

Rancang Bangun *Prototype* Sistem Kendali Atap *Smart Mosque* Berbasis *Android*

Rizky Maulana¹, Muhammad Nasir^{2*}, Anwar³

^{1,2,3} Jurusan Tekniknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe
Jln. B.Aceh Medan Km.280 Buketrata 24301 INDONESIA

¹maulanar684@gmail.com

^{2*}muhnasir.tmj@pnl.ac.id

³anwarsy@pnl.ac.id

Abstrak— Pengembangan teknologi telah banyak memberikan kontribusi penting untuk manusia dan menghemat energi, dengan seiringnya perkembangan teknologi sehingga membuat masjid Pintar, salah satu permasalahan tersebut adalah bagaimana masjid tetap nyaman tanpa menjadi lembab dan mendapatkan cahaya matahari yang cukup, pada penelitian ini bertujuan untuk membuka atap otomatis yang dapat memberikan penyediaan sinar matahari kedalam ruangan masjid. Alat ini menggunakan pengendali otomatis dengan *Sensor* cahaya yang diaktifkan secara otomatis bila cahaya matahari mengenai *Sensor*, dan atap ini juga otomatis menutup, apabila cahaya tidak mengenai *Sensor* cahaya maka otomatis atap tertutup. Hasil pengujian system secara keseluruhan adalah pembuka atap masjid mampu secara otomatis membuka dan menutup atap masjid, dan user/pemegang kendali masjid dapat juga mengendalikan atap masjid dengan menyentuh buka atau tutup berdasarkan keseluruhan data rekapitulasi parameter QOS rancang bangun *Prototype* sistem kendali atap *Smart Mosque* berbasis *Android* dimulai pada jam 15.00 pagi sampai dengan jam 19.35 sore hari dengan menggunakan metode Quality of service (QOS) dan mendapatkan keseluruhan data persentase dari hasil pengukuran beberapa parameter dari metode tersebut dengan total persentase 97,625.

Kata kunci— Otomasi, IoT, Wemos, Arduino Uno, Light Dependent Resistor, Raindrop, DC, Blynk, Smartphone.

Abstract— Technological development has made many important contributions to humans and conserves energy, along with the development of technology so as to make Smart mosques, one of these problems is how mosques stay comfortable without getting damp and get enough sunlight, in this study aims to open automatic roofs that can provide sunlight into the mosque room. This tool uses an automatic control with a light sensor which is activated automatically when the sun's light hits the sensor, and the roof also automatically closes, if the light does not hit the light sensor, the roof will automatically close. The results of the overall system test are that the mosque roof opener is able to automatically open and close the mosque roof, and the mosque user / control holder can also control the mosque roof by touching open or close based on the overall QOS parameter recapitulation data. starting at 15.00 in the morning until 19.35 in the afternoon using the Quality of Service (QOS) method and getting the overall percentage data from the measurement results of several parameters of the method with a total percentage of 97.625.

Keywords— Otomasi, IoT, Wemos, Arduino Uno, Light Dependent Resistor, Raindrop, DC, Blynk, Smartphone.

I. PENDAHULUAN

Perkembangan Zaman di dunia teknologi, beberapa tahun terakhir ini berperan penting dalam kehidupan modern. Sistem otomatis mampu mempersingkat proses dan memberikan tingkat akurasi tinggi. Pengembangan sistem Smart Mosque merupakan salah satu aplikasi otomasi pada kehidupan sehari-hari. Pengembangan sistem Smart Mosque awalnya cukup jarang digunakan dan hanya dimiliki oleh masjid-masjid tertentu. Untuk mempermudah pekerjaan pengurus masjid sehingga menimbulkan ide untuk membuat sistem Smart Mosque [1].

Sistem Smart Mosque sudah banyak dikembangkan dengan berbagai macam otomasi, contohnya untuk aplikasi otomasi kunci Pintu, sistem kendali gerbang, pendeteksi api, pendeteksi alarm, penerangan lampu masjid dan lain sebagainya. Sistem Smart Mosque dikendalikan dengan menggunakan Android melalui perantara Blynk dan Wemos. Wemos dipilih sebagai Mikrokontroler dengan pertimbangan harga yang terjangkau.

Aplikasi pada Smartphone memecahkan permasalahan biaya untuk membuat notifikasi pemberitahuan dan bahwa atap sudah terbuka atau tertutup tersendiri terhadap sistem Smart Mosque, sehingga sistem Smart Mosque dapat tercapai. Salah satu otomasi yang diperlukan untuk menambah kenyamanan mesjid adalah otomasi pada bagian atap sehingga dapat membuat seisi ruangan masjid mendapatkan cahaya yang cukup [2].

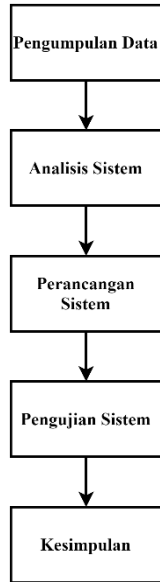
, pada penelitian ini bertujuan untuk membuka atap otomatis yang dapat memberikan penyediaan sinar matahari kedalam ruangan masjid. Alat ini menggunakan pengendali otomatis dengan Sensor cahaya yang diaktifkan secara otomatis bila cahaya matahari mengenai Sensor, dan atap ini juga otomatis menutup, apabila cahaya tidak mengenai Sensor cahaya maka otomatis atap tertutup. Hasil pengujian system secara keseluruhan adalah pembuka atap masjid mampu secara otomatis membuka dan menutup atap masjid, dan user/pemegang kendali masjid dapat juga mengendalikan atap masjid dengan menyentuh buka atau tutup berdasarkan keseluruhan data rekapitulasi parameter QOS rancang bangun *Prototype* sistem kendali atap *Smart Mosque* berbasis *Android*

dimulai pada jam 15.00 pagi sampai dengan jam 19.35 sore hari dengan menggunakan metode Quality of service (QOS) dan mendapatkan keseluruhan data persentase dari hasil pengukuran beberapa parameter dari metode tersebut dengan total persentase 97,625[3].

METODOLOGI PENELITIAN

A. Tahapan Penelitian

Tahapan metodologi yang digunakan dalam penelitian ini adalah pengumpulan data, analisis sistem, perancangan perancangan sistem, pengujian sistem, dan kesimpulan. Adapun blok diagram metodologi penelitian dapat dilihat pada gambar 1 berikut.



Gambar 1. Blok diagram metodologi penelitian

Pada gambar 1 tahapan dari penelitian yang pertama kali adalah pengumpulan data yang dilakukan untuk mengumpulkan data dari berbagai sumber, melakukan analisis kebutuhan pada data yang dibutuhkan dalam pembuatan sistem monitoring Smartmosque.

B. Pengumpulan Data

Pada tahap ini pengumpulan data dilakukan untuk mencari berbagai macam literatur seperti jurnal, buku perpustakaan maupun internet dan sumber-sumber lainnya yang terkait dengan penelitian yang akan dibuat. Data yang telah dikumpulkan dapat dijadikan sebagai referensi ketika muncul kendala pada perancangan, sehingga dengan adanya berbagai data dapat memudahkan tahap perancangan yang akan dibuat.

C. Analisis Sistem

Analisis sistem dilakukan untuk memberikan arahan dan menentukan tahap proses pengerjaan selanjutnya dalam hal penentuan kebijakan. Analisis sistem dilakukan dengan tahap sebagai berikut:

1) Analisis Data

Tahapan analisis data akan di dapatkan setelah melakukan pengujian terhadap sistem, adapun tahapan-tahapan analisis data sebagai berikut[4] :

- Melakukan pengujian terhadap alat yang dibuat untuk mendapat data QOS, keberhasilan alat saat proses *monitoring*, dan konfigurasi dari alat yang dibuat.

- Menganalisis semua data yang telah didapatkan dari hasil pengujian.
- Menganalisis perhitungan persentase *error*

2) Analisis kebutuhan

Analisis kebutuhan adalah kebutuhan dalam membuat sistem terhadap perangkat keras dan perangkat lunak yang dibutuhkan dalam membangun sistem yang dibuat.

Kebutuhan perangkat keras dalam penyusunan tugas akhir ini adalah sebagai berikut :

1. Laptop/Komputer
2. NodeMCU
3. Sensor *LDR*
4. Sensor Raindrop
5. *Motor DC*
6. Lampu LED

Kebutuhan perangkat lunak yang dibutuhkan untuk perancangan aplikasi klasifikasi jenis kulit wajah adalah sebagai berikut:

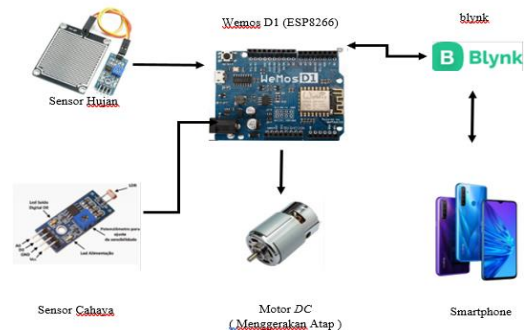
1. Software Arduino IDE
2. Blynk

D. Perancangan Sistem

Perancangan sistem merupakan tahapan yang akan dilakukan untuk membuat sistem *monitoring* Smartmosque Perancangan sistem ini dibagi dalam beberapa tahap yaitu : perancangan blok diagram sistem, *Hardware wiring* diagram dan perancangan *user interface* dan *flowchart* sistem.

1) Perancangan Blok Diagram Sistem

Blok diagram atau gambaran perancangan sistem dapat dilihat pada gambar 2.



Gambar 2. Blok diagram sistem

Berdasarkan gambar 2 Sistem yang akan dibuat adalah penerapan *IoT* pada atap masjid berbasis arduino . dalam *system* ini atap masjid akan otomatis terbuka atau tertutup pada cuaca hujan atap akan tertutup apabila cuaca cerah akan terbuka pada kondisi siang atap terbuka dan apabila hujan atap otomatis tertutup, untuk mengetahui atap sedang terbuka atau tertutup buka aplikasi *Blynk* lalu sentuh buak untuk membuka dan tutup untuk menutup atap.

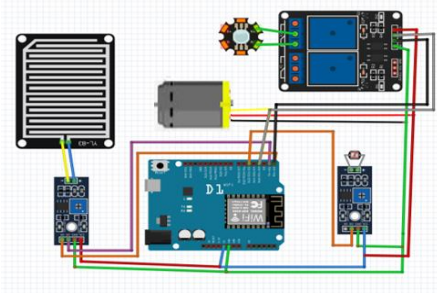
2) *Hardware Wiring* Diagram

Pada wiring diagram keseluruhan komponen merupakan gambaran letak-letak pin NodeMCU yang digunakan oleh semua komponen. Berikut tabel pin NodeMCU yang digunakan dapat dilihat ada tabel I.

TABEL I
PIN SISTEM

Pin	Relay	Sensor	Sens	Motor	Keterangan
Wemos		Raindrop	or LDR	DC	
5 V	VCC	VCC	VCC	-	Muatan +5V
Gnd	Ground	Ground	Ground	Kabel negative	Ground
D0	Inc	Tring Pin	-	Kabel positif	Output
D3	-	-	-	-	Tx Output
D5	-	Echo Pin	Tring Pin	-	Rx Input
D11	-	-	-	Tring Pin	

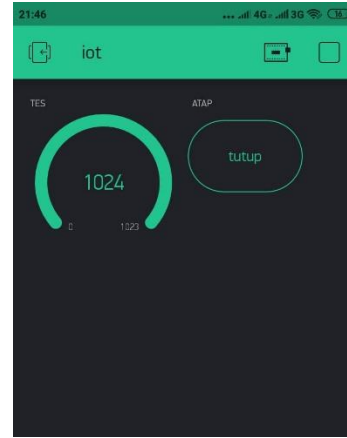
Dari tabel 1 dapat dilihat 2 pin NodeMCU yang digunakan untuk menghubungkan ke masing-masing komponen untuk mendapatkan input data dari komponen. Pada wiring diagram ini menggunakan NodeMCU, sensor LDR, Raindrop dan LED. Gambar wiring diagram dari sistem seluruh komponen dapat dilihat pada gambar 3.



Gambar 3. Wiring diagram sistem

3) User Interface

Dalam pembuatan sistem monitoring pemakaian Smartmosque ini terdapat interface yang berfungsi sebagai monitoring pada alat monitoring Smartmosque. tampilan user interface dapat dilihat pada gambar 4.



Gambar 4. Tampilan user interface

4) Flowchart Sistem

Dalam pembuatan sistem monitoring pemakaian air Smartmosque ini terdapat 1 sistem yang berjalan. Berikut ini flowchart dari sistem dapat dilihat pada gambar 5.

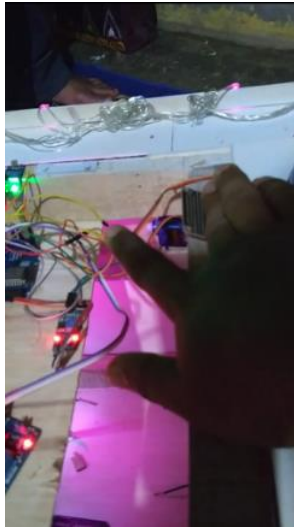


Gambar 5. Flowchart sistem

Pada gambar 5 dapat dilihat cara kerja sistem yang pertama adalah operator mengkoneksikan node mcu ke internet, kemudian node mcu akan mendeteksi air (hujan) pada sensor raindrop dan cahaya pada sensor ldr, node mcu juga akan mengirimkan data tersebut ke aplikasi blynk online secara berkala agar dapat dilihat dan dikontrol oleh operator

II. HASIL DAN PEMBAHASAN

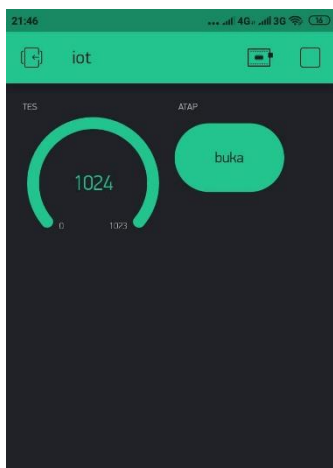
A. Implementasi Rangkaian Sistem *Monitoring* Pemakaian Smartmosque Implementasi rangkaian perangkat dilakukan untuk mengetahui apakah sistem memiliki kesalahan rangkaian atau tidak, tampilan rangkaian dapat dilihat pada gambar 6.



Gambar 6. Tampilan rangkaian sistem *monitoring* atap Smartmosque berbasis IoT

B. Implementasi Tampilan Halaman *Monitoring*
 1) Tampilan Halaman *Monitoring*

Halaman ini merupakan tampilan yang akan muncul ketika kita sudah membuat proyek baru dan mendesain tampilan *monitoring* pada Blynk. Halaman *monitoring* ini digunakan untuk menampilkan Tombol virtual, pengguna dapat mengetahui segala informasi atap terbuka atau tertutup melalui Blynk. Gambar halaman *interface* Blynk dapat dilihat pada gambar 7 berikut.

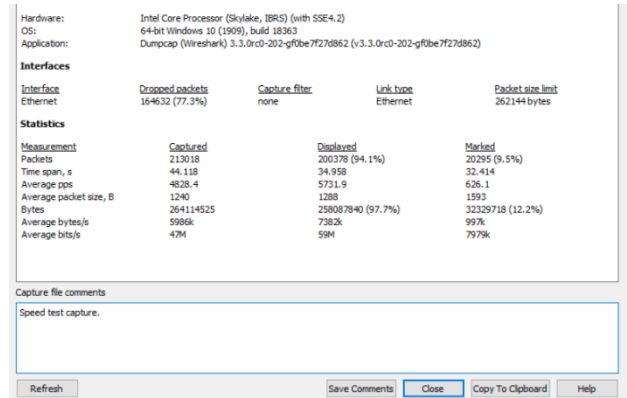


Gambar 7. Tampilan halaman *monitoring*

C. Data Hasil Pengujian

1. Pengujian Troughput

Throughput adalah bandwidth sebenarnya (actual) yang diukur dengan satuan waktu tertentu yang digunakan untuk melakukan transfer data dengan ukuran tertentu. Waktu download terbaik adalah ukuran file dibagi dengan bandwidth. Sedangkan waktu actual atau sebenarnya adalah ukuran file dibagi dengan throughput. Berikut pengujian Troughput dapat dilihat pada gambar 8 dibawah.



Gambar 8. Pengukuran parameter throughput

statistik dari komunikasi data yang digunakan, untuk mengukur throughput menggunakan persamaan dibawah ini :

Persamaan perhitungan throughput :

$$Throughput = \frac{\text{Paket data diterima}}{\text{Lama pengamatan}}$$

Berdasarkan persamaan diatas maka untuk mengukur throughput perhitungannya sebagai berikut :

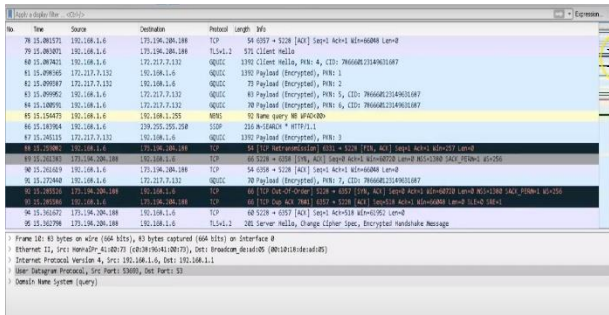
$$\begin{aligned} Throughput &= \frac{26411452 \text{ Bytes}}{44.118 \text{ s}} \\ &= 598654 \text{ Bytes/S} \\ &= 598654 \times 8 \\ &= 4.7 \text{ Mb/s} \end{aligned}$$

No	Time	Time 1	Time 2	Delay	Jitter
1	0	0	0.186654	0.186654	0.087238
2	0.186654	0.186654	0.626543	0.439889	0.000798
3	0.626543	0.626543	0.106642	0.004678	0.000358
4	0.1066702	0.106642	0.150673	0.087238	(0.057415)
5	0.1506726	0.150673	0.194675	0.000798	(0.100855)
6	0.1946750	0.194675	0.238677	0.000358	(0.144295)
7	0.2386774	0.238677	0.282680	0.334918	(0.187735)
8	0.2826798	0.282680	0.326682	0.417478	(0.231175)
9	0.3266822	0.326682	0.370685	0.500038	(0.274615)
10	0.3706846	0.370685	0.414687	0.582598	(0.318055)
11	0.4146870	0.414687	0.458689	0.665158	(0.361495)
12	0.4586894	0.458689	0.502692	0.747718	(0.404935)
13	0.5026918	0.502692	0.546694	0.830278	(0.448375)
14	0.5466942	0.546694	0.590697	0.912838	(0.491815)
15	0.5906966	0.590697	0.634699	0.995398	(0.535255)
16	0.6346990	0.634699	0.678701	1.077958	(0.578695)
17	0.6787014	0.678701	0.722704	1.160518	(0.622135)
18	0.7227038	0.722704	0.766706	1.243078	(0.665575)
19	0.7667062	0.766706	0.810709	1.325638	(0.709015)
20	0.8107086	0.810709	0.854711	1.408198	(0.752455)

Gambar 12. Tabel data hasil penujian delay dan jitter.

2. Pengujian Delay

Delay (Latency) merupakan waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. Delay dapat dipengaruhi oleh jarak, media fisik, congesti atau juga waktu proses yang lama, selanjutnya pengukuran delay menggunakan wireshark dapat dilihat pada gambar 13



Gambar 13 untuk mengukur delay

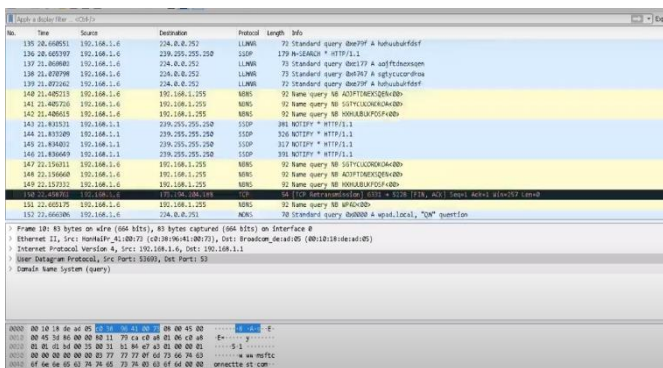
untuk mengukur delay langkah pertama adalah membuka wireshark kemudian pada bagian pencarian ketik tcp, kegunaan tcp adalah untuk memfilter delay tersebut, selanjutnya pilih file dan pilih export packet dissection untuk menjadikan data delay dalam bentuk excel supaya mudah dimengerti. maka untuk mengukur delay kita dapat menggunakan persamaan dibawah ini :

Berdasarkan persamaan diatas maka untuk mengukur delay perhitungannya sebagai berikut :

$$Delay = \frac{69823.155621 s}{213018} = 0.327780 s \times 1000 = 327.78 b/s$$

3. Pengujian Jitter

Jitter diakibatkan oleh variasi-variasi dalam panjang antrian, dalam waktu pengolahan data, dan juga dalam waktu penghimpunan ulang paket-paketdiakhir perjalanan jitter. Jitter lazimnya disebut variasi delay dan berhubungan erat dengan delay yang menunjukkan banyaknya variasi delay pada trasmisi data di jaringan. Pengukuran jitter menggunakan wireshark dapat dilihat pada gambar 14.



Gambar 14 Pengukuran jitter

Berdasarkan gambar 14 untuk mengukur jitter sebenarnya sama saja seperti mengukur delay hanya terdapat perbedaan pada

persamaan rumusnya saja untuk mengukur jitter kita dapat menggunakan persamaan dibawah ini :

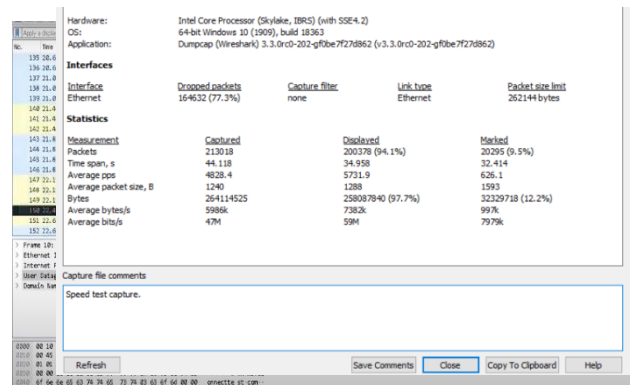
$$Jitter = \frac{\text{Total variasi delay}}{\text{Total paket yang diterima}}$$

D. Berdasarkan persamaan diatas maka untuk mengukur delay perhitungannya sebagai berikut :

$$E. Jitter = \frac{36,753.115 S}{213018} = 0.1725352 s \times 1000 = 172.5352 b/s$$

4 Pengujian Packet loss

Packet loss merupakan suatu parameter yang menggambarkan kondisi yang menunjukkan jumlah total paket yang hilang, hal ini dapat terjadi karena beberapa kemungkinan antara lain terjadinya overload didalam suatu jaringan, error yang terjadi pada sisi penerima antara lain bisa disebabkan karena router buffer over flow atau kemacetan, pengukuran packet loss menggunakan wireshark dapat dilihat pada gambar 15



Gambar 15 Pengukur packet loss

Berdasarkan gambar 15 untuk mengukur packet loss langkah pertama adalah membuka wireshark selanjutnya pada bagian pencarian tulis tcp.analysis.lost_segment kemudian pada bagian statistisc terdapat packet captured dan displayed, maka untuk mengukur packet loss kita dapat menggunakan persamaan dibawah ini :

$$Packet\ loss = \frac{(\text{Paket data dikirim} - \text{Paket data diterima}) \times 100\%}{\text{Paket data dikirim}}$$

Berdasarkan persamaan diatas maka untuk mengukur packet loss perhitungannya sebagai berikut :

$$Packet\ loss = \frac{(213018 - 213017)}{2130} \times 100\% = \frac{1}{213018} \times 100\% = 0.046$$

TABEL II
REKAPITULASI PARAMETER QOS RANCANG BANGUN *PROTOTYPE*
SISTEM KENDALI ATAP *SMART MOSQUE* BERBASIS *ANDROID*

No	Rekapitulasi parameter QOS rancang bangun <i>Prototype</i> sistem kendali atap <i>Smart Mosque</i> berbasis <i>Android</i> .				
	Jam	Packe t Loss	Delay (ms)	Jitter (ms)	Throughput (kb/s)
1	15.00	0%	13.24	15.44	1.700
2	15.25	0%	15.54	16.35	1.300
3	15.50	0%	15.87	14.19	1.400
4	16.15	0%	14.04	14.86	1.800
5	16.40	0%	16.27	15.55	3.300
6	17.05	0%	14.74	14.37	3.500
7	17.30	0%	13.65	15.88	2.500
8	18.55	0%	15.39	14.76	2.800
9	18.20	0%	15.08	14.41	4.200
10	18.45	0%	14.87	14.03	4.600
11	19.10	0%	14.56	15.08	4.700
12	19.35	0%	13.8	13.33	3.400
Total		0%	177.05	17 8.2 5	35.200

= 97,625

Berdasarkan tabel 4.5 keseluruhan data rekapitulasi parameter QOS rancang bangun *Prototype* sistem kendali atap *Smart Mosque* berbasis *Android* dimulai pada jam 15.00 pagi sampai dengan jam 19.35 sore hari dengan menggunakan metode Quality of service (QOS) dan mendapatkan keseluruhan data persentase dari hasil pengukuran beberapa parameter dari metode tersebut dengan total persentase 97,625

III. KESIMPULAN

Berdasarkan hasil pengujian dan analisa pada penelitian yang sudah dilakukan yang berjudul “ Rancang Bangun *Prototype* Sistem Kendali Atap *Smart Mosque* Berbasis *Android* “ , maka dapat diambil kesimpulan sebagai berikut :

1. Pengujian sistem *Smart Mosque* dan notifikasi *Blynk* berjalan dengan baik . Dimana *Sensor LDR* merespon adanya cahaya sehingga *DC* langsung aktif membuka atap masjid dan apabila *Sensor LDR* tidak merespon adanya cahaya sehingga *DC* langsung aktif menutup atap masjid dan menghidupkan lampu serta kipas angin, pada siang hari *Sensor Raindrop* mendeteksi object air (hujan) apabila *Sensor* ini terkena air maka secara otomatis atap tertutup dan menghidupkan lampu serta kipas angin.
2. Berdasarkan hasil pengukuran dengan Metode Qos (Quality Of Service) perancangan alat Rancang Bangun *Prototype* Sistem Kendali Atap *Smart Mosque* Berbasis *Android* berjalan dan bekerja dengan “Baik” serta memperoleh nilai,dengan persentase keberhasilan alat dan sistem 95 s/d 100% dan masuk dalam kategori Sangat memuaskan.
3. Delay waktu yang dibutuhkan hanya 1s (detik) untuk sistem mengirim data monitoring ke android.

REFERENSI

- [1] Akbar,S. (2019). “Rancang Bangun *Prototype* Payung *Florabella* Berbasis *Internet Of Things* Sebagai *Monitoring* Cuaca”.(Online). <http://library.pnl.ac.id/index.php?search=search&keywords=Syawqi+Akbar+>. Diakses 4 November 2019.
- [2] Dewi,K.A.dkk. (2017).” Sistem Kendali Buka Tutup Atap Rumah untuk *Smarthome* dengan Menggunakan *Android Smartphone*”.*Jurnal Teknologi dan Sistem Komputer*. Vol.5. No.1. Universitas Negeri Malang.hlm.43-48.
- [3] Husna,R, Nasir.M (2019).“ Rancang Bangun *Prototype* Jemuran Berbasis *IoT (Internet Of Things)*” .(Online). <http://library.pnl.ac.id/index.php?search=search&keywords=Raudhatul+Husna+>. Diakses 29 Oktober 2019.
- [4] Mufida,E.dkk. (2013).” Pengendali Atap Jemuran Otomatis Dengan *Sensor* Cahaya Berbasis *Mikrokontroler*”.*Jurnal Konferensi Nasional Ilmu Sosial & Teknologi (KNiST)*.Vol.1. No.1.Jakarta.hlm.269-274.
- [5] Udin,B. (2014).“ Perancangan *Prototype* Alat Buka Tutup Atap Otomatis Berbasis *Mikrokontroler*”.(Online). https://scholar.google.co.id/citations?user=Z_QyBKwAAAAJ&hl=id. Diakses 15 Oktober 2019.
- [6] Basyir.M, Nasir.M, dkk, (2017).“ *Emergency Reporting Application is an Android-based* ” (Online). <https://emitter.pens.ac.id/index.php/emitter/article/view/220>. Diakses 1 Januari 2020
- [7] Nasir, M, dkk, (2018), “ Sistem Pendeteksi Dini Kebakaran Menggunakan *Colour Image Processing* dan *Raspberry Pi* ” (Online). <http://ejournal.pnl.ac.id/index.php/semnaspnl/article/view/775>. Diakses 1 Januari 2020

- [8] Jetorbit, (2016), "*Internet Of Things*" (Online). <https://www.jetorbit.com/blog/apa-itu-internet-of-things/>. Diakses 22 Januari 2020
- [9] Toppng, (2018), "*Wemos D1 ESP8266*" (Online). https://toppng.com/Wemos-d1-Analog-input-PNG-free-PNG-Images_83490?search-result=gold%20numbers. Diakses 29 Februari 2020