PENERAPAN MOSQUITTO SEBAGAI SERVER MQTT LOKAL PADA KOMUNIKASI ANTARA *DATABASE* MONGODB DAN MESIN ABSENSI *MULTI-NODE*

TUJUAN

1. Menerapkan protokol MQTT pada mesin absensi di *local area network* dengan mengandalkan *access point* terdekat untuk komunikasi lebih handal dan aman.
2. Menerapkan algoritma komunikasi *bidirectional* dari protokol MQTT pada perangkat-perangkat yang terhubung pada jaringan lokal atau jaringan publik.
3. Menggunakan aplikasi *GUI designer* PyQt5 dan penerapannya untuk menginput, mengedit, dan menampilkan data dari suatu *database* dengan konsep NoSQL salah satunya MongoDB dengan *library* PyMongo pada Python serta sebagai penerapan manajemen *database* secara terpusat.
4. Menganalisa pengaruh penggunaan server MQTT (lokal dan publik), proses parsing data pada Python dan update pada database MongoDB, serta penambahan jumlah client terhadap delay pada sistem.

LATAR BELAKANG

Mesin absensi di pasaran rata-rata hanya menggunakan sistem standalone database, yaitu 1 database untuk 1 mesin walaupun sudah menerapkan prinsip IoT. Salah satu protokol yang sering digunakan dalam IoT saat ini salah satunya protokol MQTT. Protokol MQTT sudah sering dijadikan pembahasan, namun masih memanfaatkan public MQTT Server yang sudah siap pakai dan kebanyakan hanya dilakukan untuk komunikasi 1 arah. Public MQTT broker harus menggunakan internet walaupun hanya untuk perangkat yang letaknya tidak terlalu jauh. Komunikasi antar client dengan cloud broker akan terganggu bila koneksi internet mengalami gangguan terutama instansi yang berada di daerah yang belum dijangkau oleh ISP yang handal. Untuk mengatasinya perlu menerapkan MQTT menggunakan jaringan lokal yang sudah tersedia tanpa memerlukan internet. Dengan menerapkan protokol MQTT secara lokal dan komunikasi dua arah, PyQt serta MongoDB sebagai manajemen database terpusat diharapkan mempermudah dalam memonitor dan mengatur jumlah mesin absensi yang terkoneksi.

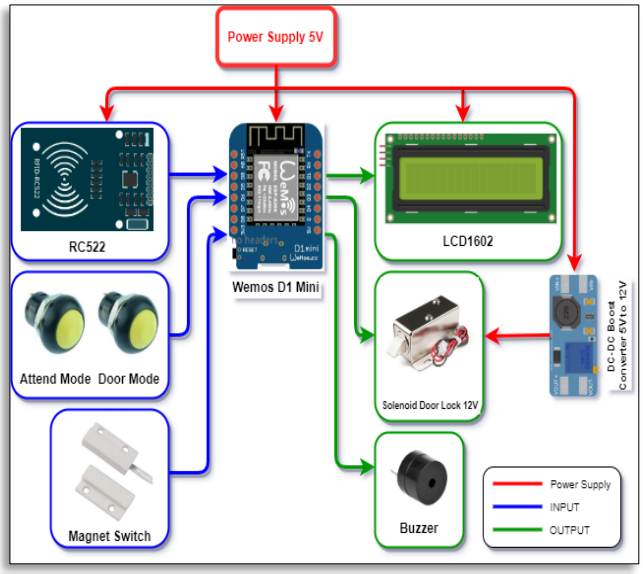
CARA KERJA ALAT

Urutan kerja sistem autentikasi dari *Client Node* ke *Client Database*, yaitu:
1. *User* melakukan *tapping* RFID tag pada mesin *client node*. Mesin akan mengirimkan UID ke server MQTT melalui topik khusus UID dan *unique client ID* dari *node* bersangkutan.
2. *Client database* akan mendeteksi *payload* yang masuk di topik tersebut. Dengan syarat *client database* sudah berlangganan ke topik tersebut.
3. Data yang diterima oleh *client database* akan dilakukan *parsing* serta pengecekan ke *database* MongoDB.
4. *Client database* akan mengirim respon balik ke server MQTT sesuai topik dari *client ID node* yang bersangkutan apakah UID *user* memiliki akses atau tidak.
5. *Client node* akan menampilkan output dan respon ke user berdasarkan *payload* respon yang sudah diterima oleh *client node*.

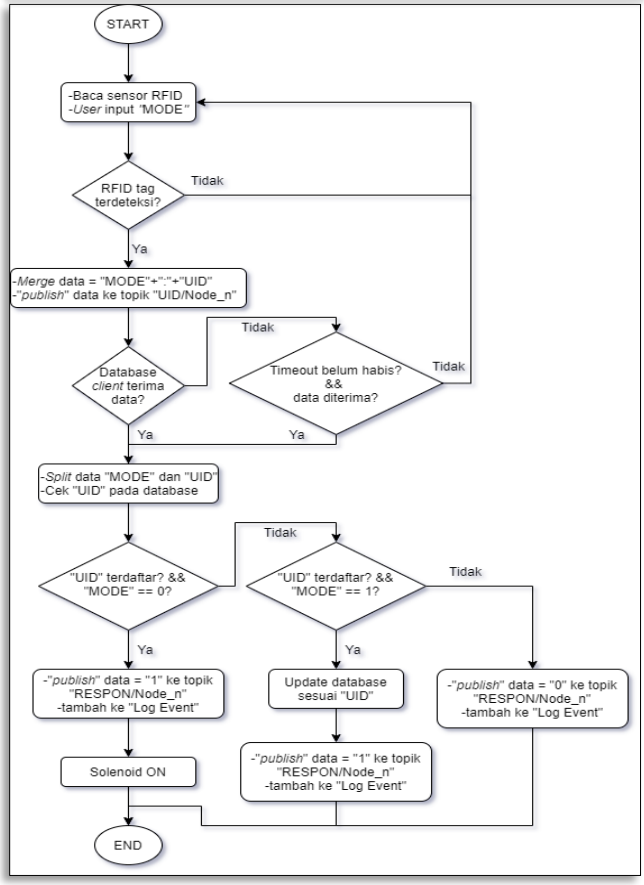
SPESIFIKASI ALAT

- **Local MQTT Server**
  1. Hardware: Raspberry Pi 3 Model B V1.2
  2. Sistem Operasi: Ubuntu Mate 18.04.4 LTS 64bit
- **Monitor dan Database Client**
  1. Hardware: Raspberry Pi 3 Model B V1.2
  2. Sistem Operasi: Ubuntu Mate 18.04.4 LTS 64bit
  3. Bahasa Pemrograman: Python 3.6
  4. GUI: PyQt5
  5. Database: MongoDB V4.2
  6. Database *library*: Pymongo
  7. MQTT Client library: Paho-Mqtt
- **Node Client**
  1. Hardware: ESP8266 Wemos D1 Mini
  2. Bahasa Pemrograman: Arduino C/C++

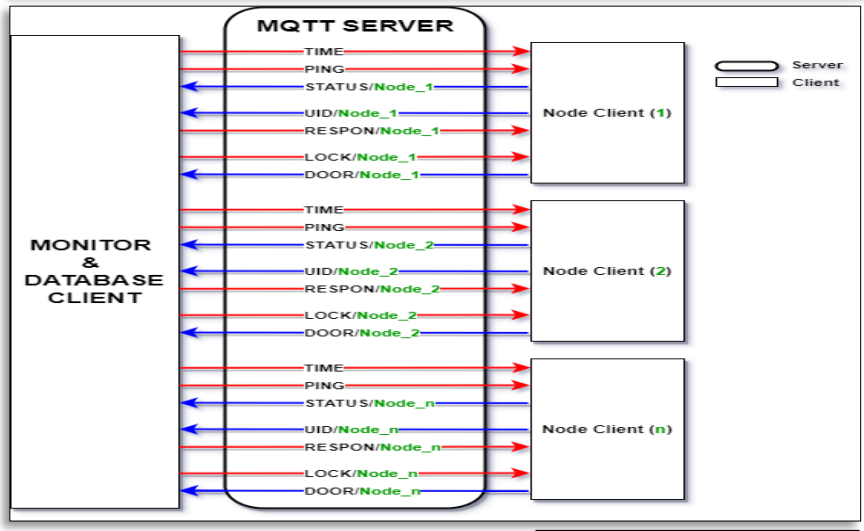
BLOK DIAGRAM HARDWARE *NODE CLIENT*



FLOWCHART SISTEM AUTENTIKASI



BLOK DIAGRAM SISTEM KOMUNIKASI



Dibuat Oleh:
Khiarul Arham
(1903423002)

Dosen Pembimbing:
Agus Wagyana, S.T., M.T.
(196808241999031002)

Tanggal Sidang:
28 Juli 2020