

## RANCANG BANGUN SISTEM PENILAIAN UJIAN PRAKTIK SIM C MENGGUNAKAN PROTOKOL KOMUNIKASI PESAN MQTT BERBASIS ANDROID

Dimas Adiputra Rahadyan<sup>1)</sup> Harianto<sup>2)</sup> Weny Indah Kusumawati<sup>3)</sup>

Program Studi/Jurusan Teknik Komputer

Universitas Dinamika

Jl. Raya Kedung Baruk 98 Surabaya, 60298

Email: 1)[14410200047@dinamika.ac.id](mailto:14410200047@dinamika.ac.id), 2) [harianto@dinamika.ac.id](mailto:harianto@dinamika.ac.id), 3) [weny@dinamika.ac.id](mailto:weny@dinamika.ac.id)

**Abstrak:** SIM C merupakan salah satu tanda yang wajib dipunyai oleh seseorang apabila ingin berkendara khususnya untuk sepeda motor. Orang yang mempunyai SIM C dianggap sudah mengetahui aturan-aturan dalam berkendara. Untuk mendapatkan SIM C harus melalui serangkaian tes yang dilakukan dari ujian tertulis sampai dengan ujian praktik. Pada ujian praktik, peserta melalui beberapa jenis ujian dimana petugas kepolisian melihat praktik tersebut dari pos penguji. Hal tersebut tidak efisien karena kemungkinan pemantauan tersebut terlewatkan. Pada penelitian sebelumnya yang dilakukan oleh Aron Octavianus Siahaan telah meminimalisir kerja kepolisian pada saat ujian praktik SIM C tersebut dengan membuat aplikasi di komputer kepolisian yang menampilkan hasil ujian praktik. Namun hal tersebut dirasa masih kurang, sehingga penulis membuat aplikasi berbasis Android dengan memanfaatkan metode komunikasi MQTT dalam pengiriman datanya. Dengan adanya aplikasi tersebut, pihak petugas kepolisian dapat melakukan pemantauan ujian secara *mobile*, sehingga tidak harus dilakukan di dalam pos pengujian. Aplikasi ini menampilkan hasil ujian praktik pada perangkat Android. Hasil yang ditampilkan berupa lampu yang menunjukkan tiap patok pada ujian. Lampu-lampu ini menyala jika patok ujian tersengkol. Aplikasi ini juga menampilkan hasil waktu dari lamanya ujian praktik. Data-data tersebut disimpan dalam sistem *database* melalui protokol *subscriber* yang mengirimkan datanya. Berdasarkan pengujian yang dilakukan sebanyak tiga kali, didapatkan tingkat keberhasilan yaitu 100% dengan tingkat kesalahan 0%. Jarak aman yang digunakan agar pengiriman data dapat berjalan dengan baik adalah maksimal 12 meter dengan waktu pengiriman data maksimal selama 0.5 detik.

### PENDAHULUAN

Kendaraan bermotor adalah salah satu kebutuhan *tersier* bagi rakyat Indonesia, tapi bagi beberapa orang adanya kendaraan bermotor adalah kebutuhan *primer*. Untuk bisa mengendarai sepeda motor membutuhkan SIM (Surat Ijin Mengemudi). SIM adalah bukti registrasi dan identifikasi yang diberikan oleh POLRI (Kepolisian Negara Republik Indonesia) kepada seseorang yang telah memenuhi persyaratan. (Website Resmi POLRI)

Dalam pembuatan SIM C, pemohon harus lulus dalam ujian teori dan praktik. Ujian teori berisikan pertanyaan pertanyaan mengenai pengetahuan dasar lalu lintas di Indonesia dan juga kasus permasalahan pada lalu lintas di Indonesia. Ujian praktik menguji kemahiran dan ketangkasan

pemohon dalam mengendarai sepeda motor di jalan raya.

Pada tahun 2017, salah satu lulusan Institut Bisnis dan Informatika STIKOM Surabaya pada Prodi Sistem Komputer, saudara Aron Octavianus Siahaan membuat penelitian rancang bangun mengenai penilaian ujian praktik SIM C yang membahas penuh mengenai sistem ujian praktik SIM C yang sudah ter-komputerisasi. Sistem ini bekerja dengan menggunakan beberapa sensor yang terhubung dengan Arduino Mega2560 dan ditampilkan pada aplikasi yang dibuat mendeteksi pelanggaran dan kecepatan kendaraan bermotor pemohon pada setiap sesi ujian praktik. Data ini diolah dan ditampilkan pada aplikasi yang *dimonitoring* oleh penguji dan membantu penguji

dalam memberikan penilaian terhadap pemohon SIM C.

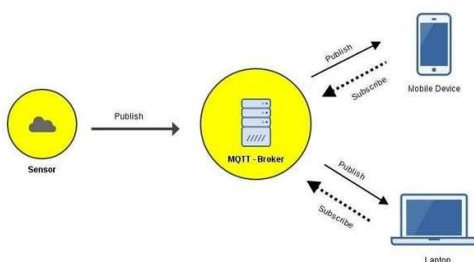
Tujuan dari penulis ialah untuk bisa mengembangkan sistem rancang bangun yang telah dibuat oleh saudara Aron Octavianus Siahaan dengan menggunakan *smartphone* atau pada aplikasi Android. Dengan harapan penelitian ini dapat membantu penguji ujian praktik SIM C dalam mengawasi peserta ujian dan menghemat tempat serta biaya operasional POLRI.

### SIM (Surat Ijin Mengemudi)

Surat izin mengemudi (SIM) adalah bukti yang diberikan oleh Polri kepada seorang yang telah memenuhi persyaratan administrasi, sehat jasmani dan rohani, memahami peraturan lalu lintas dan terampil mengemudikan kendaraan bermotor. Setiap orang yang mengemudikan kendaraan bermotor diwajibkan memiliki surat izin mengemudi sesuai dengan jenis kendaraan bermotor yang dikemudikan. (Ananto, 2014)

### Message Queuing Telemetry Transport (MQTT)

MQTT (*Message Queuing Telemetry Transport*) adalah protokol komunikasi *machine-to-machine* (M2M) "Internet of Things." Protokol ini berbasis model *publish-subscribe*, berjalan diatas TCP/IP dan mempunyai ukuran paket yang kecil. Sistem MQTT membutuhkan dua perangkat lunak utama yaitu *Client* dan *Broker*. *Client* dipasang pada perangkat keras sedangkan *Broker* berfungsi untuk menangani *publish-subscribe* data. (Equan, 2015)



Gambar 1. Jalan Komunikasi MQTT

### Android

Android merupakan sistem operasi yang berbasis linux dan bersifat kode sumber terbuka (*open source*) untuk peralatan-peralatan *mobile* seperti *smartphone* dan tablet. Tidak hanya itu, sistem operasi ini juga digunakan sebagai sistem

operasi pada *smart TV* dan perangkat-perangkat elektronik cerdas lainnya.

Sistem operasi Android menawarkan pendekatan terpadu untuk pengembangan aplikasi perangkat *mobile* yang berarti bahwa pengembang aplikasi hanya perlu mengembangkan aplikasi berbasis Android dan aplikasi tersebut dapat berjalan pada perangkat-perangkat yang berbeda yang memiliki sistem operasi Android. (Wadi)



Gambar 2. Logo Android pada Smartphone

### Android Studio

Android studio merupakan lingkungan pengembangan Android baru yang berdasarkan IntelliJ IDEA. Android studio ini mirip cara kerjanya dengan Eclipse + ADT Plugin, Android studio menyediakan alat pengembang Android yang terintegrasi dalam pengembangan dan *debugging* program. IDEA mendukung beberapa Bahasa seperti: Java, Java script, HTML/XHTML/CSS, XML/XLS, ActionScript/MXML, Phyton, Ruby/JRuby, SQL, PHP, dll. (Satyaputra & Aritohang, 2014)



Gambar 3. Logo Android Studio

### Microsoft Visual Studio Code

Visual Studio Code adalah versi Visual Studio yang ringan tetapi tetap *powerfull*. Versi ini seperti *code editor* dengan fitur-fitur tambahan untuk mempermudah penulisan kode program. Visual Studio Code tersedia pada platform Windows, Linux dan MacOS. Visual Studio Code juga mendukung banyak bahasa pemrograman

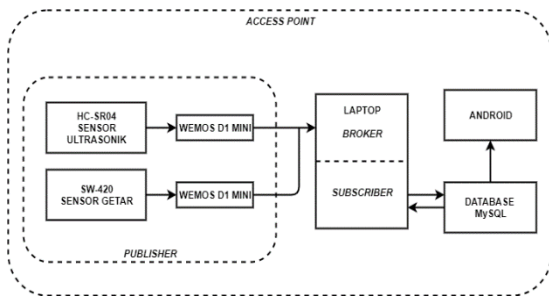
seperti halnya Visual Studio 2015 ditambah bahasa pemrograman PHP, Node.js dan lain-lain. (Faisal & Kurniawan)

### MySQL

MySQL (*My Structure Query Language*) adalah salah satu *database Management System* (DBMS). MySQL berfungsi untuk mengelola *database* menggunakan bahasa SQL. MySQL bersifat *open source*, sehingga bisa menggunakannya secara gratis. Pemrograman PHP juga sangat mendukung/*support* dengan *database* MySQL. Beberapa kelebihan yang dimiliki oleh MySQL sebagai berikut: 1. Bersifat *open source* yang memiliki kemampuan untuk dikembangkan lagi. 2. Memiliki bahasa SQL (*Structure Query Language*) yang mempunyai *standart* bahasa dunia dalam pengolahan data. 3. *Super performance* dan *reliable*, tidak bisa diragukan, pemrosesan *databasenya* sangat cepat dan stabil. 4. Sangat mudah dipelajari (*ease touse*). 5. Memiliki dukungan *support (group)* pengguna MySQL. 6. Mampu lintas platform, dapat berjalan diberbagai sistem operasi. 7. *Multiuser*, dimana MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami konflik. (Anshar, 2010)

### METODE PENELITIAN

Perancangan secara keseluruhan sistem dapat dilihat pada gambar 4 :



Gambar 4. Blok diagram sistem

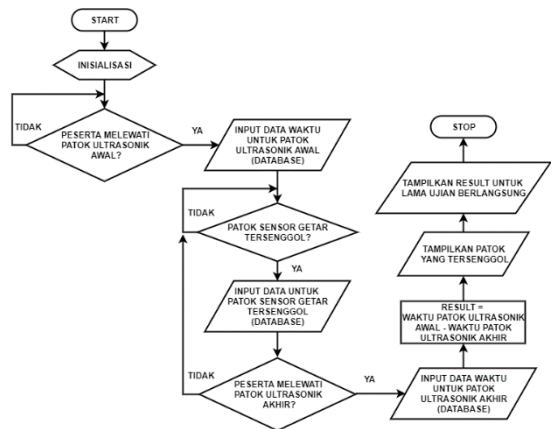
Penulis hanya berfokus pada bagian komunikasi menggunakan protokol MQTT atau *mosquitto*, pembuatan aplikasi Android, serta sinkronisasi dari protokol *mosquitto* ke aplikasi Android. Untuk perangkat masukan seperti sensor sebagai alat uji coba pada patok hanya sebagai perangkat masukan yang sudah dibuat pada penelitian sebelumnya. Patok tersebut digunakan

untuk menunjukkan hasil uji coba dan hasil sinkronisasi dari tiap protokol ke aplikasi Android.

Dapat dilihat dari blok diagram pada gambar 4 bahwa pada patok terdapat sensor getar SW-420 yang menandakan ada tidaknya kesalahan, dan sensor ultrasonic HC-SR04 yang digunakan sebagai penanda awal dan akhir sesi tiap ujian berlangsung. Data tersebut dikirim ke *broker* sebagai jembatan antara *publisher* dengan *subscriber*, lalu data tersebut diolah oleh *subscriber* sedemikian rupa, sehingga data tersebut digolongkan berdasarkan *topic* serta status data yang dibawa. Topik serta data tersebut menjadi satu informasi utuh yang disimpan dalam *database* ke dalam kolom-kolom *database* yang sudah disediakan. Kemudian data tiap *field* tersebut ditampilkan pada aplikasi Android bergantung pada apa yang ingin ditampilkan.

### Flowchart Sistem Secara Keseluruhan

Berikut adalah flowchart dari sistem secara keseluruhan.



Gambar 5. Flowchart sistem secara keseluruhan

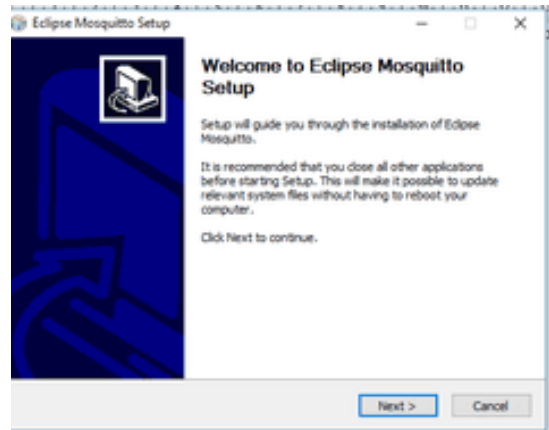
Proses dari flowchart gambar 5 adalah sebagai berikut:

1. Inisialisasi diatas adalah menyambungkan perangkat *Publisher*, *broker*, dan *subscriber* ke *Access Point* yang sama.
2. Sistem memulai dengan men-*subscribe* / men-*publish* ke *topic* yang telah ditentukan.
3. Setelah itu sensor ultrasonik awal mendeteksi adanya pergerakan. Ketika mendeteksi adanya pergerakan sistem mencatat waktu (jam, menit dan detik).
4. Waktu yang dicatat pada proses 3 dikirim dan disimpan ke dalam *database* yang sebelumnya telah diolah pada perangkat *subscriber*. Data ini masuk pada tabel sensor

- ultrasonik dengan ID 1.
5. Ketika patok sensor getar tersenggol, *subscriber* menerima data dari ID patok yang tersenggol. Lalu menyimpannya ke dalam sistem *database* pada tabel sensor getar.
  6. Sensor ultrasonik akhir yang berada digaris *finish* mendeteksi adanya gerakan ketika pengendara melewati garis. Kemudian mencatat waktu (jam, menit dan detik).
  7. Waktu yang dicatat pada proses 6 dikirim dan disimpan ke dalam *database* yang sebelumnya telah diolah pada perangkat *subscriber*. Data ini masuk pada tabel sensor ultrasonik dengan ID 2.
  8. Untuk mendapatkan waktu lamanya ujian praktik berlangsung didapatkan dari selisih antara patok sensor ultrasonik awal dan akhir.
  9. Data yang disimpan pada proses 5 diambil oleh perangkat Android lalu ditampilkan guna mengetahui kesalahan peserta yang menyenggol patok.
  10. Data yang didapat pada proses 8 ditampilkan oleh perangkat Android sebagai hasil dari lama ujian berlangsung.
  11. Sistem selesai.

### Perancangan Perangkat untuk Protokol *Mosquitto Broker*

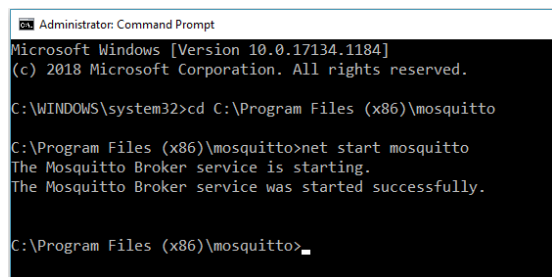
*Broker* adalah sebuah protokol yang digunakan untuk menjembatani *publisher* dengan *subscriber*. Dimana protokol ini hanya bertugas untuk menerima dan meneruskannya. Pada protokol ini tidak terdapat proses pengolahan data. Jika dalam suatu komunikasi menggunakan protokol MQTT tidak terdapat *broker*, maka komunikasi tidak terjadi atau bisa dikatakan data yang dikirim tidak tahu harus dikirim kemana, sehingga pada saat proses pengiriman data, protokol ini diwajibkan untuk aktif agar pengiriman data dapat berjalan dengan baik. Berikut adalah proses yang dilakukan untuk membuat suatu perangkat menjadi protokol *broker*. Pertama dengan menginstall protokol *mosquitto* pada komputer.



Gambar 6. Proses Instal *Mosquitto Broker*

Jika proses instalasi telah selesai, maka selanjutnya adalah dengan menjalankan *mosquitto broker* melalui *Command Prompt* (*Run as Administrator*). Proses ini mengaktifkan protokol *broker*, sehingga data yang dikirim oleh *publisher* dapat dikirim dengan baik dan diteruskan ke *subscriber* untuk diolah lebih lanjut. Selama proses komunikasi, protokol *broker* harus selalu berjalan.

Selanjutnya masuk ke tempat penyimpanan *mosquitto* yang sudah diinstall sebelumnya. Dapat dengan menyalin langsung dari perangkat komputer. Setelah masuk ke dalam penyimpanan, maka *mosquitto* dapat langsung dijalankan dengan menuliskan perintah “*net start mosquitto*” pada *command prompt*.



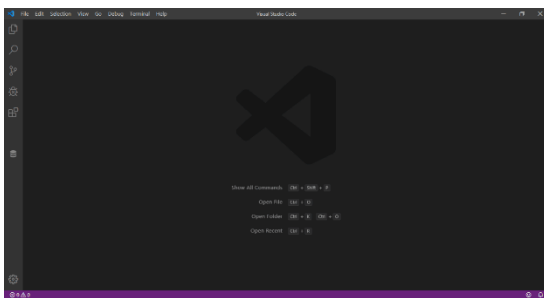
Gambar 7. Menjalankan *Mosquitto Broker*

Dapat dilihat bahwa pengaturan *broker* telah berhasil dengan adanya pesan keberhasilan “*The Mosquitto broker service was started successfully*”. Status protokol *broker* saat ini sedang berjalan, sehingga pengiriman data dapat dilakukan.

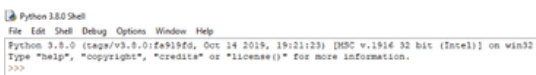
## Perancangan Perangkat untuk Protokol Subscriber

*Subscriber* adalah protokol yang digunakan untuk mengolah data masukan yang di terima. Data yang diterima di simpan pada *database* bergantung pada *topic* dan isi data yang di bawa. Pada *subscriber* juga dilakukan sinkronisasi dengan *database* sebelumnya agar data yang diolah benar-benar dapat disimpan sesuai dengan *database* yang sudah dibuat. Jika tidak melakukan proses sinkronisasi, maka data tersebut hanya disimpan dalam variabel yang dibuat pada protokol *subscriber* dan tidak diteruskan dalam sistem *database*. Dalam pengaturan perangkat menjadi *subscriber* menggunakan *software* Visual Studio Code. Dimana di *software* ini juga memerlukan beberapa pengaturan agar dapat membaca protokol *mosquito* serta membaca *database* MySQL. Pengaturan yang dilakukan yaitu dengan menambah *path* untuk membaca fungsi *mosquito* agar dapat diatur menjadi *subscriber* serta *path* untuk membaca fungsi pada MySQL. Selain itu juga ditambahkan beberapa *extensions* yang digunakan untuk membuat program dengan Bahasa python, *extensions mysql*.

Tahap pertama dalam perancangan yaitu dengan menginstal *software* Visual Studio Code dan Python version 3. Visual Studio Code merupakan *worksheet* yang digunakan untuk membuat program, sedangkan *Python* adalah Bahasa yang digunakan untuk program tersebut. Dalam VSCode tidak dapat langsung membaca program *python* tersebut, sehingga harus ditambahkan *extensions* serta *path* atau *library* tambahan agar program dapat dibaca dan dijalankan dengan baik.

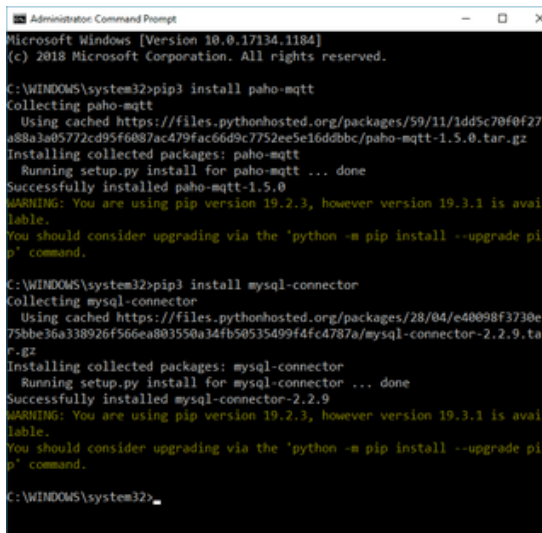


Gambar 8. Visual Studio Code

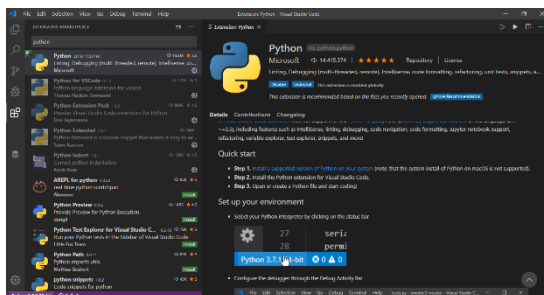


Gambar 9. Python 3

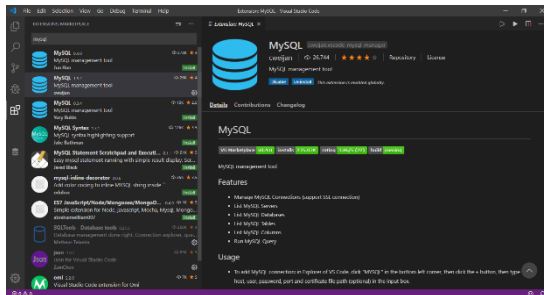
Selanjutnya dengan menambahkan *path* untuk membaca program *mosquito* serta program MySQL. Untuk menambahkan *path* dengan membuka *Command Prompt (Run as Administrator)* lalu memasukkan perintah “*pip3 install paho-mqtt*” untuk menginstall *path* untuk protokol *paho mosquito* dan “*pip3 install mysql-connector*” untuk menginstall *path* untuk pembacaan program MySQL.



Gambar 10. Menambahkan *Path* untuk MySQL



Gambar 11. Instal *Python Extensions*

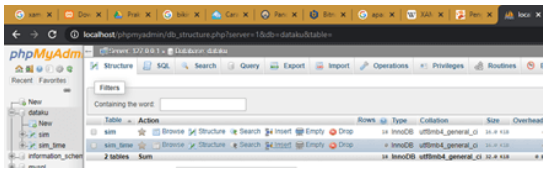


Gambar 12. Instal *MySQL Extensions*



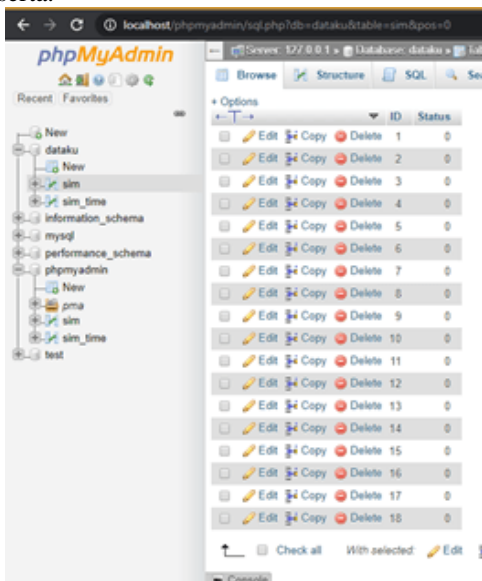
### Perancangan Sistem Database

Dalam perancangan sistem database MySQL, penulis menggunakan aplikasi XAMPP dimana di dalamnya terdapat program MySQL serta Apache yang berjalan bersama-sama. Pembuatan tabel-tabel data pada database juga dibuat melalui aplikasi ini. Dimana data tersebut dibaca dan ditampilkan pada aplikasi Android.

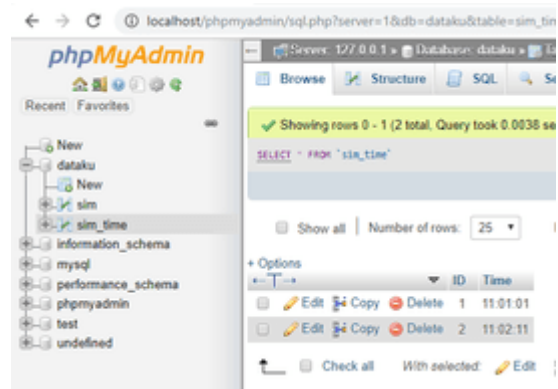


Gambar 13. Pembuatan Database

Di sini penulis membuat data sim dan sim\_time, dimana data sim menyimpan ID dari tiap patok perangkat publisher serta status dari tiap patok. Status berubah nilai dari 0 menjadi 1 apabila patok dikenai oleh peserta ujian, dan apabila tidak dikenai, maka nilainya tetap menjadi 0. Nilai tersebut berubah sesuai dengan ID patok. Misalnya patok yang dikenai adalah patok 10, maka pada database sim ID 10 dengan status 1. Pada sim\_time juga menyimpan ID dari patok pertama dan patok terakhir serta menyimpan waktu dari kedua patok tersebut. Selisih waktu dari kedua patok digunakan untuk mengetahui lama ujian praktik dilalui oleh peserta.



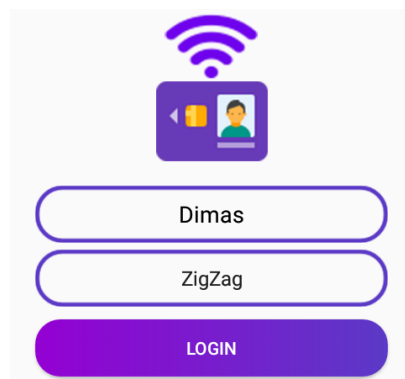
Gambar 14. Isi Data Tabel sim pada Database



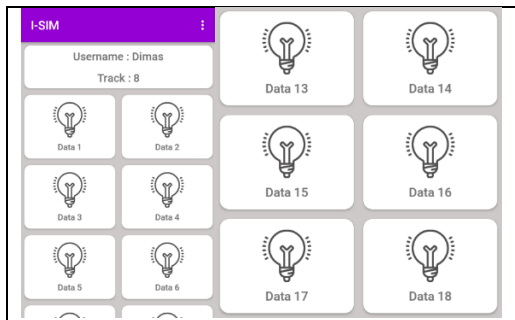
Gambar 15. Isi Data Tabel sim\_time pada Database

### Perancangan Aplikasi Android

Dalam perancangan Aplikasi Android, penulis menggunakan software Android Studio dalam pembuatan tampilan aplikasi serta sinkronisasi aplikasi dengan database. Dalam pembuatan aplikasi, hal yang harus diperhatikan adalah tipe Operating System (OS) yang dapat didukung oleh Android pengguna. Pada saat pembuatan project baru, tipe OS yang dipilih menggunakan OS 8, sehingga OS dibawah itu kesulitan dalam pengoperasian aplikasi karena tidak mendukung dalam hal spesifikasi OS nya. Apabila Android yang digunakan oleh pengguna adalah Android dibawah itu, maka aplikasi yang diinstal berjalan dengan lambat saat pengoperasiannya. Berikut adalah hasil dari perancangan aplikasi yang sudah diinstal pada perangkat Android.



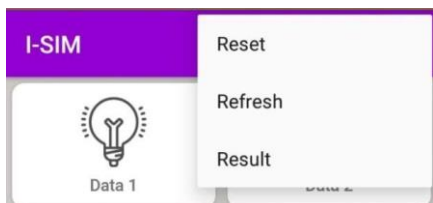
Gambar 16. Tampilan awal aplikasi



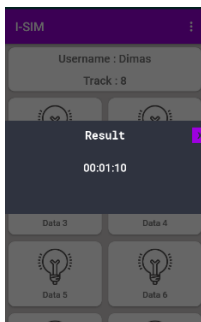
Gambar 17. Keadaan awal aplikasi saat pertama dibuka

Banyaknya lampu sama dengan banyaknya patok. Lampu-lampu tersebut menandakan hasil dari tiap penilaian ujian praktik SIM. Kondisi awal lampu adalah mati, hal itu menandakan bahwa patok yang terdapat di lapangan masih belum tersengol oleh peserta. Apabila peserta menyengol patok, lampu tersebut menyala sesuai dengan nomor patok.

Aplikasi juga dilengkapi dengan menu untuk hasil ujian yang dinyatakan dengan lamanya proses ujian berlangsung dari awal saat peserta menjalankan kendaraan sampai dengan mencapai *finish* dari masing-masing ujian praktik. Serta ada menu untuk mengulang kembali pengaturan lampu sampai dengan semua lampu mati atau dalam keadaan awal sebelum ujian. Hasil dari menu *Result* menunjukkan waktu lamanya sebuah ujian dilakukan.



Gambar 18. Menu pada Aplikasi



Gambar 19. Tampilan Menu *Result*

## HASIL DAN PEMBAHASAN

### Pengujian Protokol *Subscriber* dengan Sistem *Database*

Tujuan dari proses ini adalah agar data yang diterima dapat disimpan dan tidak langsung hilang pada protokol *subscriber*. Penyimpanan data tersebut juga bertujuan agar data tersebut dapat dipanggil kembali dan ditampilkan pada aplikasi Android.

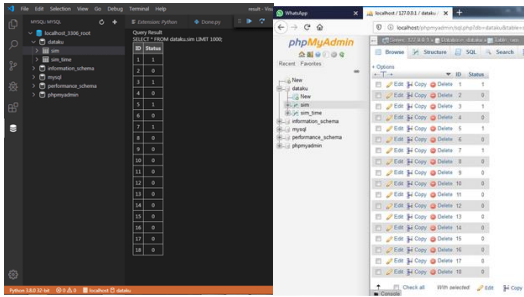
Koneksi yang dilakukan dapat dikatakan berhasil apabila tampilan MySQL pada Visual Studio Code sesuai dengan isi pada *database* yang tersedia. Jika data pada *database* berubah, maka MySQL pada VSCode pun juga berubah, begitu sebaliknya.

Tabel 1. Sinkronisasi VSCode dengan MySQL untuk Data Tabel *sim*

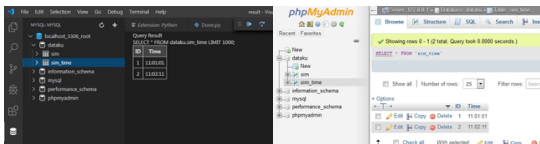
ID Patok	Percobaan 1		Percobaan 2		Percobaan 3	
	VSCode	MySQL	VSCode	MySQL	VSCode	MySQL
1	0	0	1	1	0	0
2	0	0	0	0	0	0
3	0	0	1	1	0	0
4	0	0	0	0	0	0
5	0	0	1	1	1	1
6	0	0	0	0	1	1
7	0	0	1	1	1	1
8	0	0	0	0	1	1
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	1	1
14	0	0	0	0	1	1
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	1	1
<b>Hasil</b>	<b>Berhasil</b>		<b>Berhasil</b>		<b>Berhasil</b>	

Tabel 2. Sinkronisasi VSCode dengan MySQL untuk Data Tabel *sim\_time*

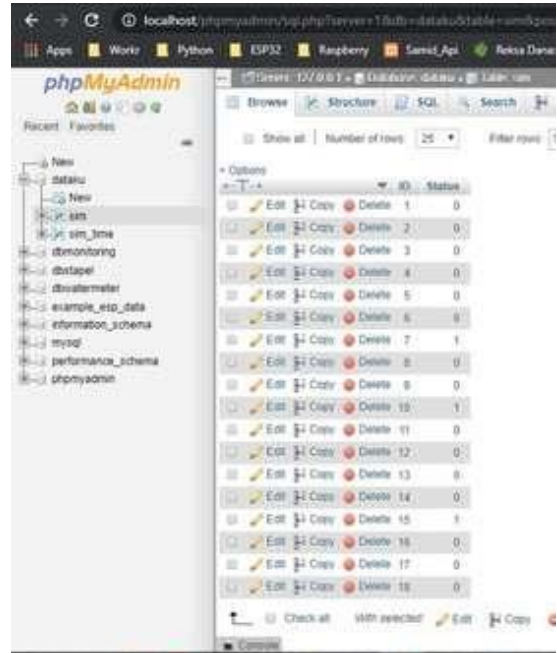
ID Patok	Percobaan 1		Percobaan 2		Percobaan 2	
	VSCode	MySQL	VSCode	MySQL	VSCode	MySQL
1	11:01:01	11:01:01	09:10:11	09:10:11	09:55:27	09:55:27
2	11:02:11	11:02:11	09:12:32	09:12:32	09:56:20	09:56:20
<b>Hasil</b>	<b>Berhasil</b>		<b>Berhasil</b>		<b>Berhasil</b>	



Gambar 20. Hasil sinkronisasi Data Tabel sim



Gambar 21. Hasil Sinkronisasi Data Tabel sim\_time

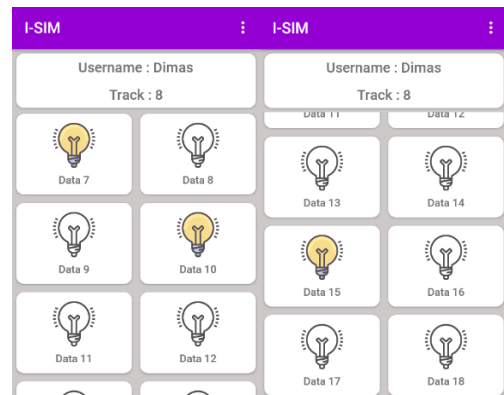


Gambar 22. Data pada Database

### Pengujian Sistem Database dengan Aplikasi Android

Tujuan dari proses ini adalah memastikan aplikasi Android dapat menampilkan data sesuai dengan data yang diambil dari sistem database. Data yang ada pada database adalah data hasil ujian praktik yaitu data kesalahan yang dilakukan dan seberapa lama ujian dilakukan. Jika data yang diambil aplikasi tidak sesuai dengan database, maka hasil ujian praktik tidak dapat dikatakan valid atau dapat dikatakan sistem tidak berjalan dengan baik.

Hasil yang diperoleh dari pengujian ini adalah berupa data lampu yang ada pada Android menyala sesuai dengan data yang ada pada database. Nyala lampu menandakan data pada database bernilai 1 atau dalam kenyataannya menunjukkan keadaan patok yang tersenggol atau terjatuh.

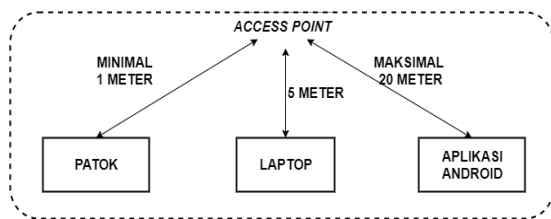


Gambar 23. Hasil yang Ditunjukkan Android

### Pengujian Jarak Android dengan Access Point Terhadap Kecepatan Pengiriman Data

Tujuan dari proses ini adalah untuk mengetahui performa dari kecepatan pengiriman data berdasarkan jarak dari perangkat Android dengan Access Point. Pada jarak kurang dari dua belas meter, perangkat Android masih dapat dengan mudah menerima data yang dikirimkan dari perangkat subscriber dengan rata-rata kecepatan kurang dari satu detik. Bila jarak lebih dari sepuluh meter, maka kecepatan pengiriman data semakin lama.





Gambar 24. Jarak tiap perangkat

Berdasarkan gambar 24 dapat dilihat bahwa terdapat jarak antara perangkat-perangkat yang terhubung. Dimana jarak ini adalah jarak peletakan tiap perangkat dengan *Access Point* yang mempengaruhi kecepatan pengiriman data. Jarak antara Laptop dengan *Access Point* adalah 5 meter. Jarak ini tidak berubah dikarenakan peletakan Laptop juga tidak berpindah-pindah. Jarak Android dengan *access point* yang terhubung mempunyai jarak maksimal sepanjang 20 meter dikarenakan perangkat Android dapat berpindah-pindah sesuai kebutuhan pengguna. Jarak ini mempengaruhi kecepatan pengiriman data dari patok ke perangkat Android. Semakin jauh jarak antar perangkat, maka lama pengiriman data juga semakin lambat.

Tabel 3. Tabel Perbandingan Jarak Android ke *Access Point* Terhadap Waktu Pengiriman Data

Percobaan 1		Percobaan 2		Percobaan 3	
Jarak (m)	Kecepatan (ms)	Jarak (m)	Kecepatan (ms)	Jarak (m)	Kecepatan (ms)
1	202	1	187	200	196,3
2	203	2	187	231	207
3	201	3	204	266	223,7
4	200	4	220	285	235
5	220	5	243	351	287
6	250	6	260	374	319,3
7	298	7	286	453	358
8	315	8	306	479	395,3
9	367	9	340	500	427
10	401	10	380	543	483,3
11	487	11	420	570	510
12	500	12	460	595	565,7
13	600	13	502	628	645
14	700	14	607	699	737,7
15	800	15	714	793	831
16	900	16	800	888	969,3
17	1020	17	1000	979	1109,6
18	1150	18	1200	1100	1223,3
19	1250	19	1320	1249	1356,3
20	1400	20	1420	351	287

Dapat dilihat dari tabel 3 bahwa semakin jauh jarak antara perangkat Android dengan *access point* yang terhubung, maka semakin lama waktu yang digunakan untuk mengirim suatu data. Waktu pengiriman yang efektif adalah kurang dari setengah detik. Dengan begitu jarak aman yang dapat digunakan adalah maksimal dua belas meter dari *access point* agar perangkat Android dapat menampilkan hasil ujian dengan cepat, sehingga

penilaian dapat dengan cepat dilanjutkan ke tahap selanjutnya.

### Pengujian Keseluruhan Sistem

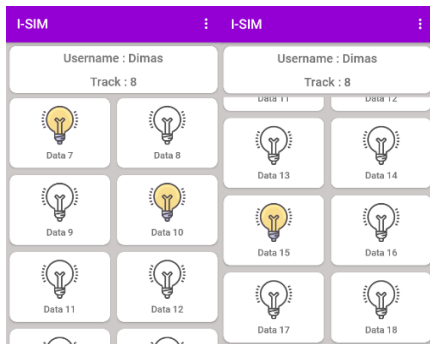
Tujuan dari proses ini adalah memastikan aplikasi Android dapat menampilkan data sesuai dengan data yang diambil dari sistem *database*. Data yang ada pada *database* adalah data hasil ujian praktik yaitu data kesalahan yang dilakukan dan seberapa lama ujian dilakukan. Data tersebut dikirim dari protokol *publisher* ke protokol broker lalu diteruskan dan diterima oleh protokol *subscriber*, sehingga data tersebut dapat diolah berdasarkan patok yang tersengol lalu disimpan pada *database*. Jika data yang diambil aplikasi tidak sesuai dengan *database*, maka hasil ujian praktik tidak dapat dikatakan valid atau dapat dikatakan sistem tidak berjalan dengan baik.

Hasil pengujian keseluruhan dapat langsung dilihat pada perangkat Android. Pada aplikasi yang terdapat pada perangkat Android memiliki 18 lampu dimana keadaan awal lampu adalah mati atau tidak berwarna. Lampu-lampu tersebut hidup atau berubah warna menjadi kuning apabila patok tersengol atau jatuh. Perubahan warna pada lampu mengikuti ID patok yang jatuh. Selain mengetahui patok-patok mana saja yang jatuh, aplikasi ini juga menampilkan hasil dari lamanya ujian praktik berlangsung untuk setiap sesimya. Lama waktu disini diperoleh berdasarkan selisih waktu dari patok yang mempunyai sensor ultrasonik 1 dan ultrasonik 2. Selisih inilah yang ditampilkan jika pengguna memilih menu “*Result*” pada aplikasi.

Tabel 4. Hasil percobaan patok dengan Android

ID	Percobaan 1		Percobaan 2		Percobaan 2	
	Patok	Patok	Patok	Android	Patok	Android
1	0	0	1	1	0	0
2	0	0	1	1	0	0
3	0	0	1	1	0	0
4	0	0	1	1	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	1	1	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	1	1	1	1	0	0
11	0	0	1	1	0	0
12	0	0	1	1	1	1
13	0	0	1	1	0	0
14	0	0	1	1	1	1
15	1	1	1	1	0	0
16	0	0	1	1	1	1
17	0	0	1	1	0	0
18	0	0	1	1	0	0
Hasil	Berhasil		Berhasil		Berhasil	

Nilai 0 menandakan bahwa patok tidak tersenggol atau jatuh dan pada Android menandakan bahwa lampu dalam kondisi mati. Dapat dilihat dari tiga kali percobaan yang dilakukan bahwa pada saat patok tersenggol atau bernilai 1, maka pada Android juga menampilkan lampu yang menyala. Hal itu berarti komunikasi serta pengiriman data yang dilakukan oleh perangkat *publisher* dikirim, diterima, disimpan, dan ditampilkan secara baik sesuai dengan sistem yang sudah ditentukan dan dibuat.

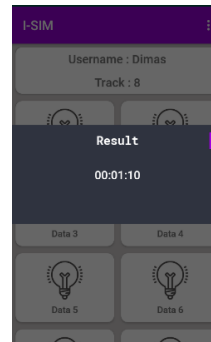


Gambar 25. Tampilan aplikasi Android berdasarkan patok

Selain kesalahan yang ditampilkan, lamanya waktu ujian juga ditampilkan. Hasil ini didapatkan dari selisih waktu patok pertama dan patok kedua. pada *database* menampilkan waktu dari masing-masing patok, kemudian diproses pada Android untuk mendapatkan selisih waktunya.

Tabel 5. Selisih waktu untuk lama ujian praktik berlangsung

Per cobaan	Ultrasonik 1	Ultrasonik 2	Selisih
1	11:01:01	11:02:11	00:01:10
2	11:05:17	11:07:01	00:01:44
3	11:41:48	11:42:59	00:01:11
4	12:13:07	12:15:05	00:01:58
5	13:22:18	13:23:52	00:01:34
6	13:31:21	13:32:56	00:01:35
7	08:30:09	08:32:00	00:01:51
8	08:47:19	08:49:01	00:01:42
9	09:10:11	09:12:32	00:02:21
10	09:55:27	09:56:20	00:00:53



Gambar 26. Hasil selisih waktu pada aplikasi android

Dapat dilihat, hasil *Result Time* didapat dari selisih waktu yang mana masing-masing waktu didapat dari data yang disimpan pada *database*. Kemudian data tersebut ditampilkan pada aplikasi Android untuk evaluasi penilaian ujian praktik SIM C.

## KESIMPULAN

Berdasarkan hasil pengujian diperoleh hasil sebagai berikut:

1. Protokol MQTT yang digunakan dibagi menjadi 3 bagian yaitu *publisher*, *broker*, dan *subscriber*. Masing-masing protokol harus terhubung dengan *access point* sebagai media pengiriman data. Data ini kemudian diolah oleh protokol *subscriber* untuk diolah dan dikelompokkan berdasarkan *topic* serta status data yang diterima. Data tersebut kemudian diteruskan dan disimpan ke dalam *database* yang nantinya diambil oleh perangkat Android.
2. Pengujian dilakukan sebanyak tiga kali dengan persentase keberhasilan 100% dimana pengiriman data yang dilakukan dapat disimpan oleh *database* dan di tampilkan pada perangkat Android.
3. Jarak aman antara perangkat Android dengan *access point* adalah maksimal 12 meter dengan kecepatan pengiriman data maksimal 0.5 detik. Dengan jarak itu, proses penampilan hasil ujian praktik SIM C dapat dilakukan secepat mungkin.
4. Aplikasi Android menampilkan hasil ujian praktik berdasarkan lampu-lampu yang hidup yang merepresentasikan tiap-tiap patok.
5. Aplikasi Android dapat menampilkan hasil waktu dari lama ujian praktik berlangsung.

## DAFTAR PUSTAKA

- (t.thn.). Dipetik Agustus 2019, dari Badan Pusat Statistik:  
<https://www.bps.go.id/linkTableDinamis/view/id/1133>
- (t.thn.). Dipetik Agustus 2019, dari Website Resmi POLRI: <https://www.polri.go.id/layanan-sim>
- Ananto. (2014, Desember 11). Dipetik Juli 2019, dari Standar Ujian Praktek Sim C (Motor) :  
<http://ananto-phc.blogspot.com/2014/12/standar-ujian-praktek-sim-c-motor.html>
- Anshar. (2010). *Panduan PHP & MySQL Secara Otodidak*. Jakarta: Mediakita.
- Equan. (2015, Oktober 6). *Mengenal MQTT*. Diambil kembali dari <https://medium.com/pemrograman/mengenal-mqtt-998b6271f585>
- Faisal, M. R., & Kurniawan, E. (t.thn.). *Seri Belajar ASP.NET Core 2MVC & MS SQL Server dengan Visual Studio 2017*.
- Satyaputra, A., & Aritohang, E. M. (2014). *Beginning Android Programming With ADT Bundle*. Jakarta: Elex Media Komputindo.
- Setiawan, R. H. (2019). *Rancang Bangun Sistem Penilaian Ujian Praktik SIM C Berbasis Wireless Menggunakan Protokol Komunikasi Pesan MQTT*. Surabaya.
- Wadi, H. (t.thn.). *Pemrograman Android Untuk Pelajar & Mahasiswa*.