

IMPLEMENTASI SISTEM PEMANTAUAN DAN PENGONTROLAN PERANGKAT LISTRIK OTOMATIS MENGGUNAKAN APLIKASI ANDROID BERBASIS INTERNET OF THINGS

IMPLEMENTATION OF AUTOMATIC ELECTRICAL DEVICES MONITORING AND CONTROLLING SISTEM USING ANDROID APPLICATIONS BASED INTERNET OF THINGS

Rizqi Surya Utama¹, Ratna Mayasari,S.T.,M.T.², Muhammad Ary Murti,S.T.,M.T.³

Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom

¹rizqisuryautama@student.telkomuniversity.ac.id, ²ratnamayasari@telkomuniversity.ac.id,

³arymurti@telkomuniversity.ac.id

Abstrak

Internet of Things atau yang dikenal dengan IoT semakin berkembang luas pada era kecerdasan komputasi dewasa ini. Pada tugas akhir ini, dibangun sebuah bentuk pengaplikasian IoT seperti smart home. Sistem yang dibangun dapat memantau keadaan rumah dan mengontrol perangkat listrik menggunakan aplikasi android. Sistem dibangun menggunakan NodeMCU dan beberapa sensor seperti sensor DHT11, sensor Light Dependent Resistor (LDR), dan sensor Passive Infrared (PIR), serta relay sebagai aktuator. Data yang diperoleh dari hasil pembacaan sensor-sensor dikirim melalui NodeMCU ke internet, kemudian akan ditampilkan pada aplikasi Android. Sistem kontrol akan dilakukan melalui aplikasi android dan akan dijalankan oleh relay yang telah dirangkai dengan lampu dan kipas angin. MQTT Broker diperlukan sebagai jembatan transfer data dari NodeMCU ke aplikasi Android yang dibangun.

Setelah dilakukan pengujian, pada pengujian kualitas jaringan didapatkan bahwa nilai rata-rata *delay* yang didapatkan dari sistem pemantauan dan pengontrolan adalah sebesar 313,05 milisekon, nilai rata-rata *throughput* yang didapatkan ketika 1 node aktif adalah 8,992 Kbps, sedangkan pada 2 node aktif adalah 15,614 Kbps, nilai rata-rata *packet loss* yang didapatkan ketika 1 node aktif adalah 0,17%, sedangkan pada 2 node aktif adalah 0,25%, nilai rata-rata *reliability* dan *availability* pada server yang didapatkan adalah sebesar 99,99%.

Kata Kunci : *Internet of Things*, NodeMCU, DHT11, LDR, PIR, Relay, MQTT Broker

Abstract

Nowadays, Internet of Things or known as IoT is growing up in the era of computing intelligence. In this final project built form IoT application such as smart house. The system built can monitor the state of the house and control electrical devices using the Android application. The system is built using NodeMCU and several sensors such as DHT11 sensors, Light Dependent Resistor (LDR) sensors, and Passive Infrared (PIR) sensors, as well as relays as actuators. The data obtained from the reading of sensors sent via NodeMCU to the internet, will then be displayed on the Android application. The control system will be done through Android applications and will be run by relays that have been connected with lamp and table fan. MQTT Brokers are required as bridge data transfers from NodeMCU to Android application.

After several tests, the network quality testing found that the highest average *delay* in the monitoring system and control was 313,05 milliseconds, the average value of *throughput* obtained by compiling 1 active node was 8,992 Kbps on 2 active nodes is 15,614 Kbps, the average value of *packet loss* obtained by compiling 1 active node was 0,17% on 2 active nodes is 0,25%, the average value of *reliability* and *availability* is 99,99%.

Keywords : *Internet of Things*, NodeMCU, DHT11, LDR, PIR, Relay, MQTT Brokers

1. Pendahuluan

IoT atau dikenal juga dengan Internet of Things suatu konsep yang sangat populer merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus [1] [2]. Pengimplementasian teknologi IoT yang sedang banyak dikembangkan saat ini adalah Smart Home. Smart home atau rumah pintar merupakan sebuah sistem yang di bangun pada rumah yang dilengkapi

dengan teknologi pintar yang memungkinkan berbagai perangkat atau peralatan di rumah tersebut dapat berkomunikasi satu sama lain. Teknologi smart home dapat digunakan untuk mengendalikan hampir semua perlengkapan dan peralatan di rumah yang perintahnya dapat dilakukan dengan kontrol jarak jauh (remote).

Dengan memanfaatkan pengaplikasian smart home pada rumah dapat memantau keadaan dan mengontrol peralatan elektronik di rumah tersebut. Pemantauan dan pengontrolan tersebut bisa dilakukan dari jarak jauh seperti mendeteksi gerakan, menghitung suhu ruangan, intensitas cahaya serta dapat mengontrol untuk menyalakan ataupun mematinya lampu melalui sebuah aplikasi pengontrolan. Hal inilah yang menjadi bahasan pokok pada tugas akhir ini dengan mengilhami implementasi dari internet of things.

Sebelum tugas akhir ini, terdapat beberapa penelitian yang terkait dan membahas tentang hal yang serupa. Tahun 2015 dari Politeknik Negeri Semarang tentang "Pengendalian Peralatan Listrik Dengan Smartphone Berbasis Android Menggunakan Komunikasi Wireless" yang lebih berfokus pada sistem pengontrolan jarak jauh menggunakan aplikasi android dari penggunaan perangkat listrik [3]. Tahun 2017 dari Universitas Telkom tentang "Sistem Kontrol dan Monitoring Genset Melalui Internet" yang berfokus pada penggunaan relay pada sistem pengontrolan menggunakan PC atau smartphone [4]. Tahun 2017 tentang "Sistem Pengawasan dan Kontrol dengan Perangkat Cerdas yang Diterapkan pada Rumah Menggunakan Internet Of Things dan Modul 3G" dari Universitas Telkom yang lebih berfokus kepada pengembangan perangkat cerdas dalam membangun sambungan ke internet menggunakan jaringan 3G dan kecerdasan sistem untuk melakukan otomasi registrasi di server [5].

Saat ini, penulis mengerjakan tugas akhir ini dengan membangun sebuah sistem yang dapat memantau keadaan serta dapat mengontrol perangkat listrik dari jarak jauh menggunakan aplikasi Android. Sistem pemantauan tersebut berupa pemantauan pada deteksi gerakan, menghitung suhu dan kelembapan, serta intensitas cahaya dalam ruangan. Sedangkan sistem pengontrolannya berupa pengontrolan pada relay yang terhubung dengan lampu dan kipas angin.

2. Dasar Teori

2.1 Internet of things

Internet of Things (IoT) dapat diartikan sebagai segala benda yang dapat berkomunikasi dengan benda lainnya, seperti komunikasi machine to machine (M2M) dan komunikasi orang dengan komputer serta akan meluas sampai komunikasi kesegalanya [6]. Perangkat IoT berprinsip bahwa benda di dunia nyata diberikan identitas unik dan dapat dikali di sistem komputer dan dapat di representasikan dalam bentuk data pada istem komputer. Perkembangannya memungkinkan sebuah benda dapat diberi tanda pengenal berupa IP address dan menggunakan jaringan internet untuk bisa berkomunikasi dengan benda lain yang memiliki pengenal IP address.

2.2 Microwifi NodeMCU

Microwifi merupakan salah satu arduino compatible development board yang dirancang khusus untuk keperluan IoT menggunakan chip SoC WiFi yang cukup terkenal saat ini yaitu ESP8266. Arduino compatible ialah sebuah program yang menggunakan Arduino IDE dengan sintaks program dan library yang banyak terdapat di internet. Microwifi akan digunakan sebagai perangkat IoT yang dapat diintegrasikan dengan perangkat yang akan di kontrol dan monitor melalui internet yang terhubung ke akses poin. Microwifi yang sering digunakan adalah NodeMCU yang berbasis ESP8266. [7]

2.3 Sensor

Sensor adalah komponen yang dapat digunakan untuk mengkonversi suatu besaran tertentu menjadi satuan analog sehingga dapat dibaca oleh suatu rangkaian elektronik. Sensor merupakan komponen utama dari suatu transduser, sedangkan transduser merupakan sistem yang melengkapi agar sensor tersebut mempunyai keluaran sesuai yang kita inginkan dan dapat langsung dibaca pada keluarannya. [8]

2.4 Aktuator Relay

Aktuator adalah sebuah peralatan mekanis yang digunakan sebagai penggerak atau pengontrol sebuah mekanisme atau sistem. Aktuator diaktifkan dengan menggunakan lengan mekanis yang dikontrolkan oleh media pengontrol otomatis yang terprogram pada mikrokontroler. Aktuator merupakan suatu elemen yang dapat mengkonversikan besaran listrik analog menjadi besaran lainnya. Aktuator yang cocok digunakan pada kontrol penggunaan perangkat listrik di rumah adalah relay. [9]

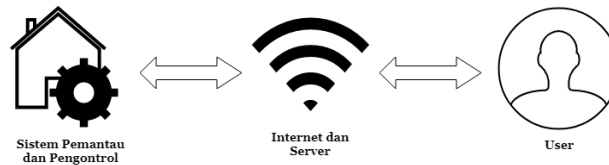
2.5 MQTT (Message Queuing Telemetry Transport)

Protokol MQTT (Message Queuing Telemetry Transport) adalah protokol yang berjalan pada diatas stack TCP/IP dan mempunyai ukuran paket data dengan low overhead yang kecil (minimum 2 bytes) sehingga berefek pada konsumsi catu daya yang juga cukup kecil. Protokol ini adalah jenis protokol data-agnostic yang

artinya anda bisa mengirimkan data apapun seperti data binary, text bahkan XML ataupun JSON dan protokol ini memakai model publish dan subscribe pada model client-server. [10]

3. Perancangan Sistem

3.1 Rancangan Umum Sistem

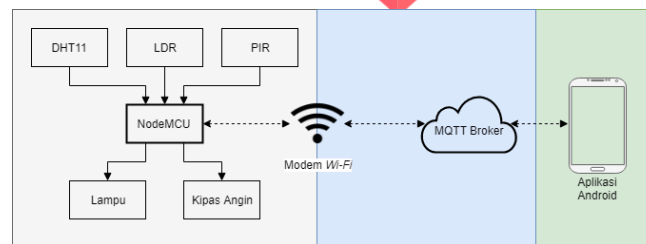


Gambar 1 Rancangan Umum Sistem

Gambar 1 ini menunjukkan sistem yang terdiri dari perangkat pemantau dan pengontrol serta internet yang menghubungkan dengan sisi pengguna yang mengalami konsep internet of things. Perangkat pada sistem pemantau dan pengontrol berupa microwifi NodeMCU sebagai pembaca dari sensor yang digunakan untuk mendeteksi dan dikontrol menggunakan relay. Data yang terbaca oleh sensor dan diterima oleh NodeMCU akan dikirimkan ke server melalui akses internet dari access point pada modem wifi yang digunakan. Pada sisi user aplikasi digunakan adalah aplikasi android. Melalui aplikasi tersebut kita dapat melakukan operasi kontrol pada relay sehingga bisa mengontrol aliran listrik pada perangkat elektronik yang terhubung dari jarak jauh melalui internet secara real-time.

3.2 Desain Model Sistem

Berikut adalah desain model perancangan sistem pemantau dan pengontrol pada rumah pintar berbasis internet of things:



Gambar 2 Diagram Blok Sistem Kendali

Gambar 2 ini merupakan desain model yang digunakan untuk merancang sistem, secara umum. Sistem terdiri dari 3 bagian berupa sistem pemantau dan pengontrol pada bagian pertama yang terdiri dari microwifi ESP8266 yang terhubung dengan sensor-sensor, aktuator dan catu daya. Sensor-sensor tersebut terdiri dari sensor DHT11 sebagai pengukur suhu ruangan, sensor LDR sebagai penghitung intensitas cahaya, dan sensor PIR sebagai pendeteksi gerakan. Aktuator yang digunakan yaitu berupa relay dimana kita dapat mengontrol perangkat elektronik tertentu dari jarak jauh. Sedangkan bagian kedua adalah konektivitas sistem, dimana ESP8266 akan diberi koneksi internet melalui akses dari sebuah access point. Apabila ESP8266 telah mendapat koneksi internet data akan diteruskan melalui internet ke server dan aplikasi android yang ada pada bagian user. Bagian ketiga adalah sisi pengguna dimana pengguna dapat melakukan instruksi tertentu terhadap perangkat elektronik miliknya hanya dengan melalui aplikasi Android. Selain melakukan pengontrolan, pada aplikasi ini pengguna juga dapat memantau kondisi suhu, kelembapan, intensitas cahaya bahkan gerakan manusia dalam ruangan.

3.3 Perancangan dan Konfigurasi Sistem

3.3.1 Perancangan Sistem Mikrokontroler

Sistem mikrokontroler pada sistem pemantau dan pengontrolan pada ruangan pintar merupakan satu kesatuan dari beberapa perangkat keras berupa NodeMCU yang dihubungkan sensor suhu dan kelembapan DHT11, sensor cahaya LDR, dan sensor PIR, serta aktuator relay yang dirangkai dengan lampu dan kipas angin, yang mana sistem ini akan terkoneksi pada jaringan internet. Berikut Akan ditampilkan beberapa perancangan rangkaian elektronika pada setiap komponen.

Arduino IDE digunakan untuk memprogram NodeMCU agar dapat bekerja sesuai dengan kebutuhan yang diinginkan, mulai dari menghubungkan NodeMCU dengan jaringan internet, menghubungkan NodeMCU dengan MQTT broker serta mengintegrasikan NodeMCU dengan sensor. Arduino IDE yang digunakan ialah Arduino IDE versi 1.8.10.0 yang dijalankan pada sistem operasi Windows 10.

3.3.2 Perancangan Server

Server pada sistem ini dapat diperankan oleh MQTT broker. MQTT broker diperlukan agar data yang dikirimkan dari sensor dapat langsung dibaca oleh user melalui aplikasi yang dibuat. MQTT broker harus selalu aktif agar data dapat dikirim secara real time, maka dari itu MQTT broker ditanam di Virtual Private Server (VPS) yang selalu online setiap saat. MQTT broker hanya akan meneruskan pengiriman data yang memiliki topic dan message yang sama. VPS yang digunakan yaitu menggunakan sistem operasi Linux Ubuntu 16.04.4 dengan RAM 1GB . Untuk mengelola VPS yang dibuat dapat menggunakan software untuk me-remote yaitu Putty.

3.3.3 Perancangan Aplikasi Android

Agar pengguna dapat melakukan pemantauan dan pengontrolan dengan sangat mudah maka dirancanglah sebuah aplikasi Android yang terintegrasi pada server secara terus-menerus. Untuk memenuhi kebutuhan akan aplikasi pada smartphone bersistem operasi Android maka diperlukan Android Studio untuk membangun aplikasi tersebut. Android Studio yang digunakan ialah Android Studio versi 3.1.2 yang dijalankan pada sistem operasi Windows 10. Berikut disertakan potongan gambar pemograman pada Android Studio :

4. Hasil Percobaan dan Analisa

4.1 Pengujian Fungsionalitas

Pada pengujian diuji fungsi-fungsi dari perangkat mulai dari pengujian fungsi perangkat pada mikrokontroler, broker pada server hingga aplikasi pada android. Pengujian bertujuan untuk mengetahui fungsi masing-masing perangkat yang digunakan telah memenuhi harapan. Pada sistem mikrokontroler akan diuji mulai dari NodeMCU dapat terkoneksi dengan internet, sensor-sensor dapat membaca data sesuai yang diharapkan dan fungsinya masing-masing, relay dapat menerima instruksi dan menjalankan instruksi tersebut pada perangkat yang dirangkaikan.

Setelah dilakukan pengujian pada sistem mikrokontroler dilakukan pengujian broker pada server guna melihat apakah data sudah dapat terkirim pada brokersehingga broker dapat menjalankan fungsinya. Data yang berhasil di-*subscribe* oleh broker berarti data siap di-*publish*-kan pada setiap *client* yang juga men-*subscribe* sesuai dengan *topic* data tersebut.

Pengujian berikutnya adalah pengujian pada sisi *user* atau pengguna. Pengujian ini dilakukan untuk menguji aplikasi dapat digunakan sesuai fungsi yang diinginkan seperti tampilan aplikasi, fungsi tombol, dan halaman *monitoring*.

4.2 Pengujian Sensor dan Aktuator

4.2.1 Sensor PIR

Tabel 1 Pengujian sensor PIR

| No | Jarak (m) | Sudut | | | | | | |
|----|--------------|-------|------|------|-----|-----|-----|-----|
| | | -60° | -45° | -30° | 0° | 30° | 45° | 60° |
| 1 | 0.00 | - | - | - | Det | - | - | - |
| 2 | 0.25 | ND | Det | Det | Det | Det | Det | Det |
| 3 | 0.50 | Det | Det | Det | Det | Det | Det | Det |
| 4 | 0.75 | Det | Det | Det | Det | Det | Det | Det |
| 5 | 1.00 | Det | Det | Det | Det | Det | Det | Det |
| 6 | 1.50 | Det | Det | Det | Det | Det | Det | Det |
| 7 | 2.00 | Det | Det | Det | Det | Det | ND | Det |
| 8 | 2.50 | ND | Det | Det | Det | Det | Det | ND |
| 9 | 3.00 | ND | Det | Det | Det | Det | Det | ND |
| 10 | 3.50 | ND | ND | Det | Det | ND | Det | ND |
| 11 | 4.00 | ND | ND | Det | Det | Det | ND | ND |
| 12 | 5.00 | ND | ND | Det | Det | ND | ND | ND |
| 13 | 6.00 | ND | ND | ND | Det | ND | ND | ND |

Dari pengujian diatas, pengujian sensor PIR dengan parameter jarak dan sudut ND menjelaskan bahwa tidak terdeteksi (Not Detected) dan Det adalah (Detected). Hasil pengujian didapatkan presisi sensor PIR pada

pengujian jarak jangkauan adalah 6 meter dan sudut cakupan 120° dengan tingkat keberhasilan pembacaan sensor :

$$H=(N-nF)/N \times 100\% = (85-25)/85 \times 100\% = 70,59\%$$

Keterangan :

H = tingkat keberhasilan pembacaan sensor

N = jumlah total percobaan

nF = jumlah percobaan yang gagal

Dari pengujian yang dilakukan diketahui bahwa objek yang berada tepat didepan sensor cenderung berhasil ditangkap oleh sensor PIR dengan tingkat keberhasilan pembacaan sensor 70,59%.

4.2.2 Sensor LDR dan Lampu

Pengujian ini bertujuan untuk mengetahui konfigurasi sensor LDR terhadap lampu pada sistem otomatis. Pengujiannya dilakukan dengan memperhatikan kondisi lampu sesuai dengan perubahan intensitas cahaya yang diterima oleh sensor LDR. Skenario yang digunakan pada pengujian ini adalah dengan memberikan intensitas cahaya yang berbeda-beda terhadap sensor LDR sehingga akan mengakibatkan kondisi lampu berubah ketika *check box automode* atau sistem otomatis diaktifkan dan sensor PIR membaca gerakan manusia.

Berdasarkan hasil pengujian sensor LDR terhadap Lampu diatas menunjukkan hasil dimana pengujiannya memenuhi logika yang telah deprogram pada NodeMCU yang digunakan. Lampu akan hidup ketika intensitas cahaya yang terbaca pada sensor ≤ 300 Lux dan lampu akan mati ketika intensitas cahaya yang terbaca pada sensor ≥ 500 Lux, sedangkan apabila intensitas cahaya antara 300-500 Lux maka tidak ada interrupt berarti kondisi sebelumnya akan dipertahankan dan user tetap bisa melakukan kontrol manual.

4.2.3 Sensor DHT11 dan Kipas Angin

Tabel 2 Pengujian sensor DHT11 dan kipas angin

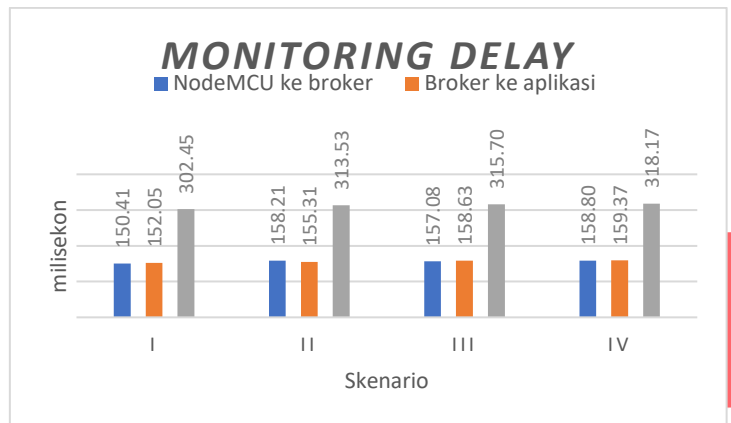
| No | Suhu pada Termometer (°C) | Suhu pada DHT11 (°C) | Selisih suhu (°C) | Akurasi | Kondisi Kipas Angin |
|----|---------------------------|----------------------|-------------------|---------|----------------------------|
| 1 | 21.9 | 22 | 0.1 | 99.545 | Mati |
| 2 | 22.2 | 23 | 0.8 | 96.522 | Mati |
| 3 | 22.8 | 23 | 0.2 | 99.130 | Mati |
| 4 | 23.4 | 24 | 0.6 | 97.500 | Tidak ada <i>interrupt</i> |
| 5 | 23.5 | 24 | 0.5 | 97.917 | Tidak ada <i>interrupt</i> |
| 6 | 24.2 | 25 | 0.8 | 96.800 | Tidak ada <i>interrupt</i> |
| 7 | 24.7 | 25 | 0.3 | 98.800 | Tidak ada <i>interrupt</i> |
| 8 | 24.3 | 25 | 0.7 | 97.200 | Tidak ada <i>interrupt</i> |
| 9 | 25.2 | 26 | 0.8 | 96.923 | Tidak ada <i>interrupt</i> |
| 10 | 25.7 | 26 | 0.3 | 98.846 | Tidak ada <i>interrupt</i> |
| 11 | 26.4 | 27 | 0.6 | 97.778 | Hidup |
| 12 | 26.8 | 27 | 0.2 | 99.259 | Hidup |
| 13 | 27.6 | 28 | 0.4 | 98.571 | Hidup |
| 14 | 27.4 | 28 | 0.6 | 97.857 | Hidup |
| 15 | 28.2 | 29 | 0.8 | 97.241 | Hidup |

Berdasarkan hasil pengujian sensor DHT11 terhadap kipas angin diatas menunjukkan hasil dimana pengujiannya memenuhi logika yang telah deprogram pada NodeMCU yang digunakan. Kipas angin akan mati ketika suhu yang terbaca pada sensor $\leq 23^{\circ}\text{C}$ dan lampu akan hidup ketika suhu yang terbaca pada sensor $\geq 26^{\circ}\text{C}$, sedangkan apabila suhu ruangan antara 23°C - 26°C maka tidak ada interrupt berarti kondisi sebelumnya akan dipertahankan dan user tetap bisa melakukan kontrol manual.

4.3 Pengujian Kualitas Jaringan

4.3.1 Delay

Dalam sebuah jaringan, *delay* dapat dijadikan sebuah acuan untuk menilai kualitas jaringan. *Delay* merupakan waktu tunda yang terjadi dalam proses pengiriman dari suatu paket mulai titik awal sampai ke titik tujuan. *Delay* yang diukur ialah *delay* antara NodeMCU dengan MQTT broker, dan *delay* antara MQTT broker dengan aplikasi. Pengukuran *delay* menggunakan *software* Wireshark yang dipasang pada MQTT broker, sehingga Wireshark dapat mengamati setiap paket yang di kirim oleh NodeMCU serta paket yang di teruskan ke aplikasi. Untuk pengujian ini terdapat skenario umum, yaitu dibedakan jeda pengiriman data dari sisi NodeMCU saat hanya 1 *node* aktif dan 2 *node* aktif dalam waktu bersamaan. Pengujian ini juga terdapat skenario tambahan diantaranya ketika hanya dilakukan sistem pantau saja dan ketika dilakukan sistem pantau serta kontrol dalam bersamaan. Berikut adalah grafik dari hasil *delay* pada sistem monitoring atau pemantauan yang merupakan *delay* rata dari nilai sensor yang dikirim oleh NodeMCU sampai MQTT broker dan diteruskan lagi ke aplikasi:

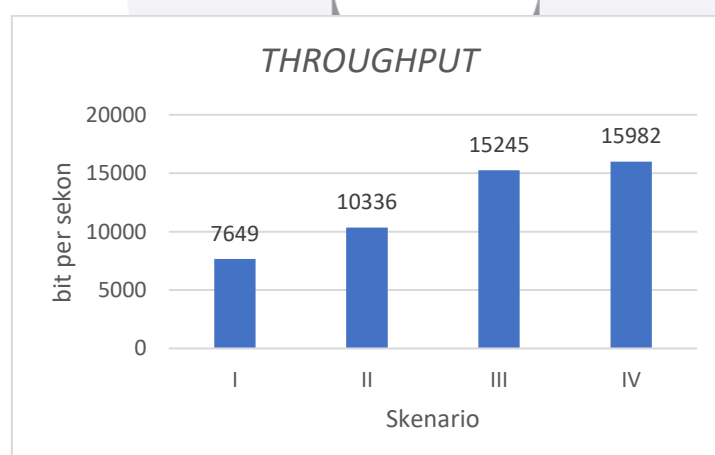


Gambar 3 Grafik *delay* sistem monitoring

Gambar diatas merupakan grafik dari rata-rata nilai *delay* dari tiap data sensor yang berhasil dikirm oleh NodeMCU ke MQTT broker dan juga diteruskan sampai ke aplikasi. Grafik berwarna biru merupakan representasi dari rata-rata nilai *delay* dari NodeMCU ke MQTT broker, sedangkan oranye adalah nilai *delay* rata-rata dari MQTT broker ke aplikasi, dan yang abu-abu adalah hasil dari rata-rata end to end *delay* seperti keterangan yang tertara. Berdasarkan hasil dari gambar grafik 4.1 diatas didapatkan *delay* sebanding dengan beban yang diterima oleh server, semakin banyaknya beban yang diberikan pada server terlihat grafik menunjukkan semakin meningkat juga *delay*-nya.

4.3.2 Throughput

Pengujian *throughput* dilakukan dengan cara menghitung ukuran paket yang berhasil ditransfer yang dibandingkan dengan waktu transfer paket tersebut. Pengujian ini dilakukan dalam kurun waktu 60 detik dan masing-masing 30 kali percobaan. *Throughput* adalah rata-rata pengiriman pesan sukses melalui saluran komunikasi.



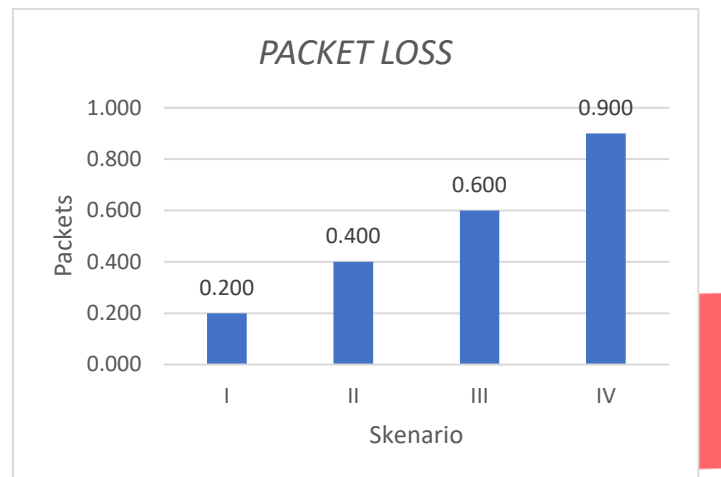
Gambar 4 Grafik *throughput*

Dari hasil pengujian diatas dapat dilihat *throughput* akan semakin meningkat ketika beban yang diberikan pada jaringan meningkat. Hal ini dikarenakan, berdasarkan penjelasan serta perumusan tentang *throughput* menjelaskan bahwa ukuran paket yang berhasil ditransfer berbanding lurus dengan nilai *throughput*. *Throughput* juga akan berbanding terbalik dengan waktu pengiriman data. Dikarenakan pada pengujian ini

dilakukan dalam durasi pengamatan 60 detik, maka interval waktu pengujian atau time span pada tiap skenario akan relatif sama maka kita bisa dengan mudah membandingkan nilai *throughput* berdasarkan banyak paket yang teramati.

4.3.3 Packet loss

Tujuan menghitung *packet loss* ini adalah untuk mengetahui banyak paket yang hilang pada suatu jaringan paket yang disebabkan oleh tabrakan (*collision*), penuhnya kapasitas jaringan, dan penurunan paket yang disebabkan oleh habisnya TTL (*Time To Live*) paket. Berikut adalah hasil perhitungan dari nilai *packet loss*.



Gambar 5 Grafik *packet loss*

Berdasarkan Gambar 5 terlihat bahwa semakin banyak beban yang dikirim ke server atau broker akan menyebabkan semakin meningkatnya kecenderungan paket yang gagal untuk dikirimkan. Banyaknya beban paket yang diterima broker menyebabkan broker akan sibuk dalam melakukan pekerjaannya. Sehubungan dengan prinsip *publish* dan *subscribe* broker akan langsung mengirimkan pesan yang diterimanya sesuai dengan topic yang diberikan dengan waktu secepat mungkin. Jadi, ketika paket yang dikirimkan ke broker tersebut datang dengan jeda waktu yang hampir bersamaan tentu hal tersebut dapat mengakibatkan adanya beberapa paket yang gagal diteruskan.

4.3.4 Reliability dan Availability

Reliability atau realibilitas merupakan kemungkinan dari suatu sistem untuk dapat memenuhi fungsi yang dibutuhkan pada kondisi tertentu dan periode tertentu. Sedangkan *availability* merupakan kemungkinan suatu sistem siap untuk beroperasi saat dibutuhkan. Untuk mencari nilai *reliability* dan *availability* pada server kita terlebih dahulu harus mengetahui *uptime* dan *downtime* server tersebut. *Uptime* merupakan total waktu suatu sistem dalam kondisi *rise/reliable*, sedangkan *downtime* adalah total waktu suatu sistem dalam kondisi *failure/unreliable*.

Tugas akhir ini menggunakan VPS dengan provider Digital Ocean. Provider tersebut mempunyai spesifikasi pada semua layanannya dengan *uptime* sebesar 99,99% dan *downtime* sebesar 0,01%. Maka dari angka tersebut sesuai dengan persamaan *reliability* dan *availability* yang terdapat pada persamaan 2.4 dan 2.5 dapat menghasilkan nilai *reliability* dan *availability* pada server sebesar :

$$Reliability = \frac{(99,99 - 0,01)}{99,99} \times 100\% \approx 99,99\%$$

$$Availability = \frac{99,99}{(99,99 + 0,01)} \times 100\% = 99,99\%$$

5. Kesimpulan dan Saran

5.1 Kesimpulan

Dari hasil yang telah diperoleh setelah dilakukannya pengujian, maka dapat diambil kesimpulan sebagai berikut :

1. Sistem pemantau dapat menampilkan hasil dari pembacaan sensor berupa suhu, kelembapan, cahaya, dan inframerah.
2. Sistem pengontrol dapat melakukan instruksi pada relay yang telah dihubungkan dengan lampu dan kipas angin
3. Aplikasi pada ponsel android dapat menampilkan hasil pemantauan dan dapat melakukan pengontrolan

4. Pada pengujian kualitas jaringan nilai rata-rata *delay* yang didapatkan dari sistem pemantauan dan pengontrolan adalah sebesar 313,05 milisekon
5. Pada pengujian kualitas jaringan nilai rata-rata *throughput* yang didapatkan ketika 1 node aktif adalah 8,992 Kbps sedangkan pada 2 node aktif adalah 15,614 Kbps.
6. Pada pengujian kualitas jaringan nilai rata-rata packet loss yang didapatkan ketika 1 node aktif adalah 0,17% sedangkan pada 2 node aktif adalah 0.25%.
7. Pada pengujian kualitas jaringan nilai rata-rata *reliability* yang didapatkan adalah sebesar 99,99% dan rata-rata nilai *availability*-nya sebesar 99,99%

5.2 Saran

Dari sistem yang telah dibangun, masih terdapat kekurangan yang dapat dikembangkan pada penelitian selanjutnya, antara lain :

1. Sistem dapat menggunakan sensor sensor yang lebih banyak dan yang lebih akurat dalam membaca data serta dapat menggunakan aktuator lain selain relay, serta dapat mengontrol kecepatan kipas angin yang digunakan, dll.
2. Sistem dapat dibangun dengan server yang lebih aman dan kompleks
3. Parameter pengujian yang digunakan lebih banyak dengan parameter-parameter tambahan lainnya.
4. Aplikasi yang digunakan lebih kompleks sehingga sangat dapat membantu user dalam melakukan pekerjaannya, dengan menambahkan fitur-fitur baru.

Daftar Pustaka

- [1] A. Homechat and T. Smarthome, "HomeChat : a way communicate with home instrument," 2011.
- [2] International Telecommunication Union, "Overview of the Internet of things," Ser. Y Glob. Inf. infrastructure, internet Protoc. Asp. next-generation networks - Fram. Funct. Archit. Model., p. 22, 2012
- [3] Syahid, A. Santoso, Y. Badruzzaman. "Pengendalian Peralatan Listrik dengan Smartphone Berbasis Android Menggunakan Komunikasi Wireless," Semarang: 2015
- [4] W. Firdaus, M. A. Murti, R. Nugraha. "Sistem Kontrol dan Monitoring Genset Melalui Internet," Bandung: 2017.
- [5] N. Cahyanti, M. A. Murti, R. Mayasari. "Sistem Pengawasan dan Kontrol dengan Perangkat Cerdas yang Diterapkan pada Rumah Menggunakan Internet of Things," Bandung: 2017.
- [6] S. D. T. Kelly, N. K. Suryadevara, and S. C. Mukhopadhyay, "Towards the implementation of IoT for environmental condition monitoring in homes," IEEE Sens. J., vol. 13, no. 10, pp. 3846–3853, 2013.
- [7] "NodeMCU V3 ESP8266", Available: <https://www.makerlab-electronics.com/product/nodemcu-v3-esp8266-esp-12e/> [Diakses 22 Oktober 2017].
- [8] "Sensor", Available: <http://zoniaelektro.net/sensor/> [Diakses 22 Oktober 2017].
- [9] "Modul Relay", Available: <https://www.gagalenyilih.com/2017/05/pengertian-dan-penjelasan-lengkap.html>. [Diakses 23 Oktober 2017].
- [10] F. Quintero, "MQTT – IoT Messaging Protocol Francisco Quintero Lead Firmware Engineer."