IMPLEMENTASI DAN ANALISIS FITUR KEAMANAN PROTOKOL MQTT PADA *TELEHEALTHCARE*

TUGAS AKHIR

Program Studi
S1 TEKNIK KOMPUTER

Oleh:
Dian Rachmadini
16410200026

FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
2020

IMPLEMENTASI DAN ANALISIS FITUR KEAMANAN PROTOKOL MQTT PADA *TELEHEALTHCARE*

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Teknik

Oleh :

Nama	**: Dian Rachmadini**
NIM	**: 16410200026**
Program Studi	**: S1 Teknik Komputer**

FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
2020

Tugas Akhir

IMPLEMENTASI DAN ANALISIS FITUR KEAMANAN PROTOKOL MQTT PADA *TELEHEALTHCARE*

Dipersiapkan dan disusun oleh

Dian Rachmadini

NIM : 16410200026

Telah diperiksa, diuji dan disetujui oleh Dewan Penguji

Pada : Kamis, 13 Agustus 2020

Susunan Dewan Penguji

Pembimbing

I. Dr. Jusak

NIDN. 0708017101

Digitally signed by Universitas Dinamika
Date: 2020.09.07 14:24:57 +07'00'

II. Ira Puspasari, S.Si., M.T.

NIDN. 0710078601

Digitally signed by Universitas Dinamika
Date: 2020.09.08 03:19:57 +07'00'

Pembahas

Dr. Susijanto Tri Rasmana, S.Kom., M.T.

NIDN. 0727097302

Digitally signed by Universitas Dinamika
Date: 2020.09.07 20:34:31 +07'00'

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

Digitally signed by Universitas Dinamika
Date: 2020.09.08 15:21:54 +07'00'

Dr. Jusak

NIDN:0708017101

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA

“Selesaikan Apa yang Sudah Kamu Mulai”

Kupersembahkan Kepada

ALLAH SWT

Kedua Orang tua dan seluruh keluarga yang selalu mendukung, memotivasi dan mendoakan yang terbaik untuk saya.

Serta rekan-rekan S1 Teknik Komputer yang selalu membantu, mendukung dan memotivasi untuk menjadi pribadi yang lebih baik lagi.

SURAT PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, saya :

Nama	:	Dian Rachmadini
NIM	:	16410200026
Program Studi	:	S1 Teknik Komputer
Fakultas	:	Fakultas Teknologi Informasi
Jenis Karya	:	Tugas Akhir
Judul Karya	:	**IMPLEMENTASI DAN ANALISIS FITUR KEAMANAN PROTOKOL MQTT PADA *TELEHEALTHCARE***

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 Royalti 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, 31 Juli 2020

Yang menyatakan

Dian Rachmadini

NIM : 16410200026

ABSTRAK

Penerapan Internet of Things saat ini sudah menjangkau ke semua bidang salah satunya dalah bidang kesehatan atau disebut dengan telehealthcare. Pada bidang kesehatan ini data setiap pasien bersifat privasi dan sangat penting, seperti halnya data sinyal EKG yang berbeda – beda setiap pasien. Pengiriman data Sinyal EKG pada menggunakan konsep IoT memerlukan protokol yang sederhana, dan aman. Protokol Message Queue Telemetry Transport (MQTT) merupakan salah satu protokol yang sering digunakan untuk menerapkan konsep ini. Pada penelitian ini, protokol MQTT diterapkan untuk pengiriman data sinyal EKG dengan mengaktifkan fitur keamanan pada protokol ini menggunakan TLS. Proses yang dilakukan untuk dapat melakukan pengiriman yang aman adalah dengan menggunakan kunci yang telah dibuat terlebih dahulu oleh broker mosquitto. Kunci (ca.crt) ini dibuat oleh menggunakan protokol TLS dan akan dipasang oleh setiap client dikarenakan enkripsi bersifat simetris. Sehingga kunci yang digunakan oleh client dan server sama. Kemudian, pengiriman dilakukan dengan 2 penerima yang dianalogikan dokter dan perawat/petugas rumah sakit. Proses pengiriman tersebut dilakukan analisis data dengan rekam menggunakan wireshark untuk mengetahui lama waktu proses enkripsi data, dan untuk melihat besar paket setelah enkripsi. Kemudian untuk pengujian integritas dilakukan menggunakan metode cross correlation pada MATLAB. Proses transmisi dilakukan pada Qos 0 dan Qos 1. Data yang diambil dengan sample 10 orang yang dikirim secara real-time. Hasil analisis perhitungan selisih besar paket sebelum dan setelah pengiriman pada qos 0 dan qos 1 menunjukkan selisih yang cukup besar, tetapi pengiriman data menjadi lebih aman. Hasil waktu proses enkripsi yang diperlukan pada qos 0 dan qos 1 menghasilkan rata – rata sedikit lebih lama pada qos 1. Hal ini, dikarenakan qos 1 terdapat PUBACK pada proses pengirimannya. Kemudian hasil pengujian integritas data dengan cross-correlation qos 0 dan qos 1 menunjukkan hasil 1 pada lag ke-0 yang dapat diartikan bahwa data yang dikirim dan diterima adalah sama.

Keywords : *Protokol MQTT, TLS, MQTT Secure, sinyal EKG.*

KATA PENGANTAR

Pertama-tama penulis panjatkan puji dan syukur atas kehadiran Allah SWT, karena berkat izin, Rahmat dan hidayah-nya penulis dapat menyelesaikan laporan penelitian ini yang merupakan salah satu syarat menempuh Tugas Akhir pada Program Studi S1 Teknik Komputer di Fakultas Teknologi dan Informatika Universitas Dinamika. Shalawat serta salam tidak lupa selalu penulis panjatkan kepada Rasulullah SAW.

Di dalam buku Tugas Akhir ini dilakukan pembahasan mengenai Implementasi Dan Analisis Fitur Keamanan Protokol Mqtt Pada *Telehealthcare*. Dalam usaha menyelesaikan Tugas Akhir ini penulis banyak mendapatkan bantuan dari berbagai pihak baik moral maupun materi. Oleh karena itu penulis mengucapkan terima kasih dan penghargaan setinggi-tingginya kepada:

1. Orang tua dan saudara-saudara tercinta yang telah memberikan dukungan dan bantuan baik moral maupun materi sehingga penulis dapat menempuh dan menyelesaikan Tugas Akhir maupun laporan ini.
2. Kepada Bapak Dr. Jusak dan juga kepada Ibu Ira Puspasari, S.Si., M.T. selaku Dosen Pembimbing. Terima kasih atas bimbingan yang diberikan sehingga penulis dapat menyelesaikan Tugas Akhir dengan baik.
3. Kepada Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi Teknik Komputer Surabaya atas ijin yang diberikan untuk mengerjakan Tugas Akhir ini.
4. Semua staf dosen yang telah mengajar dan memberikan ilmunya.
5. Terima kasih terhadap rekan-rekan S1 Teknik Komputer khususnya rekan-rekan seperjuangan angkatan 2016 khususnya Prodi S1 Teknik Komputer yang selalu memberikan semangat dan bantuannya.
6. 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.

Tugas akhir ini dikerjakan di tengah pandemi Covid-19, dimana proses yang penulis lalui tentunya berbeda dengan proses pada umumnya. Bimbingan dan sidang pun dilakukan secara daring *(online)*, meskipun demikian penulis

tetap yakin dapat menyelesaikan perjalanan hingga akhir studi di masa pandemi ini. Penulis berharap semoga laporan ini dapat berguna dan bermanfaat untuk menambah wawasan bagi pembacanya. Penulis juga menyadari dalam penulisan buku Tugas Akhir ini masih terdapat banyak kekurangan. Oleh karena itu penulis berharap adanya saran maupun kritik dalam memperbaiki kekurangan dan berusaha untuk lebih baik lagi kedepannya.

Surabaya, 18 Juni 2020

Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

DAFTAR LAMPIRAN

Halaman

BAB I
PENDAHULUAN

1.1 Latar Belakang

Istilah *Internet Of Things* atau yang biasa dikenal sebagai IoT adalah sebuah konsep di mana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke manusia atau manusia ke komputer. Penerapan IoT saat ini kerap kali diterapkan pada pengiriman data sensor, baik pengiriman data *real time* maupun tidak, tergantung penggunaan dan pengaturan pada protokol yang digunakan.

Protokol – protokol yang diciptakan untuk komunikasi *Internet Of Things* ini sangat beragam diantaranya yaitu protokol *Message Queue Telemetry Transport* (MQTT). Protokol ini sangat familiar dalam penerapan sistem IoT. Untuk pegiriman data terlebih pada konsep IoT memerlukan fitur keamanan untuk menjaga keamanan data itu sendiri. Seperti halnya jika data tersebut berupa data sinyal EKG (Elektrokardiogram) yang merupakan data sinyal jantung manusia. Sinyal EKG ini berisi informasi kesehatan penting seorang pasien yang bersifat unik untuk setiap individu dalam jangka waktu yang panjang. Sinyal EKG juga dapat bertindak sebagai identitas biometrik untuk membedakan informasi spesifik yang dimiliki orang tertentu. Fitur ini membawa konsekuensi langsung pada transmisi sinyal EKG yang membuatnya rentan terhadap serangan dari luar. *Platform e-Health* berbasis internet yang mengabaikan perlindungan informasi kesehatan merupakan ancaman bagi privasi pasien. Sayangnya, belum ada *platform e-Health* yang menerapkan perlindungan transmisi sinyal EKG. *(Setiawan, 2018).*

Pada tahun 2018, Bramasta Agnanda Setiawan melakukan penelitian tentang anonimasi sinyal elektrokardiogram untuk kemanan proses transmisi data pada node sensor. Pada penelitian tersebut proses keamanan terletak pada data sinyal Elektrokardiogram, di mana data sinyal tersebut di-anonimasi sehingga untuk proses pengiriman menjadi lebih aman.

Selanjutnya juga terdapat penelitian oleh Ayaskanta Mishra pada tahun 2018, mengenai penerapan MQTT pada *Telehealthcare* menggunakan broker *online* yaitu CloudMQTT untuk pemantauan sinyal EKG. Penelitian ini dilatar belakangi oleh orang – orang yang tinggal jauh dari perkotaan, supaya tetap mendapatkan layanan kesehatan yang cepat dan efektif.

Kemudian pada tahun 2019 Hyunwoo Lee melakukan penelitian mengenai *Secure* MQTT pada sisi broker, dimana broker dibatasi hanya bisa membaca topik yang terpublish tanpa mengetahui pesan apa yang dikirimkan. Karena broker disini dianggap *man-in-the-middle* diantara klien. Oleh karna itu penelitian ini menerapkan MQTLS untuk menjaga keamanan terhadap pesan yang dikirim sampai ke penerima. Pada tahun yang sama Neven Nikolov juga melakukan penelitian dengan judul *"Research of Secure Communication of Esp32 IoT Embedded System to.NET Core Cloud Structure using MQTTS SSL/TLS"*, dengan menerapkan protokol MQTTS untuk pengiriman data menggunakan sensor DHT22.

Pada tahun 2019, M. Reza Bintami melakukan penelitian tentang pengiriman data heartrate menggunakan protokol MQTT. Pada penelitian tersebut data heartrate bukan termasuk data streaming dan protokol MQTT yang digunakan tidak terdapat fitur keamanan, sehingga keamanan data yang dikirimkan kurang terjamin.

Berdasarkan latar belakang di atas, dalam Tugas Akhir ini akan diimplementasi dan dianalisis proses pengiriman data sinyal Elektrokardiogram (EKG) menggunakan protokol MQTT dengan mengaktifkan fitur keamanan yang menggunakan protokol TLS atau biasa disebut dengan MQTTS. Sehingga data sinyal EKG tetap terjaga keamanannya. Dari pengujian tersebut kemudian dilakukan analisis terhadap beberapa parameter antara lain : Selisih besar paket data sebelum enkripsi dan sesudah enkripsi, selisih waktu sebelum pengiriman dan sesudah pengiriman, dan pengujian integritas data.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan permasalahan sebagai berikut :

1. Bagaimana melakukan implementasi dan analisis fitur keamanan protokol MQTT pada sinyal EKG?
2. Bagaimana melakukan perbandingan ukuran paket dan selisih waktu yang dibutuhkan sebelum dan sesudah enkripsi?
3. Bagaimana melakukan pengujian korelasi data sinyal EKG untuk mengetahui validitas data?

1.3 Batasan Masalah

Dalam perancangan ini, terdapat beberapa batasan masalah, antara lain :

1. Data yang digunakan merupakan data sinyal Elektrokardiogram (EKG)
2. Metode enkripsi yang digunakan menggunakan enkripsi simetris dengan panjang kunci 2048 bit.
3. Sampel pengujian merupakan laki – laki maupun perempuan dengan batas usia antara 20 – 55 tahun dengan jumlah 10 orang dan 3 kali pengambilan data tiap orang.
4. Perangkat yang digunakan yaitu *Raspberry pi* 2 / 3, modul AD8232, dan ADS1115.
5. Komunikasi antara sensor AD8232 dengan *raspberry* menggunakan I2C (*Inter Integrated Circuit*).
6. *Broker* MQTT menggunakan mosquito

1.4 Tujuan

Adapun beberapa tujuan dilakukannya penelitian ini sebagai berikut :

1. Melakukan implementasi dan analisis fitur keamanan protokol MQTT pada *telehealthcare* seperti : pengiriman data sinyal Elektrokardiogram (EKG).
2. Melakukan analisis terhadap ukuran paket dan selisih waktu yang dibutuhkan sebelum dan setelah enkripsi

3. Melakukan pengujian korelasi data untuk mengetahui validitas datasinyal EKG sebelum dan setelah dilakukan pengiriman menggunakan MQTTS.

BAB II
LANDASAN TEORI

2.1 Perkembangan Penelitian Pengamanan Pengiriman Sinyal Jantung

Penelitian tugas akhir ini dilatar belakangi beberapa penelitian sebelumnya yaitu pada tahun 2018 Bramasta Agnanda S, dkk melakukan penelitian yaitu "Anonimasi Sinyal EKG (Elektrokardiogram) untuk Keamanan Transmisi Data pada Sebuah Node Sensor" , Metode Anonimasi yang digunakan berbasis algoritma *Jusak-Seedahmed* dengan *wavelet-packet*. Algoritma *Jusak-Seedahmed* digunakan untuk menguraikan sinyal EKG dalam domain frekuensi, yaitu dengan cara memisahkan komponen sinyal EKG frekuensi tinggi untuk proses anonimasi. Metode Anonimasi sinyal ini bertujuan untuk mengamankan sinyal EKG secara daring melalui jaringan internet sekaligus melindungi sinyal EKG dari jangkauan para peretas *(Setiawan, 2018).*

Pada tahun yang sama (2018) Sony Solehuddin juga melakukan penelitian melanjutkan penelitian Bramasta Agnanda S, mengenai "Rekonstruksi Sinyal EKG (Elektrokardiogram) Hasil Proses Anonimasi dengan Tampilan pada Aplikasi Android". Proses rekonstruksi dimulai dengan mengambil kunci rahasia terenkripsi, *K*, dari penyimpan elektronik yang ada dirumah sakit dan bagian EKG yang tersimpan pada *cloud server*. Sebelumnya kunci rahasia ini didapatkan dari sinyal EKG frekuensi rendah pada saat proses anonimasi sinyal.

Proses rekontruksi dilakukan berdasarkan algoritma FFT *(Fast Fourier Transform)* yaitu melakukan transformasi bagian sinyal EKG yang telah diambil dari *cloud server* dari domain waktu ke domain frekuensi. Kemudian terdapat langkah penting yaitu mengalihkan komponen sinyal frekuensi tinggi dengan vektor untuk mendapatkan komponen sinyal frekuensi tinggi seperti saat sebelum dimodifikasi. Hasil dari proses anonimasi dan rekonstruksi sinyal EKG juga dilakukan pengujian korelasi silang dan menghasilkan nilai 1 pada *lag* ke-0 yang menunjukkan bahwa sinyal EKG hasil rekonstruksi sama persis dengan sinyal EKG sebelum proses anonimasi.

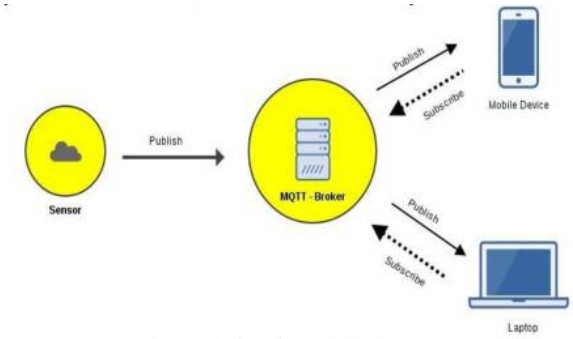
Pada tahun 2019 Adrian Febiyanto, melakukan penelitian dengan melakukan yaitu "Pengukuran dan Pengamatan Sinyal Electrocardiogram menggunakan Raspberry dengan Tampilan Aplikasi Mobile" di mana penelitian ini bertujuan untuk dapat memantau sinyal EKG seseorang secara *real-time* dengan sistem pengiriman data dari node sensor menggunakan *Raspberry pi* menuju *firebase. Firebase* disini merupakan database yang bisa diakses secara real-time. Selanjutnya dari *firebase* dikirim ke aplikasi berbasis *mobile.* Hasil dari penelitian ini menyimpulkan bahwa aplikasi berbasis *mobile* yang telah dibuat dapat menampilkan sinyal hasil pembacaan EKG dari basis data *Firebase* tidak memiliki perbedaan. Hal ini ditunjukkan dari hasil pengujian *cross-correlation* yang mana seluruh data dalam pengujian memberikan nilai 1 pada *lag* ke-0 *(Febiyanto, 2019).*

Selanjutnya pada tahun 2019 M. Reza Bintami melakukan penelitian yaitu "Rancang Bangun Transmisi Data Heart Rate menggunakan Protokol MQTT" di mana protokol MQTT ini digunakan untuk proses pengiriman data *Heart Rate. Heart Rate* juga merupakan salah satu jenis sinyal jantung yang bisa didapatkan dari sinyal Elektrokardiogram. Pengujian pengiriman data ini berdasarkan QoS 0, dan 1 berdasarkan parameter *delay, throughput, dan packet loss*. Dari hasil penelitiannya, menunjukkan hasil bahwa penggunaan protokol MQTT pada parameter *packet loss* dan *delay* tergolong sangat bagus karena *packet loss* yang dihasilkan kurang dari 1% dan rata-rata *delay* 20,21 ms pada Qos 1.

2.2 Protokol MQTTS

Protokol MQTTS merupakan versi aman dari MQTT. Sedangkan protokol MQTT *(Message Queue Telemetry Transport)* sendiri adalah protokol yang berjalan di atas TCP/IP. MQTT menggunakan metode *publish/subscribe message*. Perangkat publisher dan subscriber terhubung satu sama lain melalui penghubung yang disebut broker. Ketika publisher mengirimkan pesan, pesan tersebut akan dikirim kepada broker terlebih dahulu, kemudian akan diteruskan kepada subscriber. Dengan metode publish/subscribe, untuk mendapatkan data yang di publish oleh publisher, subscriber hanya perlu melakukan subscribe topik yang

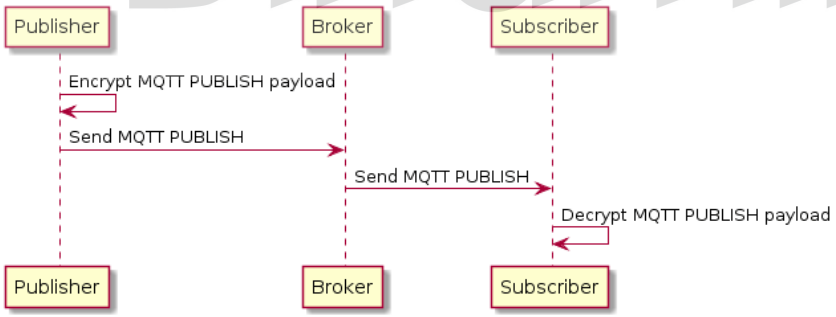
sama dengan topik yang di publish oleh publisher. Dengan demikian subscriber akan mendapatkan data tanpa harus melakukan request berulang-ulang. Gambar 2.1 meunjukkan ilustrasi proses pengiriman protokol MQTT.



Gambar 2.1 Ilustrasi Proses Kerja Protokol MQTT

(Sumber:Bintami, 2019)

Pada segi kemanan MQTT terdapat banyak cara salah satunya dengan menambahkan SSL/TLS. Penggunaan TLS pada MQTT dikarenakan protokol MQTT berada di atas TCP/IP. SSL/TLS ini akan diimplementasikan pada sisi broker. Protokol MQTT dengan TLS/SSL dapat juga disebut dengan MQTTS. Port protokol MQTTS berada pada port 8883. Cara kerja protokol MQTTS sama halnya dengan HTTPS karena menerapkan TLS/SSL untuk kemanannya. Gambar 2.2 merupakan struktur pengiriman paket data menggunakan MQTTS.



Gambar 2.2 Encryption Scenario Client to Broker
(Sumber : *HiveMQ*, 2015)

2.3 Protokol TLS

TLS *(Transport Layer Security)* dirilis pada tahun 1999 dengan tujuan untuk menjadi protokol standar komunikasi yang aman. Protokol ini didesain agar komunikasi antara client dan server tidak dapat di sadap atau diketahui oleh orang

lain. Protokol TLS bekerja pada layer transport dimana aliran data komunikasi digital dikelola. Lapisan ini merupakan merupakan bagian dari sistem transportasi yang dipisahkan dari lapisan aplikasi sehingga terpisah dari pengguna.

2.3.1 Mekanisme Keamanan TLS

TLS terdiri dari 3 kumpulan kriptografi yaitu :

- *Authentication*

  Authentication diperoleh dengan menggunakan *asimetric cipher* seperti RSA, Dife-Helman, dan lain-lain.

- *Confidentially*

  Confidentiality diperoleh dengan melakukan enkripsi *simetric* dari *plaintext* melalui transfer jaringan. Secara umum *simetric cipher* yang kuat diimplementasikan di TLS seperti AES, DES-3, RC4, dan sebagainya.

- *Integrity*

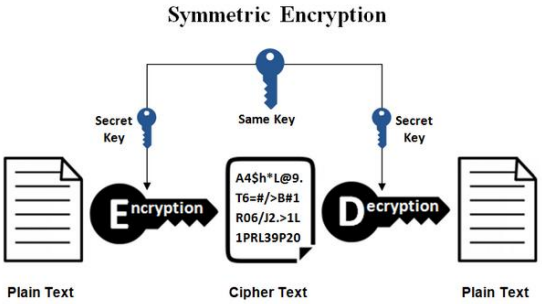
  Intergity diperoleh dengan menghitung *Message Authentication Code* (MAC) dari paket MD5 atau SHA-1. *(Fadhli, 2015)*

2.4 Kriptografi

Salah satu solusi yang dapat digunakan untuk menjamin kerahasiaan maupun keamanan dari suatu informasi adalah dengan kriptografi. Kriptografi memegang peranan penting dalam memberikan keamanan terhadap data yang dikirimkan melalui internet. Dengan kriptografi, maka pengamanan sebuah data dapat dilakukan. Suatu data yang tadinya bisa dibaca / dikenali dengan mudah, maka dengan kriptografi akan menjadi sulit dikenali karena telah melalui proses pengacakan pada tahap enkripsi. Pada kriptografi terdapat 2 tahap yang paling utama, yaitu : enkripsi dan dekripsi. Pada tahap enkripsi, akan dilakukan pengacakan sebuah data / teks ke dalam format atau bentuk yang susah untuk dikenali (cipherteks). Sedangkan tahap dekripsi adalah tahapan untuk mengubah data yang telah diacak ke dalam bentuk aslinya (plainteks) *(Fadlan, 2017)*.

2.5 Enkripsi Simetris dan Asimetris

Enkripsi simetris adalah enkripsi yang paling sederhana yang hanya melibatkan satu kunci rahasia untuk menyandikan dan menguraikan informasi. Contoh enkripsi simetris adalah Blowfish, AES, RC4, DES, RC5, dan RC6, sedangkan algoritma simetris yang paling umum digunakan adalah AES-128, AES-192, dan AES-256.



Gambar 2.3 Sistem Enkripsi Simetris

(Sumber : Naharuddin, 2018)

Dari gambar di atas, dapat dilihat bahwa untuk dapat melakukan enkripsi dan deskripsi sebuah plain text menggunakan secret key (kunci) yang sama. *Secret key* dari *plain text* pertama dilakukan untuk enkripsi data kemudian pada sisi penerima secret key tersebut digunakan untuk mendeskripsi dari bentuk *chiper text* hasil enkripsi menjadi *plain text* (sesuai dengan yang dikirimkan). Kelemahan enkripsi simetris ini bahwa semua pihak yang terlibat harus menukar kunci yang digunakan untuk mengenkripsi data sebelum mereka dapat mendekripsinya.

Sedangkan enkripsi asimetris pengembangan dari metode enkripsi simetris dan lebih menjamin keamanan. Enkripsi asimetris menggunakan dua kunci yaitu kunci publik *(public key)* dan kunci pribadi *(private key)*. Kunci publik tersedia secara bebas bagi siapa saja yang mungkin ingin mengirimkan pesan, namun kunci pribadi (kunci kedua) dirahasiakan sehingga hanya dari sisi penerima saja yang mengetahuinya. *(Naharuddin, 2018)*

2.6 *Cross Correlation*

Cross Corrrelation atau korelasi silang adalah suatu metode standar yang digunakan untuk mengukur similaritas atau kesamaan dari dua buah

sinyal dalam deret waktu dengan cara menggeser salah satu sinyal kemudian dicari tingkat kesamaannya dengan cara mengkalikan setiap nilai pada setiap fungsi sinyal terhadap fungsi sinyal yang lain. Jika nilai koefisien korelasi sebesar 1, maka kedua sinyal tersebut sama. Jika nilai koefisien korelasi sebesar 0, maka kedua sinyal tersebut tidak sama. *(Solehudin, 2018)*

Cross correlation didefinisikan dengan persamaan korelasi silang :

$$r(d) = \frac{\sum_i [(x(i) - mx) * (y(i-d) - my)]}{\sqrt{\sum_i (x(i) - mx)^2} \sqrt{\sum_i (y(i-d) - my)^2}}$$

Keterangan :

r(d) =korelasi silang

mx = nilai rata – rata pada nilai x

my = nilai rata – rata pada nilai y

x = nilai pada fungs sinyal x

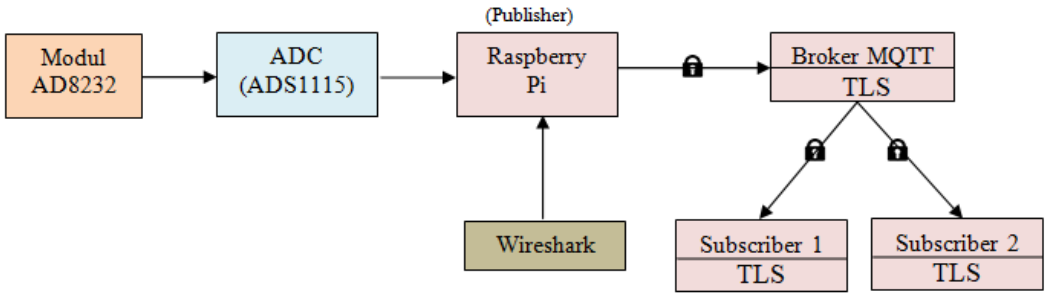
y = nilai pada fungsi sinyal y

i = indeks pada sinyal (1,2,3,…,i)

BAB III
METODOLOGI PENELITIAN

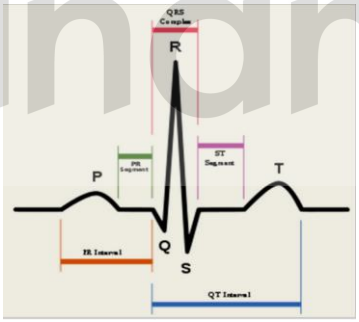
3.1 Metode Penelitian

Pada bab ini membahas tentang bagaimana proses komunikasi dan model sistem yang akan diterapkan :



Gambar 3.1 Blok Diagram Sistem

Pada Gambar 3.1, merupakan blok diagram dari penelitian ini. Modul AD8232 di atas merupakan modul sensor EKG yang berfungsi untuk mendeteksi aktifitas kelistrikan jantung manusia.



Gambar 3.2 Sinyal EKG

(Sumber:Febiyanto, 2019)

Sinyal jantung manusia terdiri atas 4 sinyal yaitu P-Q-R-S-T, di mana P adalah gelombang awal yang merupakan hasil depolarisasi di kedua atrium, Gelombang Q merupakan gelombang defleksi negatif setelah gelombang P.Gelombang R merupakan gelombang defleksi positif setelah gelombang P atau setelah gelombang Q. Gelombang S merupakan gelombang defleksi negatif setelah gelombang R atau gelombang Q. Gelombang T merupakan hasil

repolarisasi di kedua vetrikel. Normalnya positif dan *inverted* (terbalik) di AVR. *(Setiawan, 2018)*

Pengukuran sinyal EKG ini menggunakan elektroda penjapit yang dipasangkan pada kaki kanan *(left right)*, lengan kiri *(left arm)*, dan lengan kanan *(right arm)*. Kemudian dari elektroda penjapit akan mengirimkan sinyal analog yang akan dibaca oleh modul ad8232 yang berfungsi sebagai ADC. Dari modul ad8232 supaya *Raspberry pi* bisa membaca nilai adc tersebut maka dibutuhkan modul I2C yaitu ADS1115 dikarenakan *Raspberry pi* tidak memiliki pin ADC. Data ADC yang diterima oleh ADS1115 akan dikirim ke raspberry menggunakan komunikasi I2C.

Selanjutnya, untuk dapat menerima data modul ADS1115 maka perlu mengaktifkan port I2C dan install library ADS1115 dengan cara yang dijelaskan pada lampiran (1) 1.7 dan 1.8. Sedangkan skematik rangkaian untuk pengambilan data dapat dilihat pada lampiran (1) 1.9. Kemudian data yang masuk akan langsung dikirim ke broker *mosquitto*. Data yang dikirim dengan menggunakan kunci dari broker, dikarenakan sistem enkripsi pada *mosquitto* adalah enkripsi asimetris yaitu penggunaan kunci yang sama oleh semua klien.

3.1.1 Tahap pembuatan Kunci

1. Langkah pertama yang dilakukan adalah install broker mosquitto. Broker Mosquitto merupakan broker lokal dan akan diinstall pada *Raspberry pi* dengan syntax `sudo apt install mosquitto mosquitto-clients`, melalui *Command prompt* seperti pada lampiran 1 (1.1).
2. Langkah selanjutnya adalah penginstalan openssl untuk pembuatan kunci. Openssl ini nantinya juga akan diinstall pada seluruh client dengan syntax `sudo apt-get install openssl`, melalui *Command Prompt* seperti pada lampiran 1 (1.2).
3. Selanjutnya membuat file kunci melalui *command prompt* dengan tahap sebagai berikut (dapat dilihat pada lampiran 1 (1.3)):

a. langkah pertama adalah membuat CA key pair yang akan digunakan untuk membuat CA certificate dengan menuliskan syntax `openssl genrsa -des3 -out ca.key 2048`
b. Kemudian membuat file CA *certificate* dengan menggunakan file ca.key di atas, file CA *certificate* (ca.crt) ini akan digunakan untuk publish dan subscribe data. Syntax : `openssl req -new -x509 -days 1826 -key ca.key -out ca.crt`
c. Selanjutnya membuat server *key pair* (server.key) yang nantinya akan digunakan oleh broker dengan *command* seperti sebagai berikut : `openssl genrsa -out server.key 2048`
d. Kemudian membuat server *certificate request* (server.csr) dengan menggunakan server.key, dengan *command* seperti sebagai berikut : `openssl req -new -out server.csr -key server.key`
e. Kemudian membuat server.crt (*Server Certificate)* menggunakan kunci CA yang telah dibuat sebelumnya untuk verifikasi dan menandatangani server.crt , dengan *command* seperti sebagai berikut : `openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CA createserial -out server.crt -days 360`
f. Berikut adalah beberapa kunci yang sudah dibuat yaitu : ca.key, ca.crt, ca.srl, server.crt, server.csr, server.key.

4. Setelah membuat CA *file*, *key file*, dan *server certificate* pada tahap sebelumnya. Ketiga *file* ini kemudian akan digunakan untuk konfigurasi pada mosquitto untuk mengaktifkan TLS melalui mosquitto.conf dengan cara menuliskan syntax pada *command prompt* : (dapat dilihat pada lampiran 1 (1.4))

```
sudo nano /etc/mosquitto/mosquitto.conf
```

5. Kemudian aktifkan fitur TLS pada mosquitto dengan menambahkan port 8883 (port MQTTS) dan menambahkan letak *directory* dari cafile

(file.crt), keyfile (file.key), dan certfile (file.certs). (dapat dilihat pada lampiran 1 (1.4))

Kunci TLS yang dibuat ini nantinya akan digunakan para klien untuk melakukan transmisi data. File kunci yang digunakan adalah ca.crt (*certificate file).* Jika salah satu klien tidak memiliki kunci tersebut maka pengiriman atau penerimaan data tidak dapat dilakukan.

Selain memberikan file kunci ke klien. Pada sisi klien juga harus menginstall kunci tersebut supaya kunci dapat digunakan dalam proses transmisi. Karena dalam proses transmisi harus menyertakan kunci TLS tersebut baik *publisher* maupun *subscriber*.

3.1.2 Proses *install* kunci pada klien

Klien disini berupa PC/Laptop sebelumya klien juga sudah terinstall *mosquitto* dan juga openssl. Langkah – langkah ini dapat dilihat pada lampiran 1 1.4.

1. *Copy* file ca.crt dari *Raspberry pi* ke PC
2. Kemudian cari Internet Options pada icon search di taskbar, Klik Tab Content → klik Certificates → Klik Import
3. Kemudian arahkan pada letak file ca.crt
4. Setelah itu klik Next → Install → Finish

Setelah kunci sudah terpasang pada semua klien barulah pengiriman dapat dilakukan. Untuk melakukan percobaan pengiriman data dapat menggunakan syntax sebagai berikut :

Percobaan dapat dilakukan melalui terminal (*command prompt)* menggunakan syntax sebagai berikut : (dapat dilihat pada lampiran 1 (1.6))

- Syntax Publisher

```
Mosquitto_pub -h 192.168.43.121 -t tes -d -p 8883
--cafile C:\Users\Asus\Desktop\ca.crt
```

- Syntax Subscriber

```
mosquitto_sub -h 192.168.43.121 -t tes -d -p 8883
--cafile C:\Users\Asus\Desktop\ca.crt
```

Pada tugas akhir ini *publisher* menggunakan program python dengan library MQTT yaitu Paho MQTT. Pengiriman data sinyal yang dilakukan secara *real-time*. Untuk pengujian penerimaan data pada penelitian menggunakan 2 subscriber, di mana keduanya sudah terpasang TLS dan memiliki file CA yang dimiliki oleh *publisher* untuk mengirim data. Dalam realitanya, 2 *subscriber* ini diibaratkan sebagai dokter dan perawat atau petugas kesehatan yang menerima data sinyal EKG pasien. Sedangkan dari sisi *publisher* untuk mengirim data sinyal EKG dapat menggunakan topik yang berbeda supaya dapat membedakan data kepemilikan setiap pasien.

Selanjutnya pengujian dalam penelitian ini juga menggunakan *wireshark* untuk membandingkan ukuran paket data sebelum dan sesudah enkripsi dan juga untuk mengukur waktu yang dibutuhkan dalam proses enkripsi sekaligus pengiriman data tersebut.

3.2 Parameter Perbandingan

Setiap pengiriman akan dilakukan analisis terhadap 3 parameter yaitu :

1. Perbandingan ukuran paket sebelum dan setelah enkripsi data

   Data yang telah terkirim akan dienkripsi, dari hasil enkripsi data tersebut akan dilakukan perbandingan ukuran paket data sebelum dienkripsi dan setelah terenkripsi dengan menggunakan standart kunci TLS untuk MQTT yaitu dengan panjang 2048 bit.
2. Selisih waktu yang dibutuhkan untuk proses enkripsi sekaligus pengiriman data

   Selisih waktu yang dimaksud yaitu melakukan pengujian terhadap lama waktu yang dibutuhkan untuk enkripsi sekaligus mengirim data tersebut. Dengan cara menghitung selisih waktu sebelum pengiriman dan setelah pengiriman.
3. Pengujian Integritas Data

   Pengujian ini dilakukan dengan metode *cross correlation* untuk mengetahui validitas data sinyal EKG yang dikirim dengan data sinyal yang diterima. Jika nilai yang dihasilkan mendekati 1 maka data sinyal EKG tersebut mendekati asli berarti integritas data sebelum dan setelah enkripsi tetap terjaga.

BAB IV
HASIL DAN PEMBAHASAN

Hasil penelitian ini merupakan hasil dari pengamatan dan pengujian dari transmisi data EKG dengan MQTTS yang telah dilakukan penulis.

4.1 Pengujian Transmisi data EKG

4.1.1 Tujuan

Untuk dapat melakukan percobaan apakah sensor dapat berjalan dengan baik, dan apakah program dapat membaca data yang masuk dari sensor.

4.1.2 Alat yang diperlukan

1. Raspberry pi 2
2. Donggle *Wifi*
3. Adaptor Rasp pi
4. Laptop sebagai monitor rasp pi
5. 2 buah PC yang bertindak sebagai subscriber
6. Modul AD8232
7. Modul ADS1115
8. Kabel Jumper
9. 3 elektroda penjapit
10. Kabel Tunggal

4.1.3 Prosedur Pengujian

1. Menghidupkan laptop
2. Hidupkan *Raspberry pi*
3. Nyalakan hotspot, dan sambungkan ke Laptop, *Raspberry pi*, dan subscriber lainnya.
4. Sambungkan Modul AD8232, Modul ADS1115 dan pin GPIO *Raspberry pi* sesuai dengan skematik yang sudah dijelaskan pada bab sebelumnya.
5. Kemudian letakkan elektroda penjapit ke lengan kiri, lengan kanan dan kaki kanan sesuai dengan sambungan pada modul AD8232.

6. Perhatikan indikator led pada modul ad8232 apabila sudah berkedip seperti detak jantung maka dapat dipastikan data sudah dapat dikirimkan.
7. Setelah sensor sudah siap maka buka program publisher.py
8. Kemudian buka *command prompt* pada PC dan tulis syntax (sesuaikan dengan topic yang akan digunakan) :

   ```
   mosquitto_sub -h 192.168.43.121 -t tes -d -p 8883 --cafile C:\Users\Asus\Desktop\ca.crt
   ```

9. Kemudian tekan enter maka akan ada tampilan seperti berikut ini yang menandakan bahwa sudah terhubung ke broker.

```
Microsoft Windows [Version 10.0.18362.657]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\MSRF>cd C:\Program Files\mosquitto

C:\Program Files\mosquitto>mosquitto_sub -h 192.168.43.69 -t tes -d -p 8883 --cafile C:\Users\MSRF\Desktop\ca.crt
Client mosq/yvPoaf1LHkp3scu2bS sending CONNECT
Client mosq/yvPoaf1LHkp3scu2bS received CONNACK (0)
Client mosq/yvPoaf1LHkp3scu2bS sending SUBSCRIBE (Mid: 1, Topic: tes, QoS: 0, Options: 0x00)
Client mosq/yvPoaf1LHkp3scu2bS received SUBACK
Subscribed (mid: 1): 0
```

Gambar 4.1 Subscriber telah terhubung dengan broker

10. Setelah subscriber terhubung, run program publisher.py dan ubah nama file untuk menyimpan data yang dikirim dengan nama pasien tersebut, seperti gambar dibawah ini :

Gambar 4.2 File data EKG pasien

11. Kemudian dari sisi subcriber akan menerima data tersebut seperti gambar 4.3 berikut ini :

```
Microsoft Windows [Version 10.0.18362.657]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\MSRF>cd C:\Program Files\mosquitto

C:\Program Files\mosquitto>mosquitto_sub -h 192.168.43.69 -t tes -d -p 8883 --cafile C:\Users\MSRF\Desktop\ca.crt
Client mosq/yvPoaf1LHkp3scu2bS sending CONNECT
Client mosq/yvPoaf1LHkp3scu2bS received CONNACK (0)
Client mosq/yvPoaf1LHkp3scu2bS sending SUBSCRIBE (Mid: 1, Topic: tes, QoS: 0, Options: 0x00)
Client mosq/yvPoaf1LHkp3scu2bS received SUBACK
Subscribed (mid: 1): 0
Client mosq/yvPoaf1LHkp3scu2bS sending PINGREQ
Client mosq/yvPoaf1LHkp3scu2bS received PINGRESP
Client mosq/yvPoaf1LHkp3scu2bS sending PINGREQ
Client mosq/yvPoaf1LHkp3scu2bS received PINGRESP
Client mosq/yvPoaf1LHkp3scu2bS sending PINGREQ
Client mosq/yvPoaf1LHkp3scu2bS received PINGRESP
Client mosq/yvPoaf1LHkp3scu2bS sending PINGREQ
Client mosq/yvPoaf1LHkp3scu2bS received PINGRESP
Client mosq/yvPoaf1LHkp3scu2bS received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
8623
Client mosq/yvPoaf1LHkp3scu2bS received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
8791
Client mosq/yvPoaf1LHkp3scu2bS received PUBLISH (d0, q0, r0, m0, 'tes', ... (5 bytes))
10259
Client mosq/yvPoaf1LHkp3scu2bS received PUBLISH (d0, q0, r0, m0, 'tes', ... (5 bytes))
10098
Client mosq/yvPoaf1LHkp3scu2bS received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
9161
Client mosq/yvPoaf1LHkp3scu2bS received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
9651
Client mosq/yvPoaf1LHkp3scu2bS received PUBLISH (d0, q0, r0, m0, 'tes', ... (5 bytes))
10603
Client mosq/yvPoaf1LHkp3scu2bS received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
9122
Client mosq/yvPoaf1LHkp3scu2bS received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
7896
```

Gambar 4.3 Data telah diterima *subscriber*

Seperti yang terlihat diatas data EKG yang diterima ditunjukkan pada gambar 4.4 :

```
Client mosq/yvPoaf1LHkp3scu2bS received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
8623
```

Gambar 4.4 Data EKG pada *subscriber*

Sedangkan untuk besar ukuran data ditunjukkan pada gambar 4.5 dibawah ini :

```
Client mosq/yvPoaf1LHkp3scu2bS received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
8623
```

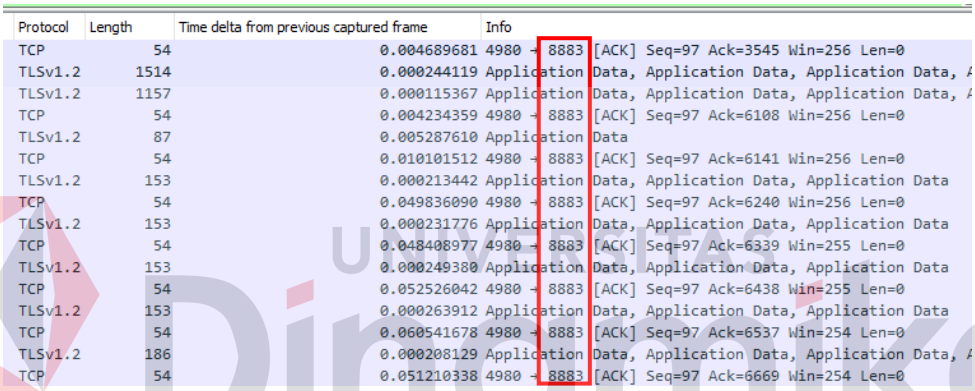
Gambar 4.5 Besar ukuran data pada *subscriber*

Data yang diterima oleh *subscriber* merupakan data yang telah didekripsi dengan menggunakan kunci yang telah terpasang. Sehingga data asli dapat terbaca, sekaligus menampilkan besar ukuran data asli tersebut.

12. Sebelum melakukan pengiriman, nyalakan wireshark pada sisi *publisher* untuk merekam trafik jaringan yang sedang berjalan. Hasil rekam *wireshark* nantinya akan digunakan untuk analisis perbandingan besar paket data dan juga lama waktu proses enkripsi.
13. Untuk dapat menyaring jaringan yang diinginkan dapat dengan cara seperti pada gambar 4.6 berikut ini :

Gambar 4.6 Filter port 8883

14. Setelah itu *traffic* jaringan pada *wireshark* akan tersaring seperti pada gambar 4.7 dibawah ini :

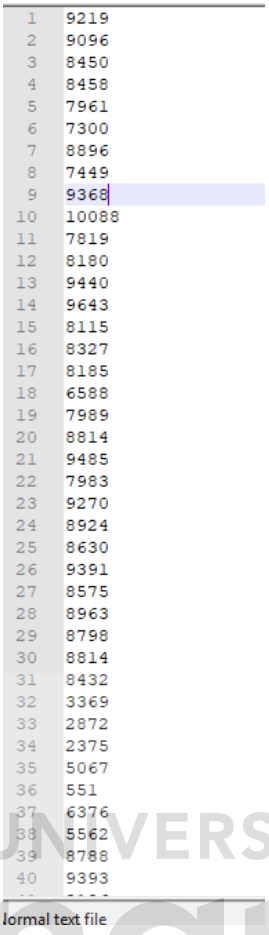


Protocol	Length	Time delta from previous captured frame	Info
TCP	54	0.004689681	4980 → 8883 [ACK] Seq=97 Ack=3545 Win=256 Len=0
TLSv1.2	1514	0.000244119	Application Data, Application Data, Application Data, A
TLSv1.2	1157	0.000115367	Application Data, Application Data, Application Data, A
TCP	54	0.004234359	4980 → 8883 [ACK] Seq=97 Ack=6108 Win=256 Len=0
TLSv1.2	87	0.005287610	Application Data
TCP	54	0.010101512	4980 → 8883 [ACK] Seq=97 Ack=6141 Win=256 Len=0
TLSv1.2	153	0.000213442	Application Data, Application Data, Application Data
TCP	54	0.049836090	4980 → 8883 [ACK] Seq=97 Ack=6240 Win=256 Len=0
TLSv1.2	153	0.000231776	Application Data, Application Data, Application Data
TCP	54	0.048408977	4980 → 8883 [ACK] Seq=97 Ack=6339 Win=255 Len=0
TLSv1.2	153	0.000249380	Application Data, Application Data, Application Data
TCP	54	0.052526042	4980 → 8883 [ACK] Seq=97 Ack=6438 Win=255 Len=0
TLSv1.2	153	0.000263912	Application Data, Application Data, Application Data
TCP	54	0.060541678	4980 → 8883 [ACK] Seq=97 Ack=6537 Win=254 Len=0
TLSv1.2	186	0.000208129	Application Data, Application Data, Application Data, A
TCP	54	0.051210338	4980 → 8883 [ACK] Seq=97 Ack=6669 Win=254 Len=0

Gambar 4.7 Hasil filter port 8883

Port 8883 diatas menunjukkan letak port MQTTS (MQTT + TLS) protokol yang tedeteksi oleh *wireshark* adalah TLS dengan versi 1.2.

15. Pada tugas akhir ini data yang diambil sebanyak 1000 data, sehingga setelah 1000 data tersebut telah terkirim maka stop wireshark, dan cek file data.txt atau file data pasien yang langsung tersimpan pada desktop raspberry pi seperti dibawah ini :



Gambar 4.8 *Publish* data

16. Kemudian simpan hasil rekam jaringan pada wireshark berupa file.csv yang nantinya akan digunakan untuk analisis data. Data yang digunakan untuk analisis ada pada kolom *length* untuk melihat besar paket dan *time delta from previous captured frame* untuk melihat lama waktu yang dibutuhkan untuk proses enkripsi, dihitung dari sebelum pengiriman sampai data masuk ke broker (setelah dienkripsi).

	A	B	C	D	E	F	G	
1	No.	Time	Source	Destination	Protocol	Length	Time delta from previous captured frame	Info
2	29	0.520523760	192.168.43.121	192.168.43.69	TLSv1.2	78	0.363020592	Application Data
3	30	0.521105478	192.168.43.69	192.168.43.121	TLSv1.2	78	0.000581718	Application Data
4	33	0.566702340	192.168.43.121	192.168.43.69	TCP	54	0.045334727	2012 > 8883 [ACK] Seq=25 Ack=25 Wir
5	846	24.840.756.513	192.168.43.69	192.168.43.121	TLSv1.2	89	0.293295944	Application Data
6	847	24.841.114.220	192.168.43.69	192.168.43.127	TLSv1.2	89	0.000357707	Application Data
7	848	24.920.611.174	192.168.43.69	192.168.43.121	TLSv1.2	649	0.079496954	Application Data, Application Data, Ap
8	849	24.920.987.892	192.168.43.69	192.168.43.127	TLSv1.2	649	0.000376718	Application Data, Application Data, Ap
9	850	24.987.477.687	192.168.43.69	192.168.43.121	TLSv1.2	124	0.066489795	Application Data, Application Data
10	851	25.007.460.194	192.168.43.69	192.168.43.127	TLSv1.2	124	0.019982507	Application Data, Application Data
11	852	25.217.491.692	192.168.43.69	192.168.43.121	TCP	754	0.210031498	[TCP Retransmission] 8883 > 2012 [PS
12	853	25.247.523.603	192.168.43.69	192.168.43.127	TCP	754	0.030031911	[TCP Retransmission] 8883 > 10248 [P
13	855	25.667.525.709	192.168.43.69	192.168.43.121	TCP	754	0.160063587	[TCP Retransmission] 8883 > 2012 [PS
14	856	25.717.515.124	192.168.43.69	192.168.43.127	TCP	754	0.049989415	[TCP Retransmission] 8883 > 10248 [P
15	863	26.356.476.245	192.168.43.121	192.168.43.69	TCP	66	0.000019792	2012 > 8883 [ACK] Seq=25 Ack=725 W
16	871	26.362.221.909	192.168.43.121	192.168.43.69	TLSv1.2	80	0.003790512	Application Data
17	872	26.362.744.824	192.168.43.69	192.168.43.121	TLSv1.2	80	0.000522915	Application Data
18	873	26.366.053.202	192.168.43.127	192.168.43.69	TLSv1.2	80	0.003308378	Application Data
19	874	26.366.444.451	192.168.43.121	192.168.43.69	TCP	66	0.000391249	[TCP Dup ACK 863#1] 2012 > 8883 [AC
20	875	26.366.639.815	192.168.43.127	192.168.43.69	TCP	66	0.000195364	10248 > 8883 [ACK] Seq=27 Ack=701 V
21	876	26.366.844.398	192.168.43.69	192.168.43.127	TLSv1.2	80	0.000204583	Application Data
22	877	26.366.663.721	192.168.43.127	192.168.43.69	TCP	66	-0.000180677	[TCP Dup ACK 875#1] 10248 > 8883 [A
23	888	26.368.059.864	192.168.43.121	192.168.43.69	TLSv1.2	548	-0.000297968	Application Data, Application Data, Ap
24	889	26.372.879.488	192.168.43.127	192.168.43.69	TLSv1.2	548	0.004819624	Application Data, Application Data, Ap
25	901	26.378.021.976	192.168.43.69	192.168.43.121	TLSv1.2	80	0.003029212	Application Data
26	902	26.378.382.913	192.168.43.69	192.168.43.127	TLSv1.2	80	0.000360937	Application Data

Gambar 4.9 Tampilan pada excel

17. Setelah itu, simpan data yang diterima oleh setiap subscriber dengan format .txt

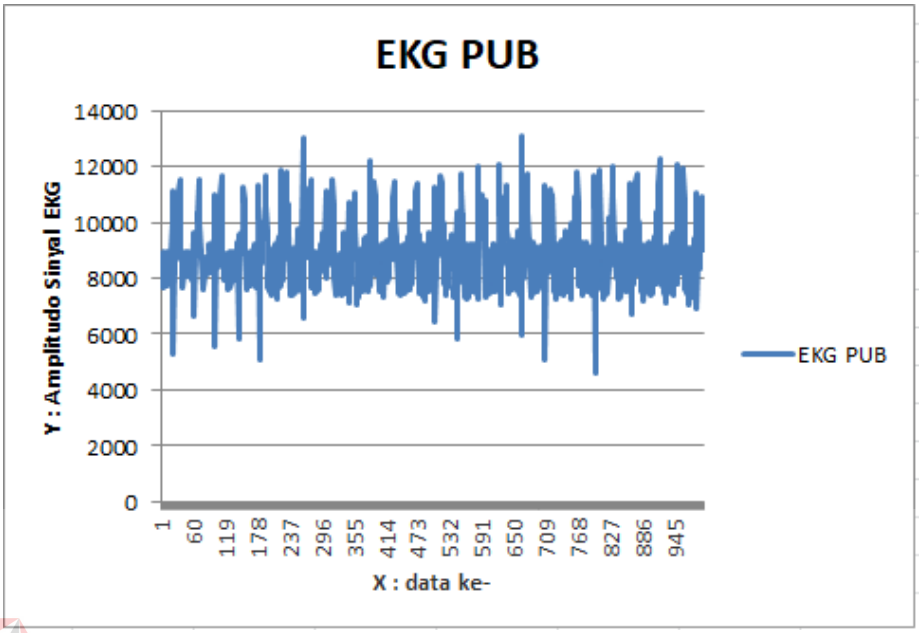
andre121.txt - Notepad

File Edit Format View Help

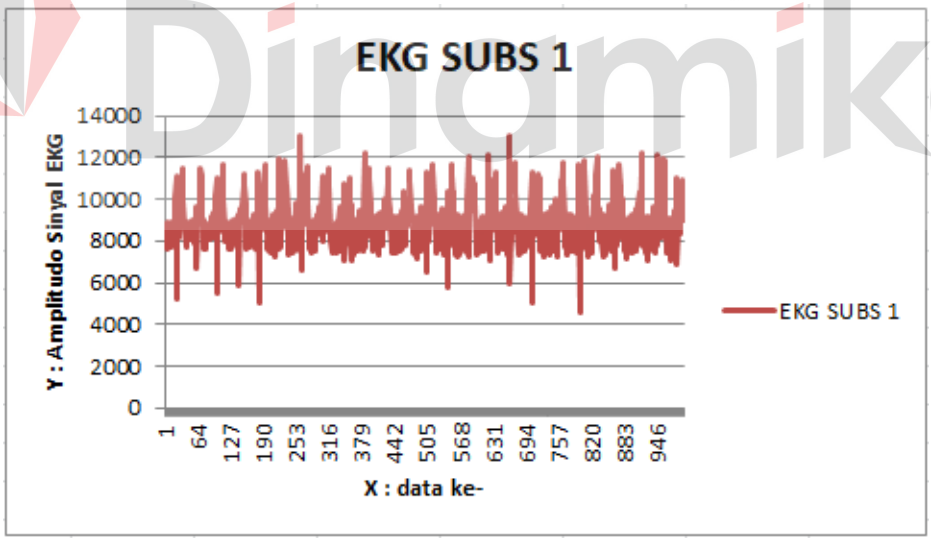
Client mosq/ZYb5v1nVltjxw6RRrc received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
7738
Client mosq/ZYb5v1nVltjxw6RRrc received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
7768
Client mosq/ZYb5v1nVltjxw6RRrc received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
8489
Client mosq/ZYb5v1nVltjxw6RRrc received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
8917
Client mosq/ZYb5v1nVltjxw6RRrc received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
7819
Client mosq/ZYb5v1nVltjxw6RRrc received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
7660
Client mosq/ZYb5v1nVltjxw6RRrc received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
8798
Client mosq/ZYb5v1nVltjxw6RRrc received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
8672
Client mosq/ZYb5v1nVltjxw6RRrc received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
7994
Client mosq/ZYb5v1nVltjxw6RRrc received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
7824
Client mosq/ZYb5v1nVltjxw6RRrc received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
8159
Client mosq/ZYb5v1nVltjxw6RRrc received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
8944
Client mosq/ZYb5v1nVltjxw6RRrc received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
8720
Client mosq/ZYb5v1nVltjxw6RRrc received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
7720
Client mosq/ZYb5v1nVltjxw6RRrc received PUBLISH (d0, q0, r0, m0, 'tes', ... (4 bytes))
8700

Gambar 4.10 Data pada *subscriber*

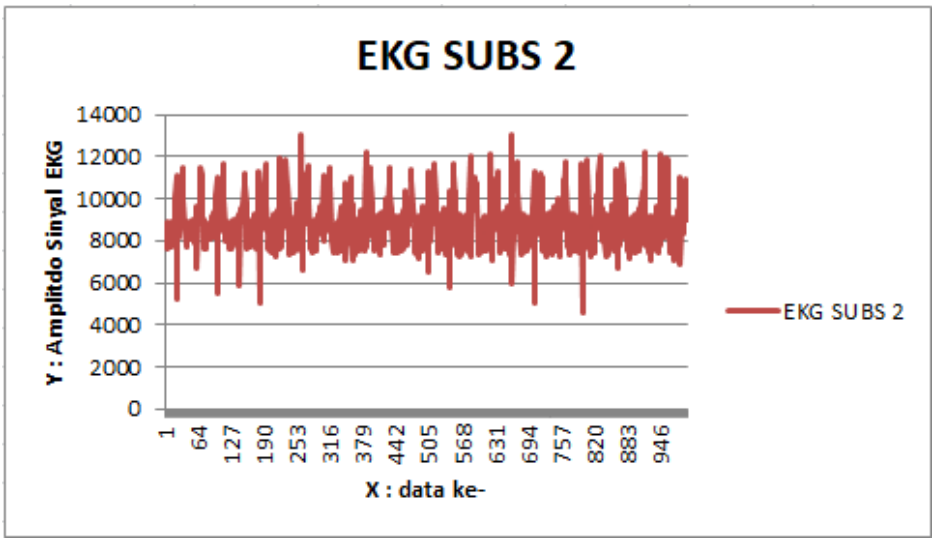
18. Berikut ini adalah tampilan data sinyal EKG sampel ke- 1 qos 0 sebelum dan setelah pengiriman, dengan sumbu y = data sinyal EKG, sumbu x = urutan data sinyal yang masuk



Gambar 4.11 Tampilan sinyal EKG sampel ke-1 qos 0 yang dikirim



Gambar 4.12 Tampilan Sinyal EKG sampel ke-1 qos 0 pada *subscriber 1*



Gambar 4.13 Tampilan Sinyal EKG sampel ke-1 qos 0 pada *subscriber 2*

19. Kemudian setelah seluruh data yang dibutuhkan sudah tersimpan maka dapat dilakukan analisis perhitungan data pada excel seperti gambar berikut ini :

	ANDRE			SELISIH	43,598	BYTE	
			RATA-RATA	4,142	47,74	BYTE	0,0003236
			192, 168,43,121				
NO	PROTOCOL	DATA EKG PUB	DATA EKG SUB	LENGTH(BYTE) SEBELUM ENKRIPSI	LENGTH(BYTE) SETELAH ENKRIPSI	BYTE	DELTA WAKTU SEBELUM DAN SESUDAH PENGIRIM
1	TLSv1,2	7738	7738	4	153	BYTE	0,000216772
2	TLSv1,2	7768	7768	4	186	BYTE	0,000322293
3	TLSv1,2	8489	8489	4	153	BYTE	0,00020521
4	TLSv1,2	8917	8917	4	186	BYTE	0,000218334
5	TLSv1,2	7819	7819	4	153	BYTE	0,00020948
6	TLSv1,2	7660	7660	4	153	BYTE	0,000315002
7	TLSv1,2	8798	8798	4	188	BYTE	0,000217657
8	TLSv1,2	8672	8672	4	153	BYTE	0,000200001
9	TLSv1,2	7994	7994	4	153	BYTE	0,000223178
10	TLSv1,2	7824	7824	4	189	BYTE	0,000232553
11	TLSv1,2	8159	8159	4	154	BYTE	0,000285054
12	TLSv1,2	8944	8944	4	153	BYTE	0,000227813
13	TLSv1,2	8720	8720	4	153	BYTE	0,000346617

Gambar 4.14 Perhitungan analisis data

Pada gambar di atas dilakukan perhitungan rata – rata untuk besar paket data sebelum dan setelah dikirim, dan juga rata – rata untuk selisih waktu yang dibutuhkan untuk proses enkripsi.

4.2 Analisis Data

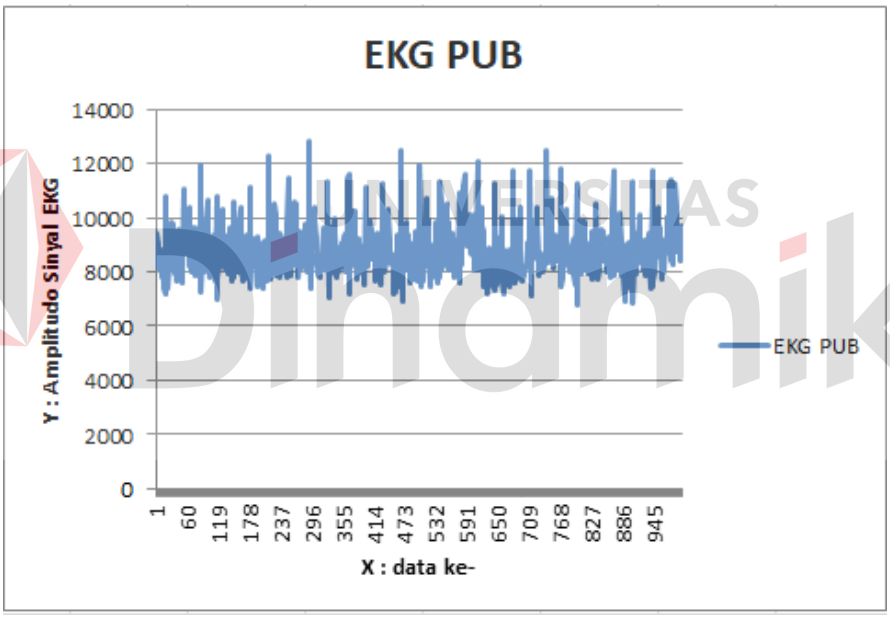
Pengujian data yang dilakukan berdasarkan 3 parameter perbandingan berikut yaitu :

1. Selisih besar paket sebelum dan sesudah pengiriman

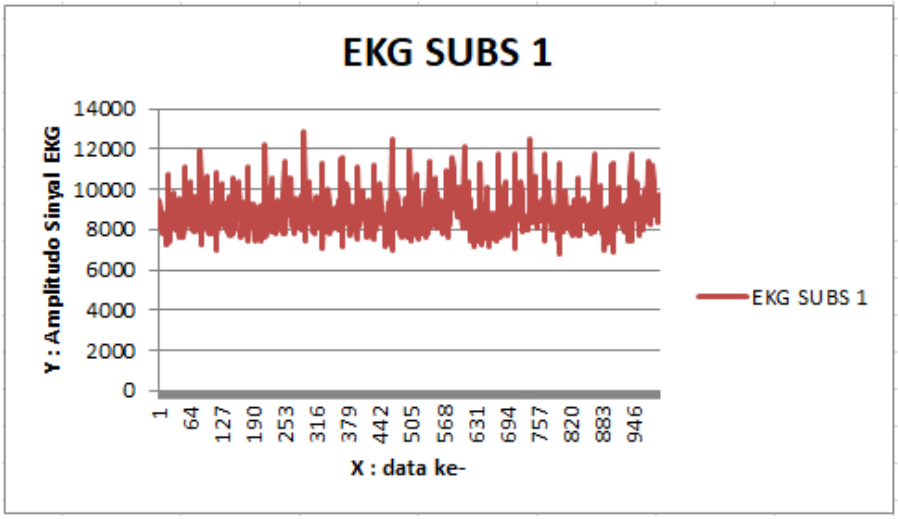
2. Waktu yang dibutuhkan untuk proses enkripsi dihitung dari selisih waktu ketika data akan dikirm dan sampai ke broker (setelah terenkripsi)
3. Pengujian integritas data dengan menggunakan metode *cross correlation* untuk mengetahui data yang diterima telah sama dengan data yang dikirim

Pengiriman data dilakukan dengan menggunakan qos 0 dan qos 1, dimana qos *(Quality of Service)* merupakan kualitas pelayanan yang pada proses transmisi. Qos 0 merupakan kualitas pelayanan dimana tidak ada jaminan pesan tersampaikan kepada sisi penerima. Sedangkan Qos 1 merupakan kualitas pelayanan dimana pesan akan dijamin sampai ke *subscriber* minimal 1x.

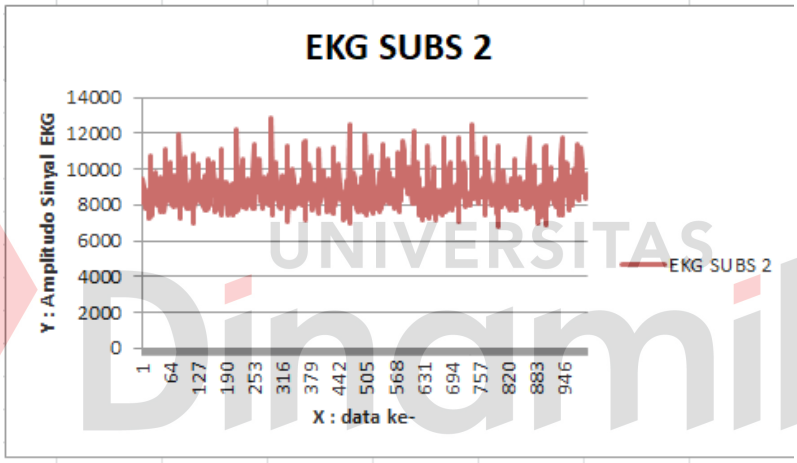
4.2.1 Hasil analisis transmisi data pada Qos 0



Gambar 4.15 Publikasi Sinyal EKG sampel ke- 3 Qos 0



Gambar 4.16 *Subscriber 1* sampel ke- 3 Qos 0



Gambar 4.17 *Subscriber 2* sampel ke- 3 Qos 0

Ketiga gambar diatas merupakan gambar sinyal EKG sampel ke- 3 pada pengiriman menggunakan Qos 0, satu dari 10 sampel. Penjelasan grafik diatas sumbu y merupakan data sinyal EKG, dan sumbu x merupakan urutan data sinyal yang masuk.

1. Selisih besar paket dari kedua *subscriber*

   a. *Subscriber* 1

   Tabel 4.1 Selisih besar paket *subscriber* 1 (qos 0)

No Sampel	Besar Paket Sebelum Enkripsi			Besar Paket Setelah Enkripsi		
	Min (byte)	Max (byte)	Rata- rata setiap sampel (byte)	Min (byte)	Max (byte)	Rata- rata setiap sampel (byte)
1	4	5	4,142	87	623	158,1918
2	4	5	4,135	87	584	149,9449
3	4	5	4,088	78	616	154,8506
4	4	5	4,12	87	619	160,8387

Tabel 4.2 Selisih besar paket *subscriber* 1 (qos 0)

5	4	5	4,161	78	619	161,6364
6	4	5	4,119	87	615	155,9046
7	4	5	4,004	87	615	160,7799
8	4	5	4,298	87	622	163,1738
9	4	5	4,009	87	582	152,8293
10	4	5	4,164	87	616	149,5476
Min/Max	4	5		78	623	
Rata-rata			4,124			156,7698
Standar Deviasi			0,0834			4,9222
Selisih rata-rata				152,6458		

Pengiriman menggunakan qos 0 pada *subscriber 1* besar paket minimum sebelum enkripsi adalah 4 byte dan maksimum 5 byte. Sedangkan besar paket minimum setelah enkripsi adalah 78 byte dan maksimum 623 byte. Rata-rata besar paket sebelum enkripsi dari 10 sampel adalah 4,124 byte. Rata-rata setelah enkripsi sebesar 156,7698 byte. Selisih rata – rata besar paket adalah 152,6458 byte dengan standar deviasi sebelum enkripsi sebesar 0,0834 byte setelah enkripsi sebesar 4,9222.

b. *Subscriber* 2

Tabel 4.3 Selisih besar paket *subscriber* 2 (qos 0)

No Sampel	Besar Paket Sebelum Enkripsi			Besar Paket Setelah Enkripsi		
	Min (byte)	Max (byte)	Rata- rata setiap sampel (byte)	Min (byte)	Max (byte)	Rata- rata setiap sampel (byte)
1	4	5	4,142	87	483	146,4832
2	4	5	4,135	87	620	140,7121
3	4	5	4,088	87	285	141,2797
4	4	5	4,12	87	652	147,5339
5	4	5	4,161	87	615	147,4113
6	4	5	4,119	87	484	142,7909
7	4	5	4,004	87	615	145,1713
8	4	5	4,298	87	622	144,4837
9	4	5	4,009	87	516	139,7377
10	4	5	4,164	87	621	140,1403
Min/Max	4	5		87	652	
Rata-rata			4,124			143,5744
Standar Deviasi			0,0834			3,0336
Selisih rata-rata				139,4504		

Pada *subscriber 2* pengiriman menggunakan qos 0 sebelum enkripsi minimum besar paket ialah 4 byte dan maksimum 5 byte. Setelah enkripsi minimum besar paket 87 byte dan maksimum 652 byte. Standar deviasi sebelum enkripsi sebesar 0,0834 dan setelah enkripsi 3,0336. Rata – rata sebelum enkripsi 4,124 byte dan rata – rata setelah enkripsi 143,5744 byte. Selisih rata-rata sebelum dan setelah enkripsi sebesar 139,4504 byte.

2. Waktu yang dibutuhkan untuk proses enkripsi

Tabel 4.4 Rata-rata proses waktu enkripsi (qos 0)

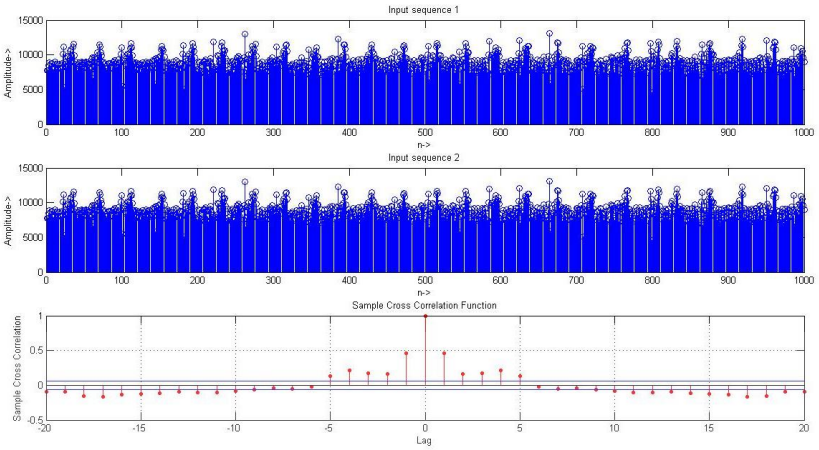
	No Sampel	Rata-rata setiap sampel (s)	Waktu Minimum (s)	Waktu Maksimum (s)
Subscriber 1	1	0,001029	0,000111	0,213207
	2	0,000668	0,000096	0,120007
	3	0,000368	0,000132	0,015458
	4	0,000888	0,000111	0,135876
	5	0,001770	0,000105	0,215640
	6	0,001284	0,000099	0,268409
	7	0,001217	0,000105	0,232037
	8	0,000663	0,000106	0,040023
	9	0,000304	0,000125	0,007440
	10	0,000339	0,000107	0,022267
Subscriber 2	1	0,000315	0,000108	0,026047
	2	0,000647	0,000131	0,090009
	3	0,000557	0,000108	0,114175
	4	0,000897	0,000088	0,223816
	5	0,000758	0,000101	0,099999
	6	0,000266	0,000125	0,001883
	7	0,000469	0,000109	0,039982
	8	0,000925	0,000110	0,231028
	9	0,000730	0,000126	0,182455
	10	0,000296	0,000106	0,011076
Min/Max			0,000089	0,268409
Rata-rata keseluruhan		0,0007195 (s)		
Standar Deviasi		0,0003946		

Hasil perhitungan pada tabel di atas dapat dilihat bahwa pada qos 0 pada kedua *subscriber*, waktu minimum keseluruhan sampel sebesar 0,000089 s dan waktu maksimum 0,268409 atau 0,26841 s. Rata-rata waktu yang dibutuhkan untuk proses enkripsi data dari 10 sampel yang dikirim adalah 0,0007195 s, dengan standar deviasi sebesar 0,0003946.

Jika dihitung waktu pada 1 sampel dengan 1000 data, maka rata-rata waktu untuk deskripsi yang dibutuhkan adalah 0,019558 s atau 0,02 ms dengan rata-rata waktu enkripsi 0,001029 s atau 0,001 ms.

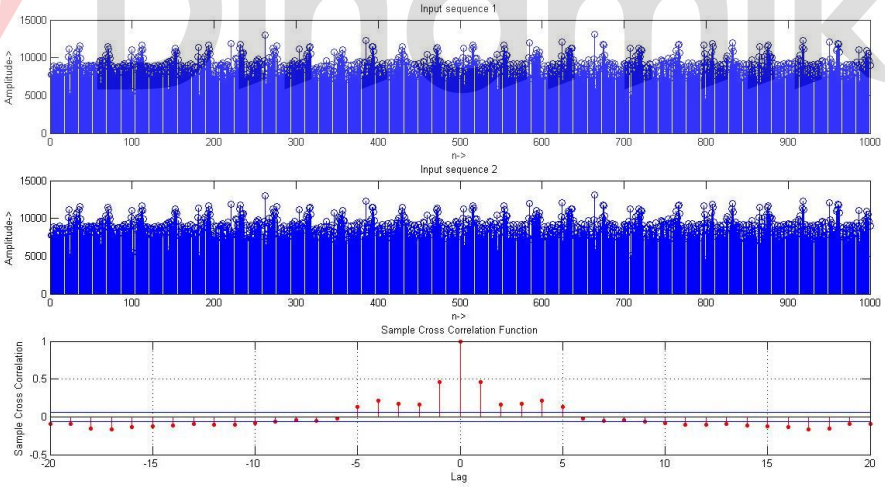
3. Hasil *cross correlation*

a. *Subscriber* 1



Gambar 4.18 Hasil *cross-correlation* qos 0 pada *subscriber 1*

b. *Subscriber* 2

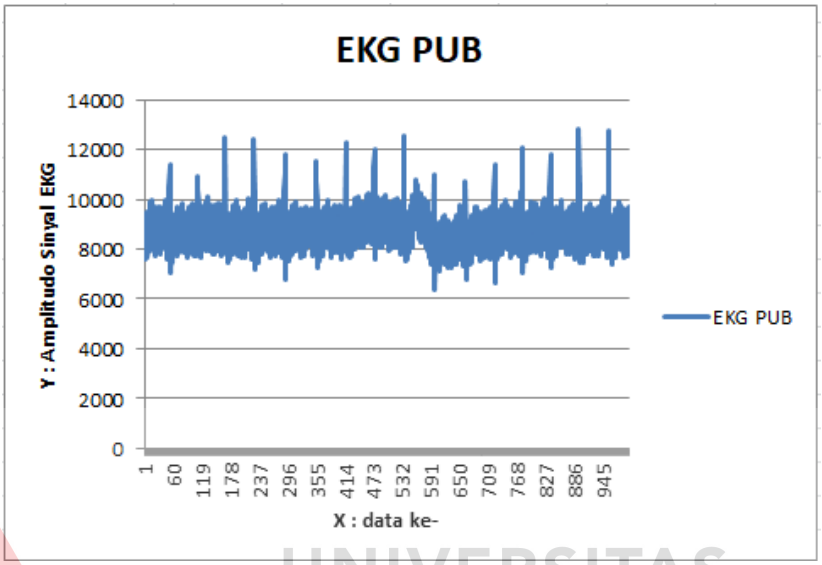


Gambar 4.19 Hasil *cross-correlation* qos 0 pada *subscriber 2*

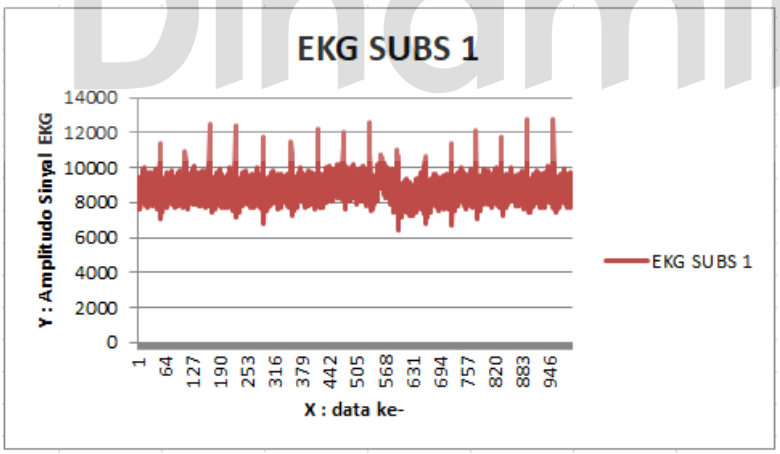
Hasil *cross-correlation* yang ditampilkan adalah hasil sampel ke-1 dari sampel 10 pada qos 0. Pengujian *cross-correlation* dilakukan menggunakan 2 input, dimana input pertama merupakan data yang dikirim, dan input kedua merupakan data yang diterima. Dilihat dari

gambar 4.17 dan 4.18 hasil dari keduanya pada lag ke-0 menunjukkan nilai 1, yang berarti data yang dikirim dan diterima adalah sama.

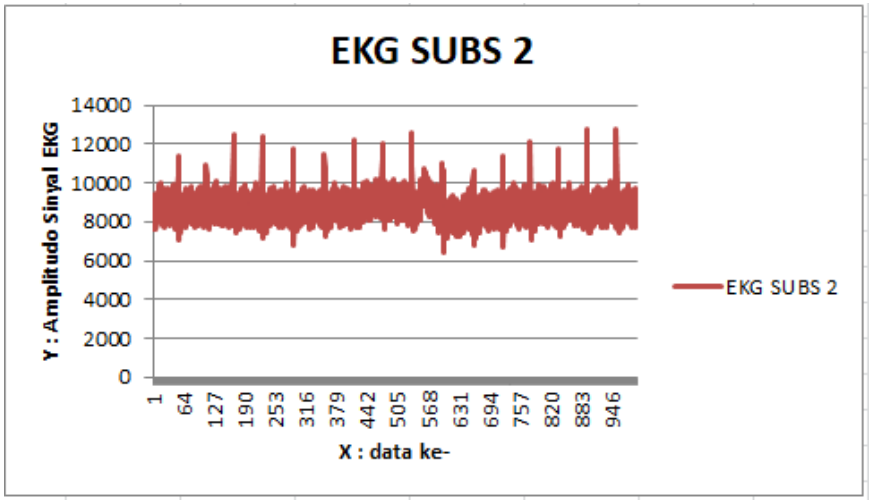
4.2.2 Hasil analisis transmisi data pada Qos 1



Gambar 4.20 Publikasi Sinyal EKG sampel ke-10 Qos 1



Gambar 4.21 *Subscriber* 1 sampel ke-10 Qos 1



Gambar 4.22 *Subscriber* 2 sampel ke-10 Qos 1

Gambar diatas merupakan gambar sampel ke-10 dari 10 sampel yang dikirim menggunakan qos 1. Keterangan sumbu y pada gambar diatas merupakan data sinyal EKG, sedangkan sumbu x merupakan urutan data sinyal yang masuk.

1. Selisih besar paket sebelum dan sesudah pengiriman

   - *Subscriber 1*

Tabel 4.5 Selisih besar paket *subscriber* 1 (qos 1)

No Sampel	Besar Paket Sebelum Enkripsi			Besar Paket Setelah Enkripsi		
	Min (byte)	Max (byte)	Rata- rata setiap sampel (byte)	Min (byte)	Max (byte)	Rata- rata setiap sampel (byte)
1	1	5	4,271	87	587	96,1483
2	4	5	4,403	89	656	96,2053
3	4	5	4,406	89	550	96,8126
4	4	5	4,401	89	551	95,1161
5	4	5	4,401	89	655	97,5437
6	4	5	4,417	89	726	104,9597
7	4	5	4,403	89	549	99,6813
8	4	5	4,421	89	727	101,8016
9	3	5	4,367	88	725	126,4734
10	4	5	4,061	89	720	126,7407
Min/Max	1	5		87	727	
Rata-rata			4,3551			104,1483
Standar Deviasi			0,1122			12,2068
Selisih rata-rata				99,7932		

Pengiriman menggunakan qos 1 pada *subscriber 1* minimum besar paket sebelum enkripsi adalah 1 byte dan maksimum adalah 5 byte. Besar paket minimum setelah enkripsi adalah 87 byte, dan maksimum 727 byte. Rata besar paket sebelum enkripsi sebesar 4,3551 byte. Rata – rata besar paket setelah enkripsi sebesar 104,1483 byte. Standar deviasi yang dihasilkan sebelum enkripsi adalah 0,1122 sedangkan standar deviasi setelah enkripsi adalah 12,2068. Selisih rata – rata keduanya adalah 99,7932 byte.

- *Subscriber 2*

Tabel 4.6 Selisih besar paket *subscriber* 1 (qos 1)

No Sampel	Besar Paket Sebelum Enkripsi			Besar Paket Setelah Enkripsi		
	Min (byte)	Max (byte)	Rata- rata setiap sampel (byte)	Min (byte)	Max (byte)	Rata- rata setiap sampel (byte)
1	1	5	4,271	86	546	96,2407
2	4	5	4,403	89	727	98,1434
3	4	5	4,406	89	656	103,9379
4	4	5	4,401	89	726	118,1540
5	4	5	4,401	89	726	129,3213
6	4	5	4,417	89	728	141,8834
7	4	5	4,403	78	727	140,5599
8	4	5	4,421	89	657	131,8484
9	3	5	4,367	78	727	116,5707
10	4	5	4,061	89	650	122,2121
Min/Max	1	5		78	728	
Rata-rata			4,3551			119,8872
Standar Deviasi			0,1122			16,4958
Selisih rata-rata			115,5321			

Pada *subscriber 2* pengiriman menggunakan qos 1 menghasilkan besar paket minimum sebelm enkrispi adalah 1 byte dan maksimum 5 byte. Besar paket setelah enkripsi adalah 78 byte sedangkan maksimum sebesar 728 byte. Hasil rata-rata besar paket sebelum enkripsi sebesar 4,3551 byte dan hasil rata- rata setelah enkripsi sebesar 119,8872 byte. Standar deviasi yang diperoleh sebelum enkripsi adalah 0,1122 dan setelah enkripsi adalah 16,4958. Untuk selisih rata-rata yang dihasilkan adalah 115,5321 byte.

2. Waktu yang dibutuhkan untuk proses enkripsi

Tabel 4.7 Rata – rata waktu enkripsi (qos 1)

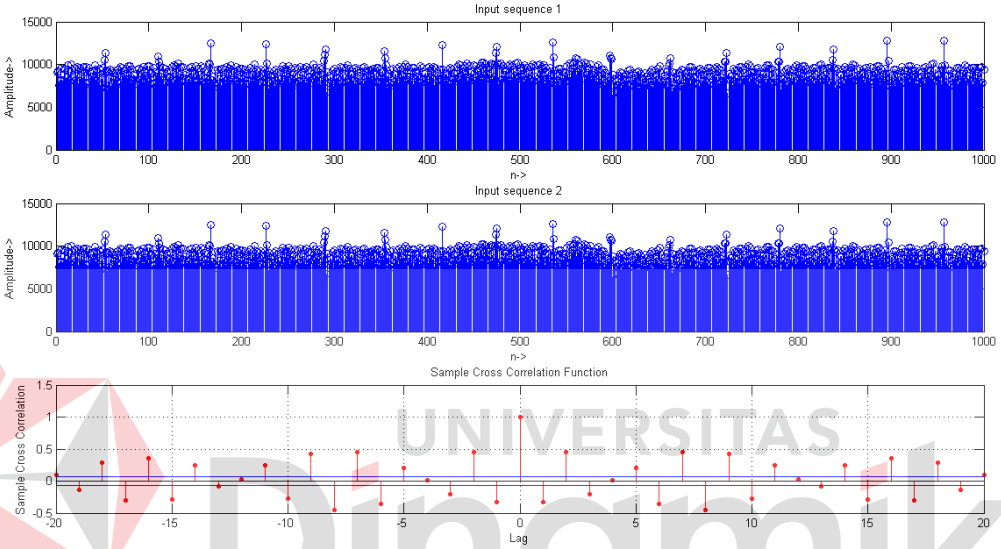
	No Sampel	Rata-rata setiap sampel (s)	Waktu Minimum (s)	Waktu Maksimum(s)
Subscriber 1	1	0,004594	0,000026	0,212009
	2	0,001119	0,000042	0,130032
	3	0,009943	0,000073	0,419589
	4	0,002654	0,000100	0,220023
	5	0,001929	0,000038	0,149941
	6	0,002021	0,000053	0,078267
	7	0,002210	0,000112	0,738426
	8	0,002098	0,000028	0,154626
	9	0,004467	0,000072	0,219614
	10	0,017735	0,000025	0,907781
Subscriber 2	1	0,001071	0,000035	0,218703
	2	0,008175	0,000021	0,346234
	3	0,000968	0,000063	0,101576
	4	0,012749	0,000050	0,508535
	5	0,018501	0,000067	0,543705
	6	0,027637	0,000107	0,682976
	7	0,017472	0,000033	0,189993
	8	0,036240	0,000088	0,911691
	9	0,013530	0,000079	0,966936
	10	0,006562	0,000042	0,229366
Min/Max			0,000021	0,966936
Rata-rata keseluruhan		0,009584 (s)		
Standar Deviasi		0,009801		

Pada qos 1 waktu minimum keseluruhan sampel sebesar 0,000021 s dan waktu maksimum yang diperlukan untuk enkripsi adalah 0,966936 s. Sedangkan waktu rata-rata yang diperlukan untuk proses enkripsi di kedua *subscriber* menghasilkan 0,009584 s dengan standar deviasi = 0,009801. Untuk waktu deskripsi dihitung dari 1 sampel dengan 1000 data, rata-rata waktu yang dibutuhkan adalah 0,016153 s atau 0,02 ms dengan rata-rata waktu enkripsi 0,007 ms.

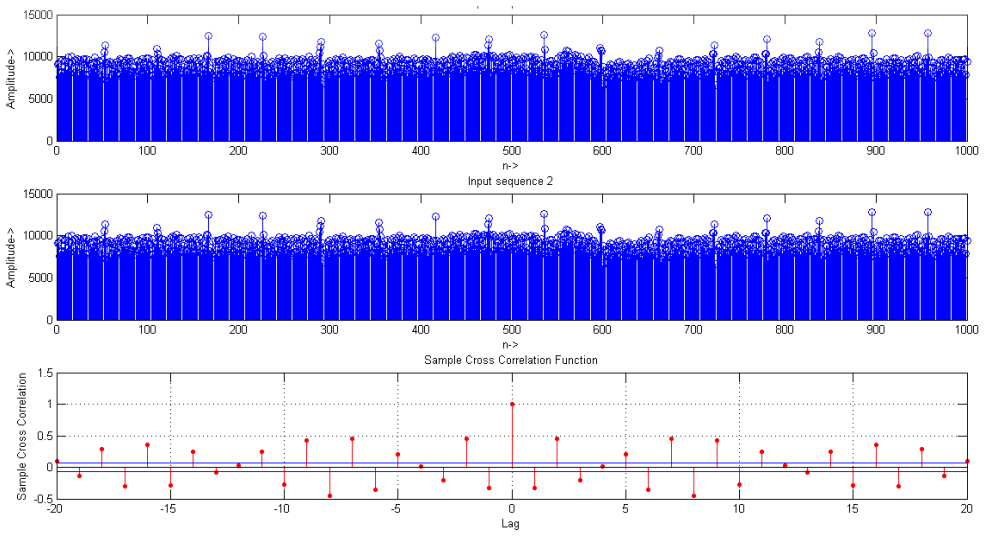
Jika dibandingkan waktu pengiriman yang dibutuhkan qos 1 sedikit lebih lama dari qos 0 dikarenakan protokol pada qos 1 berbeda dengan qos 0. Pada qos 0 sinyal kontrol hanya terdapat PUBLISH. Sedangkan pada qos 1 terdapat PUBLISH dan PUBACK. PUBLISH merupakan sinyal kontrol yang mewakili

publikasi baru atau terpisah. Sedangkan PUBACK merupakan respons qos 1 terhadap pesan publikasi. Penambahan sinyal kontrol pada qos 1 ini menyebabkan waktu yang dibutuhkan lebih lama dari pada qos 0. Tetapi, disisi lain qos 1 lebih baik dari qos 0 karena terdapat PUBACK ini. Adanya PUBACK pada qos 1 ini bertujuan untuk memastikan bahwa pesan benar – benar tersampaikan pada penerima dan pesan 1x diterima (tidak berulang).

3. Hasil *cross correlation*



Gambar 4.23 Hasil *cross-correlation* sampel ke-10 qos 1 pada *subscriber 1*



Gambar 4.24 Hasil *cross-correlation* sampel ke-10 qos 1 pada *subscriber 2*

Hasil uji *cross-correlation* dari kedua *subscriber* dilakukan dengan membandingkan data yang dikirim pada input 1 dan data yang diterima pada input

2. Pada gambar 4.22 dan 4.23 dapat dilihat hasil *cross-correlation* menunjukkan nilai 1 pada lag-0 yang berarti sama.

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan dari tugas akhir ini dapat disimpulkan beberapa hal sebagai berikut :

1. Fitur Keamanan pada protokol MQTT dapat diaktifkan salah satunya dengan menggunakan protokol TLS.
2. Penerapan pengiriman pada protokol MQTTS pada broker *mosquitto* menggunakan metode enkripsi simetris, dimana seluruh klien akan menggunakan file kunci yang sama dengan broker.
3. Protokol MQTTS dapat diterapkan untuk pengiriman data dalam bidang kesehatan *(telehealthcare)*, sebagaimana yang dilakukan pada tugas akhir ini yaitu menggunakan data sinyal Elektrokardiogram (EKG).
4. Pengiriman data pada qos 0 selisih rata – rata besar paket sebelum dan setelah pada penerima 1 dan 2 adalah 152,6458 byte dan 139,4504 byte. Sedangkan pada qos 1 penerima 1 dan 2 sebesar 99,7932 byte dan 115,5321 byte. Dari keempat hasil selisih rata-rata tersebut dapat disimpulkan bahwa relatif cukup besar dibanding dengan pengiriman tanpa menggunakan enkripsi. Tetapi, disisi lain pengiriman data menjadi lebih terjaga dan lebih aman karena adanya proses enkripsi dan juga selama proses pengambilan data, waktu pengiriman tidak membutuhkan waktu yang lama karena menerapkan metode enkripsi simetris.
5. Pada segi waktu yang dibutuhkan proses enkripsi, qos 0 menghasilkan rata – rata sebesar 0,72 ms dengan rata – rata waktu deskripsi 0,02 dihitung menggunakan 1 sampel. Sedangkan qos 1 menghasilkan rata – rata sebesar 9,58 ms dan rata – rata waktu deskripsi yaitu 0,02 ms dihitung pada 1 sampel. Sedikit lebih lama dibanding dengan qos 0 dikarenakan terdapat tambahan sinyal kontrol pada qos 1 yaitu PUBACK.
6. Pada hasil uji *cross-correlation* kedua *subscriber* pada qos 0 dan qos 1 menunjukkan hasil 1 pada lag ke-0. Sehingga dapat diartikan antara data

yang dikirim dan diterima adalah sama meskipun telah melalui proses enkripsi

5.2 Saran

Adapun saran untuk pengembangan penelitian selanjutnya adalah penerapan MQTTS pada *broker online* sehingga pengiriman yang dilakukan dapat menjangkau jarak yang jauh. Selain itu, juga dapat untuk melakukan pengujian atau tes keamanan pada protokol MQTTS sendiri ketika terdapat serangan dari luar.

DAFTAR PUSTAKA

Adnan, F. (2018, April 17). *Wireshark*. Dipetik Agustus 21, 2019, dari Apa itu WireShark, kegunaan, cara kerja, cara menggunakan wireshark: http://fauziadnan.blogspot.com/

Bintami, M. R. (2019). *Rancang Bangun Transmisi Data Heart Rate menggunakan Protokol MQTT*. Surabaya.

Bintara, H. (2017, May 19). *Mengenal SSL dan TLS Sebagai Transport Layer*. Dipetik October 22, 2019, dari NetSec.id: https://netsec.id/mengenal-ssl-dan-tls-sebagai-transport-layer/

Fadlan, M. (Desember 2017). Rekayasa Aplikasi Kriptografi Dengan Penerapan Kombinasi Algoritma Knapsack Merkle Hellman dan Affine Cipher. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) Vol. 4, No. 4*, 268-274.

Febiyanto, A. (2019). *Pengukuran dan Pengamatan Sinyal Electrocardiogram menggunakan Raspberry dengan Tampilan Aplikasi Mobile*. Surabaya.

Naharuddin, A. (2018, February 16). *Pengertian Enkripsi Simetris dan Asimetris*. Dipetik October 21, 2019, dari Web Pintar: http://webpintar.blogspot.com/2018/02/pengetian-simetris-dan-asimetris.html

Setiawan, B. A. (2018). *Anonimasi Sinyal EKG (Elektrokardiogram) untuk Keamanan Transmisi Data pada Sebuah Node Sensor*. Surabaya.

Setiawan, B. A. (2019). Implementasi Pengamanan Transmisi Sinyal EKG (Elektrokardiogram) secaraDaring dengan Metode Anonimasi. *Elkomika*, 85-96.

Solehudin, S. (2018). *Rekonstruksi Sinyal EKG (Elektrokardiogram) Hasil Proses Anonimasi dengan Tampilan pada Aplikasi Android.* Surabaya.

Syandrez. (2011, November 16). *Elektrokardiogram (EKG)*. Dipetik Juli 31, 2019, dari Syandrez Personal Blog: https://sandurezu.wordpress.com/2011/11/16/elektrokardiogram-ekg/

team, T. H. (2015, June 16). *Hive MQ*. Dipetik October 22, 2019, dari Securing MQTT Systems - MQTT Security Fundamentals: https://www.hivemq.com/blog/mqtt-security-fundamentals-securing-mqtt-systems/

Fadhli, M. (2015). Ancaman Keamanan pada Transport Layer Security. *ULTIMA Computing*, 70.

Lee, H., Lim, J., & Kwon, T. (2019). MQTLS: Toward Secure MQTT Communication with an Untrusted Broker. *International Conference on Information and Communication Technology Convergence (ICTC)*, 53.

Mishra, A., Kumari, A., Sajit, P., & Pandey, P. (2018). Remote Web Based ECG Monitoring Using MQTT Protocol. *International Journal of Advance Engineering and Research*, *5*(4), 1100.

N. Nikolov and O. Nakov, "Research of Secure Communication of Esp32 IoT Embedded System to.NET Core Cloud Structure using MQTTS SSL/TLS," 2019 *IEEE XXVIII International Scientific Conference Electronics (ET)*, Sozopol, Bulgaria, 2019, pp. 1-4, doi: 10.1109/ET.2019.8878636.