

## ***Remote PC Wake-On Design Manufacture in Mujahidin Mosque at Candi Mendut***

### **Rancang Bangun Pengaktifan PC Jarak Jauh di Masjid Mujahidin Candi Mendut**

Rizki Priya Pratama<sup>1\*</sup>

#### **Abstract**

*There are several methods to turn on a PC from other places, one of which is WOL (wake-on lan). This method requires modem configuration to enable WOL and port forwarding features. However, if it is not possible, there is one method offered by the researcher. The method is creating a system that functions to turn on the PC via the MQTT protocol. This system consists of electronic devices and android applications. Both can be connected to the internet network. The function of this device is to connect the PWRBTN and GND pins for 0.3 seconds. While the android application "Remote wake-on PC Masjid Mujahidin" is a user interface to receive and send data to the device. From the experiment, it was found that this device was controlled from the internet with an android application successfully. In addition, there is information about the status of the PC, the system voltage on the device, and the temperature and voltage of the PC room on the android application. All this information is successfully transmitted from the PC-enabled device to the android application.*

#### **Keywords**

*MQTT, ESP8266, wake-on lan, Android, port forwarding*

#### **Abstrak**

Terdapat beberapa metode untuk menghidupkan PC dari keadaan mati/*off*, salah satunya adalah WOL (*wake-on lan*). Metode ini membutuhkan konfigurasi modem untuk mengaktifkan fitur WOL dan *port forwarding*. Namun, apabila tidak bisa, terdapat satu metode yang ditawarkan oleh peneliti. Metode tersebut adalah sebuah sistem yang berfungsi untuk menghidupkan PC melalui protokol MQTT. Sistem ini terdiri dari sebuah perangkat elektronik dan aplikasi android yang dapat terhubung pada jaringan internet. Perangkat ini bertugas untuk meng-on-kan PC dengan cara menghubungkan pin PWRBTN dan GND selama 0,3 detik. Sedangkan aplikasi android "Remote wake-on PC Masjid Mujahidin" sebagai user interface untuk menerima dan mengirimkan data ke perangkat. Dari pengujian, didapatkan bahwa perangkat ini berhasil dikendalikan dari internet dengan aplikasi android. Selain itu juga, terdapat informasi mengenai status PC, tegangan sistem pada alat serta suhu dan tegangan ruang PC pada aplikasi android. Semua informasi ini berhasil dikirimkan dari peralatan *wake-on PC* ke aplikasi android.

#### **Kata Kunci**

*MQTT, ESP8266, wake-on LAN, Android, port forwarding*

<sup>1</sup> Teknik Mekatronika, Politeknik Kota Malang  
Jalan Raya Tlogowaru No 3 Malang

\*rizkipriyap@gmail.com

Submitted : August 05, 2021. Accepted : September 08, 2021. Published : September 09, 2021.

## PENDAHULUAN

Salah satu hal yang perlu diperhatikan saat ini adalah perkembangan teknologi terutama pada bidang multimedia. Saat ini, teknologi multimedia sangat berpengaruh dan menarik perhatian jama'ah masjid[1]. Masjid Mujahidin Candi Mendut Malang juga telah difasilitasi dengan multimedia untuk mesy'iarkan kegiatan-kegiatan di dalam dan luar masjid. Seperti contoh dalam penyampaian kajian-kajian Islami, jama'ah yang sekiranya tidak memiliki kesempatan untuk hadir langsung ke masjid, maka mereka dapat menyimak dan mengikuti kajian tersebut melalui sosial media yang ada, seperti Youtube[2].

Kegiatan tersebut membutuhkan sebuah PC yang dapat digunakan untuk *streaming* sekaligus untuk merekam proses kajian tersebut. Selain itu, PC tersebut juga harus dapat diremote dan dihidupkan melalui internet tanpa harus ada operator di ruang komputer.

Saat akan ada kegiatan pengajian yang biasanya jadwalnya setelah sholat fardhu, PC itu harus dalam kondisi ON / siap melakukan *streaming*. Urutan kejadian ini berlangsung cepat sekali, sedangkan operator tidak dapat mengoperasikan komputer setelah sholat fardhu. Hal ini diakibatkan karena imam masih duduk di tempat sholat, yang mana, ruangan PC yang terletak di samping imam. Akibatnya jika komputer belum siap maka proses perekaman dan *streaming* menjadi terlambat atau terpotong. Selain itu, jika kegiatan pengajian akan berlangsung sedangkan operator tidak dapat hadir pada jadwal tersebut, maka komputer tersebut dapat diremote dari jarak jauh.

Kegiatan remote PC tersebut dapat menggunakan aplikasi Microsoft Remote Desktop, TeamViewer, Ammy Admin, UltraVNC, Windows Remote Desktop, Chrome Remote Desktop dan AnyDesk[3]. PC dapat di-remote jika berada dalam kondisi on / hidup. Namun permasalahannya adalah bagaimana jika komputer masih dalam keadaan mati. Ada metode untuk menghidupkan atau membangunkan komputer dari keadaan mati yaitu WOL (*wake-on lan*) atau WOW (*wake on wan*)[4]. Teknologi WOL ini telah dirancang untuk bekerja di jaringan lokal, namun dengan beberapa pengaturan pada *router* dan *firewall*, paket WOL dapat dikirim dari jaringan eksternal atau internet[5]. Namun, besar kemungkinannya paket WOL yang dikirim dari jaringan eksternal, tidak dapat mencapai PC yang dituju[6].

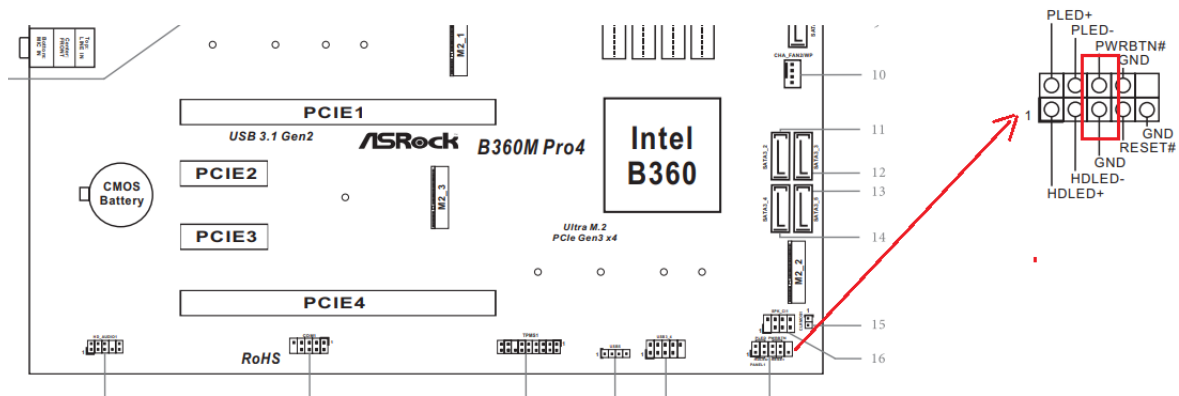
Akses internet di Masjid Mujahidin menggunakan WMS@Wifi.id. Layanan internet ini tidak memungkinkan untuk mengakses modem agar dapat mengaktifkan fitur *wake-on lan*, merubah *firewall*nya atau mengkonfigurasi fitur *port forwarding*. Jika tidak bisa mengaktifkan beberapa fitur tersebut, maka PC Masjid Mujahidin tidak dapat dihidupkan dengan WOL atau WOW.

Beberapa penelitian yang relevan dengan permasalahan ini adalah penelitian [6] dan [7]. Penelitian [6] ini menggunakan mikrokontroler PIC18F4620 dan IC lan ENC28J60 dengan *interface* berupa *web*. Sedangkan penelitian [7] merupakan *web service* yang ditanamkan pada perangkat Lantronix Xport. Penelitian-penelitian diatas menggunakan *web service* yang ditanamkan pada perangkat keras tersebut untuk mengirimkan paket WOL ke komputer tujuan. Dengan tujuan yang sama yaitu menghidupkan PC dari internet, penelitian ini menawarkan sebuah metode layaknya menekan tombol ON PC. Metodenya dengan memparalel *switch power* PC dengan perangkat *wake-on PC*. Perangkat ini dikendalikan melalui Internet dengan protokol MQTT. Dengan metode ini, modem tidak perlu diatur untuk konfigurasi pengiriman paket WOL yang ribet, sehingga perangkat ini dapat diterapkan pada jaringan apapun asalkan terhubung internet.

### Pengendali Wake-on PC

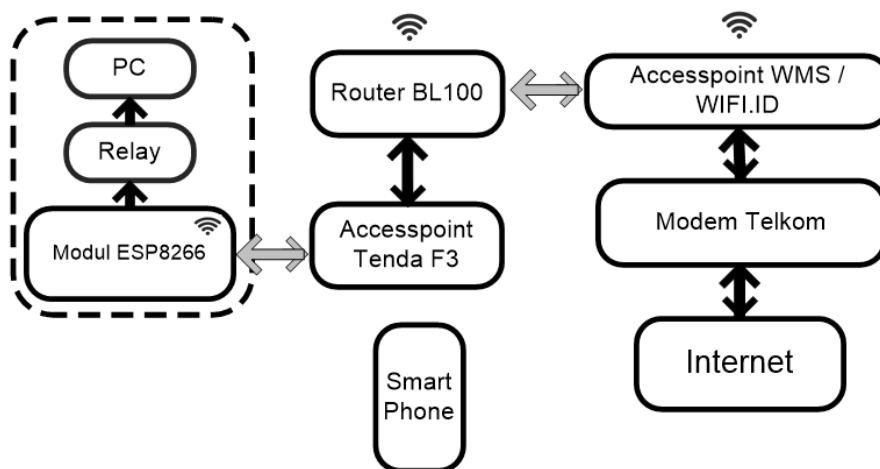
Pengendali *wake-on PC* ini merupakan perangkat yang dapat menghidupkan komputer yang sedang *off / sleep* dari internet dengan aplikasi Android. Perangkat ini bekerja dengan

menghubungkan antar pin PWRBTN dengan GND, layaknya seperti kita menekan tombol ON pada PC. *Motherboard* yang digunakan untuk PC ini adalah Asrock B360M. Gambar pin-pin ini dapat dilihat pada Gambar 1. Supaya PC ini dapat dikendalikan dengan perangkat *wake-on PC* maka pin-pin PWRBTN dan GND yang terhubung *switch* power PC, diparalel masuk ke rangkaian *relay*. Jika ada perintah dari *smartphone*, maka *relay* ini akan aktif dan PC akan hidup.



Gambar 1. Denah Motherboard Asrock B360M [8]

Perangkat *wake-on PC* ini harus terhubung jaringan internet, karena perangkat ini menggunakan *MQTT broker* yang berada pada alamat `broker.hivemq.com`. Blok diagram sistem dapat dilihat pada Gambar 2.

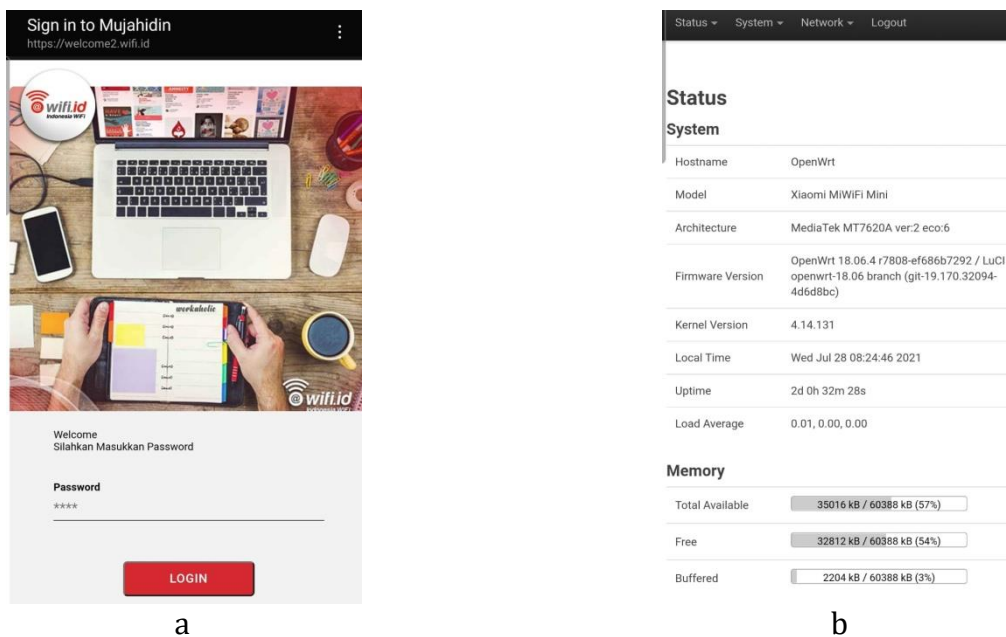


Gambar 2. Blok Diagram Sistem

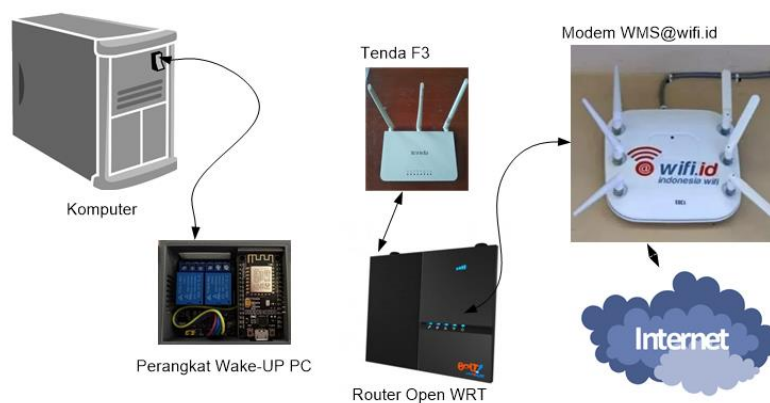
Pengendali *wake-on PC* ini menggunakan mikrokontroler ESP8266 yang dapat terhubung dengan jaringan *wifi*. Sumber internet pada jaringan di masjid ini adalah layanan WMS dari PT Telkom, yang saat mulai berselancar internet terdapat *landing page / captive portal* untuk *authentication* pengguna. Layanan *landing page* PT Telkom ini dapat dilihat pada Gambar 3.a. Namun karena ESP8266 tidak mempunyai *web browser*, maka ESP8266 ini tidak dapat terkoneksi internet. Untuk mengatasi hal ini, maka dibutuhkan sebuah perangkat *router* BL 100 yang telah di-*install* OpenWRT.

*Router* dengan OpenWRT ini dapat mengatasi *landing page* ini dengan *script* untuk *auto login* WMS, web openWRT dapat dilihat pada Gambar 3.b. Agar jaringan internet dapat dikoneksikan ke perangkat WIFI dibutuhkan sebuah aksespoint (AP) Tenda F3, sehingga

perangkat ESP8266 dapat terhubung internet. Denah jaringan untuk implementasi sistem dapat dilihat pada Gambar 4.



Gambar 3. Login Page dan web OpenWrt



Gambar 4. Denah jaringan pada Sistem

## METODE PENELITIAN

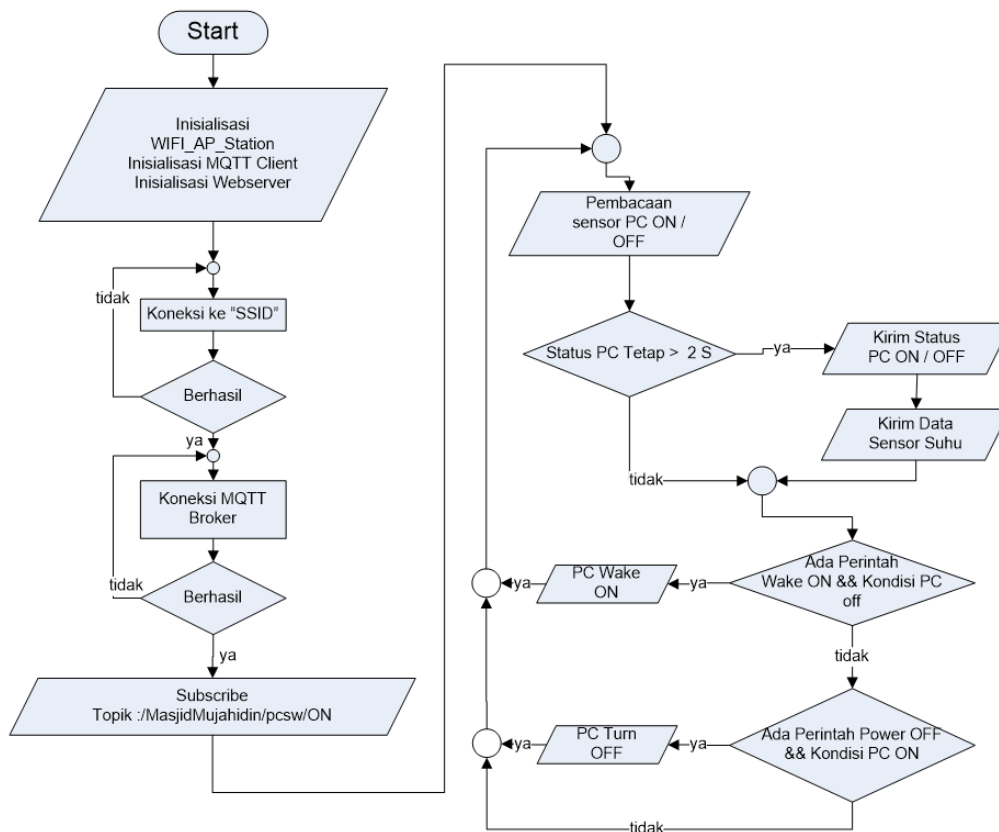
Metode yang dilakukan dalam penelitian ini adalah metode *research and development*. Peneliti membuat sebuah sistem yang berfungsi untuk menghidupkan PC melalui protokol MQTT. Prosesnya antara lain pembuatan program peralatan *wake-on* PC, pembuatan aplikasi android remote *wake-on* PC masjid dan perancangan rangkaian elektronik yang berupa mikrokontroller ESP8266, sensor-sensor LDR, DHT 22 dan rangkaian *relay*. Pengujian terdiri dari pengujian *hardware* dan pengujian *software*. Pengujian *hardware* meliputi pengujian

sensor LDR dan DHT 22, sedangkan pengujian *software* meliputi komunikasi antara perangkat dengan aplikasi remote *wake-on PC* masjid.

## HASIL DAN PEMBAHASAN

### Program Perangkat *wake-on PC*

ESP8266 yang merupakan otak dari perangkat *wake-on PC*, diprogram dengan aplikasi Arduino IDE. Program ini membutuhkan beberapa *library* antara lain ESP8266WiFi dan PubSubClient. *Library* ESP8266WiFi digunakan untuk mengaktifkan dan mengolah fitur *wifi* dalam mode *station*, sedangkan PubSubClient bertugas untuk melayani protokol MQTT. Alur program / *flowchart* perangkat dapat dilihat pada Gambar 5.



Gambar 5 . Flowcart program ESP8266

Saat mulai, ESP8266 bekerja dengan mode *station*, ESP8266 ini akan terhubung ke *router* Tenda F03. Selanjutnya akan terhubung ke MQTT broker, dan *subscribe* ke topik `"/MasjidMujahidin/pcsw/ON"`. Kemudian, ESP8266 akan *looping* untuk membaca kondisi masing-masing sensor dan menunggu perintah *wake-on* atau *turn-off*. Data sensor akan dikirim setiap 2 detik sekali ke jaringan MQTT. Jika ada data dari topik `"/MasjidMujahidin/pcsw/ON"` maka *relay* akan aktif untuk menghidupkan PC. Potongan program untuk menerima data dari topik `"/MasjidMujahidin/pcsw/ON"` ini dapat dilihat pada Gambar 6.

```

void MQTT_Callback(char* topic, byte * payload, unsigned int length)
{
  Serial.println(topic);
  char buff1[200];
  int buff2;
  memcpy(buff1, (char *)payload, length);
  Serial.println(buff1);

  if (strcmp(TOPIC_WAKE_ON, topic) == 0)
  {
    if (sscanf(buff1, "on=%d", &buff2) == 1)
    {
      if (buff2 == 90)
        PC_Wake_On();
    }
  }
}

```

Gambar 6. Potongan program untuk menampung / menerima kiriman topik

### Aplikasi Android “Remote Wake-On PC Masjid Mujahidin”

Beberapa perangkat yang harus tersedia untuk komunikasi MQTT adalah MQTT *broker* dan MQTT *client*. MQTT *Broker* yang digunakan untuk perangkat *wake-on* PC beralamatkan pada “broker.hivemq.com” [9] dan MQTT *client* adalah *smartphone* dan perangkat *wake-on* masjid. Agar dapat bekerja dengan baik sebagai MQTT *client*, maka topik yang akan digunakan harus sama, sehingga pesan yang terkirimkan dapat tersampaikan.

Topik yang digunakan untuk menghidupkan PC adalah “/MasjidMujahidin/pcsw/ON” dengan *payload* “on=90”, sedangkan topik untuk mengetahui kondisi PC dan sensor-sensor dari ESP8266 adalah “/MasjidMujahidin/pcsw/sensor”. Urutan data dari topik sensor ini adalah “Tegangan, Suhu, Kelembapan, Kondisi”. Kondisi ini adalah status PC apakah hidup atau mati, tegangan ini adalah tegangan kerja dari ESP8266, kemudian suhu dan kelembapan ruangan tempat PC berada.

Berikut adalah gambaran urutan komunikasi MQTT untuk aplikasi android, antara lain:

Proses ke 1:

Modul ESP8266 melakukan *subscribe* dengan topik “/MasjidMujahidin/pcsw/ON”. Sedangkan *smartphone* melakukan *subscribe* dengan topik “/MasjidMujahidin/pcsw/sensor”.

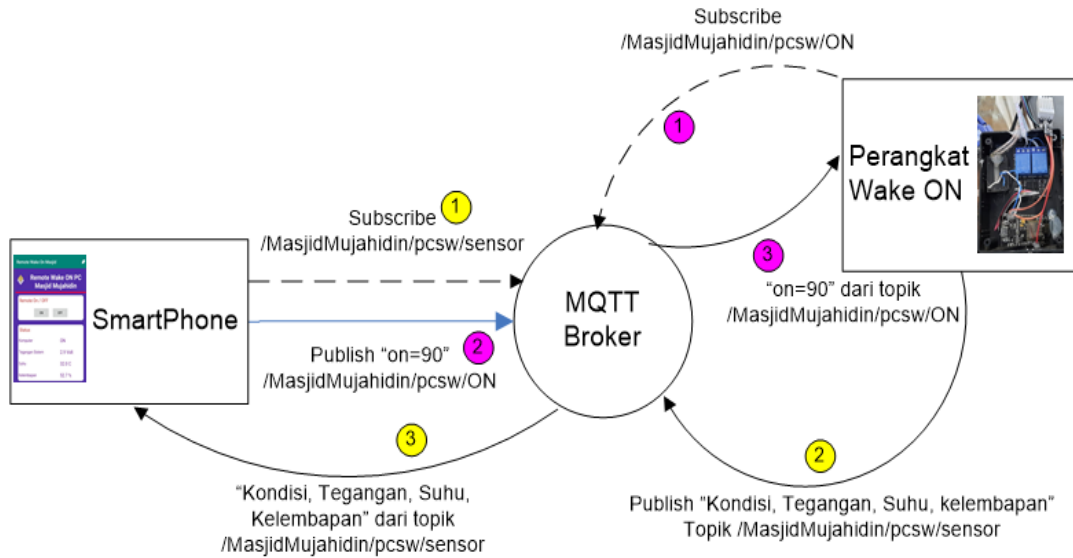
Proses ke 2:

Saat ada tombol ON pada *smartphone* ditekan, maka *smartphone* tersebut mengirimkan topik “/MasjidMujahidin/pcsw/ON” dengan *payload* “on=90” ke MQTT *broker*. Perangkat *wake-on* mengirimkan data “Kondisi, Tegangan, Suhu, dan Kelembapan” setiap 2 detik sekali dengan topik “/MasjidMujahidin/pcsw/sensor” ke MQTT *broker*.

Proses ke 3:

MQTT *broker* mengirimkan *payload* “on=90” ke perangkat yang sudah *subscribe* topik “/MasjidMujahidin/pcsw/ON”. Jika data yang diterima benar maka perangkat *wake-on* akan menghidupkan PC. MQTT *broker* juga mengirimkan data “Kondisi, Tegangan, Suhu, dan Kelembapan” kepada *smartphone*. Urutan dan proses komunikasi MQTT untuk aplikasi ini dapat dilihat pada Gambar 7.

Aplikasi ini dibuat pada Android Studio dengan bahasa Java. Aplikasi ini terdiri dari tombol *on* dan *off* serta keterangan status PC, tegangan, suhu dan kelembapan. Saat tombol *on* ditekan, maka Android akan *publish* topik “/MasjidMujahidin/pcsw/ON dengan *payload* ”on=90”. Potongan program untuk mengatasi tombol *on* dapat dilihat pada Gambar 8. Kemudian saat mendapat kiriman dari topik yang sudah *subscribe* “/MasjidMujahidin/pcsw/sensor”, maka tampilan pada aplikasi android menjadi seperti Gambar 9.



Gambar 7. Desain Komunikasi MQTT.

```

BtnON.setOnClickListener ({(view) -> {
    String topic = "/MasjidMujahidin/pcsw/ON";
    String message = "on=90"+ "\n";
    if (client.isConnected()) {
        try {
            client.publish(topic, message.getBytes(), qos: 0, retained: true);
            Toast.makeText(getApplicationContext(), text: "Published Message",
        } catch (MqttException e) {
            e.printStackTrace();
        }
    }
    else
        msg(S "MQTT Belum Konek");
});
    
```

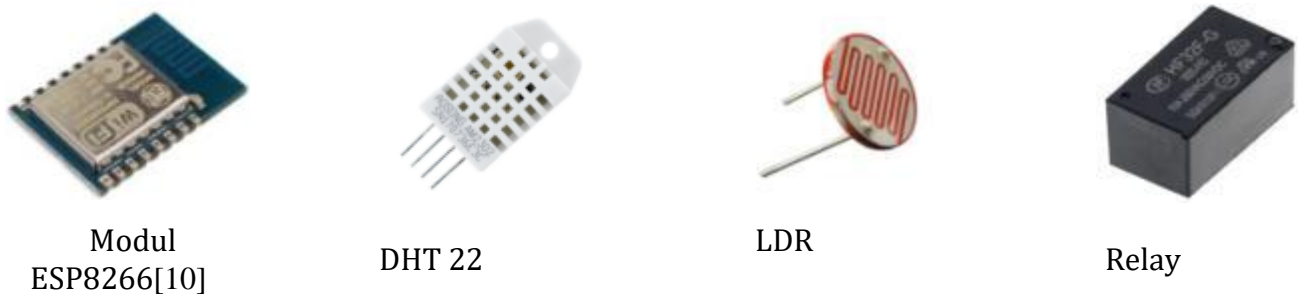
Gambar 8 . Potongan program untuk mem-publish topik “/MasjidMujahidin/pcsw/ON



Gambar 9. Tampilan aplikasi “Remote wake-on PC Masjid Mujahidin”

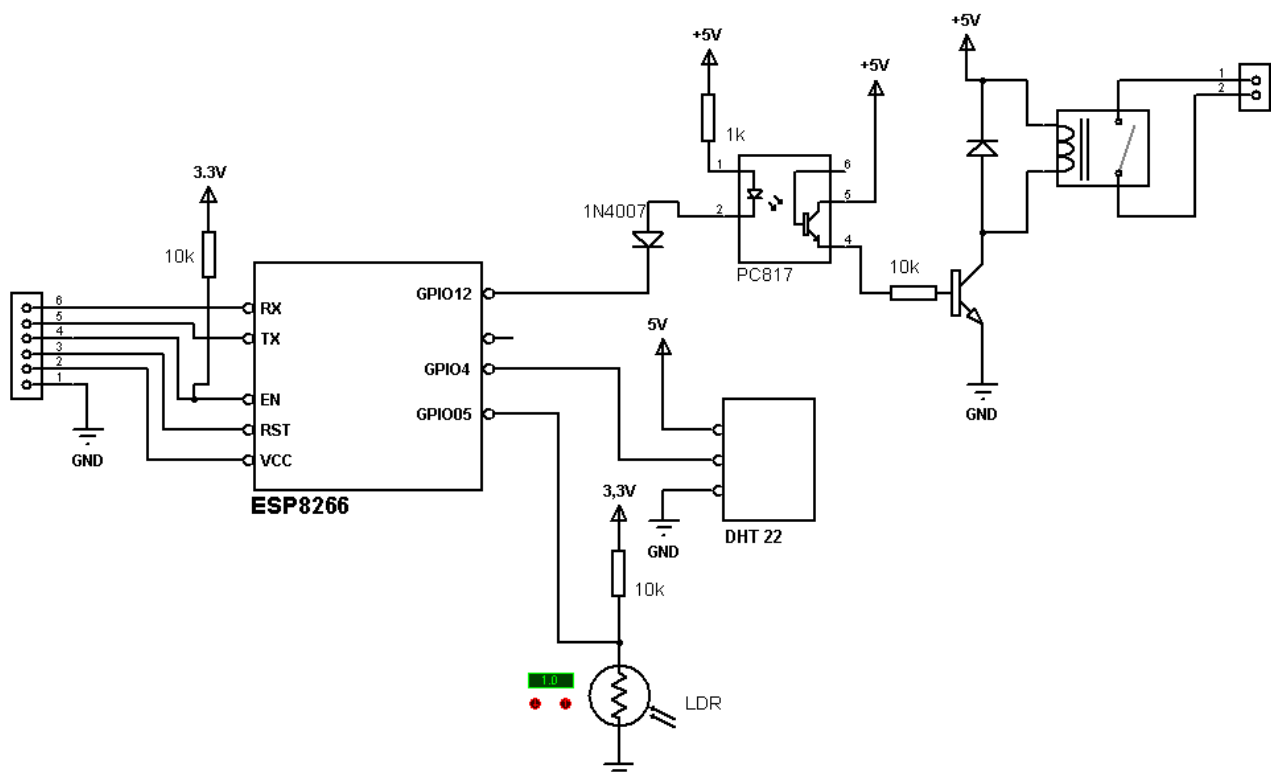
**Rangkaian Perangkat wake-on PC**

Rangkaian perangkat wake-on PC ini terdiri dari mikrokontroller ESP8266, rangkaian relay, DHT 22, rangkaian LDR dan power supply. Gambar masing- masing komponen untuk rangkaian ini dapat dilihat pada Gambar 10.

Gambar 10. Komponen Perangkat *wake-on PC*

Mikrokontroler ESP8266 merupakan pusat pengendali dari perangkat *wake-on PC* ini, Mikrokontroler ini bertugas untuk menerima dan mengirimkan pesan berupa tegangan sistem, suhu, kelembapan dan status PC melalui protokol MQTT di jaringan internet. Jika ESP8266 menerima pesan untuk menghidupkan PC maka *relay* akan aktif. Rangkaian *relay* dihubungkan ke mikrokontroler melalui pin GPIO12. Saat pin GPIO12 berlogika 1 maka *relay* akan mati, sedangkan saat berlogika 0 maka *relay* akan hidup. Rangkaian *relay* ini terdiri dari *optocoupler* PC817, transistor BC550 dan resistor.

Rangkaian LDR digunakan untuk mengetahui apakah PC dalam keadaan hidup atau mati dengan cara membaca kondisi LED *power* pada PC. Rangkaian LDR ini terdiri dari resistor 10kOhm dan LDR. Semakin banyak cahaya yang masuk pada LDR maka nilai resistansi semakin berkurang, akibatnya tegangan rangkaian LDR semakin kecil. Begitu pula sebaliknya. Rangkaian LDR ini dihubungkan melalui pin GPIO05.



Gambar 11. Rangkaian keseluruhan

Sensor DHT22 digunakan untuk mengetahui suhu dan kelembapan ruangan, agar dapat mencegah kerusakan PC akibat suhu dan kelembapan yang tinggi. Sensor DHT22 dihubungkan dengan pin GPIO4.

*Power supply* yang digunakan adalah SMPS Tenstar Robot. Tegangan keluarannya adalah 5 volt, sehingga dibutuhkan IC regulator AMS1117 untuk menurunkan tegangan catu menjadi 3.3 volt sebagai catu daya ESP8266. Skema keseluruhan rangkaian perangkat *wake-on PC* dapat dilihat pada Gambar 11.

## Pembahasan

### Pengujian Sensor LDR (*Light Dependent Resistor*)

Pengujian ini bertujuan untuk mengetahui tegangan yang dihasilkan rangkaian LDR saat didekatkan dengan LED *power* pada PC (LED PC). Pengujian ini dilakukan dengan mengukur jarak antara LED PC dengan LDR serta tegangan yang dihasilkan oleh rangkaian LDR tersebut. Hasil pengujian sensor LDR ini dapat dilihat pada Tabel 1.

Tabel 1. Pengujian tegangan rangkaian LDR saat jarak tertentu.

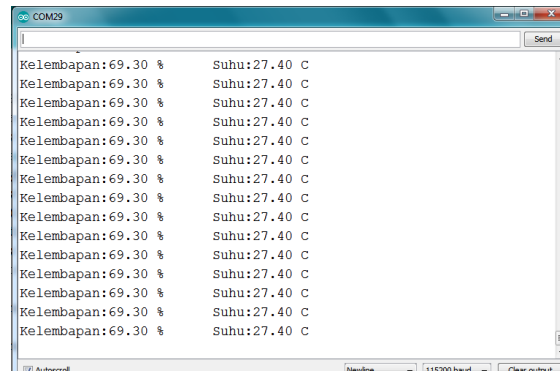
Warna : Merah (on)		Warna : Merah (Off)	
Jarak (cm)	Tegangan (V)	Jarak (cm)	Tegangan (V)
1	0,3	1	2,8
2	0,4	2	2,8
3	0,9	3	2,8
4	1	4	2,8
5	1,5	5	2,8
6	1,8	6	2,8
7	2,0	7	2,8
8	2,5	8	2,8
9	2,8	9	2,8
10	2,8	10	2,8
12	2,8	12	2,8
15	2,8	15	2,8
17	2,8	17	2,8

Jarak led dengan rangkaian LDR dapat mempengaruhi besaran tegangan keluaran. Semakin dekat LDR dengan LED PC maka tegangannya akan semakin kecil, yang artinya semakin mendekati 0 volt. Dan jika LED PC padam maka tegangan keluaran akan bernilai 2,8 Volt.

Dari lembar data, diketahui bahwa VIL (*voltage input low*) dari ESP8266 adalah 0 - 0,82 volt[10], sehingga tegangan yang memenuhi untuk kondisi ini adalah tegangan dengan jarak LED dan LDR dibawah 2 cm. Namun untuk meningkatkan pembacaan sensor LDR tersebut maka LED PC dengan LDR dapat diberi solasi agar tidak terjadi pergerakan dan tidak ada gangguan cahaya dari luar.

### Pengujian pembacaan sensor DHT 22 oleh ESP8266

Pengujian ini bertujuan untuk membuktikan bahwa sensor DHT 22 dapat dibaca oleh ESP8266 dan ditampilkan pada serial monitoring. Menurut lembar data[11], sensor DHT 22 ini dapat membaca suhu dan kelembapan dengan tingkat akurasi sebesar 2 % - 5 % kelembapan dan dibawah 0,5° celcius untuk suhu. Pengujian ini dilakukan dengan membandingkan pembacaan sensor yang dihasilkan dari ESP8266 dengan Thermohyrometer Standar. Terlihat pada Gambar 12, data suhu dan kelembapan sensor DHT22 dapat dibaca ESP8266 dan ditampilkan pada serial monitor Arduino IDE.



Gambar 12. Suhu dan Kelembapan tampilan dari serial Monitor

Tabel 2. Pengukuran suhu dan kelembaban terhadap waktu pada suhu 24° celcius

Menit	Suhu		Kelembapan	
	Standar	DHT 22	Standar	DHT
2	24,8	23,9	49	51,5
4	24,8	23,9	49	51,3
6	24,7	23,9	49	50,8
8	24,7	23,9	49	50,7
10	24,8	23,7	48	50,8

Tabel 3. Pengukuran suhu dan kelembaban terhadap waktu pada suhu 27° celcius

Menit	Suhu		Kelembapan	
	Standar	DHT 22	Standar	DHT
2	27,7	26,7	50,5	53,4
4	27,7	26,7	52,8	54,8
6	27,5	26,4	54,2	54,4
8	27,6	26,5	52,7	53,3
10	27,9	26,4	52,5	53,3

Data pengujian pada Tabel 2 dan 3 dilakukan pada suhu 24 dan 27° celcius. Data yang diambil berasal dari data Thermohygrometer dan dari tampilan serial monitoring, setiap 2 menit. Nampak pada data pengujian bahwa hasil perbandingan pembacaan suhu dan kelembapan yang berasal dari serial monitor dengan pembacaan Thermohygrometer mempunyai tingkat akurasi sebesar 2 % -5 % kelembapan dan 0,5° celcius untuk suhu, yang mana hasil ini sesuai dengan lembar data DHT22. Dari pengujian ini dapat disimpulkan bahwa sensor DHT 22 dapat dibaca dan ditampilkan pada serial monitoring oleh ESP8266.

### Pengujian Pengiriman Data dari Perangkat *wake-on* PC ke Aplikasi Android

Pengujian ini bertujuan untuk mengetahui apakah data sensor yang dikirimkan melalui jaringan MQTT dapat diterima dengan baik oleh aplikasi *remote wake-on* PC Masjid Mujahidin. Pertama, pengujian dilakukan dengan menggunakan data *payload* yang berbeda-beda seperti pada Tabel 4. Pengujian ini bertujuan untuk mengetahui format pengiriman data yang dapat diterima oleh aplikasi *remote wake-on*.

Tabel 4. Pengujian pengiriman data ke aplikasi android

No	Data dari perangkat <i>wake-on</i> PC		Tampilan aplikasi <i>remote wake-on</i>			
	<i>Payload</i>	Diterima	Tegangan	Suhu	Kelembapan	Kondisi PC
1	sensor=2.9:26.5:72.7:1	ya	2.9	26.5	72.7	ON
2	Sensor-2.9-25-56-0	tidak	-	-	-	-
3	s=2.9:26.5:72.7:1	tidak	-	-	-	-
4	sensor=2.9:26.5:72.7:1:1	tidak	-	-	-	-
5	sensor=2.9:25.0:56.2:0	ya	2.9	25.0	56.2	OFF
6	sensor=2.9-26.5-72.7-1	tidak	-	-	-	-
7	sensor=2.9/26.5/72.7/0	tidak	-	-	-	-
8	sensor=2.9:25.6:69.1:0	ya	2.9	25.6	69.1	OFF
9	Sensor:2.9:25:56:0	tidak	-	-	-	-
10	sensor=2.9:26.5:72.7:1:1	tidak	-	-	-	-

Dari hasil pengujian yang dilakukan dapat disimpulkan bahwa data yang dapat diterima dengan baik adalah data dengan format "sensor=x.y:vv.cc:zz.bb:a". Kata "sensor" diikuti dengan tanda "=", kemudian angka pecahan, dipisahkan dengan tanda ":". Format pengiriman dari perangkat *wake-on* PC ini harus mengikuti susunan seperti yang dijelaskan diatas. Sedangkan untuk data yang tidak diterima, data tersebut tidak diproses dan tampilan aplikasi *remote wake-on* akan tetap menampilkan data yang diterima sebelumnya. Potongan program untuk membaca urutan data "x.y:vv.cc:zz.bb:a" yang dilakukan oleh aplikasi *remote wake-on* dapat dilihat pada Gambar 13.

```

if (topic.equals("/MasjidMujahidin/pcsw/sensor")) {
    msg =new String(message.getPayload());

    TvKeterangan.setText(msg);
    Scanner s = new Scanner(msg);
    s.findInLine(pattern:"sensor=(\\w+).(\\w+):(\\w+).(\\w+):(\\w+):(\\w+):(\\w+)");

    try {
        MatchResult data = s.match();
        tvVolt.setText(data.group(1)+"."+data.group(2)+" Volt");
        tvSuhu.setText(data.group(3)+"."+data.group(4)+" C");
        tvKelembapan.setText(data.group(5)+"."+data.group(6)+" %");
        if (data.group(7).equals("1"))
            msg= "ON";
        else
            msg= "OFF";
        tvStatus.setText(msg);
    }
}

```

Gambar 13 . Potongan program untuk membacaan urutan data sensor

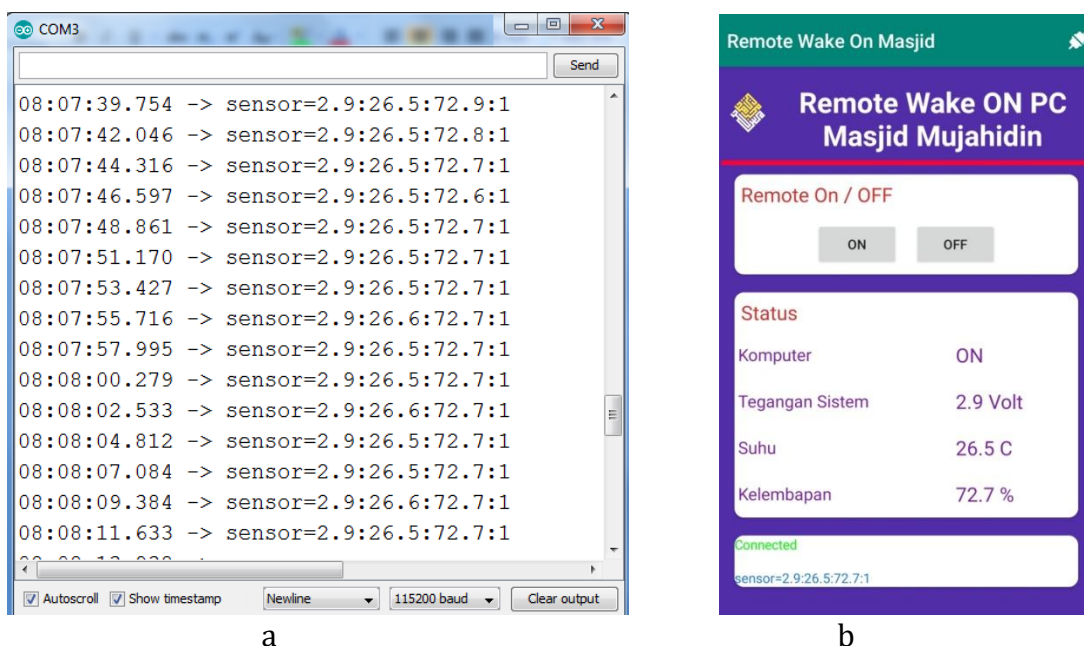
Kedua, pengujian dilakukan dengan menggunakan topik yang berbeda-beda, namun dengan data *payload* yang sama yaitu "sensor=2.9:26.5:72.7:1". Pengujian ini digunakan untuk mengetahui apakah topik lain selain topik "/MasjidMujahidin/pcsw/sensor" dapat mempengaruhi data-data sensor yang dikirimkan oleh perangkat *wake-on*. Tabel pengujian ini dapat dilihat pada Tabel 5.

Tabel 5. Pengujian pengiriman data ke perangkat wake-on PC

No	Topik	Diterima
1	/MasjidMujahidin/pcsw/sensor	ya
2	/MasjidMujahidin/pcsw/	tidak
3	/MasjidMujahidin/sensor	tidak
4	/Masjid/pcsw/sensor	tidak
5	/MasjidMujahidin/pcsw/sensor	ya
6	/masjidmujahidin/pcsw/sensor	tidak
7	MasjidMujahidin/pcsw/sensor	tidak
8	/MasjidMujahidin/pcsw/sensor	tidak
9	-MasjidMujahidin-pcsw-sensor	tidak
10	MasjidMujahidin/Pcsw/Sensor	tidak

Pengujian ini dapat disimpulkan bahwa topik yang dapat diterima oleh aplikasi *remote wake-on* adalah /MasjidMujahidin/pcsw/sensor, sedangkan topik-topik selain itu tidak diterima.

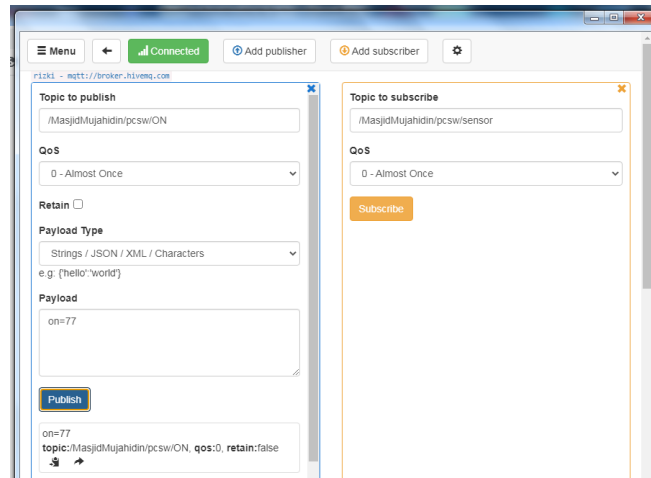
Data sensor yang dikirimkan oleh perangkat *wake-on* PC dapat dilihat pada serial monitoring Gambar 14a. Gambar 14b merupakan aplikasi android saat menerima topik "/MasjidMujahidin/pcsw/sensor" dengan *payload* "sensor=2.9:26.5:72.7:1".

Gambar 14. Gambar Serial Monitoring dan Aplikasi Remote *wake-on* PC

### Pengujian Pengiriman Data dari Aplikasi Remote *wake-on* ke Perangkat *wake-on* PC

Pengujian ini bertujuan untuk mengetahui apakah data yang akan dikirimkan oleh aplikasi *remote wake-on* dapat diterima oleh perangkat *wake-on* PC. Namun sebelum pengujian dengan aplikasi *remote wake-on*, peneliti menggunakan aplikasi MQTT Box untuk menguji beberapa topik dan data *payload* yang berbeda. Aplikasi MQTT Box dapat dilihat pada Gambar 15, sedangkan data pengujiannya dapat dilihat pada Tabel 6. Tabel 6a berisi data pengujian dengan topik "/MasjidMujahidin/pcsw/ON" dengan *payload* yang berbeda-beda,

sedangkan Tabel 6b berisi pengujian dengan *payload* “on=90” dengan topik yang berbeda-beda.

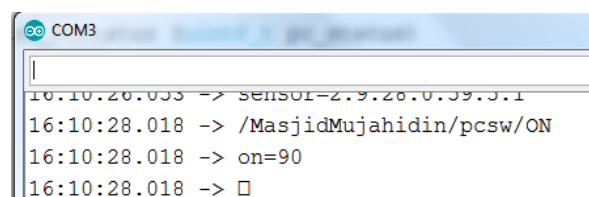


Gambar 15. Gambar Aplikasi MQTT Box.

Tabel 6. Pengujian pengiriman data ke perangkat wake-on PC

a.			b.		
Topik "/MasjidMujahidin/pcsw/ON"			Data Payload "on=90"		
No	Payload	Kondisi Relay	No	Topik	Kondisi Relay
1	on=90	Hidup	1	/MasjidMujahidin/pcsw/	Mati
2	on=94	Mati	2	/MasjidMujahidin/pcsw/ON	Hidup
3	on=23	Mati	3	/MasjidMujahidin/pcsw/oN	Mati
4	off=90	Mati	4	/masjidmujahidin/pcsw/ON	Mati
5	on=50	Mati	5	/MasjidMujahidin/	Mati
6	on=40	Mati	6	/MasjidMujahidin/PCSW/ON	Mati
7	on=10	Mati	7	/MasjidMujahidin/ON	Mati
8	on=20	Mati	8	MasjidMujahidin-ON	Mati
9	on=90	Hidup	9	/MasjidMujahidin/pcsw/ON	Hidup
10	off=100	Mati	10	-MasjidMujahidin-pcsw-ON	Mati

Dari data hasil pengujian dengan aplikasi MQTT box dapat disimpulkan bahwa data yang dapat menghidupkan *relay* adalah topik "/MasjidMujahidin/pcsw/ON" dengan *payload* "on=90". Sehingga jika ingin menghidupkan *relay* menggunakan aplikasi *remote wake-on*, maka saat tombol "ON" ditekan, *smartphone* harus mengirimkan topik "/MasjidMujahidin/pcsw/ON" dengan *payload* "on=90". Data pengiriman *smartphone* yang diterima oleh perangkat, ditampilkan pada serial monitoring seperti pada Gambar 16.



Gambar 16. Gambar serial monitoring saat tombol On ditekan.

## SIMPULAN DAN SARAN

### Simpulan

Seluruh sistem yang berfungsi untuk menghidupkan PC melalui protokol MQTT dapat bekerja dengan baik. Perangkat *wake-on* PC telah dapat bekerja dengan baik dengan jarak efektif antara LDR dengan LED *power* dibawah 2 cm. Data sensor yang berupa tegangan, suhu, kelembapan dan status PC dapat dikirimkan dari perangkat *wake-on* PC melalui jaringan MQTT ke aplikasi *remote wake-on* PC dan berhasil diterima dengan baik. Data sensor dikirimkan dengan topik `"/MasjidMujahidin/pcsw/sensor"` dan format data *payload* adalah `"sensor=x.y:vv.cc:zz.bb:a"`. Perintah yang dikirimkan dari aplikasi *remote wake-on* PC dapat diterima oleh perangkat *wake-on* PC dan rangkaian *relay* dapat bekerja sehingga PC akan hidup. Perintah ini dikirim melalui topik `"/MasjidMujahidin/pcsw/ON"` dengan *payload* `"on=90"`.

### Saran

Untuk pengembangan lebih lanjut, router OpenWRT dapat ditiadakan, diganti dengan *auto login* WMS yang dipasang pada ESP8266. Penggunaan LDR dapat diganti dengan *optocoupler* yang masukannya langsung dari terminal LED *power* pada PC.

## DAFTAR RUJUKAN

- [1] Mannuhunung S. Tenrigan A.M dan Didiharyono D.. "Manajemen Pengelolaan Masjid dan Remaja Masjid di Kota Palopo". Jurnal Pengabdian Masyarakat. 1 (1) : 14-21, 2018
- [2] Q. Shobikhul, "MANAJEMEN FASILITAS MULTIMEDIA MASJID AR-RAHMAH PERAK SURABAYA", Jurnal Masjiduna : Jurnal Ilmiah Stidki ar-Rahmah, Vol 3, no 2, pp 63-81, 2020
- [3] N. Aathmika, R. Shanta "Review of Existing Remote Desktop Protocols", International Research Journal of Engineering and Technology (IRJET) Volume: 07 Issue: 04 Apr 2020
- [4] Tran Hoang Vu. "Measure and Evaluate Delay time for Wakeup on Lan (Wol) Method of OpenFlow Switch", Journal of Electrical and Electronic Engineering. Vol. 4, No. 3, 2016, pp. 68-72. doi: 10.11648/j.jee.20160403.15
- [5] AMD, "Magic Packet Technology," November 1995. [Online]. Available: <http://support.amd.com/TechDocs/20213.pdf>.
- [6] S. Gencho, G. Zhelyan , "Server for Remote Generation of Wake On LAN Packets in the LAN " International Journal of Recent Development in Engineering and Technology Website: [www.ijrdet.com](http://www.ijrdet.com) (ISSN 2347 - 6435 (Online)) Volume 3, Issue 2, August 2014)
- [7] J. A. Gil-Martinez-Abarca, F. Macia-Perez, D. Marcos-Jorquera and V. Gilart-Iglesias, "Wake on LAN over Internet as Web Service," 2006 IEEE Conference on Emerging Technologies and Factory Automation, 2006, pp. 1261-1268, doi: 10.1109/ETFA.2006.355397.
- [8] Asrock, Motherboard Asrock B360M User Manual, Maret 2018
- [9] Tegrif, MQTT-broker Documentation, April 2016
- [10] Espressif Systems, ESP8266EX Datasheet Version 6.6 , 2020
- [11] Aosong Electronics Co.,Ltd, Digital-output relative humidity & temperature sensor/module DHT22 Datasheet