketika pengujian visualisasi, dan pengujian identifikasi kegagalan menghasilkan kesalahan 0% dari 7 sampel kegagalan yang diuji.

**Kata Kunci:** Augmented Reality, Pemantauan Kondisi, Mobil, *OBD*, Mesin

### 1. PENDAHULUAN

Kendaraan bermotor adalah alat transportasi yang banyak digunakan di Indonesia. Menurut Badan Pusat Statistik, jumlah kendaraan bermotor di Indonesia pada tahun 2016 adalah 112.205.711 unit, tahun 2017 sebanyak 118.922.708 unit, dan tahun 2018 sebanyak 126.508.776 unit [1]. Rata-rata penjualan sepeda motor di Indonesia adalah 6 juta unit per tahun, membuat Indonesia menjadi peringkat ketiga di dunia [2]. Selain itu, terdapat fakta bahwa Indonesia menjadi negara dengan jumlah kendaraan bermotor terbanyak di kawasan Asia Tenggara pada tahun 2011 [3]. Dari data-data diatas terlihat bahwa perkembangan dan populasi kendaraan bermotor di Indonesia sangat besar.

Pada era modern saat ini, kondisi kendaraan injeksi dapat dipantau dengan mudah melalui *android* menggunakan aplikasi *scanner OBD* yang tersedia di *play store* dan *scanner OBD* seperti *ELM327*. Aplikasi *scanner OBD* yang tersedia saat ini memiliki antarmuka yang susah untuk dipahami bagi beberapa konsumen, terutama konsumen yang awam dibidang kendaraan karena tampilan kondisi di aplikasi berupa angka, grafik, dan *gauge*.

Selain itu, aplikasi hanya memberikan informasi kondisi dan kegagalan sensor, tidak memberikan informasi mengenai kegagalan yang lebih lengkap untuk mempermudah proses *maintenance*.

Dari latar belakang tersebut, penulis memiliki solusi yaitu sebuah aplikasi yang dapat memantau kondisi kendaraan dalam bentuk *Augmented Reality*. *Augmented Reality* (*AR*) dapat menampilkan visualisasi seperti animasi, gambar, atau desain 3D berdasarkan data-data dari sensor dan memproyeksikan ke dunia nyata secara *real-time* [4]. Selain dapat memberikan informasi data dan kegagalan sensor, aplikasi ini juga dapat memberikan informasi mengenai letak komponen yang berkaitan dengan kegagalan, sistem lain yang berdampak, dan prosedur perbaikan kegagalan untuk mempermudah proses *maintenance*. Aplikasi *AR* ini juga dapat dijadikan sebagai sarana untuk edukasi mengenai kendaraan karena informasi disajikan dalam bentuk visualisasi yang mudah dipahami oleh semua kalangan.

### 2. METODE

Kendaraan yang digunakan pada penelitian

ini adalah *Toyota Kijang Innova 2012 2.0 Bensin*, sehingga metode yang dijabarkan dibawah ini dirancang sedemikian rupa agar cocok dengan kendaraan tersebut. Semua kalimat yang mengacu pada “kendaraan” memiliki arti *Toyota Kijang Innova 2012 2.0 Bensin*.

## 2.1 Kebutuhan Sistem Pemantauan Kondisi

*OBD (On Board Diagnostic)* adalah sistem yang pada kendaraan yang berfungsi untuk diagnostik mandiri [5]. Sistem *OBD* menyimpan data-data kondisi kendaraan secara waktu nyata dan hasil diagnosa dari data kondisi tersebut. Data pada sistem *OBD* berasal dari hasil pembacaan sensor-sensor yang terhubung pada *ECU* kendaraan injeksi. Data kondisi kendaraan pada sistem *OBD* akan diambil untuk sistem pemantauan kondisi kendaraan penelitian ini. Dibutuhkan alat bernama *Mini VCI J2534* dan perangkat lunak *Global Techstream* untuk membaca data sistem *OBD* pada kendaraan ini. Dikarenakan sensor-sensor bawaan pada kendaraan ini terbatas, sehingga penulis menambahkan beberapa sensor tambahan (sensor yang tidak berkaitan dengan sistem *OBD*) untuk membaca beberapa kondisi krusial yang datanya tidak terdapat pada sistem *OBD*.

**Tabel 2.1** Kondisi yang dipantau

No.	Nama Variabel	Satuan Data	
		Mentah	Jadi
1	Kecepatan Kendaraan	-	km/j
2	Putaran Kruk As	-	rpm
3	Suhu Cairan Pendingin	-	°C
4	Suhu Udara Intake	-	°C
5	Posisi Pedal Gas	-	%
6	Ketinggian Cairan Pendingin Cadangan	cm	%
7	Ketinggian Air Aki	cm	%
8	Tekanan Oli	bool	kPa
9	Suhu Gas Buang	-	°C



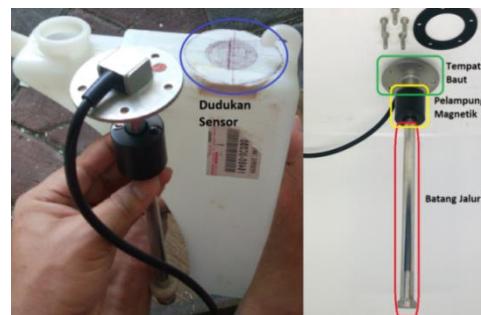
**Gambar 2.1** Mini VCI J2534 (kiri), Global Techstream (kanan)



**Gambar 2.2** Sensor Termokopel K untuk membaca suhu gas buang



**Gambar 2.3** Sensor HCSR04 untuk membaca ketinggian air aki



**Gambar 2.4** Sensor Level Magnetik untuk membaca ketinggian cairan pendingin cadangan



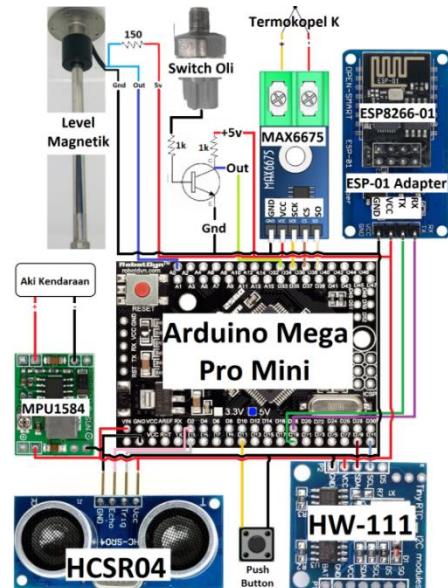
**Gambar 2.5** Sensor Switch Oli untuk membaca tekanan oli

Tabel 2.1 di atas menjelaskan mengenai variabel kondisi kendaraan yang akan dipantau, kondisi nomor 1-5 datanya berasal dari sistem *OBD*, sedangkan kondisi nomor 6-9 berasal dari sensor tambahan yang dipasangkan ke kendaraan. Seluruh data hasil pembacaan kondisi oleh sensor dikirim ke *database* agar dapat terintegrasi oleh sistem *AR* pada aplikasi *android*.

Sensor-sensor tambahan ini datanya dikelola oleh sistem yang penulis sebut dengan *sistem pemantauan non-OBD*. *Sistem Pemantauan Sensor non-OBD* adalah sistem

yang membaca, mengelola, dan mengirim hasil pembacaan kondisi sensor tambahan ke database. Berikut ini adalah komponen dan rangkaian perkabelan sistem ini.

- MP1584 - Arduino Mega
- MAX6675 - Termokopel K
- HCSR04 - Level
- Magnetik*
- Push Button - Transistor
- NPN
- ESP8266-01 - Resistor 150 Ω
- RTC HW-111 -
- Resistor 1k Ω
- ESP-01 Adapter - Switch Oli



Gambar 2.6 Rangkaian kabel untuk sistem pemantauan sensor OBD.

Selain itu, terdapat juga kegagalan-kegagalan yang telah ditentukan yang dapat diidentifikasi dengan parameter tertentu oleh sistem untuk mempermudah proses *maintenance* kendaraan. Berikut ini adalah daftar kegagalan beserta parameter identifikasi dan rincian informasi tambahan yang akan dimasukkan ke dalam program pada aplikasi *android*.

Tabel 2.2 Rincian Kegagalan

K	N	P	PL	SB	SP	I
1	Motor bakar tidak bisa ke suhu optimal	SC< 77 °C selama >5 menit saat PM >0 rpm	Termostat	kipas radiator	Pengecekan katup termostat dan visco kipas radiator	Suhu Cairan 77-95°C
2	Motor bakar terlalu panas	SC> 97 °C	cairan pendingin, radiator, kipas radiator, termostat	sistem pelumasan	Pengecekan level air pendingin, visco kipas radiator dan katup termostat. penggantian filter oli. penambahan/penggan-ian oli	Suhu Cairan 77-95°C
3	Stasioner terlalu tinggi	PM> 1000 rpm ketika PP= 0 % dan suhu CP> 77 °C	Thro-ttle body, sensor MAF, sensor SC	motor bakar	pembersihan injektor, pengecekan kebocoran katup throttle body, sensor MAF dan SC	Puta-ran Stasi-oner 600-1000 rpm
4	Penguapan air aki	KA turun 90% selama	Aki	sistem pengisian	Pengecekan umur aki dan	Air Aki 70-100%

	terlalu berlebihan	1-2 bulan		aki	alternator	
5	Tekanan oli rendah	TO <29 kPa ketika PM>0 rpm	filter oli, penampung oli	-	Pengecekan ketinggian dan kondisi oli	Teka-nan Oli >=29 kPa
6	Cairan pendingin mau habis	KC<20 %	tabung radiator	sistem pendinginan, sistem pelumasan	Pengecekan radiator, kipas radiator, termostat, filter oli, oli	Ketinggi anCairan Pendingin 32-100%
7	Suhu gas buang terlalu tinggi	SG>600 °C	throttle body, filter udara	-	Pembersihan filter udara dan katup throttle body. Pengecekan kebocoran katup throttle body	Suhu Gas Buang 300-600°C

Keterangan Tabel:

K = Kode kegagalan  
 N = Nama kegagalan  
 P = Parameter dalam mengidentifikasi kegagalan  
 PL = Prediksi lokasi kegagalan  
 SB = Sistem/komponen lain yang berkaitan dengan kegagalan  
 SP = Saran Perbaikan  
 I = Informasi mengenai kondisi ideal

SC = Suhu Cairan Pendingin

SG = Suhu Gas Buang

KK = Kecepatan Kendaraan

PM = Putaran Motor Bakar

PP = Posisi Pedal Gas

KA = Ketinggian Air Aki

KC = Ketinggian Cairan Pendingin (di Tabung Cadangan)

TO = Tekanan Oli

## 2.2 Kebutuhan Augmented Reality

AR (*Augmented Reality*) adalah penggabungan benda maya dan benda nyata di lingkungan yang nyata serta *real-time*. Terdapat 2 hal penting pada AR, yaitu *marker* dan visualisasi. Visualisasi adalah objek maya(gambar, model 3D, animasi 2D/3D) yang ditampilkan pada layar ponsel pintar atau komputer, sedangkan *marker* adalah objek nyata yang digunakan sebagai patokan posisi, rotasi, dan ukuran dalam menampilkan visualisasi [6].

AR pada penelitian ini akan diimplementasikan pada aplikasi *android*. *Unity 3D* dipilih untuk mengembangkan aplikasi *android*. *Unity 3D* adalah sebuah perangkat lunak untuk mengembangkan sebuah *game*. *Unity 3D* dapat digunakan untuk membuat sebuah game yang bisa digunakan pada komputer, *android*, *iOS*, *PlayStation*, *Xbox*, dan lain-lain [7]. Selain *game*, *Unity 3D* juga bisa digunakan untuk membuat aplikasi yang berkaitan dengan model dan animasi 3D seperti AR.

Untuk mengembangkan AR di *Unity 3D*, dibutuhkan *SDK* tambahan. *Vuforia SDK* dipilih untuk mengembangkan AR pada *Unity 3D* di penilitian ini. Pada *Vuforia*, *marker* yang

digunakan pada AR disebut dengan *target*. *Vuforia* memiliki banyak pilihan jenis *target* yang dapat digunakan, penelitian ini menggunakan jenis *ImageTarget* [8]. Untuk menghindari rancu pada pembahasan selanjutnya, *target* akan disebut dengan *objek target*.



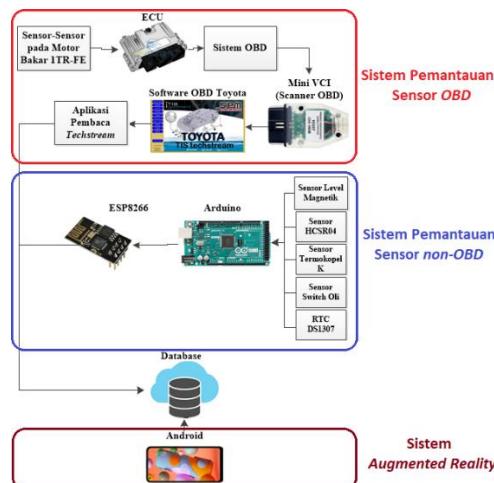
Gambar 2.7 Visualisasi dan Objek Target pada AR di *Unity 3D*

Model dan animasi 3D untuk visualisasi pada AR dibuat menggunakan *Blender 3D* sebelum dimasukkan ke *Unity 3D*. *Blender 3D* adalah perangkat lunak yang digunakan untuk membuat sebuah desain 3D. Desain 3D yang bisa dibuat tidak hanya sebuah gambar yang statis (model), tetapi juga dapat berbentuk dinamis atau animasi [9]. Model dan animasi 3D yang akan dibuat adalah benda-benda yang berkaitan dengan kondisi yang dipantau.



**Gambar 2.8** Proses Pembuatan Desain 3D pada Blender

### 2.3 Alur Kerja Perangkat Keras dan Lunak



**Gambar 2.9** Alur Kerja

Pada alur kerja, terdiri 3 sistem yang saling berhubungan. Pertama, *sistem pemantauan sensor OBD*. Sistem ini membaca kondisi kendaraan menggunakan sensor-sensor yang terhubung dengan sistem *OBD*. Sensor-sensor bawaan kendaraan membaca kondisi dan mengirim informasi ke *ECU* yang terintegrasi dengan sistem *OBD*, lalu informasi pada sistem *OBD* dibaca oleh *Mini VCI* dan ditampilkan melalui *Techstream*. Informasi kondisi yang ditampilkan oleh *Techstream* dibaca oleh *Aplikasi Pembaca Techstream* agar dapat dikirim ke *database*.

Kedua, *sistem pemantauan sensor non-OBD*. Sistem ini membaca kondisi kendaraan menggunakan sensor yang tidak terhubung dengan sistem *OBD* (sensor tambahan yang dipasang pada kendaraan). Sensor-sensor ini terhubung dengan *arduino* yang berfungsi untuk mengelola data mentah sensor menjadi data jadi yang dapat digunakan untuk aplikasi *android*. *Arduino* mengirim data jadi ke *ESP8266*, lalu *ESP8266* mengirim data tersebut ke *database*.

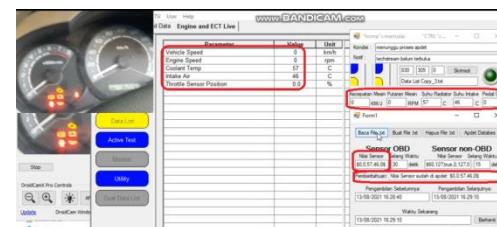
Ketiga, sistem *AR*. Sistem ini terintegrasi dengan aplikasi *android*. Sistem ini mengambil data kondisi kendaraan pada *database*, lalu mengatur visualisasi pada *AR* dan

mengidentifikasi kegagalan dari data tersebut.

### 3. HASIL DAN PEMBAHASAN

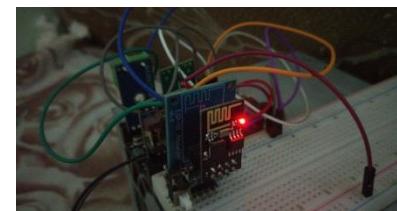
#### 3.1 Proses Pembuatan

Yang pertama dilakukan adalah pembuatan *aplikasi pembaca techstream*. Aplikasi ini dikembangkan menggunakan *VB.NET*.



**Gambar 3.1** Aplikasi Pembaca Techstream (kanan), Global Techstream (tengah)

Setelah itu, dilakukan pembuatan program dan merangkai perangkat keras *Sistem Pemantauan Sensor non-OBD*.



**Gambar 3.2** Perangkat Keras Sistem Pemantauan Sensor non-OBD

Setelah seluruh sistem pemantauan sudah berhasil membaca dan mengirim kondisi kendaraan ke *database*, selanjutnya dilakukan pembuatan aplikasi *android*.



**Gambar 3.3** Pembuatan AR pada aplikasi android

#### 3.2 Pengujian Visualisasi pada Augmented Reality

Pengujian dilakukan dengan cara

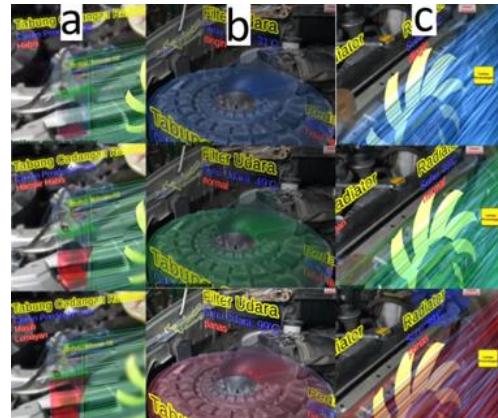
mengarahkan kamera *android* ke objek target terdekat lalu mengarahkan kamera ke objek yang ingin ditampilkan visualisasi nya. Berikut adalah hasil pengujianya.

**Tabel 3.1** Hasil Pengujian Visualisasi

No	Objek Visualisasi	Hasil Pengujian						Kesalahan	
		Pengujian Ke-							
		1	2	3	4	5	6		
1	Aki Basah	✓	✓	-l	✓	-l	✓	33%	
2	Alternator	✓	-j	-j	✓	✓	✓	33%	
3	Filter Oli	✓	-p	✓	-p	✓	-l	50%	
4	Filter Udara	-p	✓	✓	✓	✓	✓	17%	
5	Header Knalpot	✓	✓	✓	✓	-j	✓	17%	
6	Kopel Gardan	✓	✓	✓	✓	✓	✓	0%	
7	Kruk As & Piston	✓	-p	✓	✓	✓	✓	17%	
8	Pedal Gas	✓	-l	✓	-l	-l	✓	50%	
9	Radiator	✓	-j	-j	✓	✓	✓	33%	
10	Switch Oli	✓	✓	✓	✓	✓	-p	17%	
11	Wadah Cairan Pendingin Cadangan	✓	-c	✓	-p	-p	✓	50%	
12	Termostat	✓	✓	✓	✓	✓	✓	0%	
13	Throttle Body	✓	-p	-p	✓	✓	✓	33%	
<b>Rata-Rata Kesalahan</b>						<b>27%</b>			

#### Keterangan:

- ✓ = visualisasi tampil dan sesuai
- -p = visualisasi tidak tampil karena perbedaan arah kamera terhadap objek target terlalu besar
- -l = visualisasi tidak tampil karena lensa kamera tidak fokus secara otomatis ketika arah kamera berpindah
- -j = visualisasi tidak tampil karena jarak objek target terlalu dekat atau jauh dari kamera
- -c = visualisasi tidak tampil karena cahaya disekitar kamera terlalu terang atau gelap



**Gambar 3.4** Visualisasi cairan pendingin (a), suhu intake (b), suhu cairan pendingin (c)



**Gambar 3.5** Visualisasi putaran kruk as (d), katup termostat (e), switch oli (f)

Hasil pengujian menunjukkan bahwa sistem *AR* mayoritas berhasil mendeteksi objek target dan menampilkan visualisasi sesuai dengan kondisi objek nyata di kendaraan, tetapi terkadang di beberapa sampel pengujian tidak berhasil dalam mendeteksi objek. Rata-rata kesalahan yang dihasilkan dari 78 sampel percobaan adalah 27%. Terdapat 4 faktor yang penyebab sistem *AR* tidak mendeteksi objek target, tetapi faktor yang paling sering terjadi adalah perbedaan arah kamera terhadap objek terlalu besar, terjadi sebanyak 43% dari total keseluruhan pengujian yang tidak berhasil.

#### 3.3 Pengujian Kegagalan

Pengujian ini bertujuan untuk melihat respon aplikasi *android* ketika kegagalan teridentifikasi. Pengujian dilakukan dengan cara merubah nilai pembacaan sensor sesuai dengan parameter kegagalan-kegagalan yang telah dimasukkan kedalam program untuk dapat diidentifikasi oleh sistem dan membuka panel kegagalan di aplikasi *android* untuk melihat respon dari aplikasi apakah sesuai dengan rancangan atau tidak. Berikut hasil

pengujinya.

**Tabel 3.2** Hasil Pengujian Kegagalan

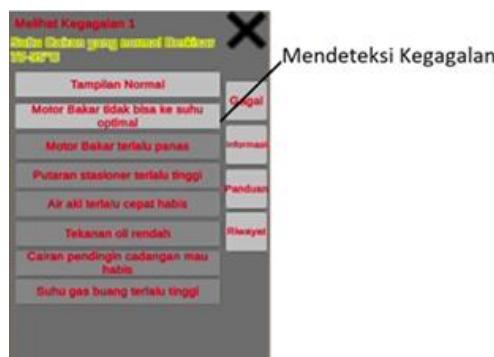
No	Kondisi	Hasil Pengujian				
		M	I	P	V	K
1	SC < 77 °C selama >5 menit saat PM > 0 rpm	✓	✓	✓	✓	0%
2	SC > 97 °C	✓	✓	✓	✓	0%
3	PM > 1000 rpm ketika PP = 0 % dan suhu CP > 77 °C	✓	✓	✓	✓	0%
4	KA turun 90% selama 1-2 bulan	✓	✓	✓	✓	0%
5	TO < 29 kPa ketika PM > 0 rpm	✓	✓	✓	✓	0%
6	KC < 20 %	✓	✓	✓	✓	0%
7	SG > 600 °C	✓	✓	✓	✓	0%
<b>Rata-Rata Kesalahan</b>		<b>0%</b>				

Keterangan:

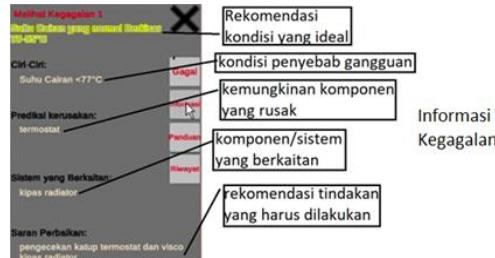
- M = Mendeteksi kegagalan
- I = Menampilkan informasi rincian mengenai kegagalan
- P = Menampilkan dokumen pedoman perbaikan
- V = Menampilkan visualisasi objek-objek yang berkaitan dengan kegagalan
- K = Presentase kesalahan



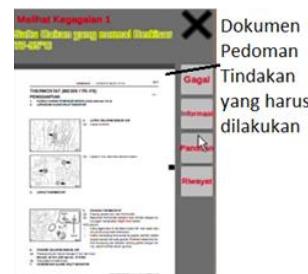
**Gambar 3.6** Cuplikan tampilan saat pengujian



**Gambar 3.7** Cuplikan pengujian pendekslan kegagalan



**Gambar 3.8** Cuplikan pengujian respon penampilan informasi kegagalan



**Gambar 3.9** Cuplikan pengujian respon penampilan dokumen pedoman perbaikan

Setelah dilakukan pengujian sebanyak 7 sampel kegagalan, aplikasi *android* berhasil mendeteksi seluruh kegagalan sesuai parameter tertentu yang telah ditentukan saat permrograman. Aplikasi *android* juga berhasil menampilkan informasi-informasi penting mengenai kegagalan, menampilkan dokumen pedoman perbaikan, dan menampilkan visualisasi AR sesuai kegagalan yang terdeteksi. Kesalahan yang dihasilkan adalah 0%.

## 4. PENUTUP

### 4.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, kesimpulan yang dapat diambil yaitu sebagai berikut.

1. Sistem AR gagal menampilkan visualisasi sebanyak 21 kali dari 78 total sampel pengujian visualisasi. Kesalahan yang dihasilkan adalah 27%.
2. Terdapat 4 faktor yang menyebabkan sistem AR gagal mendeteksi objek target, diantaranya adalah sebagai berikut.
  - perbedaan arah kamera terhadap arah permukaan objek target terlalu besar.
  - lensa kamera tidak fokus secara otomatis.
  - jarak objek dengan kamera terlalu dekat atau jauh.
  - cahaya disekitar kamera terlalu

terang atau gelap.

3. Aplikasi *android* berhasil mendeteksi seluruh kegagalan yang telah ditentukan sesuai dengan rancangan, menampilkan rekomendasi kondisi yang ideal, menunjukkan informasi kegagalan (ciri-ciri, prediksi komponen yang mengalami kegagalan, sistem yang berkaitan, rekomendasi tindakan untuk memperbaiki), dan dokumen panduan tindakan untuk memperbaiki. Hasil pengujian kegagalan menghasilkan kesalahan 0%.

#### 4.2 Saran

Terdapat beberapa hal yang harus diperhatikan dan dikembangkan lagi untuk penelitian selanjutnya yaitu untuk mengatasi 4 faktor yang menyebabkan sistem *AR* gagal mendeteksi objek target, dengan cara otomatis atau terprogram.

#### 5. DAFTAR PUSTAKA

- [1] Badan Pusat Statistik. (2019, Januari) Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis, 1949-2018. [Online]. [www.bps.go.id/linkTableDinamis/vie w/id/1133](http://www.bps.go.id/linkTableDinamis/view/id/1133)
- [2] Ghulam Muhammad Nayazri. (2017, Desember) Indonesia Masih Jadi Pasar Sepeda Motor Ketiga di Dunia. [Online]. [www.otomotif.kompas.com/read/2017/12/02/082200615/indonesia-masih-jadi-pasar-sepeda-motor-ketiga-di-dunia](http://www.otomotif.kompas.com/read/2017/12/02/082200615/indonesia-masih-jadi-pasar-sepeda-motor-ketiga-di-dunia)
- [3] Arif Arianto. (2011, Agustus) Kendaraan Bermotor di Indonesia Terbanyak di ASEAN. [Online]. [www.gooto.com/read/352572/kendaraan-bermotor-di-indonesia-terbanyak-di-asean](http://www.gooto.com/read/352572/kendaraan-bermotor-di-indonesia-terbanyak-di-asean)
- [4] Ulva Erida Nur Rochmah and Aris Rakhmadi, "PENGUNAAN AUGMENTED REALITY UNTUK MENSIMULASIKAN DEKORASI RUANGAN SECARA REALTIME," *Techno.COM*, vol. 15, no. 4, pp. 312-319, November 2016.
- [5] Paul Baltusis, "On board vehicle diagnostics," *SAE Technical Paper*, no. 2004-21-0009, 2004.
- [6] Martin Hirzer, "Marker Detection for Augmented," *Seminar/Project Image Analysis Graz*, vol. 25, Oktober 2008.
- [7] Alex Okita, *Learning C# Programming with Unity 3D*, 2nd ed. Boca Raton: A K Peters/CRC Press, 2019.
- [8] Vuforia. Getting Started with Vuforia Engine in Unity. [Online]. <https://library.vuforia.com/articles/Training/getting-started-with-vuforia-in-unity.html>
- [9] Jubilee Enterprise, *BLENDER UNTUK PEMULA*. Jakarta: Elex Media Komputindo, 2016.