Technologia : Jurnal Ilmiah
p-ISSN: 2086-6917, e-ISSN: 2656-8047
Homepage: https://ojs.uniska-bjm.ac.id/index.php/JIT/index
Vol. 15 No. 1 januari 2024

RANCANG BANGUN SISTEM NURSE CALL SECARA NIRKABEL MENGGUNAKAN MQTT

Zaiyan Ahyadi*, Imansyah Noor
[1] Program Studi Elektronika, Politeknik Negeri Banjarmasin, Indonesia.
[2] Program Studi Elektronika, Politeknik Negeri Banjarmasin, Indonesia.

Informasi Artikel:
Dikirim: 07-12-2023; Diterima: 09-12-2023; Diterbitkan: 18-01-2024
Doi : http://dx.doi.org/10.31602/tji.v15i1.13369

ABSTRAK

Sistem Nurse Call digunakan pada Rumah Sakit atau Klinik untuk meningkatkan waktu respon perawat terhadap permintaan bantuan dari pasien. Sistem Nurse Call konvensional memerlukan pengkabelan sedangkan sistem Nurse Call modern adalah nirkabel, sehingga sistem jaringannya lebih flesksibel untuk dikembangkan.

Penelitian ini bertujuan untuk membuat prototipe sistem Nurse Call sederhana secara nirkabel. Peralatan utama yang digunakan pada sistem ini adalah nodemcu Esp8266 yang merupakan mikrokontroller untuk sistem IoT. Sebagai pendukung adalah router wireless. Sistem komunikasi antar esp8266 menggunakan protokol MQTT.

Hasil ujicoba sistem meperlihatkan rancangan prototipe berfungsi dengan baik dengan kemampuan dapat menangani penekanan tombol yang bersamaan tanpa kegagalan proses. Empat macam server broker diuji coba untuk diukur waktu teransfer datanya. Sistem juga dilengkapi dengan tampilan panel dalam bentuk halaman web jika tersedia komputer dengan jaringan internet.

Kata Kunci: Sistem Nurse Call, Nodemcu esp8266, MQTT

This is an open-access article under a Creative Commons Attribution 4.0 International (CC-BY 4.0) License. Copyright © 2023 by author.

Pendahuluan

Sistem Nurse Call adalah sistem teknologi komunikasi topografi antar jaringan yang biasa digunakan di Rumah Sakit atau Klinik. Sistem ini saling berhubungan baik itu melalui kabel-kabel yang dipasang dan ada juga yang menggunakan sistem *wireless* (nirkabel). Sistem ini dapat bekerja tanpa menggunakan jaringan internet untuk menghubungkan sistem Nurse Call. Sistem ini diterapkan pada bangunan rawat inap pasien. Sistem Nurse Call digunakan untuk meningkatkan waktu tanggap (respon time) perawat terhadap panggilan dari pasien yang dirawat.

Sistem Nurse Call konvensional masih menggunakan kabel, sehingga instalasi kabel nurse call biasanya dilakukan pada saaat bersamaan dengan pembangunan gedung. Untuk bangunan rumah sakit / klinik yang ketika dibangun belum disediakan instalasi kabel untuk sistem Nurse Call, maka akan menjadi suatu masalah. Demikian juga jika bangunan dilakukan renovasi atau perubahan tata letak maka ada kemungkinan akan mengganggu sistem pengkabelan Nurse Call. Sistem dengan kabel ini menggunakan sistem LAN yang memerlukan server dan juga yang menggunakan mikrokontroller dengan menggunakan protokol modbus (Sharma & Gautam, 2015).

Sistem Nurse Call modern yang menggunakan komunikasi nirkabel menjawab permasalahan pengkabelan. Sistem nirkabel lebih fleksibel dan sistem jaringan dengan

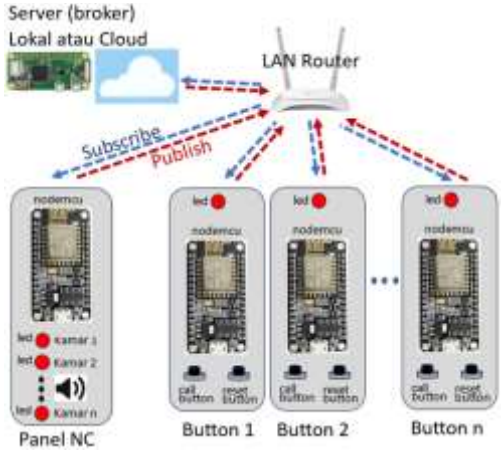
mudah dapat diperbesar dalam hal jumlah. Peralatan tombol elektronik untuk ON/OFF lampu yang tersedia di pasaran seperti keluaran Sonoff (Sonoff, 2021), tidak dapat digunakan dengan mudah untuk tombol pemanggil pasien, sehingga penulis medapatkan ide untuk merancang prototipe tombol dan sistem nirkabel untuk sistem Nurse Call. Beberapa penelitian telah mengembangkan sistem Nurse Call secara wireless dengan dengan menggunakan protokol http request (Aswin et al., 2011; Nursuwars & Rahmatulloh, 2019). Pada sistem ini juga digunakan RFID sebagai kartu identitas perawat yang merespon panggilan untuk disimpan dalam log pada database.

Perkembangan pesat teknologi elektronika dan sistem jaringan komputer serta teknologi Internet of Thing (IoT) dewasa ini menjadikan sistem Nurse call secara nirkabel banyak dikembangkan. Hal ini didukung oleh tersedianya komponen pendukung di pasaran dengan harga yang murah. Salah satunya adalah nodemcu esp8266 yang merupakan mikrokontroller yang dilengkapi fasilitas wifi (Schwartz, 2016; Weher et al., 2016). Perkembangan sistem komputer dewasa ini yang menghasilkan berbagai jenis komputer mini turut mendukung sistem menjadi lebih fleksibel dalam hal ukuran fisik server. Ini seperti penelitian Mahmud et al. (2017), yang menggunakan jenis komputer mini Raspberry Pi sebagai server pada sistem Nurse Call. Artikel Patnaikuni (2017), memaparkan kemampuan berbagai mikrokontroller dan komputer mini seperti Arduino, Raspberry Pi dan esp 8266 dalam hal penggunaannya dalam sistem IoT. Menurut artikel Mehta (2015), kelebihan esp8266 adalah harganya yang sangat murah sebagai komponen IoT secara nirkabel.

Salah satu protokol IoT adalah MQTT, yang merupakan singkatan dari Message Queuing Telemetry Transport, yang telah menjadi standard protokol transfer data dalam sistem Internet of Thing. Arsitektur sistem MQTT terdiri dari tiga bagian yaitu Broker, Publisher dan Subsciber. Broker adalah server tempat bertemunya publisher dengan subscriber. Publisher adalah komponen IoT yang mengirimkan data sedangkan Subscriber dalah komponen yang menerima data. Transfer data dari publisher ke subscriber memerlukan suatu Topic. Setiap subscriber yang mendaftar pada broker dengan akan menerima data dari publisher dengan topik yang sama. Suatu peralatan dapat berfungsi sebagai sekaligus sebagai subscriber. Penelitian Yokotani & Sasaki, 2017 memaparkan keunggulan protokol MQTT dibandingkan dengan protokol http dalam hal ukuran overhead.

Metode

Penelitian ini bertujuan untuk merancang sistem Nurse Call secara Nirkabel dengan menggunakan MQTT, seperti yang ditunjukkan pada gambar 1. Sistem tersusun atas empat bagian.



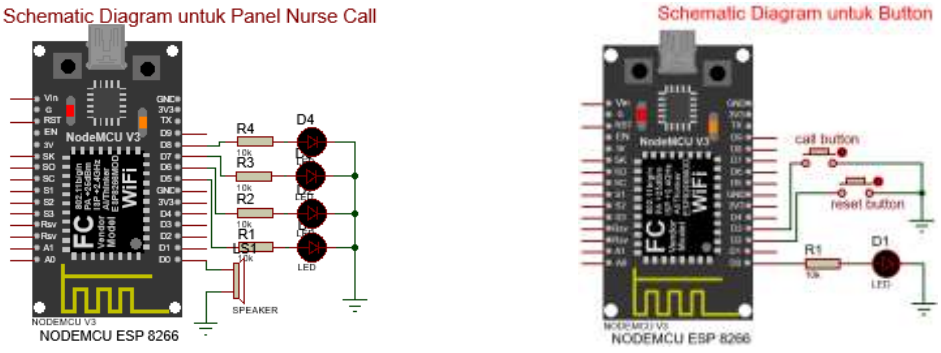
Gambar 1. Desain Sistem NC dengan MQTT

Bagian yang pertama adalah Panel Station Nurse Call (atau disebut Panel NC) seperti yang ditunjukkan oleh gambar 2a merupakan panel yang memperlihatkan status semua kamar pasien apakah terdapat panggilan atau tidak. Komponen utama pada sistem ini adalah nodemcu esp8266 atau esp32 yang merupakan mikrokontroller yang dilengkapi dengan fasilitas wifi. Selain itu juga terdapat led dan buzzer.

Bagian yang kedua adalah Tombol Nurse Call (Tombol NC). Bagian ini adalah modul tombol yang digunakan untuk melakukan panggilan. Tombol ini terdapat pada setiap kamar pasien. Komponen utama dari modul ini adalah nodemcu esp8266 juga yang dilengkapi dengan dua buah tombol, call dan reset seperti yang diperlihatkan pada gambar 2b. Modul ini juga dilengkapi dengan sebuah led yang akan menyala jika dilakukan proses call, dan akan mati jika ditekan tombol reset.

Bagian ketiga adalah router yang digunakan untuk media komunikasi wireless antar bagian lainnya. Bagian keempat adalah server broker yang menjadi jembatan antar modul Panel NC dan modul Tombol NC. Server ini bisa berupa server lokal atau dapat juga menggunakan server cloud yang terdapat pada jaringan internet.

Penerapan arsitektur MQTT pada sistem ini adalah sebagai berikut. Server Lokal atau Cloud berfungsi sebagai broker. Panel NC dan Button NC berfungsi sebagai publisher sekaligus sebagai subscriber. Topik "BtoP" digunakan untuk transfer *payload* (data) dari Button ke Panel, dan Topik "PtoB" digunakan untuk transfer data dari Panel ke Button.



(a) (b)

Gambar 2 (a) Rangkaian Station Panel Nurse Call (b) Rangkaian Button Nurse Call

Agar modul Tombol NC berfungsi maka nodemcu esp 8266 perlu diprogram dengan program seperti kode 1 sedangkan program untuk modul Panel ditunjukkan pada kode 2. Beberapa fungsi pada program tidak diperlihatkan karena merupakan fungsi umum yang dapat dilihat pada contoh yang terdapat pada Arduino untuk esp8266, yaitu fungsi setup_wifi() dan fungsi reconnect. Fungsi setup_wifi berfungsi untuk pengaturan awal agar nodemcu terhubung dengan SSID (hotspot) dan fungsi reconnect berfungsi untuk menghubungkan kembali jika komunikasi dengan SSID terputus.

Kode 1. Program untuk nodemcu pada Button

```
1  //Program untuk nodemcu button
2  #include <ESP8266WiFi.h>
3  #include <PubSubClient.h>
4  #include <string.h>
5  #define pinReq 0
6  #define pinRst 4
7  #define thisReq "S1"
8  #define thisRst "R1"
9  #define receiveMsg "P1"
10 #define tokenReset "1RES"
11
12 // Update these with values suitable for  network.
13 const char* ssid = "nama_SSID";
14 const char* password = "password_SSID";
15 const char* mqtt_server = "broker.mqtt-dashboard.com";
16 WiFiClient espClient;
17 PubSubClient client(espClient);
18 unsigned long lastMsg = 0;
19 #define MSG_BUFFER_SIZE  (10)
20 char msg[MSG_BUFFER_SIZE];
21 int value = 0;
22 char* kata=receiveMsg;
23
24 void setup_wifi() {...}
25
26 void callback(char* topic, byte* payload, unsigned int length) {
27   for (int i = 0; i < length; i++) {
28     msg[i]=(char)payload[i];
29   }
30   msg[length]='\0';
31   if (!strcmp(msg,kata)){
32     digitalWrite(BUILTIN_LED, LOW);
33   }
34   else if(!strcmp(msg,tokenReset)){
35     digitalWrite(BUILTIN_LED, HIGH);
36   }
37   else if(!strcmp(msg,"ALLRES")){
38     digitalWrite(BUILTIN_LED, HIGH);
39   }
40 }
41
```

```
42  void reconnect() {...}
43
44  void setup() {
45    pinMode(BUILTIN_LED, OUTPUT);
46    digitalWrite(BUILTIN_LED,HIGH);
47    pinMode(pinReq, INPUT_PULLUP);
48    pinMode(pinRst, INPUT_PULLUP);
49    Serial.begin(115200);
50    setup_wifi();
51    client.setServer(mqtt_server, 1883);
52    client.setCallback(callback);
53  }
54
55  bool bReq,bReqold, bRst,bRstold;
56  void loop() {
57    if (!client.connected()) {
58      reconnect();
59    }
60    client.loop();
61    bReqold=bReq; bReq=!digitalRead(pinReq);
62    bRstold=bRst; bRst=!digitalRead(pinRst);
63    if(bReq and !bReqold) {
64      client.publish("BtoP", thisReq);
65    }
66    if(bRst and bRstold){
67      client.publish("BtoP", thisRst);
68      digitalWrite(BUILTIN_LED,!LOW);
69    }
70  }
```

Program pada modul Tombol yang akan dijalankan jika tombol Call atau Reset ditekan dapat dilihat pada fungsi loop pada baris 61 sampai dengan baris 68. Sedangkan fungsi callback akan dijalankan jika terdapat data dengan topik yang sama yang diterima. Data kemudian dibandingkan secara bertahap. Data yang diterima merupakan data respon balik ketika Panel menerima data setelah tomboll call ditekan. Pada bagian awal program (baris 7 – 10) terdapat preprocessor define angka yang mengacu pada alamat Button. Sehingga hanya dengan mengganti angka tersebut maka program dapat digunakan untuk modul button dengan alamat yang berbeda.

Kode 2. Program untuk nodemcu Panel

```
1  //Program untuk nodemcu Panel NC
2  #include <ESP8266WiFi.h>
3  #include <PubSubClient.h>
4  #define led1 14
5  #define led2 12
6  #define led3 13
7  #define led4 15
8  #define Buzzer D1
9  // Update these with values suitable for network.
```

```
10  const char* ssid = "SSID";
11  const char* password = "password";
12  const char* mqtt_server = "broker.mqtt-dashboard.com";
13
14  WiFiClient espClient;
15  PubSubClient client(espClient);
16  unsigned long lastMsg = 0;
17  #define MSG_BUFFER_SIZE  (10)
18  char msg[MSG_BUFFER_SIZE];
19  int value = 0;
20
21  void setup_wifi() {...}
22
23  void callback(char* topic, byte* payload, unsigned int length) {
24   for (int i = 0; i < length; i++) {
25    msg[i]=(char)payload[i];
26   }
27   msg[length]='\0';
28   if((char)payload[0]=='S'){
29    digitalWrite(Buzzer, HIGH);
30    switch((char)payload[1]){
31     case '1': digitalWrite(led1, HIGH);
32           client.publish("PtoB","P1"); break;
33     case '2': digitalWrite(led2, HIGH);
34           client.publish("PtoB","P2"); break;
35     case '3': digitalWrite(led3, HIGH);
36           client.publish("PtoB","P3"); break;
37     case '4': digitalWrite(led4, HIGH);
38           client.publish("PtoB","P4"); break;
39    }
40   }
41
42   if((char)payload[0]=='R'){
43    switch((char)payload[1])
44    {
45     case '1': digitalWrite(led1, LOW); break;
46     case '2': digitalWrite(led2, LOW); break;
47     case '3': digitalWrite(led3, LOW); break;
48     case '4': digitalWrite(led4, LOW); break;
49    }
50   }
51
52   if((char)payload[0]=='c') {
53    client.publish("PtoB","ALLRES");
54    digitalWrite(led1, LOW); digitalWrite(led2, LOW);
55    digitalWrite(led3, LOW); digitalWrite(led4, LOW);
56   }
57  }
58
59  void reconnect() {...}
```

```
60
61  void setup() {
62    pinMode(BUILTIN_LED, OUTPUT);
63    pinMode(led1,OUTPUT); pinMode(led2,OUTPUT);
64    pinMode(led3,OUTPUT); pinMode(led4,OUTPUT);
65    Serial.begin(115200);
66    setup_wifi();
67    client.setServer(mqtt_server, 1883);
68    client.setCallback(callback);
69  }
70
71  bool b1,b2,b3,b4,b1old,b2old,b3old,b4old;
72  void loop() {
73
74    if (!client.connected()) {
75      reconnect();
76    }
77    client.loop();
78  }
```

Pada Panel Nurse Call tidak terdapat tombol maka programnya bersifat menunggu kiriman data dari modul Button. Jika data diterima maka fungsi callback akan dijalankan. Data akan dibandingkan secara bertahap. Kemudian menjalankan respon berupa transfer data ke modul Button. Jadi komunikasi dalam sistem ini seperti sistem *request* dan *response*.

Cara kerja sistem dapat diijelaskan sebagai berikut.

- Suatu Tombol NC ditekan, misal pada kamar pasien 1, maka nodemcu akan mengirimkan data *string* "S1" (maksudnya set dari kamar 1) dengan topik "BtoP" (maksudnya button to panel).
- Data yang dikirimkan diterima oleh nodemcu pada panel. Data ini dibandingkan secara bertahap perhuruf. Jika huruf pertama adalah 'S' (artinya set) , akan dibandingkan huruf kedua berupa angka 1. Respon akan diberikan dengan menyalakan led pada nomor 1 dan membunyikan buzzer, selanjutnya data "R1" dikirim balik ke broker dengan topik "PtoB".
- Data "P1" diterima seluruh nodemcu. Selanjutnya semua nodemcu membandingkan data yang diterima. Selain nodemcu 1 tidak memproses lebih lanjut karena angka berbeda dengan alamat mereka. Hanya nodemcu nomor 1 yang akan merespon dengan menyalakan led yang terdapat pada modul Button1. Proses *call-respon* selesai.
- Jika perawat telah selesai melayani call, maka akan ditekan tombol reset pada modul Button. Dan data "R1" dikirim ke Panel sekaligus mematikan led pada modul button tersebut.
- Nodemcu pada panel akan merespon data "R1" dengan mematikan led nomor 1.

Pada sistem ini juga dirancang Websocket yang dapat menampilkan semua kondisi indikator led yang terdapat pada Panel Nurse-Call dalam bentuk halaman web.

Karena MQTT tidak mendukung html maka diperlukan websocket menggunakan *library paho* (Sutton, 2015).

Hasil

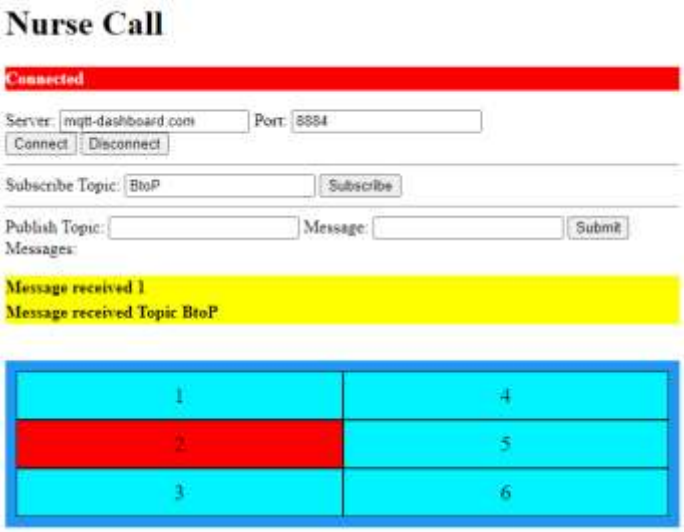
Hasil ujicoba sistem bekerja seperti yang diinginkan. Ketika tombol call ditekan data dikirimkan ke nurse panel. Panel Station Nurse Call akan menyalakan led dengan nomor yang sesuai kemudian mengirimkan data ke nodemcu tombol call. Selanjutnya data yang diterima tombol call akan memerintahkan led untuk menyala. Dengan menyalanya led pada Tombol Nurse Call menyatakan bahwa panggilan berhasil dilakukan.

Berbagai jenis server broker diujicoba pada sistem ini untuk melihat keberhasilan transfer data dan waktu antara tombol ditekan sampai led menyala. Empat macam server broker tersebut adalah applikasi broker MQTT pada smatphone (MQTT, 2023), program broker picoMQTT pada nodemcu esp8266 (Leśniewski & Fechner, 2023), program broker mosquitto yang diinstall pada Laptop (Mosquitto, 2019), dan broker online mqtt-dashboard.com (HiveMQ, 2023). Hasil ujicoba dapat dilihat pada Tabel 1.

Tabel 1. Ujicoba sistem Nurse Call terhadap beberapa jenis server broker

Server	Keberhasilan	Waktu rerata diukur 100 kali	Lokal /online	Mendukung websocket
Smartphone	100%	2305,68 milidetik	Lokal	Tidak
picomqtt pada esp8266	100 %	164,36 milidetik	Lokal	Tidak
mosquitto pada laptop	100%	1624,12 milidetik	Lokal	Ya
mqtt.dashboard.com	100%	516,93 mili detik	Online (Cloud)	Ya

Tampilan websocket dari sistem dapat dilihat pada gambar 3. Terlihat pada gambar kotak nomor 2 berwarna merah artinya terdapat panggilan dari Tombol 2.



Gambar 3. Tampilan Websocket Station Panel Nurse Call

Pembahasan

Rancangan prototipe telah dibuat dengan menggunakan komponen yang dirakit di atas papan project-board sebanyak 4 buah modul Button dan 1 buah model panel. Broker

yang digunakan adalah broker cloud pada internet dengan yaitu "broker.mqtt-dashboard.com". Untuk sumber tegangan tiap modul menggunakan charger smartphone dengan konektor usb mikro.

Uji coba pertama terhadap sistem hasil rancangan adalah dengan menekan Tombol NC 1 pada modul ditekan maka pada modul Panel NC akan terlihat led 1 akan menyala. Hal ini karena nodemcu pada modul Tombo mengirimkan data (payload) "S1" (publish) dengan topik "BtoP". Nodemcu pada modul Panel NC yang subscribe terhadap topic "BtoP" menerima data tersebut. Ketika terdapat data yang diterima program akan menjalankan fungsi callback. Fungsi callback akan memerintahkan led yang bersesuaian menjadi menyala. Dan menjalankan perintah untuk mengirim balik data berupa "P1" dengan topik "PtoB". Semua modul Button subscribe dengan "PtoB", namun hanya modul Tombol yang sesuai yang merespon, dalam ujicoba ini modul Tombol-1 yang merespon, dalam bentuk menyalakan led.

Ujicoba kedua adalah dengan adalah dengan menekan Tombol reset 1. Hasilnya terlihat led yang berada pada modul Tombol-1 menjadi padam, demikian pula led 1 pada modul Panel juga padam. Proses yang terjadi adalah ketika Tombol reset ditekan maka nodemcu mengirim data "R1" dengan topik "BtoP" dan menjalankan perintah untuk mematikan led pada modul Tombol-1. Data ini diterima oleh modul Panel dan akan merespon dengan mematikan led-1. Ujicoba seperti pertama dan kedua ini dilakukan juga terhadap modul Tombol NC lainnya. Dan hasilnya memperlihatkan sistem berjalan dengan baik.

Pada tabel 1 memperlihatkan hasil ujicoba sistem dengan menggunakan empat server broker yang berbeda. Hasilnya menunjukkan bahwa server broker lokal berupa nodemcu esp8266 dengan program yang menggunakan *library* picoMQTT menjadi yang paling cepat dalam hal transfer data. Server broker online mqtt-dashboard.com menjadi yang kedua tercepat. Sedangkan untuk broker lokal yang menggunakan laptop dan samrtphone memerlukan waktu rerata yang agak lama (di atas 1 detik). Hal ini kemungkinan disebabkan peralatan laptop atau smartphone tersebut tidak hanya menjalankan program broker saja, melainkan program dan service lainnya. Peneltian Soni & Makwana (2017), memaparkan beberapa fitur broker online dengan beberapa keterbatasannya masing-masing. Namun sayangnya tidak memperlihatkan kecepatan transfer data.

Ujicoba berikutnya dilakukan dengan terlebih dahulu mematikan semua led pada modul Panel menyala yang menandakan bahwa terdapat call dari seluruh modul Button. Selanjutnya dikirimkan data dari smartphone melalui aplikasi MQTT-Dashboard. Data yang dikirimkan adalah data berupa huruf "c" dengan topic publish "BtoP". Hasilnya memperlihatkan bahwa seluruh led pada modul Panel dan pada modul Button semuanya menjadi padam. Ujicoba ini bermaksud untuk mereset sistem dan hanya boleh dilakukan pada saat proses perawatan sistem.

Hasil ujicoba menggunakan websocket juga memperlihatkan hasil yang benar. Tampilan web akan memperlihatkan kotak yang sesuai dengan nomor kamar (nomer button call) akan berubah menjadi merah jika led pada Panel NC menyala, dan berwarna biru jika padam.

Ujicoba lainnya adalah bermaksud untuk menguji kehandalan sistem jika ada tombol yang ditekan secara bersamaan. Ini untuk mengetahui apakah protokol MQTT

dapat menangani call yang bersamaan. Ujicoba ini dilakukan dengan menghubungkan semua pin tombol tehubung menjadi satu pada satu tombol saja. Ini untuk memastikan bahwa data yang berbeda di-publish bersamaan untuk satu nodemcu pada Panel NC. Hasilnya memperlihatkan bahwa sistem tetap berjalan dengan baik. Semua led yang ditekan Button-call menjadikan lednya menyala.

Dari seluruh ujicoba, dapat dibuktikan sistem Nurse Call secara nirkabel menggunakan protokol MQTT telah bekerja dengan baik. Kelebihan protokol mqtt dibandingkan dengan protokol http request adalah data yang ditransfer lebih ringan, tidak seperti protokol http request yang harus me-*load* satu halaman web setiap pengiriman data. Dalam hal server (broker) juga tidak memerlukan pemrograman yang rumit.

Kesimpulan

Prototipe sistem Nurse Call secara nirkabel menggunakan MQTT hasil rancangan pada penelitian berupa sistem Nurse Call sedernana yang hanya berfungsi untuk memperlihatkan indikator led akan aktif pada Panel Station sehingga perawat jaga mengetahui kamar mana yang memerlukan bantuan. Komunikasi data dibuat dalam bentuk request dan respon untuk memastikan panggilan telah suskses dilakukan. Sistem hasil rancangan mampu menangani penekanan tombol secara bersamaan yang menunjukkan tidak ada proses *call* yang gagal.

Referensi:

Aswin, S., Gopalakrishnan, N., Jeyender, S., Prasanna, R. G., & Kumar, S. P. (2011). Design development and implementation of wireless nurse call station. *Proceedings - 2011 Annual IEEE India Conference: Engineering Sustainable Solutions, INDICON-2011*. https://doi.org/10.1109/INDCON.2011.6139633

HiveMQ. (2023). *HiveMQ Public Broker MQTT Dashboard*. HiveMQ. Retrieved from https://www.mqtt-dashboard.com/

Leśniewski, M., & Fechner, M. (2023). *PicoMQTT*. GitHub, Inc. Retrieved from https://github.com/mlesniew/PicoMQTT

Mahmud, M. S., Majumder, M. A., Tushar, A. K., Kamal, M. M., Ashiquzzaman, A., & Islam, M. R. (2017). Real-time feedback-centric nurse calling system with archive monitoring using Raspberry Pi. *Proceedings of 2017 International Conference on Networking, Systems and Security, NSysS 2017*, *2018-Janua*, 1–5. https://doi.org/10.1109/NSYSS2.2017.8267799

Mehta, M. (2015). ESP 8266: A Breakthrough in Wireless Sensor Networks and Internet of Things. *International Journal of Electronics and Communication Engineering &*

Technology, *6*(8), 7–11.

Mosquitto. (2019). *Mosquitto: An Open Source MQTT Broker*. Retrieved from https://mosquitto.org/

MQTT. (2023). *MQTT Broker: Dotworld Technologies Private Limited*. Google Play. Retrieved from https://play.google.com/store/apps/dev?id=6816313912734667814&hl=en&gl=US

Nursuwars, F. M. S., & Rahmatulloh, A. (2019). RFID for nurse activity monitoring in the hospital's nurse call system with Internet of Thing (IoT) concept. *IOP Conference Series: Materials Science and Engineering*, *550*(1), 1–7. https://doi.org/10.1088/1757-899X/550/1/012025

Patnaikuni, D. R. P. (2017). A Comparative Study of Arduino, Raspberry Pi and ESP8266 as IoT Development Board. *International Journal of Advanced Research in Computer Science*, *8*(5), 2350–2352.

Schwartz, M. (2016). Internet of Things with ESP8266. In *European University Institute* (1st ed., Issue 2). Packt Publishing.

Sharma, C., & Gautam, D. K. (2015). Design Development and Implementation of Wired Nurse Calling System. *International Conference on Green Computing and Internet OfThings (ICGCIoT)*, 1258–1262. https://doi.org/10.4018/978-1-7998-7607-6.ch010

Soni, D., & Makwana, A. (2017). A Survey on MQTT: a Protocol of Internet of Things (IOT). *International Conference on Telecommunication, Power Analysis and Computing Techniques (ICTPACT-2017)*, *20*(April), 173–177.

Sonoff. (2021). *SLAMPHERR2 Product Documents*. Sonoff. Retrieved from https://sonoff.tech/product-document/diy-smart-switches-doc/slampherr2-doc/

Sutton, J. (2015). *Paho JavaScript - MQTT Client Library Encyclopedia*. Eclipse Public Licence. https://www.hivemq.com/article/mqtt-client-library-encyclopedia-paho-js/

Weher, P., Seneviratne, P., Russell, B., & Duron, D. Van. (n.d.). *IoT: Building Arduino-Based Projects*. Packt Publishing.

Yokotani, T., & Sasaki, Y. (2017). Comparison with HTTP and MQTT on required network resources for IoT. *ICCEREC 2016 - International Conference on Control, Electronics, Renewable Energy, and Communications 2016, Conference Proceedings*, 1–6. https://doi.org/10.1109/ICCEREC.2016.7814989