BAB 1 METODOLOGI

Bab ini akan menjelaskan tentang langkah-langkah dalam penyusunan skripsi seperti yang ditujukkan gambar 3.1 dimana alur penyusunan tersebut dimulai dari identifikasi masalah, studi literatur, persiapan testbed, pengujian, pengolahan data, dan pembahasan hingga pengambilan kesimpulan.

Gambar 1.1 Diagram Alir Penelitian

1.1 Identifikasi Masalah

Identifikasi masalah digunakan untuk menentukan penelitian yang akan dilakukan. Mengidentifikasi masalah celah keamanan pengiriman data dari sensor nodeMcu ke middleware dalam penelitian.

1.2 Studi Literatur

Studi literatur digunakan untuk mempelajari tentang dasar-dasar teori yang digunakan untuk mendukung pengerjaan penelitian ini, didapat dari berbagai disumber diantaranya: jurnal, e-book, thesis, artikel, dan website.

1. Internet of Things (IoT) adalah sekumpulan perangkat yang saling terkoneksi dengan internet.
2. End-to-end Encryption adalah sebuah protokol dan mekanisme yang berfokus pada keamanan di titik akhir koneksi.
3. MQTT adalah protokol komunikasi yang berdasar pada TCP terdiri dari Broker, publisher dan subscriber.
4. CoAP adalah versi HTTP yang dibuat ulang untuk memenuhi kebutuhan IoT
5. TLS adalah protokol keamanan pada transport layer
6. Crypto adalah sebuah metode untuk menyimpan dan mengirim data dalam bentuk yang rahasia
7. Pengujian Pengiriman Data adalah metode yang dilakukan untuk mendapatkan suatu data.

1.3 Persiapan Testbed

Persiapan testbed adalah sebuah tahap yang disiapkan untuk melakukan penelitian. Tahap ini membutuhkan beberapa hal yang dibagi menjadi, perangkat keras dan perangkat lunak.

1.3.1 Kebutuhan Perangkat Keras

Menjelaskan spesifikasi dari perangkat keras yang digunakan untuk testbed dalam penelitian ini. Kebutuhan perangkat keras adalah:

1. Laptop dengan spesifikasi:

Merk	: ASUS-A450C
RAM	: 2 GB
Sistem Operasi	: Linux Blackbox

2. Raspberry Pi 2

Chipset	: Broadcom BCM2836
CPU	: 900 MHz quad-core ARM Cortex A7
RAM	: 1 GB
Sistem Operasi	: Raspbian Jessie

3. NodeMCU

   Developer : ESP8266 Opensource Community

   Type : Single-board microcontroller

   Operating System : XTOS

   CPU : ESP8266 (LX106)

   Memory : 128kBytes

   Storage : 4Mbytes

   Power : USB

1.3.2 Kebutuhan Perangkat Lunak

Menjelaskan spesifikasi dari perangkat lunak yang digunakan dalam testbed. Kebutuhan perangkat keras dalam penelitian ini adalah:

1. Wireshark, aplikasi untuk meng-capture paket dari jaringan
2. Nodejs v6, program untuk menjalankan kode javascript.
3. Tcpdump, program untuk meng-capture paket dari jaringan
4. ESPlorer, aplikasi untuk transfer source code ke dalam nodeMcu
5. esptool.py, aplikasi untuk flash firmware binary ke dalam nodeMcu
6. Microsoft Excel, aplikasi untuk mengolah data dengan kolom dan baris
7. NodeMCU custom build by frightanic.com

   branch: 1.5.4.1-final

   commit: 1885a30bd99aec338479aaed77c992dfd97fa8e2

   SSL: true

   modules: cjson, coap, crypto, dht, file, gpio, http, mqtt, net, node, rtctime, sntp, tmr, uart, wifi, tls

   build built on: 2018-01-04 03:38

   powered by Lua 5.1.4 on SDK 1.5.4.1(39cb9a32)
   firmware untuk sensor nodeMcu

1.4 Perancangan Lingkungan

Perancangan lingkungan dilakukan sebagai tahap untuk membuat setiap lingkungan yang dibutuhkan pada pengujian.

1.5 Pengujian dan Pengambilan Data

Pengujian dan pengambilan data dilakukan untuk mengetahui bahwa mekanisme keamanan yang ditambahkan pada komunikasi antara sensor nodeMcu dan *middleware* berhasil. Program tcpdump untuk meng-*capture* pengiriman data digunakan dalam pengujian dan pengolahan data ini.

1.5.1 Pengujian Pengiriman Data

Pengujian ini dilakukan untuk mengetahui apakah sensor nodeMcu dan middleware bekerja dengan baik sebelum dan sesudah ditambahkan mekanisme keamanan pada kedua protokol MQTT dan CoAP. Berikut adalah skenario pengujian yang akan dilakukan:

1.5.1.1 Skenario 1: Pengiriman data protokol MQTT tanpa mekanisme keamanan

Pengiriman data dengan protokol MQTT tanpa mekanisme keamanan. Pengiriman dilakukan dengan mengirimkan data dari sensor nodeMcu selama 10 menit ke *middleware* dengan jeda 30 detik dan lima kali pengujian. Dan menjalankan program tcpdump untuk meng-*capture* transmisi data yang terjadi. Hasil dari program tcdump kemudian diolah dengan wireshark. *Output* yang diharapkan adalah data bisa dikirim dari nodeMcu ke *middleware* dengan protokol MQTT.

1.5.1.2 Skenario 2: Pengiriman data protokol CoAP tanpa mekanisme keamanan

Pengiriman data dengan protokol CoAP tanpa mekanisme keamanan. Pengiriman dilakukan dengan mengirimkan data dari sensor nodeMcu selama 10 menit ke *middleware* dengan jeda 30 detik dan lima kali pengujian. Dan menjalankan program tcpdump untuk meng-*capture* transmisi data yang terjadi. Hasil dari program tcdump kemudian diolah dengan wireshark. *Output* yang diharapkan adalah data bisa dikirim dari nodeMcu ke *middleware* dengan protokol CoAP.

1.5.1.3 Skenario 3: Pengiriman data protokol MQTT dengan crypto

Pengiriman data dengan protokol MQTT dengan AES-CBC 128 bits. Pengiriman dilakukan dengan mengirimkan data dari sensor nodeMcu selama 10 menit ke *middleware* dengan jeda 30 detik dan lima kali pengujian. Dan menjalankan program tcpdump untuk meng-*capture* transmisi data yang terjadi. Hasil dari program tcdump kemudian diolah dengan wireshark. *Output* yang diharapkan adalah data bisa dikirim dari nodeMcu ke *middleware* dengan protokol MQTT dan telah dienkripsi.

1.5.1.4 Skenario 4: Pengiriman data protokol CoAP dengan crypto

Pengiriman data dengan protokol CoAP dengan AES-CBC 128 bits. Pengiriman dilakukan dengan mengirimkan data dari sensor nodeMcu selama 10 menit ke *middleware* dengan jeda 30 detik dan lima kali pengujian. Dan menjalankan program tcpdump untuk meng-*capture* transmisi data yang terjadi. Hasil dari program tcdump kemudian diolah dengan wireshark. *Output* yang diharapkan adalah data bisa dikirim dari nodeMcu ke *middleware* dengan protokol CoAP dan telah dienkripsi.

1.5.1.5 Skenario 5: Pengiriman data protokol MQTT dengan TLS

Pengiriman data dengan protokol MQTT dengan TLS. Pengiriman dilakukan dengan mengirimkan data dari sensor nodeMcu selama 10 menit ke *middleware* dengan jeda 30 detik dan lima kali pengujian. Dan menjalankan program tcpdump untuk meng-*capture* transmisi data yang terjadi. Hasil dari program tcdump kemudian diolah dengan wireshark. *Output* yang diharapkan adalah data bisa dikirim dari nodeMcu ke *middleware* dengan protokol MQTT setelah terjadi pertukaran sertifikat dalam koneksi yang aman dan data telah dienkripsi.

1.5.1.6 Skenario 6: Pengiriman data protokol CoAP denganTLS

Pengiriman data dengan protokol CoAP dengan TLS. Pengiriman dilakukan dengan mengirimkan data dari sensor nodeMcu selama 10 menit ke *middleware* dengan jeda 30 detik dan lima kali pengujian. Dan menjalankan program tcpdump untuk meng-*capture* transmisi data yang terjadi. Hasil dari program tcdump kemudian diolah dengan wireshark. *Output* yang diharapkan adalah data bisa dikirim dari nodeMcu ke *middleware* dengan protokol CoAP setelah terjadi pertukaran sertifikat dalam koneksi yang aman dan data telah dienkripsi.

1.5.2 Pengambilan Data

Pengambilan data dilakukan menggunakan hasil dari skenario-skenario pengujian. Data tersebut diolah dengan menggunakan aplikasi wireshark. Kemudian data tersebut di-export dalam format excel

1.6 Pembahasan

Analisa dilakukan pada hasil pengujian dan pengambilan data. Hasil analisis tersebut kemudian dibahas dan menjadi dasar dalam penentuan apakah mekanisme keamanan TLS dan crypto AES-CBC berhasil sebagai *end-to-end security* pada lingkungan middleware ini.

1.7 Penutup

Kesimpulan didapatkan dari analisis hasil seluruh skenario pengujian dan pengolahan data.