



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



Oleh:

**Dimas Adiputra Rahadyan
14410200047**

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
2020**

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

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Teknik**



Oleh:

**Nama : Dimas Adiputra Rahadyan
NIM : 14410200047
Program Studi : S1 Teknik Komputer**

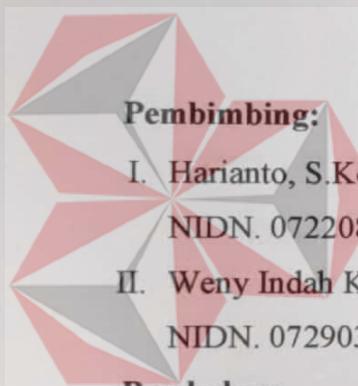
**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
2020**

Tugas Akhir

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

Dipersiapkan dan disusun oleh
Dimas Adiputra Rahadyan
NIM: 14410200047

Telah diperiksa, diuji dan disetujui oleh Dewan Pembahasan
Pada: Senin, 20 Januari 2020



Susunan Dewan Pembahasan

Pembimbing:

- I. Harianto, S.Kom., M.Eng.
NIDN. 0722087701
- II. Weny Indah Kusumawati, S.Kom., M.MT.
NIDN. 0729038504

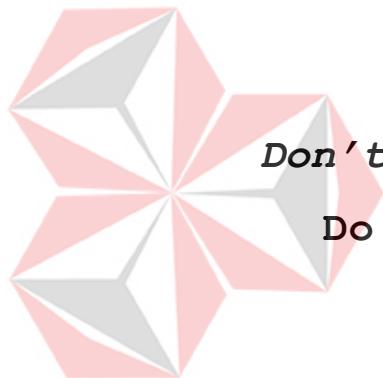
Pembahasan:

- Heri Pratikno, M.T., MTCNA., MTCRE.
NIDN. 0716117302

Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana



Dekan Fakultas Teknologi dan Informatika
UNIVERSITAS DINAMIKA



*Don't ask for it. Go out and win it.
Do that and you'll be rewarded.*

UNIVERSITAS
Dinamika

**Kupersembahkan Kepada
Ibu, Bapak, Kakak, Adik dan semua keluarga tercinta,
Yang selalu mendukung, memotivasi dan menyisipkan nama saya dalam
doa-doa terbaiknya.**



UNIVERSITAS
Dinamika

SURAT PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, saya :

Nama : Dimas Adiputra Rahadyan
NIM : 14.41020.0047
Program Studi : S1 Teknik Komputer
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Tugas Akhir
Judul Karya : **RANCANG BANGUN SISTEM PENILAIAN UJIAN PRAKTIK SIM C MENGGUNAKAN PROTOKOL KOMUNIKASI PESAN MQTT BERBASIS ANDROID**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, Februari 2020



Dimas Adiputra Rahadyan

Nim: 14.41020.0047

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 hingga ujian praktik. Pada ujian praktik, peserta melalui beberapa jenis ujian dimana petugas kepolisian akan melihat praktik tersebut dari pos pengujian. Hal tersebut tidak efisien karena kemungkinan pemantauan tersebut terlewatkan.

Pada penelitian sebelumnya memanfaatkan media *wireless* dimana pengiriman data menggunakan *Wifi* dan penampilan hasil ujian masih berupa tampilan desktop di komputer. Penampilan hasil ujian yang masih berupa aplikasi desktop pada komputer dirasa masih kurang maksimal dikarenakan pengawas ujian harus terpaku pada pos pengawasan untuk melihat hasil ujian praktik dan tidak dapat mengawasi peserta secara *mobile* atau bergerak bebas di lapangan ujian praktik. Karena hal tersebut penulis membuat aplikasi berbasis *Android* dengan memanfaatkan metode komunikasi *MQTT* dalam pengiriman datanya, sehingga petugas kepolisian dapat melakukan pemantauan ujian secara *mobile*.

Aplikasi ini akan menampilkan hasil ujian praktik pada perangkat *Android*. Hasil yang ditampilkan berupa lampu yang menunjukkan tiap patok pada ujian. Lampu-lampu ini akan menyala jika patok ujian tersenggol. Aplikasi ini juga akan menampilkan hasil waktu dari lamanya ujian praktik. Data-data tersebut akan 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.

Kata Kunci: *Ujian Praktik SIM C, MQTT, Database, Aplikasi Android.*

KATA PENGANTAR

Segala puji syukur kehadirat Allah SWT, karena dengan rahmat-Nya dan hidayah-Nya penulis dapat menyelesaikan penyusunan Laporan Tugas Akhir yang berjudul “RANCANG BANGUN SISTEM PENILAIAN UJIAN PRAKTIK SIM C MENGGUNAKAN PROTOKOL KOMUNIKASI PESAN MQTT BERBASIS ANDROID”. Laporan Tugas Akhir ini disusun dalam rangka penulisan laporan untuk memperoleh gelar Sarjana Komputer pada Program Studi S1 Teknik Komputer Universitas Dinamika.

Dalam penyusunan Tugas Akhir ini banyak hambatan serta rintangan yang penulis hadapi namun pada akhirnya dapat melaluinya berkat adanya bimbingan dan bantuan dari berbagai pihak baik secara moral maupun spiritual. Untuk itu pada kesempatan ini, penulis ingin mengucapkan rasa terima kasih kepada pihak-pihak yang memberi dukungan dan masukan dalam menyelesaikan laporan Tugas Akhir ini. Oleh karena itu penulis ingin mengucapkan kepada:

1. Orang Tua dan saudara-saudara tercinta yang telah memberikan dorongan dan bantuan baik moral maupun materi sehingga penulis dapat menempuh dan menyelesaikan Tugas Akhir ini.
2. Bapak Dr. Jusak selaku Dekan Fakultas Teknologi dan Informatika (FTI) Universitas Dinamika telah membantu proses penyelesaian Tugas Akhir yang dibuat oleh penulis dengan baik.
3. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer Universitas Dinamika.
4. Bapak Harianto, S.Kom., M.Eng. selaku Dosen Pembimbing 1 yang selalu

- 
- memberi arahan dan bimbingan dalam menyelesaikan Tugas Akhir ini.
5. Ibu Weny Indah Kusumawati, S.Kom., M.MT. selaku Dosen Pembimbing 2 yang selalu memberi arahan dan bimbingan dalam menyelesaikan Tugas Akhir ini.
 6. Bapak Heri Pratikno, M.T., MTCNA., MTCRE. selaku Dosen Penguji atas ijin dan masukan dalam menyusun Tugas Akhir ini.
 7. Ibu Yosefine Triwidayastuti, M.T. selaku Dosen Wali yang selalu memberi arahan dan bimbingan dalam menyelesaikan Tugas Akhir ini.
 8. Semua staf dosen S1 Teknik Komputer yang telah mengajar dan memberikan ilmunya.
 9. Teman-teman seperjuangan Teknik Komputer angkatan 2014 yang selalu membantu, mendukung dan memotivasi agar tetap berusaha menjadi lebih baik.
 10. Serta semua pihak lain yang tidak dapat disebutkan secara satu per satu, yang telah membantu dalam menyelesaikan Tugas Akhir ini baik secara langsung maupun tidak langsung,

Penulis menyadari bahwa Laporan Tugas Akhir ini jauh dari kata sempurna, masih banyak kekurangan dalam menyusun laporan ini. Oleh karena itu dalam kesempatan ini, penulis meminta maaf apabila dalam Laporan Tugas Akhir ini masih banyak kesalahan baik dalam penulisan maupun Bahasa yang digunakan. Penulis juga memerlukan kritik dan saran dari para pembaca yang sifatnya membangun untuk kesempurnaan laporan yang telah penulis susun.

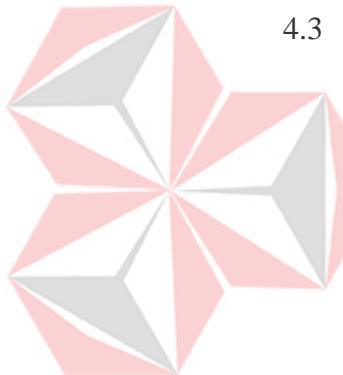
Surabaya, Februari 2020

Penulis

DAFTAR ISI

Halaman

ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Batasan Masalah	4
1.4 Tujuan	4
BAB II LANDASAN TEORI	5
2.1 Surat Izin Mengemudi (SIM).....	5
2.1.1 Syarat Pembuatan.....	5
2.1.2 Ujian Praktik	5
2.2 <i>Message Queuing Telementary Transport</i> (MQTT).....	6
2.3 Android	8
2.3.1 Fitur-fitur Android	8
2.3.2 Arsitektur Android	9
2.4 Microsoft Visual Studio Code	9
2.5 MySQL	10
BAB III METODE PENELITIAN.....	11
3.1 Diagram Blok Sistem.....	12
3.2 <i>Flowchart</i> Keseluruhan Sistem.....	14
3.3 Rancangan Perangkat.....	15
3.3.1 Perancangan Perangkat untuk Protokol <i>Mosquito Broker</i> ...	15
3.3.2 Perancangan Perangkat untuk Protokol <i>Subscriber</i>	17
3.3.3 Perancangan Sistem <i>Database</i>	18
3.3.4 Perancangan Aplikasi Android	20

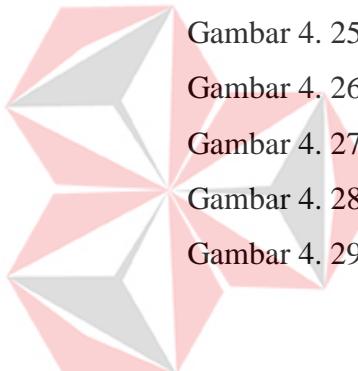


BAB IV HASIL PENGUJIAN DAN PEMBAHASAN	24
4.1 Pengujian Protokol <i>Subscriber</i> dengan Sistem <i>Database</i>	24
4.1.1 Tujuan	24
4.1.2 Alat Yang Digunakan.....	24
4.1.3 Langkah-Langkah	24
4.1.4 Hasil Pengujian	27
4.2 Pengujian Sistem <i>Database</i> dengan Aplikasi Android.....	31
4.2.1 Tujuan	31
4.2.2 Alat Yang Digunakan.....	31
4.2.3 Prosedur Pengujian	31
4.2.4 Hasil Pengujian	36
4.3 Pengujian Jarak Android dengan <i>Access Point</i> Terhadap Kecepatan Pengiriman Data	38
4.3.1 Tujuan	38
4.3.2 Alat Yang Digunakan.....	38
4.3.3 Prosedur Pengujian	39
4.3.4 Hasil Pengujian	40
4.4 Pengujian Keseluruhan Sistem	41
4.4.1 Tujuan	41
4.4.2 Alat Yang Digunakan.....	42
4.4.3 Prosedur Pengujian	42
4.4.4 Hasil Pengujian	43
BAB V PENUTUP.....	47
5.1 Kesimpulan	47
5.2 Saran	48
DAFTAR PUSTAKA	49
LAMPIRAN	50
BIODATA PENULIS	86

DAFTAR GAMBAR

	Halaman
Gambar 2. 1 Jalan Komunikasi MQTT	7
Gambar 2. 2 Arsitektur Android	9
Gambar 3. 1 Blok Diagram Sistem	13
Gambar 3. 2 <i>Flowchart</i> Keseluruhan Sistem	14
Gambar 3. 3 <i>Drive</i> dimana <i>Mosquitto</i> di <i>Install</i>	16
Gambar 3. 4 Menjalankan <i>Mosquitto</i>	16
Gambar 3. 5 Program <i>Subscriber</i> menggunakan Visual Studio Code	18
Gambar 3. 6 Menghidupkan Apache dan MySQL	19
Gambar 3. 7 Pembagian Data pada <i>Database</i>	19
Gambar 3. 8 Isi pada Data SIM	20
Gambar 3. 9 Isi pada Data SIM_Time	20
Gambar 3. 10 Pemilihan Jenis API yang digunakan	21
Gambar 3. 11 Menu <i>Login</i> Aplikasi	21
Gambar 3. 12 Keadaan Awal Lampu saat Pertama Aplikasi dibuka	22
Gambar 3. 13 Menu pada Aplikasi	22
Gambar 3. 14 Kondisi Lampu setelah Patok Dilewati Peserta Ujian Praktik	23
Gambar 3. 15 Tampilan Menu <i>Result</i>	23
Gambar 4. 1 Menjalankan MySQL	25
Gambar 4. 2 <i>Database</i> Hasil Ujian	25
Gambar 4. 3 Memilih MySQL <i>Extensions</i>	26
Gambar 4. 4 Menambahkan Koneksi ke <i>Database</i>	26
Gambar 4. 5 Koneksi <i>Database</i> Berhasil	27
Gambar 4. 6 <i>Database</i> tabel sim pada Visual Studio Code sebelum Data Dikirim	28
Gambar 4. 7 <i>Database</i> tabel sim pada MySQL sebelum Data Dikirim	29
Gambar 4. 8 <i>Database</i> tabel sim pada Visual Studio Code setelah Data Dikirim	29
Gambar 4. 9 <i>Database</i> tabel sim pada MySQL setelah Data Dikirim	30
Gambar 4. 10 <i>Database</i> tabel sim_time pada Visual Studio Code	30
Gambar 4. 11 <i>Database</i> tabel sim_time pada MySQL	30

Gambar 4. 12 <i>Syntax</i> untuk OS Android di atas 9.0	32
Gambar 4. 13 <i>Syntax</i> menambahkan IP <i>Database</i>	32
Gambar 4. 14 <i>Syntax</i> Inisialisasi Alamat <i>Database</i>	32
Gambar 4. 15 <i>Syntax Json parse</i> untuk Data Patok	33
Gambar 4. 16 <i>Syntax Json_Parse_Data_After_Wecall</i> untuk Data Patok.....	33
Gambar 4. 17 <i>Syntax</i> Menampilkan <i>Card View</i>	34
Gambar 4. 18 <i>Syntax</i> Pengambilan Data Waktu Lama Ujian	35
Gambar 4. 19 <i>Syntax Json_Parse_Data_After_Wecall</i> untuk Data Waktu	35
Gambar 4. 20 Hasil Data yang diambil oleh Android.....	36
Gambar 4. 21 Data ID Patok pada <i>Database</i>	36
Gambar 4. 22 Data Waktu pada <i>Database</i>	37
Gambar 4. 23 Hasil Pengambilan Data pada Aplikasi Android.....	37
Gambar 4. 24 Hasil Pengambilan Data Waktu pada Aplikasi Android.....	38
Gambar 4. 25 Jarak Tiap Perangkat	40
Gambar 4. 26 Hasil Tampilan Kesalahan pada Android.....	44
Gambar 4. 27 Hasil Pada <i>Database</i>	45
Gambar 4. 28 Data waktu pada <i>Database</i>	46
Gambar 4. 29 Hasil Selisih Waktu dari Dua Patok pada Android	46



UNIVERSITAS
Dinamika

DAFTAR TABEL

Halaman

Tabel 4. 1 Sinkronisasi Visual Studio Code dengan MySQL untuk Data tabel sim	27
Tabel 4. 2 Sinkronisasi Visual Studio Code dengan MySQL untuk Data tabel sim_time	28
Tabel 4. 3 Tabel Perbandingan Jarak Android ke <i>Access Point</i> Terhadap Waktu Pengiriman Data	40
Tabel 4. 4 Hasil Percobaan Pada Android	43
Tabel 4. 5 Selisih Waktu untuk Lamanya Ujian Praktik	45

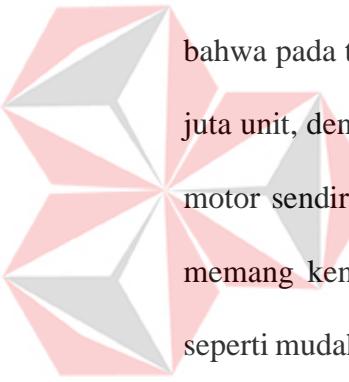


BAB I

PENDAHULUAN

1.1 Latar Belakang

Kendaraan bermotor adalah salah satu kebutuhan *tersier* bagi rakyat Indonesia, tapi bagi beberapa orang adanya kendaraan bermotor adalah kebutuhan *primer*. Dengan kendaraan bermotor, kita dapat berpergian menuju tempat tujuan dengan mudah, dan bahkan menjadi sarana untuk mencari nafkah bagi profesi tertentu.

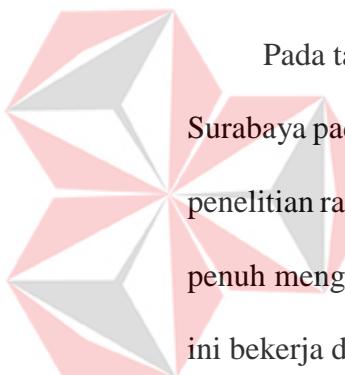


Data yang diterbitkan oleh BPS (Badan Pusat Statistik) memperlihatkan bahwa pada tahun 2015 jumlah kendaraan bermotor telah mencapai lebih dari 129 juta unit, dengan sepeda motor saja mencapai jumlah 105 juta unit. Jumlah sepeda motor sendiri mencapai 81.34% dari total kendaraan di Indonesia. Sepeda motor memang kendaraan yang paling digemari di Indonesia dengan beberapa alasan seperti mudah dipelajari, tidak menghabiskan banyak tempat, mudah melewati jalan jalan sempit dan masih banyak alasan lainnya. (Badan Pusat Statistik)

Untuk bisa mengendarai sepeda motor kita 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 baik syarat administrasi, sehat jasmani dan rohani, memahami peraturan lalu lintas, dan terampil dalam mengemudikan kendaraan bermotor. Di Indonesia ada 6 macam kategori SIM yaitu SIM A, SIM A Khusus, SIM B I, SIM B II, SIM C, dan SIM D. Agar dapat mengendarai sepeda motor dengan legal, kita harus mempunyai SIM C sebagai bukti telah mengikuti dan lulus

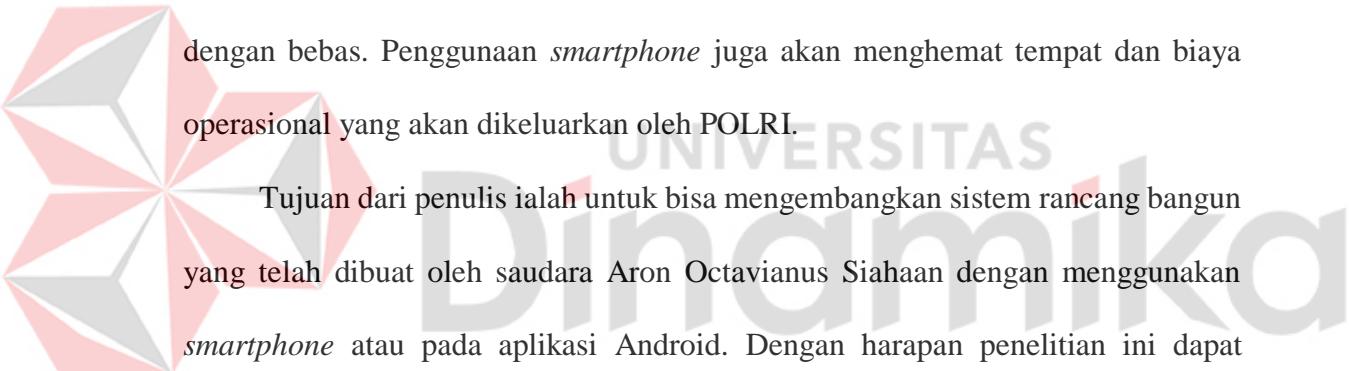
pada syarat administrasi dan ujian yang diberikan oleh POLRI. (Website Resmi POLRI)

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. Setelah pemohon dinyatakan lulus pada ujian teori, pemohon dapat melanjutkan ke ujian selanjutnya, yaitu ujian praktik. Ujian praktik menguji kemahiran dan ketangkasan pemohon dalam mengendarai sepeda motor pada jalan raya. Terdapat 5 sesi ujian yang akan dihadapi pemohon, yaitu uji keseimbangan, uji slalom (zig zag), uji membentuk angka 8, uji reaksi, dan uji berbalik arah (*U Turn*).



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 dengan Microsoft Visual Basic 6.0. Sistem akan mendeteksi pelanggaran dan kecepatan kendaraan bermotor pemohon pada setiap sesi ujian praktik. Data ini akan diolah dan ditampilkan pada aplikasi yang dimonitoring oleh penguji dan membantu penguji dalam memberikan penilaian terhadap pemohon SIM C.

Pada kesempatan kali ini, penulis mempunyai keinginan untuk mengembangkan penelitian yang telah diselesaikan oleh saudara Aron Octavianus Siahaan, dengan membawa aplikasi untuk pengawasan dan penilaian dari komputer ke *smartphone* atau lebih tepatnya mengaplikasikannya pada aplikasi Android.



Penggunaan aplikasi Android disini dilatar belakangi dengan beberapa alasan. Pada ujian praktik SIM C penguji harus mengawasi pelanggaran yang dibuat oleh peserta ujian, salah satu pelanggaran yang bisa terjadi adalah peserta ujian menurunkan kakinya dan menyentuh tanah ditengah sesi ujian, sedangkan aplikasi yang dirancang oleh saudara Aron Octavianus Siahaan terdapat pada komputer, hal ini menyebabkan penguji harus memfokuskan diri pada 2 hal di 2 tempat yang berbeda, yaitu memperhatikan peserta ujian di lapangan dan memperhatikan aplikasi di pos ujian. Dengan memindahkan aplikasi pada *smartphone*, penguji dapat bergerak bebas di area ujian praktik sambil mengawasi peserta ujian dan aplikasi secara bersamaan karena tidak terikat dengan komputer yang tidak dapat dibawa bergerak dengan bebas. Penggunaan *smartphone* juga akan menghemat tempat dan biaya operasional yang akan dikeluarkan oleh POLRI.

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.

1.2 Perumusan Masalah

Berdasarkan latar belakang di atas, perumusan masalah dalam Tugas Akhir ini antara lain sebagai berikut:

1. Bagaimana protokol MQTT dapat mengirim data ke perangkat Android sehingga dapat menampilkan hasil ujian praktik SIM C?
2. Bagaimana pengaruh jarak terhadap komunikasi yang dilakukan antara Android dengan perangkat *publisher* terhadap kecepatan pengiriman data?

1.3 Batasan Masalah

Dalam penelitian ini, agar tidak menyimpang dari tujuan yang nantinya akan dicapai maka pembahasan masalah dibatasi pada hal-hal sebagai berikut:

1. Simulasi ujian praktik menggunakan perangkat patok yang sudah dibuat pada Penelitian sebelumnya.
2. Komunikasi data menggunakan protokol MQTT.
3. Penggunaan Sistem *Database* dalam penyimpanan Hasil Ujian Praktik.
4. Lintasan ujian praktik yang digunakan adalah lintasan ujian keseimbangan, lintasan ujian zig zag, lintasan ujian putar balik.
5. Android menggunakan *Operating System* maksimal 8.

1.4 Tujuan

Tujuan dari penelitian Tugas Akhir ini adalah sebagai berikut:

1. Menampilkan hasil ujian praktik SIM C secara *mobile* sehingga pengawasan tidak hanya dilakukan di pos pengawasan.
2. Mengetahui apakah jarak perangkat *publisher* dengan perangkat Android mempengaruhi kecepatan pengiriman data.

BAB II

LANDASAN TEORI

2.1 Surat Izin Mengemudi (SIM)

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. SIM digolongkan menjadi beberapa bagian, yaitu SIM A, SIM A Khusus, SIM B1, SIM B2, SIM C, SIM D.

2.1.1 Syarat Pembuatan

Syarat-syarat yang harus dipenuhi untuk pembuatan SIM adalah sebagai berikut:

1. Usia minimal 17 tahun untuk pembuatan SIM C.
2. Menyelesaikan administrasi yang dibutuhkan untuk pembuatan baru (mengisi form pendaftaran dan melampirkan KTP).
3. Surat pengantar kesehatan jasmani dan rohani.
4. Mengikuti ujian teori dan ujian praktik.

2.1.2 Ujian Praktik

1. Uji keseimbangan

Mengendarai sepeda motor dengan kecepatan 30km/jam dengan persneling 2 dan berhenti pada garis *stop*. Jarak dari *start* sampai *stop* adalah 9 buah patok

dengan jarak $1\frac{1}{2}$ panjang motor untuk setiap patok dan lebar 2 kali panjang motor.

2. Uji slalom/*zig-zag*

Mengendarai sepeda motor secara *zig-zag* dengan kecepatan stabil melewati 5 patok dengan jarak 3 kali panjang motor untuk setiap patok dan berhenti pada garis *stop*.

3. Uji membentuk angka 8

Mengendarai sepeda motor membentuk angka 8 sebanyak 3 kali mengikuti petunjuk arah, tidak boleh berhenti dan kaki tidak boleh menginjak lapangan. Jarak antara patok adalah $1\frac{1}{2}$ panjang motor.

4. Uji reaksi rem menghindar

Mengendarai sepeda motor dengan kecepatan 30km.jam, kemudian melakukan penggereman pada garis kuning dan lepas rem pada garis hijau lalu membelok sesuai petunjuk pengawas dan berhenti pada garis *stop*.

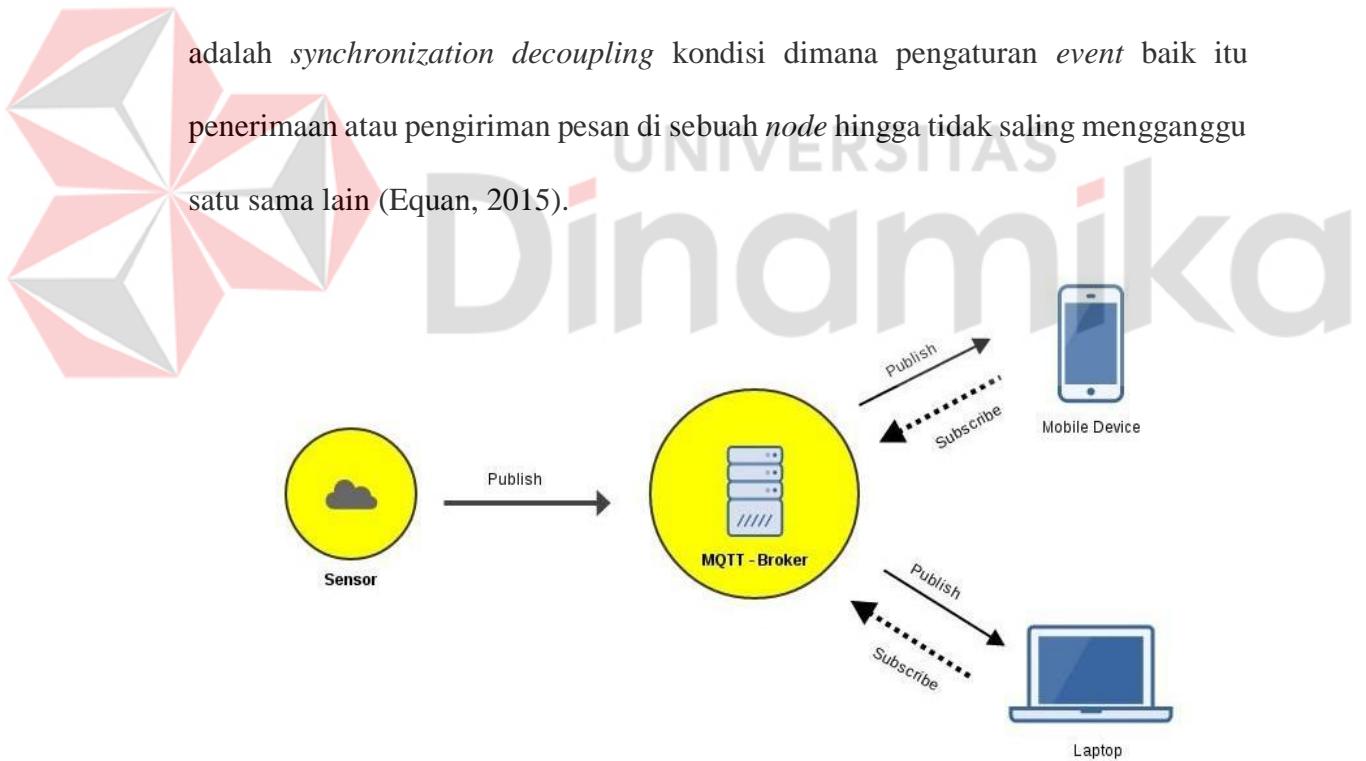
5. Uji berbalik arah (*U-turn*)

Mengendarai sepeda motor dan berputar arah dengan membentuk huruf U dijalan sempit dengan lebar 2 kali panjang motor tanpa menginjakkan kaki ke lapangan. (Ananto, 2014)

2.2 Message Queuing Telementary 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* akan dipasangkan pada perangkat keras sedangkan *Broker* berfungsi untuk menangani *publish-subscribe* data.

Publish/subscribe sendiri adalah sebuah pola pertukaran pesan di dalam komunikasi jaringan dimana pengirim data disebut *publisher* dan penerima data disebut dengan *subscriber*, metode *publish/subscribe* memiliki beberapa kelebihan salah satunya yaitu *loose coupling* atau *decouple* dimana berarti antara *publisher* dan *subscriber* tidak saling mengetahui keberadaannya, terdapat 3 buah *decoupling* yaitu *time decoupling*, *space decoupling* dan *synchronization decoupling*, *time decoupling* adalah sebuah kondisi dimana *publisher* dan *subscriber* tidak harus saling aktif pada waktu yang sama, *space decoupling* adalah dimana *publisher* dan *subscriber* aktif di waktu yang sama akan tetapi antara *publisher* dan *subscriber* tidak saling mengetahui keberadaan dan identitas satu sama lain, dan yang terakhir adalah *synchronization decoupling* kondisi dimana pengaturan *event* baik itu penerimaan atau pengiriman pesan di sebuah *node* hingga tidak saling mengganggu satu sama lain (Equan, 2015).



Gambar 2. 1 Jalan Komunikasi MQTT

(Sumber: <https://medium.com/pemrograman/mengenal-mqtt-998b6271f585>)

2.3 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 dikembangkan pertama kali oleh Google.

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.

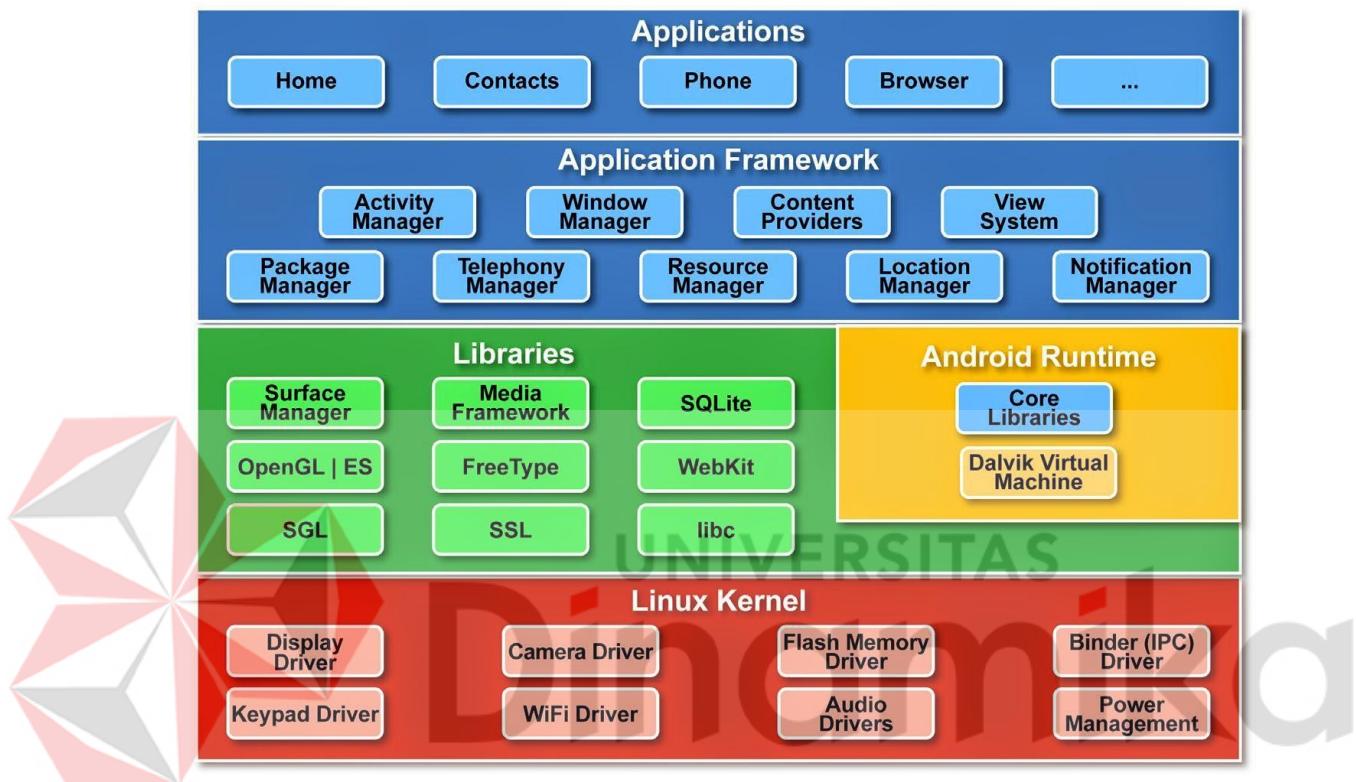
2.3.1 Fitur-fitur Android

Android memiliki fitur-fitur yang sangat menarik sehingga mampu memikat hati para pengembang. Selain fitur-fitur, sistem operasi ini juga memiliki kelebihan yang menjadi daya tarik dan alasan kenapa harus memilih Android. Berikut adalah kelebihan-kelebihan yang dimiliki Android:

1. Kode sumber terbuka (*open source*).
2. Siapapun bisa mengembangkan *platform*.
3. Memiliki komunitas pengembang yang besar dan terus berkembang.
4. Memiliki pangsa pasar yang terus berkembang.
5. Mampu menekan atau mengurangi biaya pengembangan.
6. Memiliki rasio sukses lebih tinggi.
7. Memiliki lingkungan pengembangan (*development environment*) yang luas.
8. Kaya akan fitur-fitur.

2.3.2 Arsitektur Android

Sistem operasi Android memiliki arsitektur yang dapat dikelompokkan menjadi lima bagian dan empat lapis/layer. Berikut adalah gambar yang akan memperlihatkan struktur atau arsitektur dari sistem operasi Android (Gambar 2.2).



Gambar 2.2 Arsitektur Android

(Sumber: <http://dasawarnadunia.blogspot.com/2014/04/fitur-dan-arsitektur-Android.html>)

Dari Gambar 2.2 dapat dilihat lima bagian dan empat layer dari arsitektur sistem operasi Android yang dimaksud.

2.4 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)

2.5 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 kita bisa menggunakanya 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 to use*). 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).

BAB III

METODE PENELITIAN

Dalam pengerjaan Tugas Akhir ini mempunyai tujuan untuk membuat sebuah aplikasi berbasis Android untuk penilaian ujian praktik pembuatan SIM C. Dimana aplikasi tersebut digunakan untuk memasukkan dan menampilkan data serta hasil ujian praktik dari peserta yang disimpan dalam sistem *database MySQL*.

Untuk sistem penilaian ujian yang menggunakan perangkat simulasim patok atau pembatas ujian praktik, penulis menggunakan beberapa perangkat yang sudah dibuat pada penelitian sebelumnya yang dilakukan oleh Rizki Haris Setiawan dan Aron Octavianus Siahaan dimana terdapat delapan belas buah patok yang sudah dilengkapi dengan sensor getar, dua buah patok dengan sensor ultrasonik dan semua patok dilengkapi dengan wemos. Sensor getar digunakan untuk mengetahui apakah peserta ujian membuat kesalahan atau pelanggaran dengan menyenggol atau mengenai patok tersebut. Sensor ultrasonik dipasang pada patok pertama dan patok terakhir, digunakan untuk mengetahui bahwa kendaraan yang digunakan peserta ujian sudah memulai atau mengakhiri sesi dalam salah satu ujian praktik. Wemos digunakan sebagai perangkat komunikasi yang akan mengirimkan data ke perangkat *broker*.

Broker menerima data dari wemos melalui jaringan yang sama yang sebelumnya sudah diatur pada wemos dan *broker* itu sendiri, selanjutnya data yang diterima *broker* akan dilanjutkan ke *subscriber* untuk diolah. Komunikasi *broker* dan *subscriber* serta perangkat input lain atau disebut *publisher* menggunakan metode MQTT atau yang lebih dikenal dengan nama *mosquitto*. *Subscriber* akan

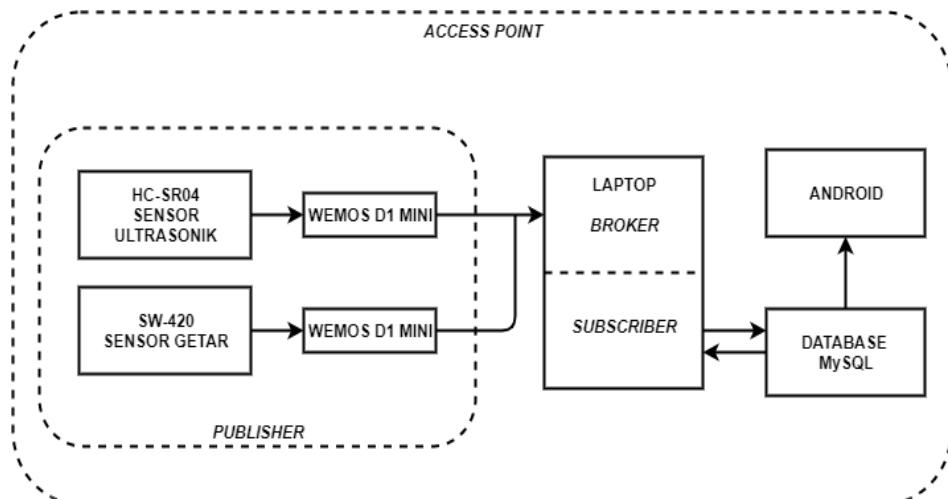
mengolah data berdasarkan kesalahan yang dilakukan oleh peserta ujian praktik yang dikirim oleh sensor getar dan berdasarkan waktu yang dikirim oleh patok ultrasonik awal dan akhir. Apabila peserta melakukan kesalahan, secara otomatis *subscriber* akan menerima data “*true*”, kemudian data tersebut diolah dan disimpan dalam *database* berdasarkan nomor patok dan status kesalahan yang dilakukan.

MySQL bertugas untuk menyimpan data tersebut yang nantinya akan digunakan kembali untuk ditampilkan dalam aplikasi Android yang dibuat. Pada MySQL data tersebut akan dibagi menjadi dua tabel, yaitu tabel sensor getar yang akan menyimpan nomor ID dan status kesalahan dari tiap patok serta tabel sensor ultrasonik yang akan menyimpan nomor ID dan waktu dari dua patok. Nomor patok akan digunakan sebagai *Topic* dalam protokol *mosquito* sedangkan status kesalahan atau waktu akan digunakan sebagai data atau isi dari *topic* yang dibawa. Setiap tipe ujian praktik akan ditampilkan pada perangkat Android data dipilah sesuai dengan pilihan pemakai atau polisi yang bertugas mengawasi ujian praktik.

3.1 Diagram Blok Sistem

Terdapat beberapa proses yang dikerjakan dalam perancangan Tugas Akhir ini mulai dari pengiriman data masukan dari patok yang diterima oleh perangkat *broker* yang kemudian diteruskan ke perangkat *subscriber* dan diolah ke dalam *database* yang akan digunakan dan ditampilkan dalam aplikasi Android. Semua perangkat yang ada harus dihubungkan dalam satu jaringan yang sama. Data yang ditampilkan berupa data diri peserta serta hasil penilaian ujian praktik SIM C yang dilakukan. Hasil penilaian ujian praktik akan menunjukkan kesalahan peserta dimana peserta akan menyenggol atau menjatuhkan patok. Penilaian dilakukan satu persatu sesuai dengan jenis ujian praktik yang dipilih, sehingga hasil penilaianya

juga akan satu persatu atau tidak ditampilkan secara bersamaan untuk ketiga ujian praktik. Berikut ini adalah blok diagram pada sistem yang ditunjukkan oleh Gambar 3.1:



Gambar 3. 1 Blok Diagram Sistem

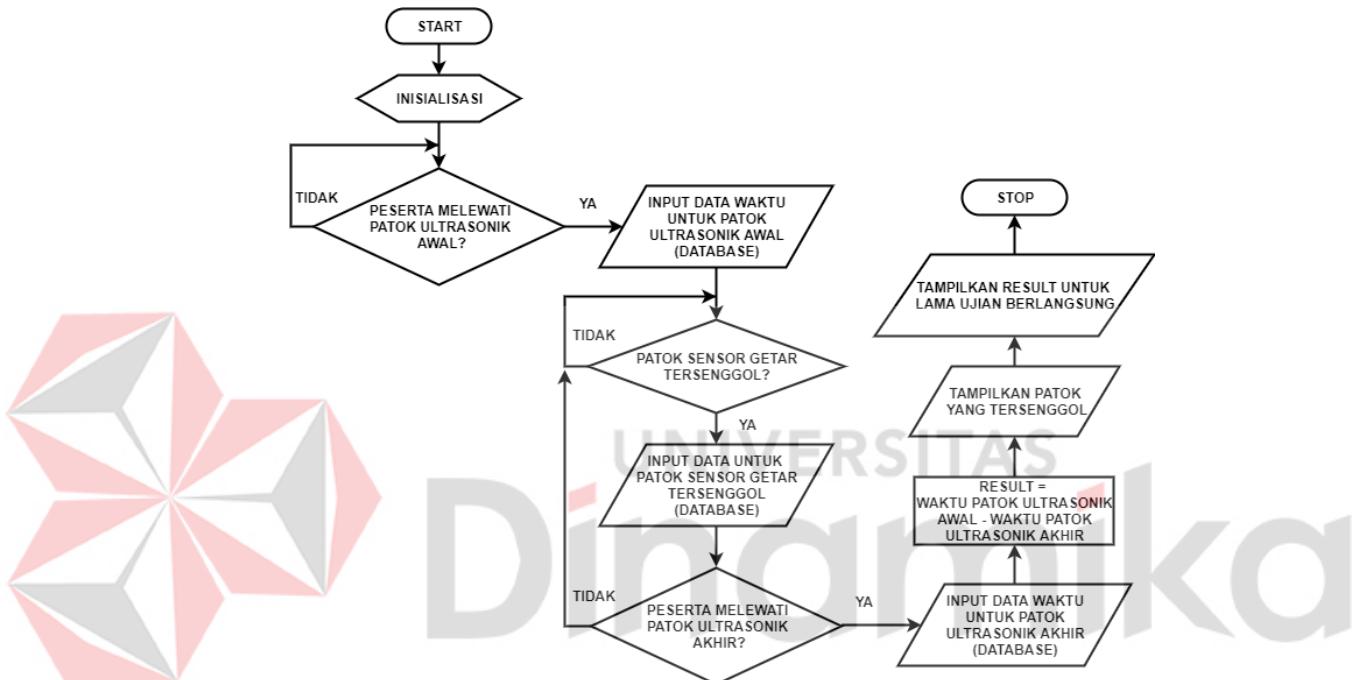
(Sumber: Olahan Penulis)

Penulis hanya berfokus pada bagian komunikasi menggunakan protokol MQTT atau *mosquitto*, pembuatan aplikasi Android, serta sinkronisasi dari protokol *mosquito* 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 3.1 bahwa pada patok terdapat sensor getar SW-420 yang akan menandakan ada tidaknya kesalahan, dan sensor ultrasonic HC-SR04 yang digunakan sebagai penanda awal dan akhir sesi tiap ujian berlangsung. Data tersebut akan dikirim ke *broker* sebagai jembatan antara *publisher* dengan *subscriber* melalui perangkat wemos, lalu data tersebut akan diolah oleh *subscriber* sedemikian rupa sehingga data tersebut digolongkan

berdasarkan *topic* serta status data yang dibawa. *Topic* serta data tersebut menjadi satu informasi utuh yang akan disimpan dalam *database* ke dalam kolom-kolom *database* yang sudah disediakan. Kemudian data tiap *field* tersebut akan ditampilkan pada aplikasi Android bergantung pada apa yang ingin ditampilkan.

3.2 Flowchart Keseluruhan Sistem



Gambar 3.2 Flowchart Keseluruhan Sistem

(Sumber: Olahan Penulis)

1. Insialisasi di atas adalah menyambungkan perangkat *Publisher*, *broker*, *subscriber* dan Android 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 akan dikirim dan disimpan ke dalam

database yang sebelumnya telah diolah pada perangkat *subscriber*. Data ini akan masuk pada tabel sensor ultrasonik dengan ID 1.

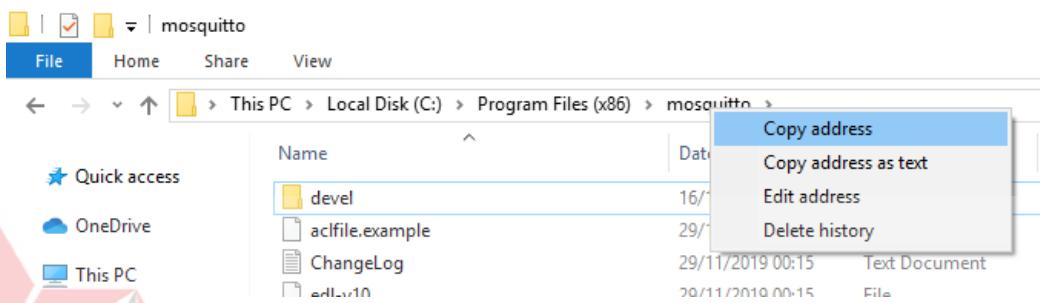
5. Ketika patok sensor getar tersenggol, *subscriber* akan menerima data dari ID patok yang tersenggol. Lalu menyimpannya ke dalam sistem *database* pada tabel sensor getar.
6. Sensor ultrasonik akhir yang berada digaris *finish* mendeteksi adanya gerakan ketika pengendara melewati garis. Kemudian mencatat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

3.3 Rancangan Perangkat

3.3.1 Perancangan Perangkat untuk Protokol *Mosquito Broker*

Broker adalah sebuah protokol yang 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 akan dapat dilakukan. Sehingga pada saat proses pengiriman data,

protokol ini diwajibkan untuk aktif agar pengiriman data dapat berjalan dengan baik. Langkah pertama yang dilakukan adalah dengan menginstal *protocol mosquito* pada komputer. Setelah *mosquito* terinstal, selanjutnya adalah menjalankan *mosquito broker* melalui *Command Prompt (Run as Administrator)*. Proses ini akan mengaktifkan protokol *broker*. Selanjutnya masuk ke tempat penyimpanan *mosquito* yang sudah diinstal sebelumnya. Dapat dengan menyalin langsung dari perangkat komputer.



Gambar 3. 3 Drive dimana Mosquitto di Install
(Sumber: Olahan Penulis)

Setelah masuk ke dalam penyimpanan, maka *mosquito* dapat langsung dijalankan dengan menuliskan perintah “*net start mosquito*” pada *command prompt*.

```

Administrator: Command Prompt
Microsoft Windows [Version 10.0.17134.1184]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Program Files (x86)\mosquitto

C:\Program Files (x86)\mosquitto>net start mosquito
The Mosquitto Broker service is starting.
The Mosquitto Broker service was started successfully.

C:\Program Files (x86)\mosquitto>

```

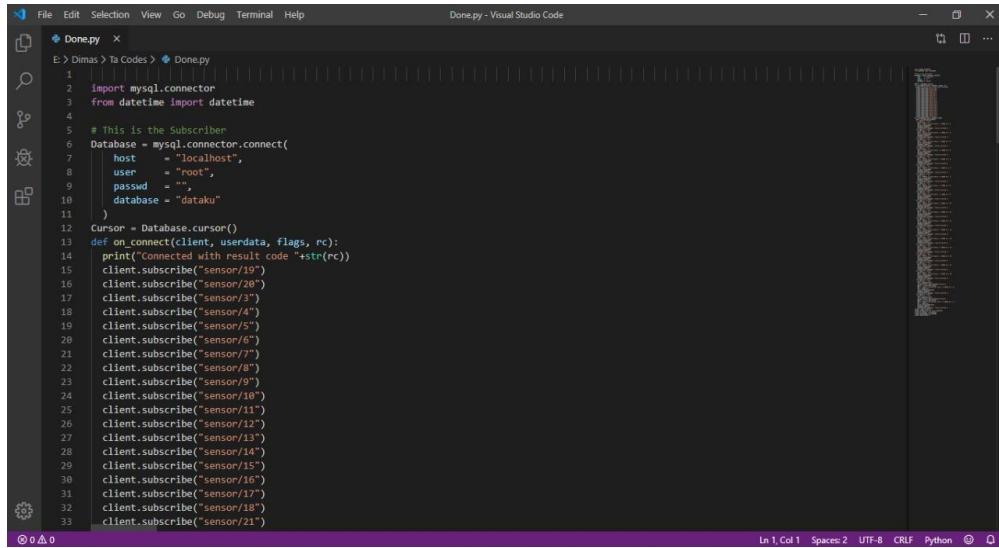
Gambar 3. 4 Menjalankan Mosquitto
(Sumber: Olahan Penulis)

Dapat dilihat bahwa pengaturan *broker* telah berhasil dengan adanya pesan keberhasilan “*The Mosquitto broker service was started successfully*”.

3.3.2 Perancangan Perangkat untuk Protokol *Subscriber*

Subscriber adalah protokol yang digunakan untuk mengolah data masukan yang di terima. Data yang diterima akan di simpan pada *database* bergantung pada *topic* dan isi data yang di bawa. Pada *subscriber* juga akan 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 akan disimpan dalam variabel yang dibuat pada protokol *subscriber* dan tidak akan diteruskan dalam sistem *database*. Dalam pengaturan perangkat menjadi *subscriber* menggunakan *software* Visual Studio Code dan Python . Dimana di *software* Visual Studio Code 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*.

Setelah semua pengaturan tambahan dilakukan, Visual Studio Code dapat digunakan untuk membuat program yang dapat mengolah data yang diterima.



```

1  File Edit Selection View Go Debug Terminal Help
2  Done.py - Visual Studio Code
3  E > Dimas > Ta Codes > Done.py
4
5  # This is the Subscriber
6  Database = mysql.connector.connect(
7      host = "localhost",
8      user = "root",
9      passwd = "",
10     database = "dataku"
11 )
12 Cursor = Database.cursor()
13 def on_connect(client, userdata, flags, rc):
14     print("Connected with result code "+str(rc))
15     client.subscribe("sensor/19")
16     client.subscribe("sensor/20")
17     client.subscribe("sensor/3")
18     client.subscribe("sensor/4")
19     client.subscribe("sensor/5")
20     client.subscribe("sensor/6")
21     client.subscribe("sensor/7")
22     client.subscribe("sensor/8")
23     client.subscribe("sensor/9")
24     client.subscribe("sensor/10")
25     client.subscribe("sensor/11")
26     client.subscribe("sensor/12")
27     client.subscribe("sensor/13")
28     client.subscribe("sensor/14")
29     client.subscribe("sensor/15")
30     client.subscribe("sensor/16")
31     client.subscribe("sensor/17")
32     client.subscribe("sensor/18")
33     client.subscribe("sensor/21")

```

Gambar 3. 5 Program *Subscriber* menggunakan Visual Studio Code

(Sumber: Olahan Penulis)

3.3.3 Perancangan Sistem *Database*

Dalam perancangan sistem *database* MySQL, penulis menggunakan aplikasi XAMPP. Pembuatan tabel-tabel data pada *database* juga dibuat melalui aplikasi ini. Dimana data tersebut akan dibaca dan ditampilkan pada aplikasi Android. Pembuatan *database* MySQL dengan menggunakan XAMPP, pertama menjalankan Apache serta MySQL dengan memilih *start*. Aktifnya program Apache dan MySQL ditandai dengan adanya indikator berwarna hijau dan adanya pemberitahuan pada *message box*.



Gambar 3. 6 Menghidupkan Apache dan MySQL
(Sumber: Olahan Penulis)

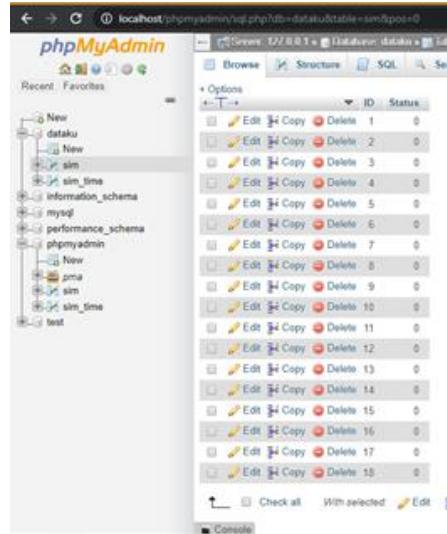
Selanjutnya pilih Admin pada MySQL untuk masuk ke dalam sistem

database. Pada saat memilih Admin, *user* akan langsung diarahkan ke dalam Server melalui *Web Browser*. Untuk membuat *database* baru dapat dengan klik *New* lalu memasukkan nama dari *database* yang ingin dibuat lalu *Create*.

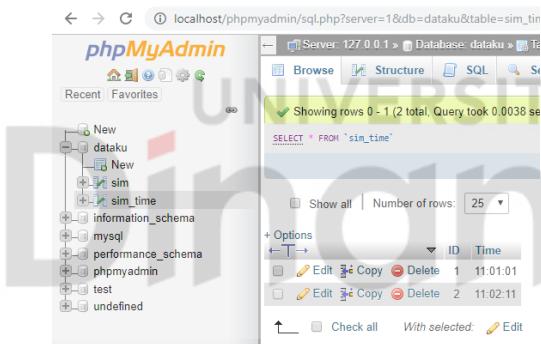
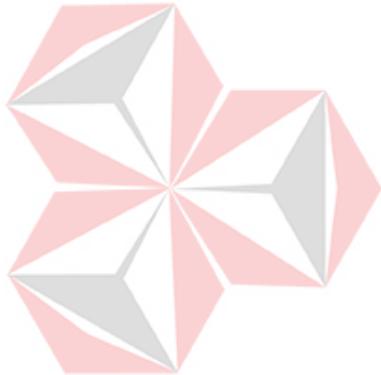


Gambar 3. 7 Pembagian Data pada Database
(Sumber: Olahan Penulis)

Di sini penulis akan membuat data sim dan sim_time, dimana data sim akan menyimpan ID dari tiap patok perangkat *publisher* serta status dari tiap patok. Status akan berubah nilai dari 0 menjadi 1 apabila patok dikenai oleh peserta ujian, dan apabila tidak dikenai maka nilainya akan tetap menjadi 0. Nilai tersebut berubah sesuai dengan ID patok. Berikut adalah isi dari data sim dan sim_time.



Gambar 3. 8 Isi pada Data SIM
(Sumber: Olahan Penulis)

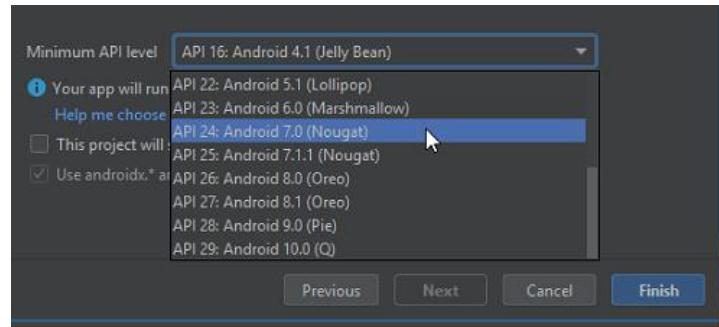


Gambar 3. 9 Isi pada Data SIM_Time
(Sumber: Olahan Penulis)

3.3.4 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 di bawah itu akan kesulitan dalam pengoperasian aplikasi karena tidak mendukung

dalam hal spesifikasi OS nya, misalnya aplikasi dapat dijalankan akan tetapi lambat dalam performanya.



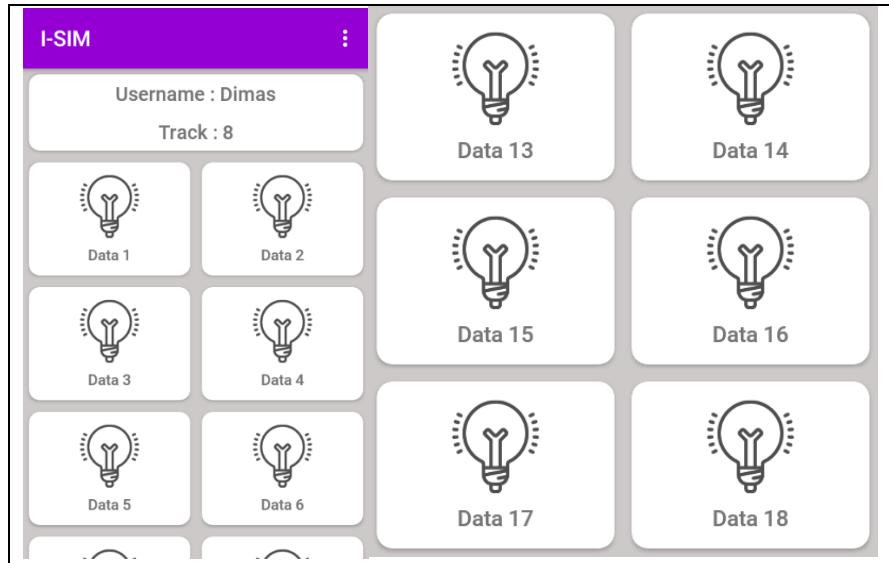
Gambar 3. 10 Pemilihan Jenis API yang digunakan
(Sumber: Olahan Penulis)

Gambar 3.11 adalah hasil dari perancangan aplikasi yang sudah diinstal pada perangkat Android. Tampilan awal aplikasi digunakan untuk memasukkan data diri peserta serta lintasan yang digunakan.



Gambar 3. 11 Menu Login Aplikasi
(Sumber: Olahan Penulis)

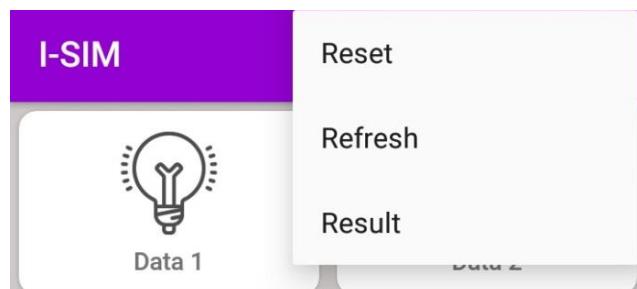
Setelah masuk, aplikasi akan menunjukkan beberapa lampu dimana tiap lampu memiliki nomor sendiri yang menandakan ID dari tiap patok. ID pada lampu akan sama dengan ID pada patok yang sudah diatur.



Gambar 3. 12 Keadaan Awal Lampu saat Pertama Aplikasi dibuka

(Sumber: Olahan Penulis)

Kondisi awal lampu adalah mati, hal itu menandakan bahwa patok yang terdapat di lapangan masih belum tersenggol oleh peserta. Apabila peserta menyenggol patok, lampu tersebut akan menyala sesuai dengan nomor patok dimana kita dapat memilih menu *refresh* untuk menampilkan lampu yang menyala tersebut.



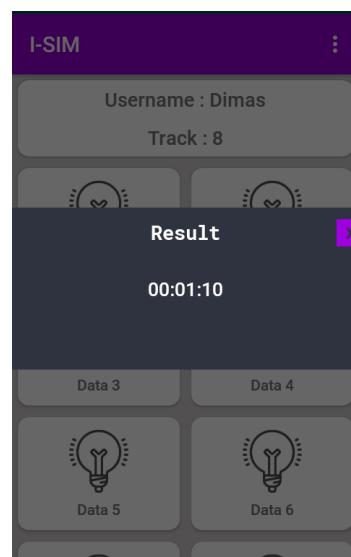
Gambar 3. 13 Menu pada Aplikasi

(Sumber: Olahan Penulis)



Gambar 3. 14 Kondisi Lampu setelah Patok Dilewati Peserta Ujian Praktik
(Sumber: Olahan Penulis)

Aplikasi juga dilengkapi dengan menu untuk hasil ujian yang dinyatakan dengan lamanya proses ujian berlangsung dari awal saat peserta menjalankan kendaraan hingga mencapai *finish* dari masing-masing ujian praktik. Serta ada menu untuk mengulang kembali pengaturan lampu hingga semua lampu mati atau dalam keadaan awal sebelum ujian (Gambar 3.12). Hasil dari menu *Result* akan menunjukkan waktu lamanya sebuah ujian dilakukan.



Gambar 3. 15 Tampilan Menu *Result*
(Sumber: Olahan Penulis)

BAB IV

HASIL PENGUJIAN DAN PEMBAHASAN

4.1 Pengujian Protokol *Subscriber* dengan Sistem *Database*

4.1.1 Tujuan

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.

4.1.2 Alat Yang Digunakan

Berikut adalah alat yang digunakan untuk menghubungkan *subscriber* dengan sistem *database*:

- a. Laptop atau Komputer
- b. *Software* Microsoft Visual Studio Code
- c. *Software* XAMPP
- d. *Access point* (Andromax Mifi/ *Handphone*)

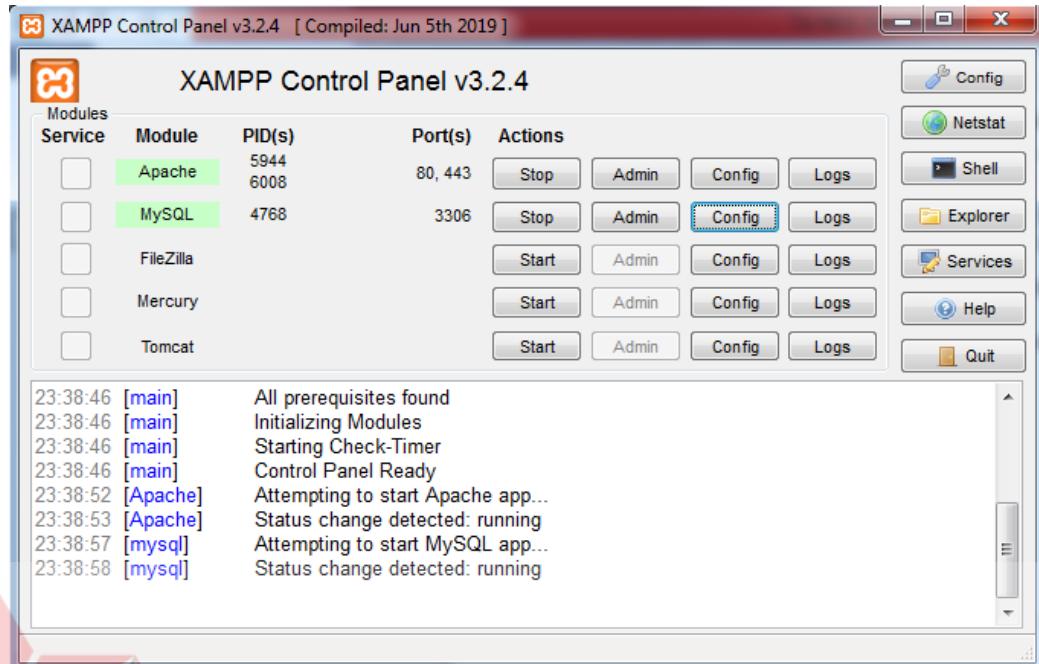
4.1.3 Langkah-Langkah

Langkah-langkah yang dilakukan untuk menghubungkan *subscriber* dengan sistem *database* adalah seperti berikut:

1. Menghidupkan Laptop atau Komputer.
2. Menghubungkan Laptop dengan *Access point*, dimana *Access point* ini akan dihubungkan juga dengan protokol *publisher*.
3. Membuka XAMPP.

4. Menjalankan Apache.

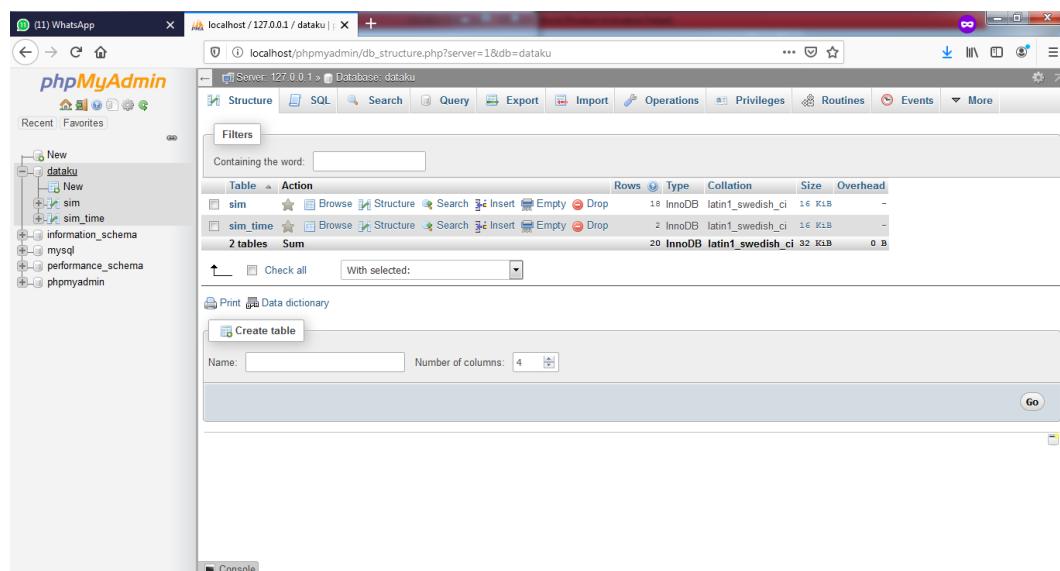
5. Menjalankan MySQL.



Gambar 4. 1 Menjalankan MySQL

(Sumber: Olahan Penulis)

6. Pilih Admin pada MySQL, sehingga akan masuk dalam PhPMyAdmin dimana terdapat *database* yang telah kita buat.



Gambar 4. 2 Database Hasil Ujian

(Sumber: Olahan Penulis)

7. Membuka Visual Studio Code.

8. Memilih *Extensions MySQL* yang telah diinstal sebelumnya.



```
File Edit Selection View Go Debug Terminal Help Done.py - Visual Studio Code

Done.py ×
C:\Users\YOS\Downloads> Done.py > ...
1 import paho.mqtt.client as mqtt
2 import mysql.connector
3
4 # This is the Subscriber
5 Database = mysql.connector.connect(
6     host      = "localhost",
7     user      = "root",
8     passwd   = "",
9     database = "dataku"
10 )
11 Cursor = Database.cursor()
12 def on_connect(client, userdata, flags, rc):
13     print("Connected with result code "+str(rc))
14     client.subscribe("sensor/1")
15     client.subscribe("sensor/2")
16     client.subscribe("sensor/3")
17     client.subscribe("sensor/4")
18     client.subscribe("sensor/5")
19     client.subscribe("sensor/6")
20     client.subscribe("sensor/7")
21     client.subscribe("sensor/8")
22     client.subscribe("sensor/9")
23     client.subscribe("sensor/10")
24     client.subscribe("sensor/11")
25     client.subscribe("sensor/12")
26     client.subscribe("sensor/13")
27     client.subscribe("sensor/14")
28     client.subscribe("sensor/15")
29     client.subscribe("sensor/16")
30     client.subscribe("sensor/17")
31     client.subscribe("sensor/18")
32 def on_message(client, userdata, msg):
33     Data = msg.payload.decode()
34     Cursor.execute("INSERT INTO dataku VALUES (%s)", (Data,))

Python 3.8.0 32-bit (Python 3.8.0) 0 localhost dataku
In 148, Col 22 Spaces: 2 UTF-8 CRLF Python
```

Gambar 4. 3 Memilih MySQL Extensions

(Sumber: Olahan Penulis)

9. Menambahkan koneksi dengan menekan simbol “+” pada kanan atas.

A screenshot of the MySQL extension in Visual Studio Code. The interface shows a 'Connect to SQL server' dialog with the following fields:

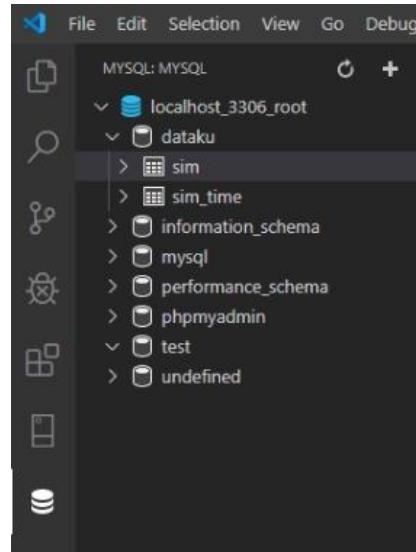
- host: localhost
- port: 3306
- username: root
- password: (redacted)

A 'Connect' button is at the bottom. The 'host' field is highlighted with a red box. The 'password' field is redacted with a black box. The 'MySQL' extension icon is in the top left, and the 'connect' icon is in the top right.

Gambar 4.4 Menambahkan Koneksi ke *Database*

(Sumber: Olahan Penulis)

10. Mengisi *Host*, *Port*, *Username*, dan *Password*, setelah itu akan muncul tampilan *database* sesuai dengan *database* yang telah dibuat pada MySQL.



Gambar 4. 5 Koneksi Database Berhasil

(Sumber: Olahan Penulis)

4.1.4 Hasil Pengujian

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 akan berubah, begitu sebaliknya.

Tabel 4. 1 Sinkronisasi Visual Studio Code 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

ID Patok	Percobaan 1		Percobaan 2		Percobaan 3	
	VSCode	MySQL	VSCode	MySQL	VSCode	MySQL
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	1	1
18	0	0	0	0	1	1
Hasil	Berhasil		Berhasil		Berhasil	

(Sumber: Olahan Penulis)

Tabel 4. 2 Sinkronisasi Visual Studio Code dengan MySQL untuk Data tabel sim_time

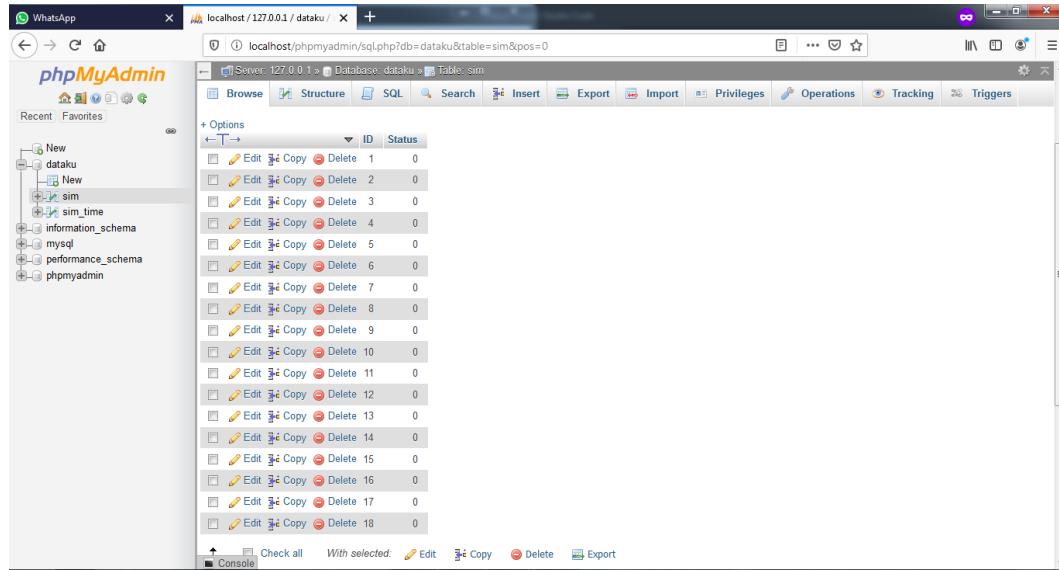
ID Patok	Percobaan 1		Percobaan 2		Percobaan 3	
	VSCode	MySQL	VSCode	MySQL	VSCode	MySQL
1	11:01:01	11:01:01	09:10:11	09:10:11	09:55:27	09:55:27
2	11:02:11	11:02:11	09:12:32	09:12:32	09:56:20	09:56:20
Hasil	Berhasil		Berhasil		Berhasil	

(Sumber: Olahan Penulis)

ID	Status
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0

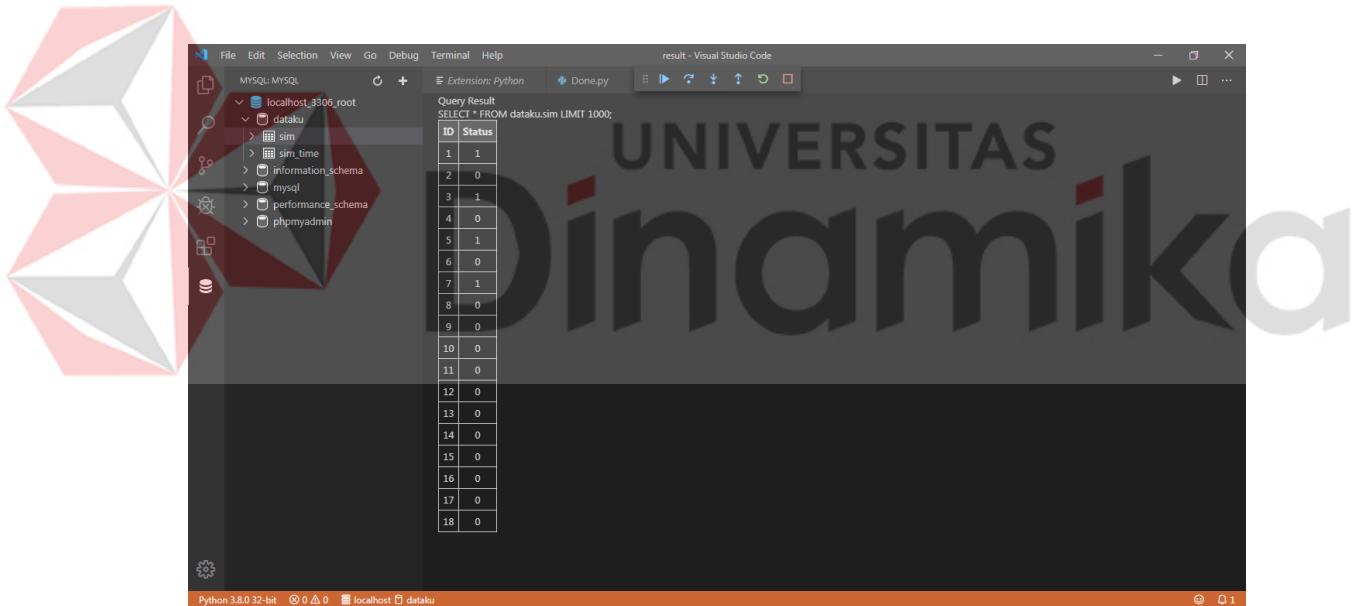
Gambar 4. 6 Database tabel sim pada Visual Studio Code sebelum Data Dikirim

(Sumber: Olahan Penulis)



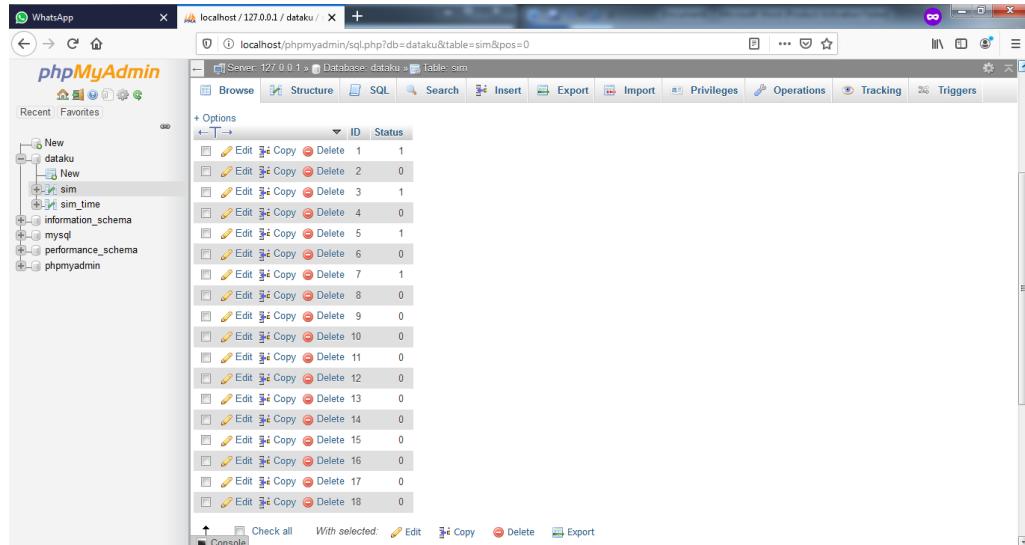
Gambar 4. 7 Database tabel sim pada MySQL sebelum Data Dikirim

(Sumber: Olahan Penulis)

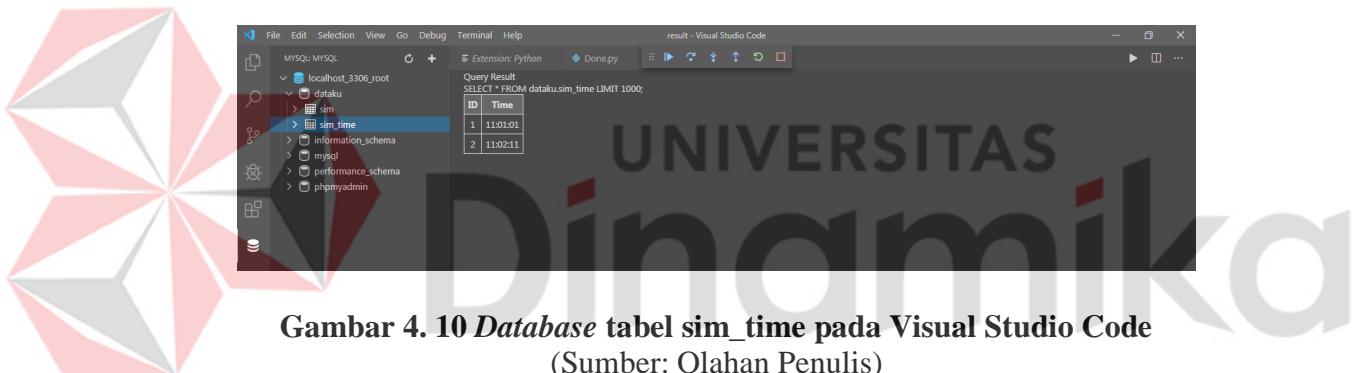


Gambar 4. 8 Database tabel sim pada Visual Studio Code setelah Data Dikirim

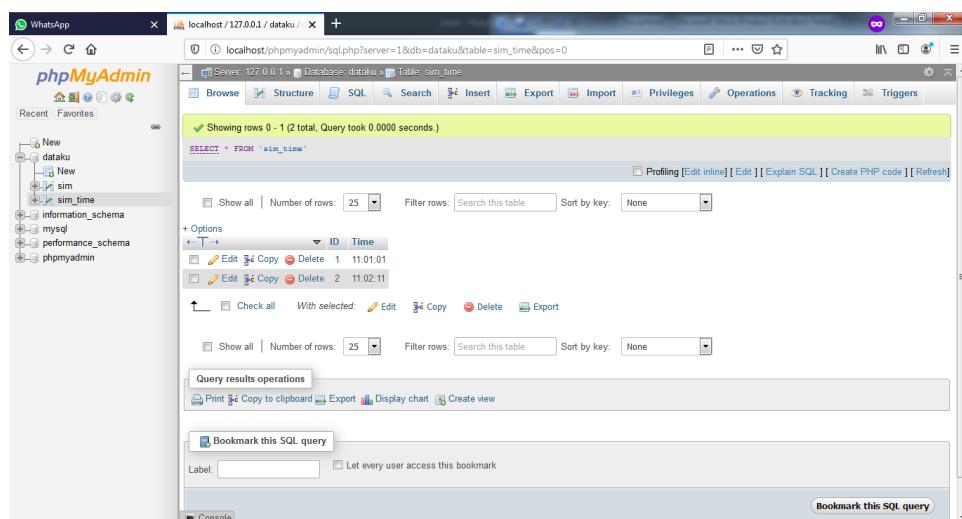
(Sumber: Olahan Penulis)



Gambar 4. 9 Database tabel sim pada MySQL setelah Data Dikirim
(Sumber: Olahan Penulis)



Gambar 4. 10 Database tabel sim_time pada Visual Studio Code
(Sumber: Olahan Penulis)



Gambar 4. 11 Database tabel sim_time pada MySQL
(Sumber: Olahan Penulis)

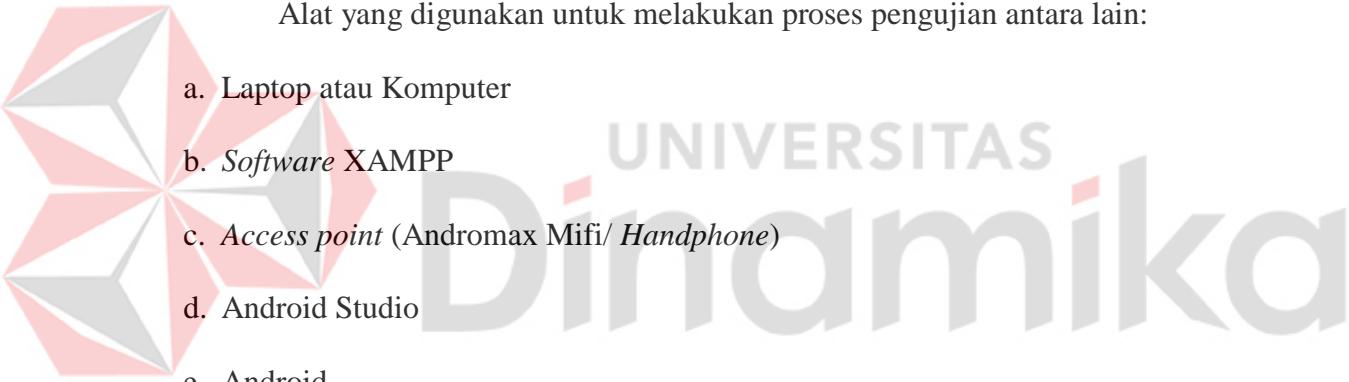
4.2 Pengujian Sistem *Database* dengan Aplikasi Android

4.2.1 Tujuan

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.

4.2.2 Alat Yang Digunakan

Alat yang digunakan untuk melakukan proses pengujian antara lain:

- 
- a. Laptop atau Komputer
 - b. *Software XAMPP*
 - c. *Access point (Andromax Mifi/ Handphone)*
 - d. *Android Studio*
 - e. *Android*

4.2.3 Prosedur Pengujian

Langkah-langkah yang dilakukan untuk melakukan pengujian adalah seperti berikut:

1. Menghidupkan Laptop atau Komputer.
2. Membuka XAMPP.
3. Menjalankan Apache.
4. Menjalankan MySQL.
5. Memilih Admin pada MySQL, sehingga masuk pada PhPMyAdmin.

6. Memperhatikan perubahan yang terjadi pada *database*.
7. Membuka Android Studio, masuk pada *manifest*. Khusus untuk Android yang mempunyai OS diatas 9.0, perlu menambahkan *syntax* berikut.

```

<meta-data
    android:name="com.google.android.actions"
    android:resource="@xml/network_security_config" />
<uses-library
    android:name="org.apache.http.legacy"
    android:required="false" />

```

Gambar 4. 12 Syntax untuk OS Android di atas 9.0
(Sumber: Olahan Penulis)

8. Menambahkan file di res/xml dengan *syntax* berikut. Tag *domain* dengan mengisi IP server *database*.



```

<?xml version = "1.0" encoding = "utf-8"?><!-- Learn More about how to use App Actions: https://developer.android.com/guide/actions/include-->
<network-security-config>
  <domain-config cleartextTrafficPermitted="true">
    <domain includeSubdomains="true">192.168.1.64</domain>
  </domain-config>
</network-security-config>

```

Gambar 4. 13 Syntax menambahkan IP Database
(Sumber: Olahan Penulis)

9. Menginisialisasi alamat *database*.

```

String HTTP_JSON_URL_GETDATA = "http://192.168.0.11:8080/sim-php/getdata.php";
String HTTP_JSON_URL_RESETDATA = "http://192.168.0.11:8080/sim-php/resetdata.php";
String HTTP_JSON_URL_GETTIMEDATA = "http://192.168.0.11:8080/sim-php/gettimedata.php";
String HTTP_JSON_URL_SETTIME = "http://192.168.0.11:8080/sim-php/settime.php";

```

Gambar 4. 14 Syntax Inisialisasi Alamat Database
(Sumber: Olahan Penulis)

10. Data yang diterima dari server merupakan data dalam bentuk *json* sehingga diperlukan *json parse* untuk mengolah data tersebut. Untuk *json parse* pada menu utama dituliskan sebagai berikut.

```
jsonArrayRequest = new JsonArrayRequest(HTTP_JSON_URL_GETDATA,
{
    (response) -> {
        JSON_PARSE_DATA_AFTER_WEBCALL(response, mode);
    },
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
        }
    });
requestQueue = Volley.newRequestQueue( context: this );
requestQueue.add(jsonArrayRequest);
```

Gambar 4. 15 *Syntax Json parse* untuk Data Patok
(Sumber: Olahan Penulis)

11. Pada *syntax* di atas terdapat *function json_parse_data_after_webcall* yang digunakan untuk *json parse*. *Syntax* dapat dituliskan sebagai berikut.

```
for(int i = 0; i<array.length(); i++) {
    Mydata GetDataAdapter2 = new Mydata();
    JSONObject json = null;
    try {
        json = array.getJSONObject(i);
        GetDataAdapter2.setId(json.getString(GET_JSON_ID));
        GetDataAdapter2.setStatus(json.getString(GET_JSON_STATUS));
        status.add(json.getString(GET_JSON_STATUS));
    } catch (JSONException e) {
        e.printStackTrace();
    }
    statusList.add(GetDataAdapter2);
}
recyclerViewadapter = new RecyclerViewChilder(statusList, context: this);
recyclerView.setAdapter(recyclerViewadapter);
```

Gambar 4. 16 *Syntax Json_Parse_Data_After_Wecall* untuk Data Patok
(Sumber: Olahan Penulis)

12. Setelah dilakukan *json parse*, maka data akan ditampilkan dalam bentuk *card view*. *Syntax* untuk menampilkan *card view* pada tiap data dituliskan sebagai berikut.

```
@Override
public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.cardview, parent, false);
    ViewHolder viewHolder = new ViewHolder(view);
    return viewHolder;
}

@Override
public void onBindViewHolder(ViewHolder holder, int position) {
    Mydata getDataAdapter1 = mydata.get(position);
    String data = "Data " + getDataAdapter1.getId();
    holder.NameData.setText(data);
    String status = getDataAdapter1.getStatus();
    if(status.equals("1")){
        holder.ImgData.setBackgroundResource(R.drawable.light_on);
    } else {
        holder.ImgData.setBackgroundResource(R.drawable.light_off);
    }
}

class ViewHolder extends RecyclerView.ViewHolder{
    public TextView NameData;
    public ImageView ImgData;

    public ViewHolder(View itemView) {
        super(itemView);
        NameData = itemView.findViewById(R.id.tv_id);
        ImgData = itemView.findViewById(R.id.img_logo);
    }
}
```

Gambar 4. 17 Syntax Menampilkan Card View
(Sumber: Olahan Penulis)

13. Untuk pengambilan data *result* atau waktu lama ujian berlangsung dapat dituliskan dengan *syntax* berikut.

```
jsonArrayRequest = new JSONArrayRequest(HTTP_JSON_URL_GETTIMEDATA,
    (response) -> [
        JSON_PARSE_DATA_AFTER_WEBCALL(response, mode);
    ],
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
        }
    });
requestQueue = Volley.newRequestQueue(context: this);
requestQueue.add(jsonArrayRequest);
```

Gambar 4. 18 Syntax Pengambilan Data Waktu Lama Ujian

(Sumber: Olahan Penulis)

14. Pada *syntax* diatas terdapat *function json_parse_data_after_webcall* yang digunakan untuk *json parse*. *Syntax* dapat dituliskan sebagai berikut.

```
for(int i = 0; i<array.length(); i++) {
    JSONObject json = null;
    try {
        json = array.getJSONObject(i);
        if(i == 0) starttime = json.getString(GET_JSON_TIME);
        else if (i == 1) endtime = json.getString(GET_JSON_TIME);
    } catch (JSONException e) {
        e.printStackTrace();
    }
}
try{
    SimpleDateFormat sdf = new SimpleDateFormat( pattern: "hh:mm:ss");
    Date starting = sdf.parse(starttime);
    Date ending = sdf.parse(endtime);
    long difference = ending.getTime() - starting.getTime();
    String hms = String.format("%02d:%02d:%02d", TimeUnit.MILLISECONDS.toHours(difference),
        TimeUnit.MILLISECONDS.toMinutes(difference) - TimeUnit.HOURS.toMinutes(TimeUnit.MILLISECONDS.toHours(difference)),
        TimeUnit.MILLISECONDS.toSeconds(difference) - TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(difference)));
    resulttext.setText(hms);
} catch (ParseException ex) {
    Log.v( tag: "Exception", ex.getLocalizedMessage());
}
```

Gambar 4. 19 Syntax Json_Parse_Data_After_Wecall untuk Data Waktu

(Sumber: Olahan Penulis)

15. Jalankan program pada Android Studio.
 16. Mengamati perubahan lampu yang ada pada Android dengan *Database*.

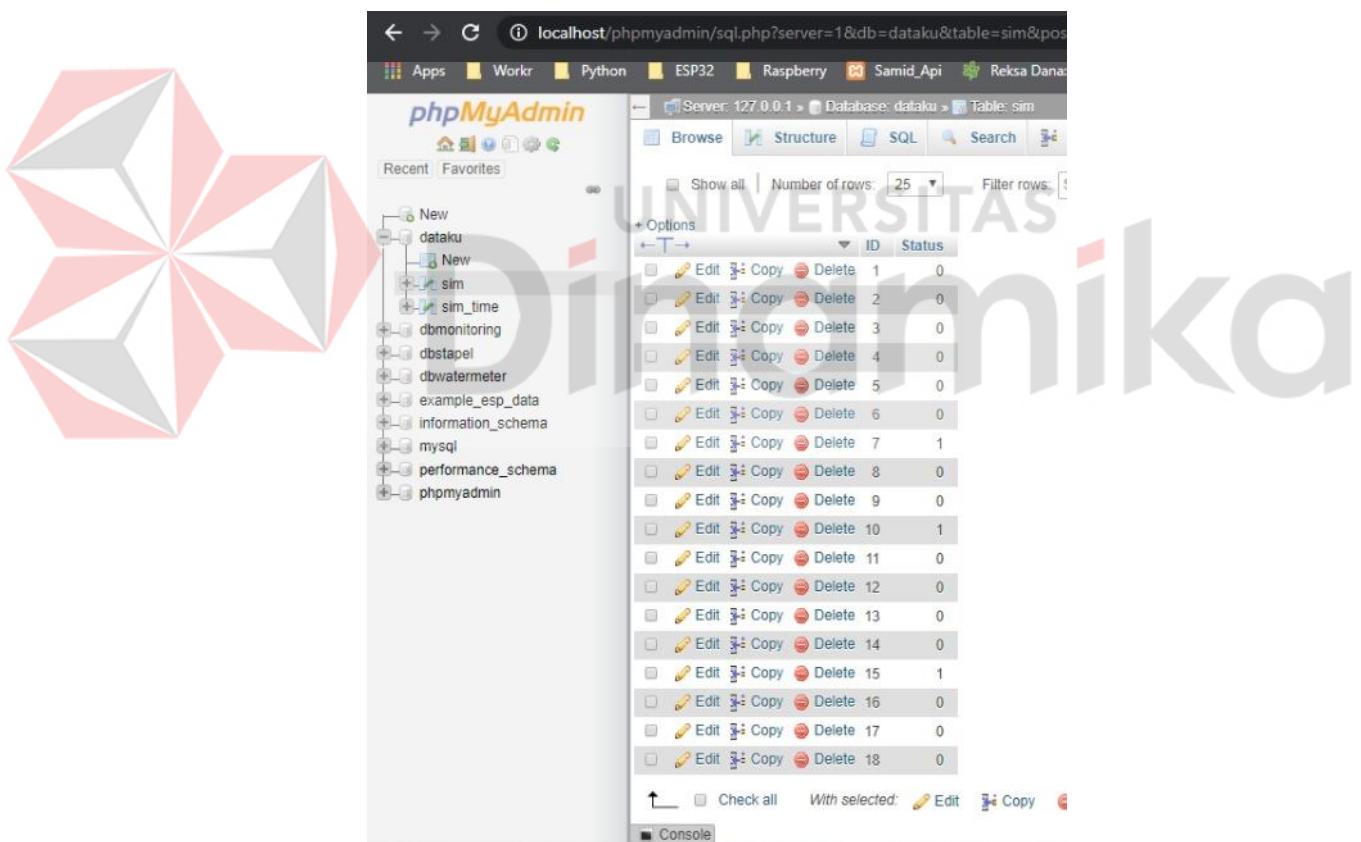
4.2.4 Hasil Pengujian

Hasil yang diperoleh dari pengujian ini adalah berupa data lampu yang ada pada Android akan 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.



```
localhost/sim-php/getdata.php
[{"id": "1", "status": "1"}, {"id": "2", "status": "1"}, {"id": "3", "status": "1"}, {"id": "4", "status": "1"}, {"id": "5", "status": "0"}, {"id": "6", "status": "0"}, {"id": "7", "status": "0"}, {"id": "8", "status": "0"}, {"id": "9", "status": "0"}, {"id": "10", "status": "1"}, {"id": "11", "status": "1"}, {"id": "12", "status": "1"}, {"id": "13", "status": "1"}, {"id": "14", "status": "1"}, {"id": "15", "status": "1"}, {"id": "16", "status": "1"}, {"id": "17", "status": "1"}, {"id": "18", "status": "1"}]
```

Gambar 4. 20 Hasil Data yang diambil oleh Android
(Sumber: Olahan Penulis)

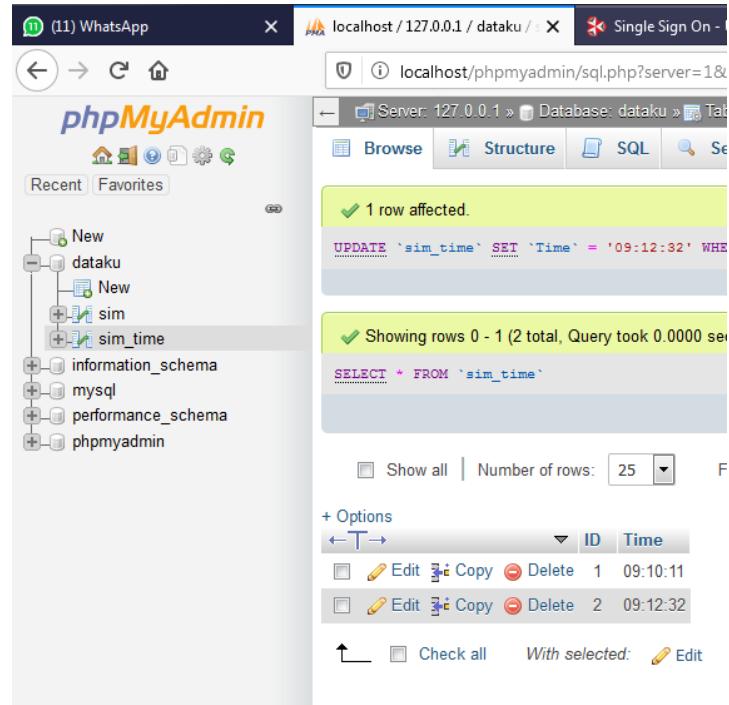


The screenshot shows the phpMyAdmin interface with the following details:

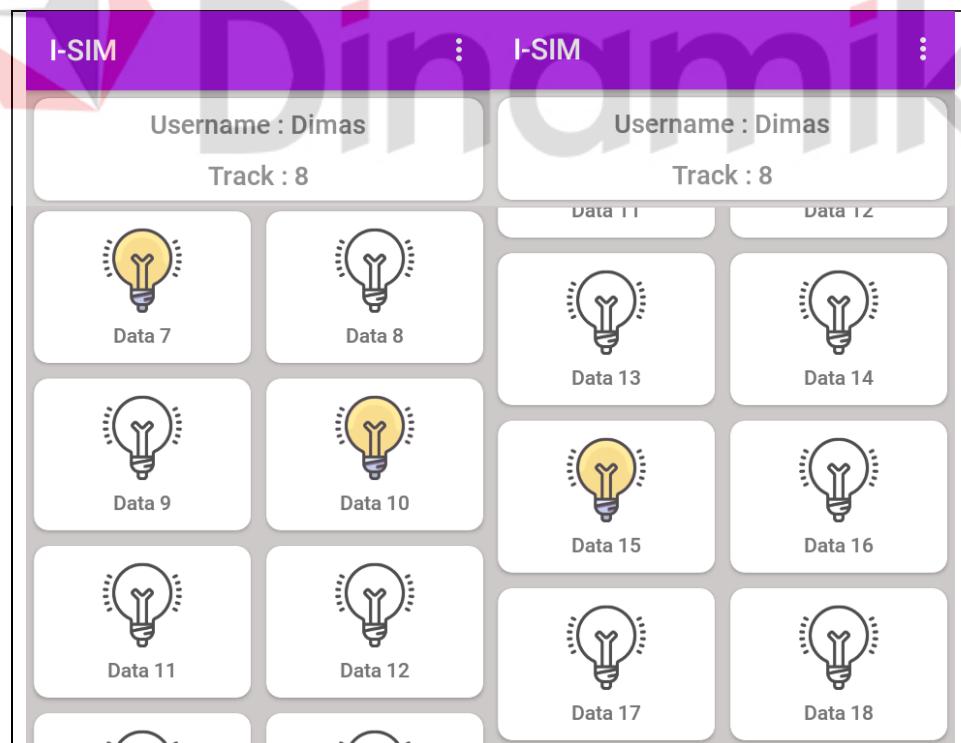
- Server:** 127.0.0.1
- Database:** dataku
- Table:** sim
- Structure:** (not shown in the screenshot)
- SQL:** (not shown in the screenshot)
- Search:** (not shown in the screenshot)
- Filter rows:** (set to 25)
- Options:** (dropdown menu)
- Rows:** 18 rows are listed, each with columns **ID** and **Status**.

ID	Status
1	0
2	0
3	0
4	0
5	0
6	0
7	1
8	0
9	0
10	1
11	0
12	0
13	0
14	0
15	1
16	0
17	0
18	0

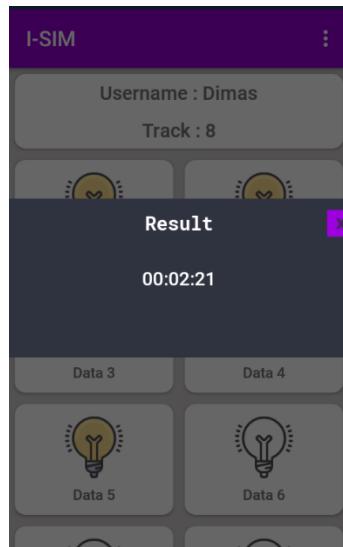
Gambar 4. 21 Data ID Patok pada Database
(Sumber: Olahan Penulis)



Gambar 4. 22 Data Waktu pada Database
(Sumber: Olahan Penulis)



Gambar 4. 23 Hasil Pengambilan Data pada Aplikasi Android
(Sumber: Olahan Penulis)



Gambar 4. 24 Hasil Pengambilan Data Waktu pada Aplikasi Android
(Sumber: Olahan Penulis)

4.3 Pengujian Jarak Android dengan *Access Point* Terhadap Kecepatan Pengiriman Data

4.3.1 Tujuan

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 akan semakin lama.

4.3.2 Alat Yang Digunakan

Alat yang digunakan untuk melakukan proses pengujian antara lain:

- Laptop atau Komputer
- Access point* (Andromax Mifi/ Handphone)

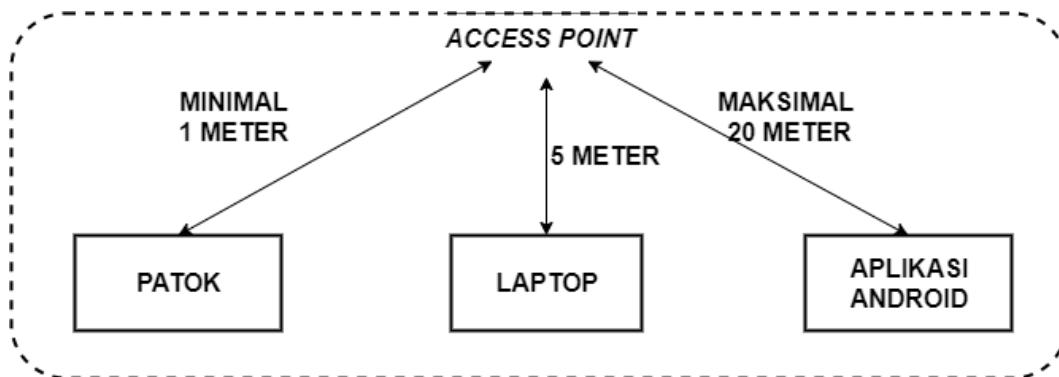
- c. Android
- d. Patok
- e. *Software XAMPP*

4.3.3 Prosedur Pengujian

Langkah-langkah yang dilakukan untuk melakukan pengujian adalah seperti berikut:

1. Menghidupkan Laptop atau Komputer.
2. Hubungkan Laptop dengan *Access point*.
3. Membuka *Software XAMPP*.
4. Menjalankan Apache.
5. Menjalankan MySQL.
6. Masuk menu Admin pada MySQL.
7. Mengamati perubahan isi dari tiap *database* hasil ujian praktik.
8. Menyalakan satu patok.
9. Menyenggol patok.
10. Membuka Aplikasi Penilaian Ujian Praktik pada perangkat Android.
11. Mengamati perubahan warna lampu sesuai dengan patok yang tersenggol dan *database* yang dibuka.
12. Menghitung waktu yang diperlukan saat patok jatuh hingga muncul pada aplikasi Android.

4.3.4 Hasil Pengujian



Gambar 4. 25 Jarak Tiap Perangkat
(Sumber: Olahan Penulis)

Berdasarkan Gambar 4.25 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 akan 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 akan mempengaruhi kecepatan pengiriman data dari patok ke perangkat Android. Semakin jauh jarak antar perangkat, maka lama pengiriman data juga akan semakin lambat.

Tabel 4. 3 Tabel Perbandingan Jarak Android ke *Access Point* Terhadap Waktu Pengiriman Data

No.	Percobaan 1		Percobaan 2		Percobaan 3		Rata-rata
	Jarak (m)	Kecepatan (ms)	Jarak (m)	Kecepatan (ms)	Jarak (m)	Kecepatan (ms)	
1	1	202	1	187	1	200	196,3
2	2	203	2	187	2	231	207
3	3	201	3	204	3	266	223,7
4	4	200	4	220	4	285	235
5	5	220	5	243	5	320	261
6	6	250	6	260	6	351	287

No.	Percobaan 1		Percobaan 2		Percobaan 3		Rata-rata
	Jarak (m)	Kecepatan (ms)	Jarak (m)	Kecepatan (ms)	Jarak (m)	Kecepatan (ms)	
7	7	298	7	286	7	374	319,3
8	8	315	8	306	8	453	358
9	9	367	9	340	9	479	395,3
10	10	401	10	380	10	500	427
11	11	487	11	420	11	543	483,3
12	12	500	12	460	12	570	510
13	13	600	13	502	13	595	565,7
14	14	700	14	607	14	628	645
15	15	800	15	714	15	699	737,7
16	16	900	16	800	16	793	831
17	17	1020	17	1000	17	888	969,3
18	18	1150	18	1200	18	979	1109,6
19	19	1250	19	1320	19	1100	1223,3
20	20	1400	20	1420	20	1249	1356,3

(Sumber: Olahan Penulis)

Dapat dilihat dari Tabel 4.3 bahwa semakin jauh jarak antara perangkat Android dengan *access point* yang terhubung maka akan 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.

4.4 Pengujian Keseluruhan Sistem

4.4.1 Tujuan

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 tersenggol 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.

4.4.2 Alat Yang Digunakan

Alat yang digunakan untuk melakukan proses pengujian antara lain:

- a. Laptop atau Komputer
- b. *Access point* (Andromax Mifi/ *Handphone*)
- c. Android
- d. Patok
- e. *Software XAMPP*

4.4.3 Prosedur Pengujian

Langkah-langkah yang dilakukan untuk melakukan pengujian adalah seperti berikut:

1. Menghidupkan Laptop atau Komputer.
2. Hubungkan Laptop dengan *Access point*.
3. Membuka *Software XAMPP*.
4. Menjalankan Apache.
5. Menjalankan MySQL.
6. Masuk menu Admin pada MySQL.
7. Mengamati perubahan isi dari tiap *database* hasil ujian praktik.
8. Menyalakan tiap patok dan susun sesuai dengan lintasan ujian.
9. Menyenggol beberapa patok.

10. Membuka Aplikasi Penilaian Ujian Praktik pada perangkat Android.
11. Mengamati perubahan warna lampu sesuai dengan patok yang tersenggol dan *database* yang dibuka.

4.4.4 Hasil Pengujian

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 berwana. Lampu-lampu tersebut akan hidup atau akan berubah warna menjadi kuning apabila patok tersenggol atau jatuh. Perubahan warna pada lampu akan mengikuti ID patok yang jatuh. Selain mengetahui patok-patok mana saja yang jatuh, aplikasi ini juga akan menampilkan hasil dari lamanya ujian praktik berlangsung untuk setiap sesinya. Lama waktu disini diperoleh berdasarkan selisih waktu dari patok yang mempunyai sensor ultrasonik awal dan ultrasonik akhir. Selisih inilah yang akan ditampilkan jika pengguna memilih menu “*Result*” pada aplikasi.

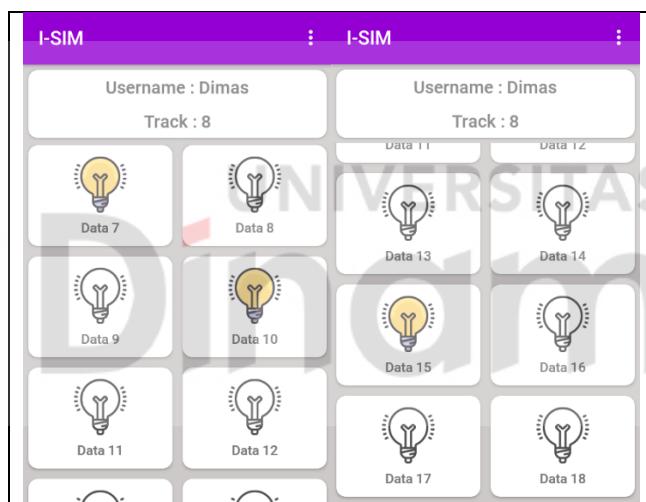
Tabel 4. 4 Hasil Percobaan Pada Android

ID Patok	Percobaan 1		Percobaan 2		Percobaan 3	
	Patok	Android	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

ID Patok	Percobaan 1		Percobaan 2		Percobaan 3	
	Patok	Android	Patok	Android	Patok	Android
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	

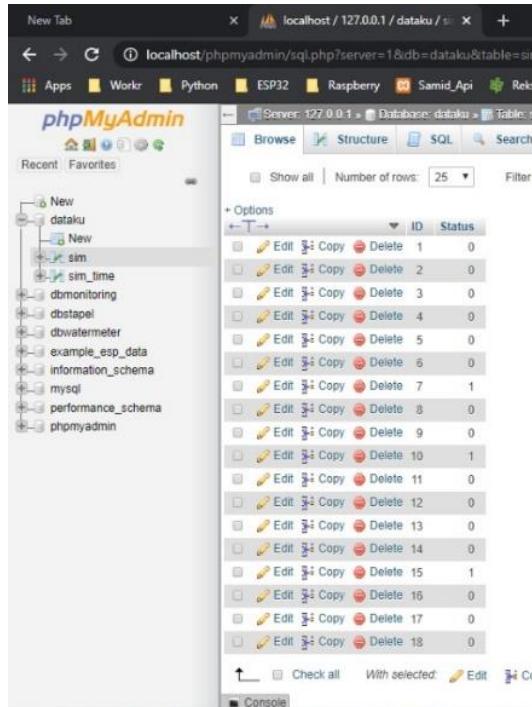
(Sumber: Olahan Penulis)

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 akan menampilkan lampu yang menyala.



Gambar 4. 26 Hasil Tampilan Kesalahan pada Android

(Sumber: Olahan Penulis)



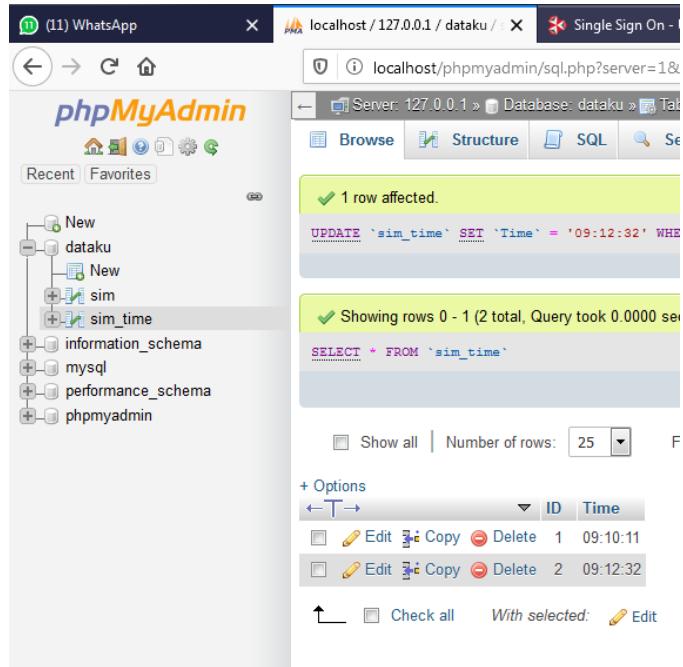
Gambar 4. 27 Hasil Pada Database
(Sumber: Olahan Penulis)

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

Tabel 4. 5 Selisih Waktu untuk Lamanya Ujian Praktik

Percobaan	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

(Sumber: Olahan Penulis)



Gambar 4. 28 Data waktu pada Database
(Sumber: Olahan Penulis)



Gambar 4. 29 Hasil Selisih Waktu dari Dua Patok pada 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.

BAB V

PENUTUP

Dalam bab ini berisi tentang kesimpulan dan saran yang berdasarkan pada hasil dari pengujian yang telah dilakukan pada Tugas Akhir ini, maka dapat diperoleh beberapa kesimpulan dan saran untuk pengembangan sistem berikutnya.

5.1 Kesimpulan

Berdasarkan hasil dari pengujian yang dilakukan pada penelitian ini, didapatkan beberapa poin kesimpulan 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 tersebut kemudian akan diteruskan dan disimpan ke dalam *database* yang nantinya akan diambil oleh perangkat Android. Aplikasi Android menampilkan hasil ujian praktik berdasarkan status tiap-tiap patok dan hasil waktu dari lama ujian praktik berlangsung.
2. Pengujian dilakukan sebanyak tiga kali dengan persentase keberhasilan 100% dimana pengiriman data yang dilakukan dapat disimpan oleh *database* dan ditampilkan 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 dengan cepat.

5.2 Saran

Untuk melakukan pengembangan lebih lanjut mengenai Tugas Akhir ini, maka penulis memberikan beberapa saran sebagai berikut:

1. Untuk melakukan pengembangan selanjutnya, dapat dilakukan dengan menggunakan baterai tipe 2x18650 atau baterai yang lebih tahan lama pada perangkat *publisher* atau patok sehingga dalam pengambilan data dapat berjalan lebih lama.
2. Penggunaan rangkaian LDO (*Low Drop Out*) Voltage dari 5V menjadi 3,3V sehingga tidak boros untuk kapasitas baterai yang digunakan.



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*.
- 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*.

