BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan dari keseluruhan metode penelitian yang telah dibahas dan dilakukan pada bab sebelumnya yang meliputi pendahuluan, kajian pustaka, metodologi, rekayasa kebutuhan, perancangan, implementasi dan pengujian dari Implementasi Access Control List Berbasis Protokol MQTT pada Perangkat NodeMCU, dapat dibuat kesimpulan bahwa:

1. Sistem berbasis protokol MQTT berhasil diimplementasi dan menerapkan *access control list* (ACL) yang diterapkan pada broker mosquitto menggunakan plugin mosquitto auth-plug, untuk melakukan mekanisme autentikasi dan otorisasi dilakukan dengan menggunakan auth-server yang berkomunikasi dengan broker MQTT melalui HTTP-*method* yang diterapkan menggunakan *framework* bottle pada python, dimana sistem dapat menangani lebih dari 100 pesan per detik di setiap konfigurasi broker.
2. Mekanisme autentikasi berhasil diimplementasi menggunakan token JWT sebagai *user-credential* untuk pemeriksaan *username* dan *password* klien MQTT di auth-server dengan waktu yang dibutuhkan untuk melakukan koneksi sekitar 0,0662 detik tanpa menggunakan TLS, dan mekanisme otorisasi berhasil di-implementasi menggunakan ACL yang diterapkan pada database MySQL untuk mengatur dan memeriksa hak akses berdasarkan peran pengguna dan topik yang akan diakses dengan *publish* atau *subscribe,* dengan waktu pemeriksaan sekitar 0,4883 detik tanpa menggunakan TLS.
3. NodeMCU yang digunakan dalam sistem berhasil berkomunikasi menggunakan protokol MQTT yang berperan sebagai salah satu klien MQTT dalam sistem, dengan waktu koneksi sekitar 0,0662 detik tanpa menggunakan TLS. Serta berhasil menerapkan token JWT yang akan digunakan sebagai parameter autentikasi, dan memiliki peran dalam ACL. NodeMCU dengan LED berhasil menangani pesan masuk untuk mengaktifkan LED dan mengirimkan *publish* setelahnya, dan nodeMCU dengan DHT11 berhasil membaca data suhu dan kelembaban lalu mengirimkan *publish* data tersebut.
4. Berdasarkan pada hasil pengujian keamanan sistem, diketahui bahwa sistem dengan konfigurasi broker tanpa auth-server memiliki celah keamananan dimana setiap pengguna yang mengakses sistem dapat melakukan koneksi tanpa adanya mekanisme autentikasi. Serta pengguna dapat mengakses setiap topik yang ada dalam sistem, baik sesuai dengan perancangan *topic tree* dan diluar dari perancangan tersebut, hal ini dikarenakan tidak ada mekanisme otorisasi yang digunakan untuk mengatur dan membatasi hak akses. Sedangkan konfigurasi broker dengan auth-server berhasil melakukan seleksi pengguna melalui mekanisme autentikasi menggunakan token JWT, dan juga dapat diseleksi saat mengakses topik tertentu yang ada dalam sistem melalui mekanisme otorisasi yang akan memeriksa ACL yang sudah dirancang, sehingga pengguna tertentu hanya dapat mengakses topik yang sesuai

dengan peran mereka pada database dan berdasarkan pada *publish* atau *subscribe* terkait hak akses tersebut.

5. Berdasarkan pada hasil pengujian keamanan sistem, diketahui bahwa sistem dengan konfigurasi broker tanpa menerapkan TLS memiliki celah keamanan dimana hasil *sniffing* paket berhasil mendapatkan setiap paket data yang dikirimkan ke broker dan isi paket data, berupa *user-credentials* dan isi pesan *publish-subscribe*, dapat dibaca secara *plain-text*. Sedangkan konfigurasi broker dengan menerapkan TLS berhasil mengamankan setiap paket data dari *sniffing* karena setiap paket data yang dikirimkan ke broker melalui pengguna tersebut dalam bentuk enkripsi melalui TLS, sehingga informasi didalam paket tersebut tidak dapat dilihat atau dicuri.
6. Berdasarkan pada hasil pengujian performa sistem berdasarkan waktu koneksi, waktu publish dan jumlah pesan per detik. Diketahui bahwa jika dibandingkan dengan konfigurasi *default* yang memiliki waktu 0,0288 detik untuk melakukan koneksi, lalu 0,1379 detik untuk melakukan publish dan mampu menangani 925 pesan per detik, kedua konfigurasi broker dengan auth-server memiliki performa yang jauh lebih rendah. Meskipun begitu performa tersebut masih dalam keadaan yang sangat baik dan cukup ideal, jika dibandingkan dengan aspek keamanan yang sangat *vulnerable* di konfigurasi broker *default*. Sedangkan konfigurasi broker yang menggunakan TLS jika dibandingkan dengan konfigurasi tanpa menggunakan TLS performanya tidak mengalami penurunan yang terlalu signifikan.

7.2 Saran

Setelah didapatkan kesimpulan dari penelitian ini, terdapat beberapa saran yang disampaikan oleh penulis mengenai penelitian selanjutnya yang terkait dengan permasalahan yang ada dalam penelitian ini. Beberapa saran tersebut meliputi:

1. Sistem yang dibuat dapat dirancang lebih baik lagi menggunakan aplikasi *front-end* bagi pengguna untuk mengakses setiap fungsional yang ada.
2. Perlu dilakukan penelitian lebih lanjut untuk membandingkan performa dan keamanan terkait berbagai *backend* yang dapat digunakan pada mosquitto auth-plug yang lain selain menggunakan token JWT.
3. Perlu dilakukan penelitian lebih lanjut mengenai perbandingan auth-server yang digunakan pada protokol MQTT menggunakan *tools* selain mosquitto auth-plug seperti OAuth yang dapat diterapkan pada protokol MQTT.
4. Perlu dilakukan penelitian lebih lanjut mengenai perbandingan beberapa desain *access control list* (ACL) sebagai mekanisme otorisasi di sistem berbasis protokol MQTT untuk melihat pengaruh desain ACL dalam sistem.
5. Pelu dilakukan penelitian lebih lanjut mengenai keamanan protokol MQTT pada *layer transport* menggunakan TLS/SSL, baik dari perancangan TLS dan perancangan sertifikat *Cetificate of Authority* (CA) dan key.