

IMPLEMENTASI MQTT PROTOCOL PADA SMART HOME SECURITY BERBASIS WEB

Bekti Maryuni Susanto¹, Ery Setiyawan Jullev Atmadji², Willy Laurent Brenkman³

^{1,2,3} Program Studi Teknik Komputer, Jurusan Teknologi Informasi, Politeknik Negeri Jember

¹bekti@polije.ac.id, ²setiyawanjullev@gmail.com, ³laurentbrenkmanwilly@gmail.com

Abstrak

Penggunaan IoT semakin berkembang dalam waktu singkat, hal ini dikarenakan oleh semakin berkembangnya teknologi informasi. Hal ini menyebabkan keterlibatan banyak device yang saling terkoneksi dengan sensor yang terpasang pada lingkungan. Sehingga dengan adanya banyak device maka menyebabkan permasalahan interoperabilitas dari masing-masing alat. Untuk mengatasi hal tersebut maka diperlukan sebuah gateway atau protocol yang mampu menjembatani interoperabilitas tersebut. Salah satu tugas Gateway atau protocol tersebut adalah harus mampu menangani permasalahan interoperabilitas serta mampu menangani permintaan maupun device profile dari masing-masing sensor maupun device yang terkoneksi. MQTT sendiri adalah sebuah protocol konektivitas *machine to machine* (M2M) yang didesain mampu mengirimkan data dengan sangat ringan menggunakan arsitektur TCP/IP. Pada MQTT sendiri mempunyai keunggulan yaitu dapat mengirimkan data dengan bandwidth yang ringan, konsumsi listrik yang sedikit, latensi serta konektivitas yang sangat tinggi, ketersediaan variable yang banyak serta jaminan pengiriman data yang dapat dinegosiasikan. Paper ini membahas tentang implementasi MQTT protocol pada smart home security berbasis web. Topik ini dipilih karena keamanan rumah merupakan permasalahan yang sangat penting, apalagi saat kita meninggalkan rumah

Kata kunci : MQTT, smart home securiy, internet of things

1. Pendahuluan

Internet Of Things atau yang biasa dikenal dengan IoT adalah salah satu teknologi pintar yang menggabungkan antara lingkungan dengan device-device melalui media internet. Penggunaan IoT semakin berkembang dalam waktu singkat, hal ini dikarenakan oleh semakin berkembangnya teknologi informasi. Hal ini menyebabkan keterlibatan banyak device yang saling terkoneksi dengan sensor yang terpasang pada lingkungan. Sehingga dengan adanya banyak device maka menyebabkan permasalahan interoperabilitas dari masing-masing alat. Interoperabilitas adalah karakteristik produk atau sistem, yang antarmukanya benar-benar dipahami, bekerja dengan produk atau sistem lain, sekarang atau masa depan, baik dalam implementasi maupun akses, tanpa batasan apapun (Kim, Choi, & Rhee, 2015). Berdasarkan survey yang telah dilakukan oleh gartner dalam (Grgic, Speh, & Hedi, 2016) pada tahun 2020 sebanyak 20 milyar object akan terkoneksi antara satu dengan yang lain.

Untuk mengatasi hal tersebut maka diperlukan sebuah gateway atau protocol yang mampu menjembatani interoperabilitas tersebut. Salah satu tugas Gateway atau protocol tersebut adalah harus mampu menangani permasalahan interoperabilitas serta mampu menangani permintaan maupun device

profile dari masing-masing sensor maupun device yang terkoneksi. Salah satu teknologi yang sering digunakan dalam menangani permasalahan tersebut adalah *Device Profile for web services* (DPWS) dan *Universal Plug and play* (UPnP) yang mana DPWS sendiri mengadopsi teknologi dari *Web Service Definition Language* (WSDL) (Govindan and Azad, 2015), tetapi DPWS sendiri memiliki kelemahan yaitu sulit untuk diterapkan, khususnya pada device dengan arsitektur ARM misal Arduino. Arduino sendiri adalah salah satu platform hardware open source yang dapat dikembangkan dan mendukung banyak *interface* seperti zigbee, Ethernet, Bluetooth ataupun yang lain.

Oleh karena itu maka diperlukan sebuah mekanisme akuisisi data dan pengiriman data yang stabil dan ringan, salah satu yang dapat digunakan adalah mekanisme *Representational State Transfer* (REST) yang menggunakan *Unique Resource Identifier* (URI), namun REST masih belum mampu menangani interoperabilitas dari masing-masing device, oleh karena itu diperlukan protocol tambahan yang mampu menangani hal tersebut, salah satu yang sering digunakan adalah *Message Queue Telemetry Transport* (MQTT).

MQTT sendiri adalah sebuah protocol konektivitas *machine to machine* (M2M) yang didesain mampu mengirimkan data dengan sangat

ringan menggunakan arsitektur TCP/IP (Dürkop et al., 2015). Pada MQTT sendiri mempunyai keunggulan yaitu dapat mengirimkan data dengan bandwidth yang ringan, konsumsi listrik yang sedikit, latensi serta konektivitas yang sangat tinggi, ketersediaan variable yang banyak serta jaminan pengiriman data yang dapat dinegosiasikan.

Paper ini membahas tentang implementasi MQTT protocol pada smart home security berbasis web. Topik ini dipilih karena keamanan rumah merupakan permasalahan yang sangat penting, apalagi saat kita meninggalkan rumah. Menurut data Polrestabas Makassar terdapat tujuh kasus Kejahatan atau kriminalitas yang dikategorikan menonjol selama tahun 2016. Yakni aniaya berat, pembunuhan, pencurian dan pemberatan atau bobol rumah, pencurian dan kekerasan atau begal, Curanmor, curi hewan dan Narkoba. Total jumlah kasus dari tujuh kategori ini sebanyak 2570 laporan dengan indeks penyelesaian atau telah dilimpahkan ke Kejaksaan sebanyak 1335 kasus. Jumlah ini menurun dari tahun 2015 sebanyak 1,53 persen atau yang mana pada tahun lalu itu sebanyak 2610 kasus. Mayoritas dari kategori menonjol ini mengalami trend penurunan kecuali kasus pembobolan rumah yang mana tahun lalu 452 laporan sedangkan tahun ini naik hingga 613 laporan atau 35,62 persen. Pembobolan rumah pun menempati posisi kedua dengan jumlah kasus terbanyak (Alfian, 2016).

Sistem terdiri dari sensor *Passive Infra Red* (PIR) untuk mendeteksi intruder, sensor gas untuk mendeteksi kebocoran gas elpiji dan sensor api untuk mendeteksi api. Data dari sensor diteruskan ke Gateway *Internet of Things* (IoT). Selanjutnya data tersebut disimpan dalam database *Mysql*. Gateway IoT menggunakan *Raspberry Pi* dengan sistem operasi *Raspbian*.

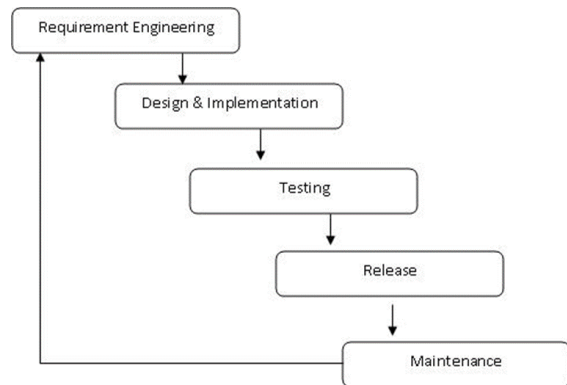
2. Metode Penelitian, Gambar dan Tabel

2.1 Metode Penelitian

Tahapan dalam penelitian ini ditunjukkan pada Gambar 1. Penelitian ini menggunakan metode waterfall, yang terdiri dari tahapan requirement engineering, design and implementation, testing, release dan maintenance (Petersen, Wohlin, & Baca, 2009). Metode penelitian yang digunakan adalah penelitian eksperimen. Penelitian eksperimen dilakukan dimana penelitian melibatkan investigasi hubungan sebab akibat menggunakan tes yang dikendalikan oleh peneliti (Dawson, 2009).

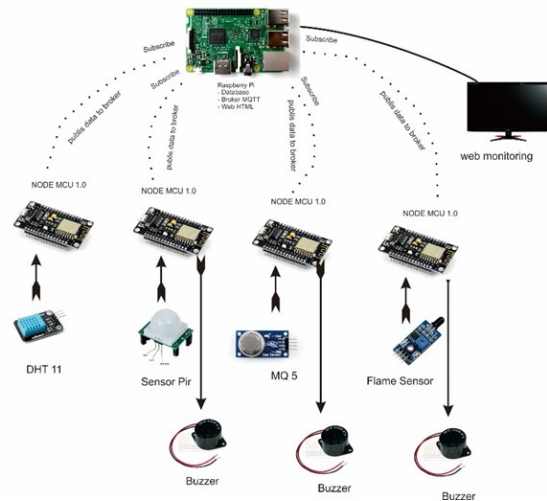
2.2 Gambar

Gambar 1 menunjukkan metode penelitian yang digunakan pada penelitian ini. Tahapan-tahapan metode penelitian yang digunakan yaitu requirement engineering, design and implementation, testing, release dan maintenance.

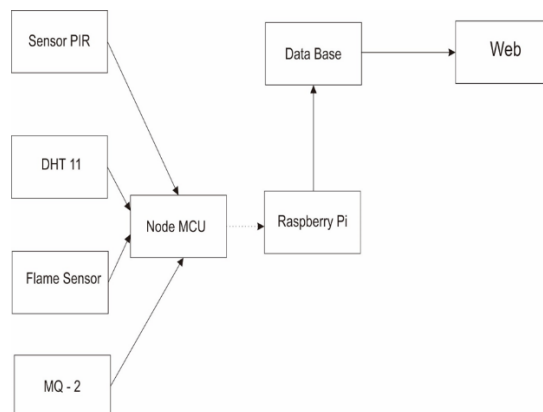


Gambar 1. Tahapan Penelitian

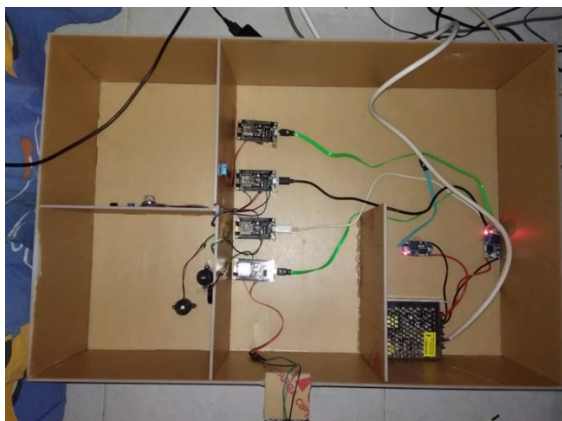
Desain alat pada penelitian ini ditunjukkan pada Gambar 2. Beberapa sensor digunakan untuk melakukan akuisisi data. Sensor-sensor tersebut adalah sensor suhu dan kelembapan, Sensor PIR, sensor gas dan sensor api. Sebuah mikrokontroler terhubung langsung dengan masing-masing sensor. Mikrokontroler ini bertugas untuk mengolah data sensor dan meneruskan data sensor tersebut ke gateway IoT menggunakan protokol komunikasi MQTT. Selanjutnya data tersebut disimpan pada database *Mysql*. Diagram blok alat yang dikembangkan ditunjukkan pada Gambar 3.



Gambar 2. Desain Alat



Gambar 3. Diagram Blok



Gambar 4. Prototipe smart home security

Gambar 4 menunjukkan prototipe smart home security yang merupakan miniatur dari sebuah rumah yang dilengkapi dengan berbagai macam sensor. Sebuah website dibangun untuk menampilkan data-data sensor yang disimpan pada database Mysql. Mikrokontroler akan mengirimkan data setiap jangka waktu tertentu ke database Mysql. Tampilan website untuk monitoring keamanan rumah ditunjukkan pada gambar 5.

Gambar 8 menunjukkan komunikasi protokol MQTT antara mikrokontroler dan raspberry. Mekanisme komunikasi pada protokol MQTT menggunakan protokol TCP/IP. Protokol MQTT memiliki dua komponen utama yaitu MQTT broker dan MQTT client. Keduanya saling berkomunikasi dalam bentuk subscribe dan publish. MQTT broker berfungsi sebagai server yang akan menerima semua informasi dari client serta akan melakukan publish ke client yang mensubscribe jenis topik tertentu.

Data Pir	
Pir	Waktu
Tidak Ada Intruder	2018-07-07 11:40:52
Tidak Ada Intruder	2018-07-07 11:40:50
Tidak Ada Intruder	2018-07-07 11:40:48
Tidak Ada Intruder	2018-07-07 11:40:46
Tidak Ada Intruder	2018-07-07 11:40:43
Tidak Ada Intruder	2018-07-07 11:40:41
Tidak Ada Intruder	2018-07-07 11:40:39
Tidak Ada Intruder	2018-07-07 11:40:36

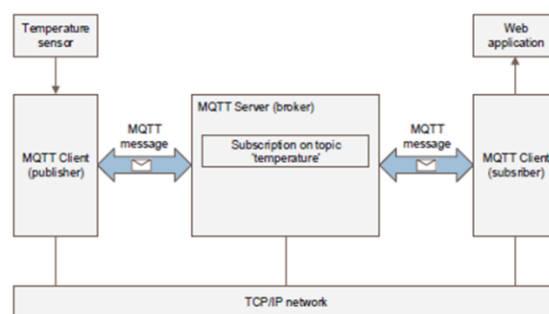
Gambar 5. Database sensor PIR

3. Pembahasan

Message Queuing Telemetry Transport (MQTT) adalah protokol transport yang bersifat client server publish/subscribe. Protokol yang ringan, terbuka dan sederhana, dirancang agar mudah diimplementasikan. Karakteristik ini membuat MQTT dapat digunakan di

banyak situasi, termasuk penggunaannya dalam komunikasi machine-to-machine (M2M) dan Internet of Things (IoT), protokol ini berjalan pada TCP/IP. Protokol MQTT membutuhkan transportasi yang menjalankan perintah MQTT, byte stream dari client ke server atau server ke client. Protokol transport yang digunakan adalah TCP/IP. TCP/IP dapat digunakan untuk MQTT, selain itu TLS dan WebSocket juga dapat menggunakan TCP/IP.

Pada protokol MQTT terdapat dua komponen utama yaitu MQTT client dan MQTT server. MQTT client bertindak sebagai publisher dan MQTT server bertindak sebagai subscriber dari sebuah topik. Pada penelitian ini yang bertindak sebagai MQTT client adalah mikrokontroler yang terhubung langsung dengan sensor untuk akuisisi data. Sedangkan yang bertindak sebagai MQTT server adalah Raspberry Pi.



Gambar 6. Protokol komunikasi MQTT

Hasil pengukuran throughput menggunakan software wireshark sebesar 1004 bit/s atau kurang dari 1 Kbps (1 Kbps = 1024 bps). Hal ini menunjukkan pengiriman data antara client MQTT dan server MQTT yang sangat kecil. Mikrokontroler yang bertugas untuk melakukan akuisisi data berfungsi sebagai client MQTT. Sedangkan, Raspberry Pi yang bertugas untuk menyimpan data sensor ke dalam database Mysql berfungsi sebagai server MQTT. Hasil pengukuran throughput ditunjukkan pada Gambar 7.

Interface	Dropped packets	Capture filter	Link type	Packet size limit
wlan0	Unknown	none	Ethernet	262144 bytes

Measurement	Captured	Displayed	Marked
Packets	128	128 (100.0%)	N/A
Time span, s	61.075	61.075	N/A
Average pps	2.1	2.1	N/A
Average packet size, B	59.5	59.5	N/A
Bytes	7670	7670 (100.0%)	0
Average bytes/s	125	125	N/A
Average bits/s	1004	1004	N/A

Gambar 7. Pengukuran throughput

4. Kesimpulan dan Saran

Penerapan protokol MQTT pada smart home security memungkinkan untuk pengiriman data yang ringan sehingga tidak membebani bandwidth gateway IoT. Penelitian berikutnya dapat menerapkan push notification pada aplikasi Android jika data hasil akuisisi data sensor menunjukkan

anomali. Sebagai contoh misalnya jika sensor PIR mendeteksi adanya intruder maka selain menyimpan data ke dalam sensor juga memberikan notifikasi dalam bentuk aplikasi Android sehingga user bisa langsung mengetahui kondisi rumah secara realtime. Selain itu penelitian berikutnya dapat menerapkan Virtual Private Network (VPN) ataupun cloud database sehingga monitoring gateway IoT dapat dilakukan dimana saja.

Daftar Pustaka:

- Alfian. (2016, Desember Senin, 26). *Begal Menurun, Pembobolan Rumah Meningkatkan 35 Persen di Tahun 2016*. Retrieved from Tribun Timur: <http://makassar.tribunnews.com/2016/12/26/begal-menurun-pembobolan-rumah-meningkat-35-persen-di-tahun-2016>
- Dawson, C. W. (2009). *Projects in Computing and Information Systems A Students Guide*. Essex: Pearson Education Limited.
- Durkop, L., Czybik, B., & Jasperneite, J. (2015). Performance evaluation of M2M protocols over cellular networks in a lab environment. *2015 18th International Conference on Intelligence in Next Generation Networks* (pp. 70-75). Paris, France: IEEE Communications Society.
- Govindan, K., & Azad, A. P. (2015). End-to-end service assurance in IoT MQTT-SN. *12th Annual IEEE Consumer Communications and Networking Conference, CCNC 2015*, (pp. 290-296). Las Vegas, USA: IEEE.
- Grgic, K., Speh, I., & Hedi, I. (2016). A Web Based IoT Solution for Monitoring Data Using MQTT Protocol. *2016 International Conference on Smart Systems and Technologies (SST)* (pp. 249-253). Osijek, Croatia: IEEE.
- Kim, S.-M., Choi, H.-S., & Rhee, W.-S. (2015). IoT Home Gateway for Auto-Configuration and Management of MQTT Devices. *2015 IEEE Conference on Wireless Sensors (ICWiSe)* (pp. 12-17). Melaka Malaysia : IEEE.
- Petersen, K., Wohlin, C., & Baca, D. (2009). The Waterfall Model in Large-Scale Development . *Lecturer Notes in Bussiness Information Processing Vol. 32* , 386-400.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.42.12	192.168.42.1	MQTT	67	Publish Message
2	0.000187916	192.168.42.1	192.168.42.12	TCP	54	1883 → 49153 [ACK] Seq=1 Ack=14 Win=29200 Len=0
3	0.009458437	192.168.42.11	192.168.42.1	MQTT	62	Publish Message
4	0.009633125	192.168.42.1	192.168.42.11	TCP	54	1883 → 49153 [ACK] Seq=1 Ack=9 Win=29200 Len=0
5	0.015371770	192.168.42.10	192.168.42.1	MQTT	70	Publish Message
6	0.015514166	192.168.42.1	192.168.42.10	TCP	54	1883 → 49153 [ACK] Seq=1 Ack=17 Win=29200 Len=0
7	0.017509479	192.168.42.10	192.168.42.1	MQTT	64	Publish Message
8	0.017570833	192.168.42.1	192.168.42.10	TCP	54	1883 → 49153 [ACK] Seq=1 Ack=27 Win=29200 Len=0
9	0.132044635	192.168.42.13	192.168.42.1	MQTT	62	Publish Message
10	0.132155312	192.168.42.1	192.168.42.13	TCP	54	1883 → 49153 [ACK] Seq=1 Ack=9 Win=29200 Len=0
11	0.540974270	192.168.42.10	192.168.42.1	MQTT	70	Publish Message
12	0.541074739	192.168.42.1	192.168.42.10	TCP	54	1883 → 49153 [ACK] Seq=1 Ack=43 Win=29200 Len=0
13	0.543719791	192.168.42.10	192.168.42.1	MQTT	64	Publish Message
14	0.543814426	192.168.42.1	192.168.42.10	TCP	54	1883 → 49153 [ACK] Seq=1 Ack=53 Win=29200 Len=0
15	0.633066614	192.168.42.13	192.168.42.1	MQTT	62	Publish Message
16	0.633168333	192.168.42.1	192.168.42.13	TCP	54	1883 → 49153 [ACK] Seq=1 Ack=17 Win=29200 Len=0
17	0.999931822	192.168.42.12	192.168.42.1	MQTT	67	Publish Message
18	1.000026822	192.168.42.1	192.168.42.12	TCP	54	1883 → 49153 [ACK] Seq=1 Ack=27 Win=29200 Len=0
19	1.010623905	192.168.42.11	192.168.42.1	MQTT	62	Publish Message
20	1.010679947	192.168.42.1	192.168.42.11	TCP	54	1883 → 49153 [ACK] Seq=1 Ack=17 Win=29200 Len=0
21	1.066184374	192.168.42.10	192.168.42.1	MQTT	70	Publish Message
22	1.066290676	192.168.42.1	192.168.42.10	TCP	54	1883 → 49153 [ACK] Seq=1 Ack=69 Win=29200 Len=0
23	1.068304374	192.168.42.10	192.168.42.1	MQTT	64	Publish Message
24	1.068367916	192.168.42.1	192.168.42.10	TCP	54	1883 → 49153 [ACK] Seq=1 Ack=79 Win=29200 Len=0
25	1.134293437	192.168.42.13	192.168.42.1	MQTT	62	Publish Message
26	1.134367135	192.168.42.1	192.168.42.13	TCP	54	1883 → 49153 [ACK] Seq=1 Ack=25 Win=29200 Len=0
27	1.591032239	192.168.42.10	192.168.42.1	MQTT	70	Publish Message
28	1.591146249	192.168.42.1	192.168.42.10	TCP	54	1883 → 49153 [ACK] Seq=1 Ack=95 Win=29200 Len=0
29	1.593957603	192.168.42.10	192.168.42.1	MQTT	64	Publish Message
30	1.594078697	192.168.42.1	192.168.42.10	TCP	54	1883 → 49153 [ACK] Seq=1 Ack=105 Win=29200 Len=0
31	1.632993905	192.168.42.13	192.168.42.1	MQTT	62	Publish Message

▶ Frame 1: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface 0
 ▶ Ethernet II, Src: dc:4f:22:11:04:a5 (dc:4f:22:11:04:a5), Dst: Raspberr_15:b6:3a (b8:27:eb:15:b6:3a)
 ▶ Internet Protocol Version 4, Src: 192.168.42.12, Dst: 192.168.42.1
 ▶ Transmission Control Protocol, Src Port: 49153, Dst Port: 1883, Seq: 1, Ack: 1, Len: 13
 ▶ MQ Telemetry Transport Protocol

Gambar 8. Komunikasi MQTT Protocol