

Pengembangan Aplikasi *DVR Driving* dengan Fitur *Remote Live Streaming* dan *Speedometer* Berbasis *Android*

Nico Gufron, Justinus Andjarwirawan, Agustinus Noertjahyana
Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) – 8417658
E-Mail: nicogufron@gmail.com, agust@petra.ac.id, justin@petra.ac.id

ABSTRAK

Di era sekarang, kendaraan bermotor sudah membantu manusia dalam kegiatan sehari-harinya, membantu manusia untuk mengantarkan barang, ataupun orang. Tetapi ketika dalam perjalanan, ada musibah yang menunggu, entah dalam kecelakaan antar kendaraan atau pun kejadian lainnya.

Oleh karena itu, dilakukan sebuah pengembangan aplikasi DVR berbasis Android, dilengkapi dengan tambahan fitur yaitu *GPS*, *Speedometer* dan *Live Streaming*. Dengan aplikasi ini, dapat merekam *video* dan mengambil foto disertai dengan lokasi dan deskripsi. Lebih dari itu, adanya fitur *live streaming* untuk membantu pengguna lainnya agar dapat bergabung langsung dengan memasukkan nama *channel*. *Live streaming* dibantu oleh *library Agora Video Broadcasting*.

Pengujian dilakukan dengan menambahkan aplikasi ke dalam *smartphone device* yang berbeda-beda.

Kata Kunci: *DVR*, *Live streaming*, *Agora Video Broadcast*, *GPS*, *Speedometer*

ABSTRACT

On this era, motor vehicles already help people on daily basis, helping people to deliver goods, or transporting people. But on the way, there is a disaster waiting, either accidents on other vehicles or other accidents.

Therefore, in this thesis will develop an application for DVR based on Android, equipped with additional features, such as GPS, Speedometer, and Live Streaming. In this application, users are able to record and take pictures with location and description about the videos and photos they take. Furthermore, live streaming feature added to help other users to join live streaming just entering channel name. Live streaming library helped by Agora Video Broadcasting library.

The tests were carried out by adding applications to different smartphone devices

Keywords: *DVR*, *Live streaming*, *Agora Video Broadcast*, *GPS*, *Speedometer*

1. PENDAHULUAN

Pada zaman sekarang, kendaraan sudah menjadi hal yang penting untuk umat manusia di kegiatan sehari-harinya, dari transportasi barang, atau berpergian dari satu tempat ke tempat lainnya dengan mudah. Tetapi dari semua hal yang telah membantu manusia, ada musibah yang menunggu. Menurut

Korlantas Polri, jumlah kecelakaan selama triwulan terakhir adalah dengan rata-rata sekitar 26,800 kasus kecelakaan berkendara. Untuk merekam bagaimana kejadian kecelakaan tersebut, adanya teknologi *DVR*, yang singkatan dari *Digital Video Recorder*, merupakan sebuah perangkat elektronik yang merekam *video* ke dalam *format digital* ke dalam *disk drive*, *USB Flash Drive*, *SD Memory Card*, *SSD* atau tempat penyimpanan lainnya.

Untuk membantu pengguna agar merekam perjalanan mereka dengan *DVR* berbasis *Android*, dikembangkan sebuah aplikasi *DVR Driving* dengan fitur *Remote Live Streaming* dan *Speedometer* berbasis *Android*. Aplikasi ini dikembangkan untuk dapat *live streaming* dan merekam perjalanan, sehingga pengguna lain mampu bergabung *live stream* tersebut dengan memasukkan nama *channel*, dan merekam lokasi serta melacak pengemudi dari jarak jauh secara *real time*, serta dibuat sebuah *speedometer* untuk melacak atau membatasi kecepatan dengan memberikan notifikasi *popup* untuk mengurangi kecepatan agar pengemudi dapat berhati-hati.

2. LANDASAN TEORI

2.1 Digital Video Recorder (DVR)

DVR merupakan sebuah perangkat elektronik yang mampu merekam *video* dalam *format digital*, dan disimpan dalam *disk drive*, *USB Flash Drive*, *SD Memory Card*, dan tempat penyimpanan lainnya. *Recorder* tersebut dihubungkan dengan *media input device*, seperti *satellite* atau kotak kabel dan televisi, *unit* ini merupakan tempat penyimpanan yang besar dengan *file management software* dan sebuah *remote control* untuk merekam sebuah acara. *DVR* dibuat pada tahun 1999 yang bernama *ReplayTV* dan *TiVo* di Las Vegas, Nevada.

PVR adalah sebuah pelanggan jaringan televisi yang menerima layanan lainnya dari jaringan televisi layanan. *PVR* disesuaikan untuk memberikan akses bersyarat ke layanan yang direkam.[2]

2.2 Global Positioning System (GPS)

GPS adalah sebuah *global navigation satellite system* yang menyediakan lokasi, kecepatan dan sinkronisasi waktu. *GPS* menggunakan setidaknya ada 24 satelit, sebuah penerima dan algoritma yang menerima lokasi, kecepatan dan sinkronisasi untuk udara, laut dan daratan. Secara umum, ada 5 kunci penting dalam *GPS*, antara lain:

1. Lokasi, untuk menentukan posisi
2. Navigasi, mendapatkan satu lokasi ke lainnya
3. Melacak, memonitor sebuah objek atau pergerakan
4. Pemetaan, membuat peta untuk dunia

5. Waktu, memungkinkan untuk melakukan pengukuran waktu yang tepat.

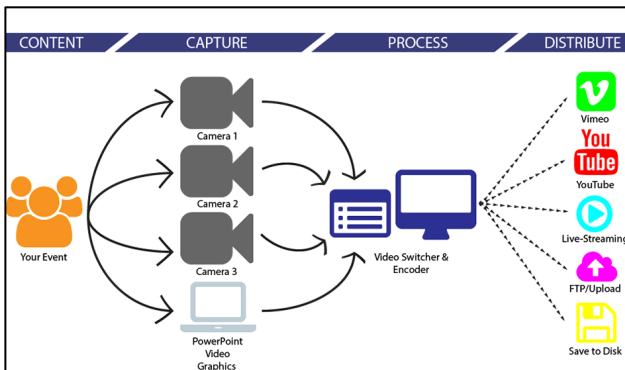
GPS sangat dibutuhkan untuk zaman sekarang dan digunakan di banyak industri untuk mempersiapkan survei dan peta, mengambil perkiraan waktu yang tepat melacak posisi atau lokasi dan navigasi.[6]

2.3 Speedometer

Speedometer atau speed meter adalah sebuah alat pengukuran yang mengukur dan menampilkan kecepatan sesaat pada kendaraan. Sekarang sudah ditempatkan diseluruh kendaraan bermotor, *speedometer* mulai tersedia sebagai opsi di tahun 1900, dan sebagai peralatan standar dari tahun 1910 dan seterusnya.[5]

2.4 Live Streaming

Streaming merupakan sebuah teknologi untuk memanikan *file video*, atau *audio* secara langsung ataupun dengan perekam dari sebuah mesin *server (web server)*. Dengan kata lain, *file video* yang berada pada mesin *server* atau *computer server* dapat langsung dijalankan dan dimainkan oleh klien sesaat proses permintaan dilakukan sehingga klien tidak perlu menghabiskan sekian waktu untuk *download file video* dan *audio* tersebut.[1] Cara kerja *live streaming* dapat dilihat pada Gambar 1.



Gambar 1. Live Streaming

Beberapa hal yang diperlukan ketika menjalankan sebuah *live streaming* berdasarkan dari Gambar 1, antara lain:

1. Kamera, menjadi hal utama yang digunakan untuk menjalankan *live streaming*, dan menjadi hal utama untuk mendapatkan gambar dan suara.
2. *Encoder*, merupakan sebuah *device* atau aplikasi yang mengambil konten dan mengubahnya menjadi *digital format* menjadi sebuah *stream*.
3. *Internet Connection*, koneksi *internet* dibutuhkan untuk dapat melakukan *streaming* dan membutuhkan *bitrate*.

Bitrate adalah jumlah *bit* yang diproses per *unit* waktu[4]. Dalam Gambar 2, untuk mendapatkan resolusi tertentu dibutuhkan *minimum upload speed*, sehingga jika ingin mendapatkan resolusi yang lebih tinggi harus memiliki *upload speed* yang berbeda.

Resolution	Recommended Upload Speed*	Minimum Upload Speed*
480p30	3 Mbps	600 Kbps
720p30	6 Mbps	125 Mbps
1080p30	13 Mbps	2.75 Mbps

*These values are estimates based on stable network environments, calculating exact bandwidth requirements is dependent on the type of content, encoder and a number of other factors.

Mbps: Megabits per second

BXCAST

Gambar 2. Bitrate upload speed

Dalam penelitian Mantoro. T, dkk, dilakukan sebuah pengembangan *mobile application* yang mengutilisasi *digital broadcast media server* dari tvOne. Penelitian ini menggunakan *Wowza Media Server 3* untuk *media streaming server*[7]. Penelitian ini mengutamakan penggunaan *Wowza Media Server 3* sebagai *media server* untuk mendapatkan hasil performa dari *streaming* tersebut melalui *Quality of Service*.

Penelitian lainnya berdasarkan I Diwi, dan dkk, dilakukan sebuah penelitian tentang analisis kualitas layanan *video live streaming* pada jaringan lokal Universitas Telkom. Penelitian ini bertujuan untuk mengetahui kebutuhan *bandwidth* untuk layanan *video live streaming*. Metode yang digunakan dalam penelitian ini dengan mengadakan sebuah pengukuran untuk kerja jaringan secara langsung di lapangan[3]. Hasil dari penelitian ini melakukan pengujian pada performansi *Quality of Service* dalam implementasi *live streaming* menggunakan *network analyzer Wireshark*, serta dengan adanya *frame* hingga 15 fps, *jitter* dan *throughput* yang besar.

2.5 Agora.io

Agora.io adalah sebuah *service provider* untuk *real-time* secara *global* untuk *video* dan *audio*. Teknologi utamanya adalah sebuah kemungkinan secara *real-time* atau *Real Time Communication (RTC)*, dimana menggunakan suara secara *real-time* dan layanan dengan komunikasi secara *offline*. [8]

Agora.io memiliki sebuah produk bernama *Agora Video Broadcasting*, dimana produk ini memprioritaskan kualitas suara yang tinggi. Pengguna dapat menjadi seorang *broadcaster* ataupun *audience*, dimana *broadcaster* yang mampu berbicara, dan jika seorang pengguna ingin berbicara maka harus mengubah perannya menjadi *host*.

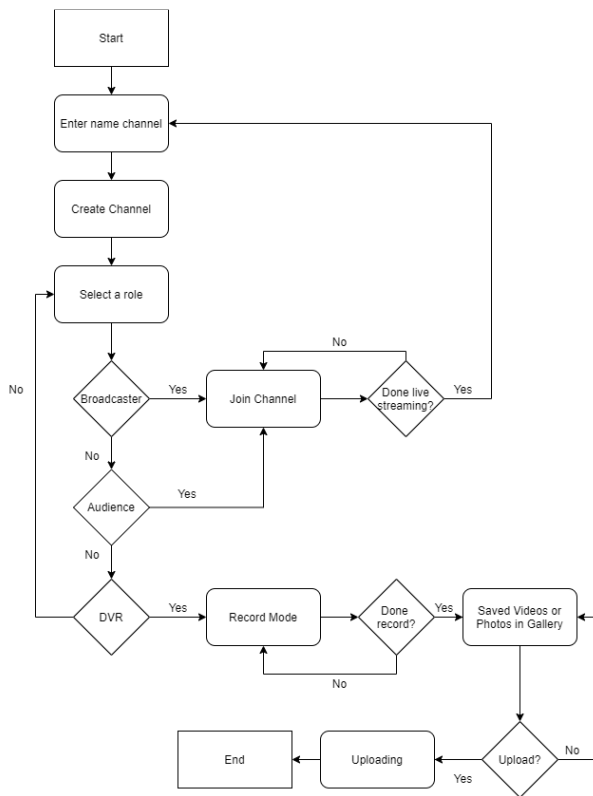
3. DESAIN SISTEM

3.1 Desain Arsitektur Sistem

Desain sistem akan ditambahkan dengan beberapa komponen, antara lain:

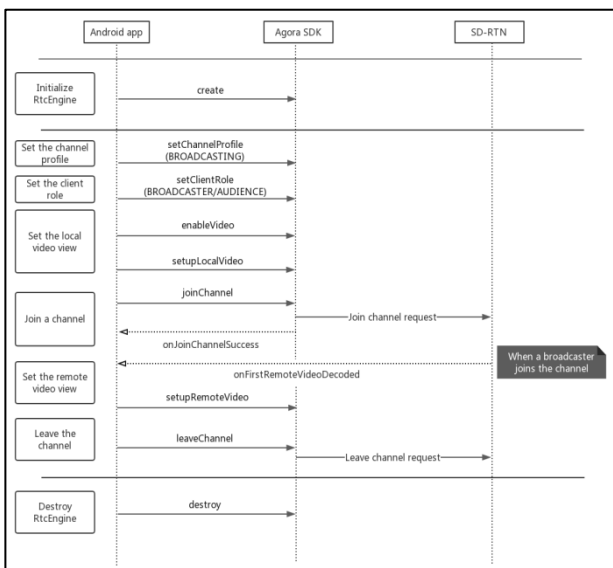
- *Start Record*,
- *Live Stream*,
- *Settings*

Desain sistem ini dapat dilihat *workflow program* seluruhnya, hal ini dapat dilihat pada Gambar 3.



Gambar 3. Workflow Program

Ketika sistem berjalan, *smartphone* akan mengambil lokasi *latitude* dan *longitude* dari GPS untuk sebagai koordinat awal pengemudi memulai *live stream*, lalu Google Maps akan menampilkan lokasi peta pada aplikasi. Ketika pengguna ingin menggunakan fitur *live stream*, pengguna harus memasukkan nama *channel* yang unik dan dapat memilih *role* sebagai *broadcaster* dan *audience*. Arsitektur untuk *live streaming* dapat dilihat pada Gambar 4.



Gambar 4. Langkah inisialisasi Agora Video Broadcast

Dalam Gambar 4, dijelaskan bagaimana *live streaming* bekerja. Ada beberapa hal yang harus dilakukan ketika melakukan *live streaming*, antara lain: menginisialisasi RTC

Engine, menambahkan sebuah *channel profile*, mengatur sebuah *role*, mengatur tampilan dari kamera *local video*, bergabung dengan *channel*, pengaturan *remote video*, melakukan *leave channel*, dan *destroy*.

Ketika pengguna menggunakan *role DVR*, pengguna mampu merekam perjalanan ataupun mengambil foto, lalu disimpan di dalam *Gallery*, lalu pengguna dapat melakukan *upload* ke dalam *website*, dalam bentuk foto ataupun *video*.

4. IMPLEMENTASI SISTEM

4.1 Proses Pengujian

Sebelum melakukan proses pengujian, *smartphone* harus memiliki spesifikasi minimum agar aplikasi dapat berjalan secara optimal, yaitu:

- *Operating System: Android 6.0* dan seterusnya
- *CPU: Qualcomm MSM8992, Snapdragon 808*, dan seterusnya.
- *Front Camera: 12 MP* atau lebih.
- *Rear Camera: 5 MP* atau lebih.
- *Memory: 3GB RAM* dan seterusnya.

4.2 Hasil Pengujian

Pada penelitian ini, tujuan yang ingin dicapai adalah pengguna mampu melakukan *live stream* dengan *smartphone* yang berbeda-beda, mendapatkan lokasi dan melacak pengemudi, mendapatkan kecepatan sebagai *speedometer*, dan adanya *notification* untuk mengingat pengemudi agar berhati-hati di jalan. Pengujian fitur *live stream* ini akan dinilai dengan 3 kategori, antara lain:

- Baik, kualitas *video* lancar dan tidak patah-patah
- Sedang, kualitas *video* lancar dan patah-patah
- Buruk, kualitas *video* tidak lancar dan patah-patah

Tabel 1. Pengujian live stream

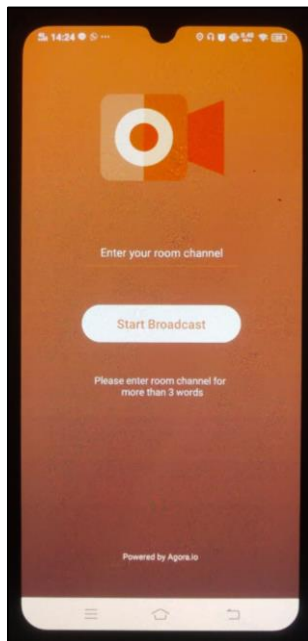
Resolusi	Delay	FPS	Bitrate (kbps)	Upload Speed (kbps)	Download Speed (kbps)	Keterangan
320 x 420	1	15	200	60 – 90	75 – 85	Sedang
480 x 360	1	30	980	130 – 160	135 – 150	Sedang
640 x 360	1	30	1.200	160 – 180	160 – 185	Sedang
640 x 480	1	30	1.500	190 – 210	190 – 245	Sedang
960 x 720	2	30	2.760	300 – 415	250 – 280	Baik
1280 x 720	2	30	3.420	350 – 500	250 – 270	Baik

Berdasarkan dari Tabel 1, pengguna yang memiliki *role* sebagai *audience* dapat menyesuaikan resolusi dengan apa yang telah dipilih oleh *broadcaster*, oleh karena itu *live streaming* memiliki fluktuasi beberapa puluh kbps ketika sedang melihat *live stream* tersebut, begitu juga dengan *broadcaster*. *Bitrate* untuk *Video Broadcasting* dua kali lipat dari *bitrate* pada umumnya, dikarenakan untuk fokus ke kualitas *video*.

Adanya penggunaan data yang drastis untuk *audience*, dalam *audience* tetap bertahan kualitas yang tinggi untuk 1280 x 720 dengan data 250 kbps hingga 270 kbps sedangkan untuk *broadcaster* mengirimkan data yang besar untuk *broadcast*.

Dalam kolom *Broadcaster* adalah seberapa banyak data yang digunakan ketika sedang *live stream*, atau *upload live stream* tersebut secara *real-time*, sedangkan untuk *audience*, dalam kolom *Audience* dapat dilihat seberapa banyak kbps yang dipakai untuk melihat *live stream* tersebut, *live stream* memiliki *delay* sekitar 2 detik.

Sebelum menjalankan aplikasi *live stream*, pengguna harus terlebih dahulu memasukkan sebuah *channel name* dengan sembarang, dengan minimal 3 huruf, terdiri dari angka, huruf, dan simbol, contoh: { . }, (,), [,], dll. Untuk pengujian halaman mengisi *channel name* dapat dilihat pada Gambar 5.



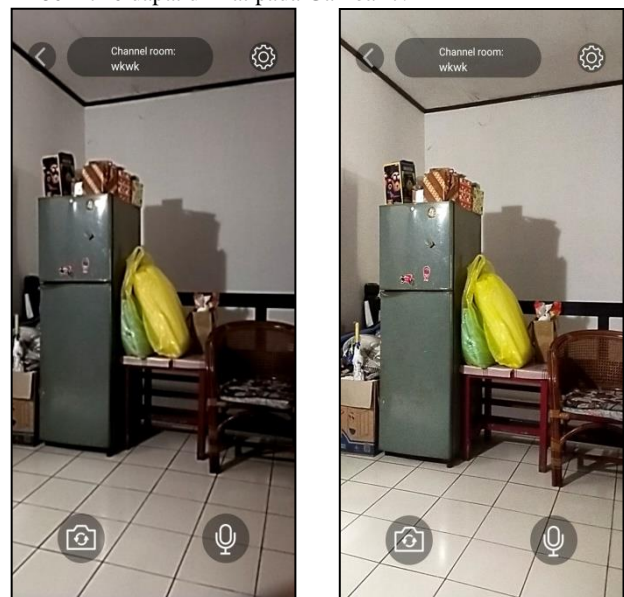
Gambar 5. Pengujian *Channel Name*

Lalu, ketika pengguna telah memasukkan *channel name* sembarang, pengguna diarahkan ke halaman lainnya untuk memilih sebuah *role*, antara lain: *broadcaster*, *audience*, dan *DVR*. Tiap *role* memiliki fungsinya sendiri, khusus untuk *live stream* harus memilih *Broadcaster*, atau *Audience*. Sedangkan untuk pengguna yang ingin merekam perjalanannya, harus memilih *role DVR*. Pengujian halaman ini dapat dilihat pada Gambar 6.



Gambar 6. Pengujian Halaman Memilih *Role*

Resolusi yang berbeda mampu mempengaruhi kejelasan dari *live stream*, sehingga rekomendasi resolusi yang terbaik adalah 1280 x 720. Perbedaan kualitas resolusi dari 640 x 480 dan 1280 x 720 dapat dilihat pada Gambar 7.



Gambar 7. Resolusi 640 x 480 dan 1280 x 720

Berdasarkan dari Gambar 7, perubahan resolusi dari 640 x 480 ke 1280 x 720 memiliki perubahan yang drastis. Perubahan warna yang terlihat jelas, kabur / *blur* untuk resolusi berukuran 640 x 480 daripada resolusi berukuran 1280 x 720.



Gambar 8. Empat Broadcaster

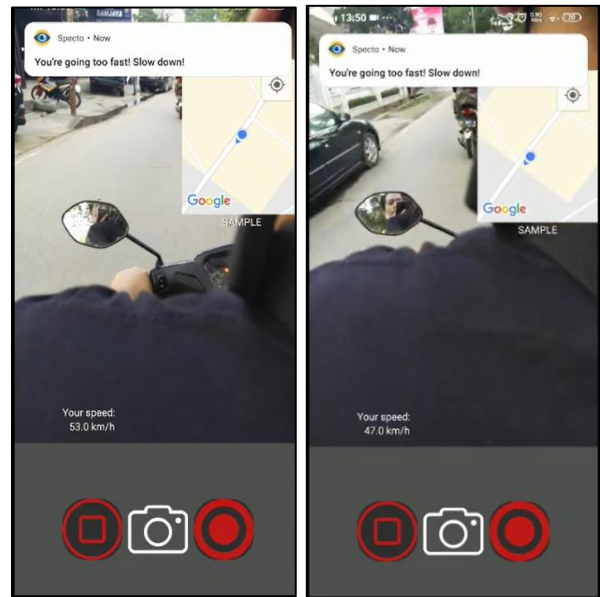
Berdasarkan dari Gambar 8, adanya pengujian yang membutuhkan 4 *broadcaster*, dimana semua *broadcaster* memiliki resolusi yang sama, yaitu 1280 x 720 dan memasuki satu *channel name* secara bersamaan. Lalu *broadcaster* memiliki sebuah fungsi yaitu *Switch Camera*, dan *Mute*. Fungsi *Switch Camera* berguna untuk membalikkan kamera dari *front camera* ke *rear camera*, sedangkan untuk fungsi *Mute*, berguna untuk mematikan suara.

Ketika pengguna ingin merekam perjalanan mereka, pengguna harus memilih *role DVR* untuk terlebih dahulu, aplikasi mengarahkan pengguna ke halaman *DVR* dimana pengguna dapat merekam atau mengambil foto dalam aplikasi lalu disimpan dalam *Gallery*.

Ketika pengguna di dalam halaman *DVR*, pengguna dapat melihat sebuah tombol *Settings*, di dalam halaman tersebut terdapat sebuah resolusi yang tersedia hanya untuk *live stream*, dan di bawahnya terdapat juga sebuah dua tombol untuk melihat isi *folder* dari masing-masing foto dan *video* yang telah diambil sebelumnya.

Apabila pengguna ingin mengambil foto, dapat dilakukan hanya dengan menggunakan *Photo Mode*, tombol tersebut terdapat pada ditengah antara tombol *stop* dan *record* yang ada di halaman *Record Mode*.

Dalam halaman *DVR*, aplikasi melacak pengguna sedang berada dimana menggunakan bantuan *GPS*, lalu menghitung kecepatan gerak, lalu adanya notifikasi untuk berhati-hati ketika pengguna melebihi dari kecepatan yang telah ditentukan.. Pengujian fitur ini dapat dilihat pada Gambar 9.



Gambar 9. Pengujian Fitur Notifikasi, Melacak, dan Kecepatan

Berdasarkan dari Gambar 9, telah dilakukan sebuah pengujian notifikasi untuk memberitahukan pengguna agar membawa kendaraan lebih pelan. Dalam gambar tersebut, jika pengguna membawa kendaraan melebihi dari kecepatan yang telah ditentukan, yaitu 35 km/h, maka aplikasi memberikan sebuah notifikasi yang memberitahukan pengguna agar membawa kendaraan lebih lambat dan berhati-hati.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan dari hasil pengujian yang dilakukan pada sistem, maka dapat disimpulkan bahwa:

1. *Live streaming* dapat dilakukan dengan baik hanya dengan memasukkan nama *channel*. Rekomendasi resolusi *live streaming* untuk terlihat jelas dan baik dengan memilih resolusi 1280 x 720.
2. Lokasi pengemudi dapat dilacak dengan menggunakan bantuan *GPS* setelah mendapatkan *longitude* dan *latitude*, serta mendapatkan kecepatan.
3. Aplikasi ini mampu mengambil foto dan video dengan hasil yang jelas.
4. *Live streaming* tidak dapat merekam perjalanan, sehingga dibuat sebuah alternatif baru dengan *role DVR*, yang berguna untuk merekam perjalanan dan mengambil foto.

5.2 Saran

Saran yang dapat diberikan untuk menyempurnakan dan mengembangkan program ini lebih lanjut antara lain:

1. Fitur *live stream* dapat dikembangkan lagi untuk merekam dan mengambil foto ketika sedang dalam perjalanan.
2. Pengembangan teknologi untuk *Camera2 API* dalam *smartphone* yang berbeda, kebanyakan dari *smartphone* tidak mendukung *API* tersebut.

6. DAFTAR PUSTAKA

- [1] Agil S, Alief Chandra. 2010. Cara Cepat Bikin Live TV di Blog dan Website. Yogyakarta, Indonesia: C.V.Andi Offset.
- [2] Akins, G. 2003, March 6. Conditional Access Personal Video Recorder. Retrieved from <https://patents.google.com/patent/US20040177369A1/en>. (25/11/2019 20:59).
- [3] Diwi, A. I., Mangkudjaja, R.R., & Wahidah, I. 2014. Analisis Kualitas Layanan Video Live Streaming pada Jaringan Lokal Universitas Telkom. *Buletin Pos dan Telekomunikasi*, 12(3), 207-216.
- [4] Gupta, Prakash C. 2006. Data Communications and Computer Networks. PHI Learning
- [5] Harris, Williams. 2007, 10 July. How Speedometers Work. Retrieved from <https://auto.howstuffworks.com/car-driving-safety/safety-regulatory-devices/speedometer.htm>. (28/11/2019 22:20).
- [6] Kyes, J. 2017.<https://www.geotab.com/blog/what-is-gps/>
- [7] Mantoro, T., Ayu, M. A., & Jatikusumo, D. 2012, August. Live video streaming for mobile devices: An application on android platform. In 2012 2nd International Conference on Uncertainty Reasoning and Knowledge Engineering (pp. 119-122). IEEE.
- [8] Zekun, W. 2019, 12 April. How Agora.io Became the Leading Player in Video and Voice. Retrieved from <https://equalocean.com/ai/20190412-dancing-with-the-video-and-voice-a-story-of-agoario>. (14/06/2020 12:29)