

IMPLEMENTASI *INTERNET OF THINGS (IoT)* PADA RANCANG BANGUN TEMPAT SAMPAH OTOMATIS MENGGUNAKAN *PLATFORM MY DEVICE CAYENNE*

Mochamad Donny Rinaldi¹, Herwin Junion Siboro², Ir. Arief Syaichu Rohman, M.Eng.Sc.,Ph.D³, Syamsyarief Baqaruzi, S.T., M.T.⁴

^{1,2,3,4}*Program Studi Teknik Elektro, Jurusan Teknik Elektro, Informatika dan Sistem Fisika, Institut Teknologi Sumatera*

Herwin.13116003@student.itera.ac.id, Rinald1.13116013@student.itera.ac.id, ariefsyaichu@gmail.com, syamsyarief.baqaruzi@el.itera.ac.id,

Abstract— Pada era modern saat ini pemanfaatan *Internet of Things* atau IoT telah banyak diterapkan diberbagai bidang seperti kesehatan, perindustrian, perkotaan dan bahkan pengolahan lingkungan. Konsep IoT sendiri sangat dibutuhkan sekali sebagai sarana efisiensi pekerjaan, IoT memberikan kemudahan dalam pemantauan *real time* dan juga proses kontrol secara berulang. Dengan semakin berkembangnya *platform-platform* penyedia layanan IoT kami memiliki inovasi untuk mengkombinasikan konsep IoT dengan pengolahan sampah. Dari ide tersebut terbentuklah tempat sampah otomatis untuk pengolahan limbah plastik berbasis IoT, tempat sampah ini menggunakan layanan *open source mydevice cayenne* sebagai salah satu penyedia *platform* IoT. *My Device Cayenne* sendiri berfungsi sebagai server yang menyimpan project kontrol dan memonitoring sebuah alat serta mendukung untuk koneksi dengan berbagai jenis mikrokontroler, *platform* ini sangat *user-friendly* serta menerapkan sistem MQTT (*Message Queuing Telemetry Transport*) sebagai protokol komunikasi antara *device* mikrokontroler dan *cloud server*. Dengan adanya konsep IoT pada tempat sampah ini, nantinya pengguna dapat melakukan pemantauan kondisi tempat sampah, seperti besarnya kapasitas dan volume sampah tertampung, pengontrolan fitur kompresi dan pemanas serta pemberian notifikasi. Semua kemudahan tersebut cukup dengan satu buah aplikasi pada *smartphone*, pengguna dan dapat dijalankan secara *real time* kapanpun dan dimanapun.

Kata kunci— *Iot, My Device Cayenne, MQTT*

I. PENDAHULUAN

Pada umumnya sampah selalu dikumpulkan secara kolektif dikumpulkan hingga akhirnya ditumpuk pada suatu tempat penumpukan yang lebih besar. Metode yang demikian membuat lahan penumpukan sampah setiap harinya semakin memprihatinkan. Secara tidak langsung sampah hasil tumpukan ini menyita banyak lahan yang harusnya dapat dipakai untuk kepentingan yang lebih baik. Ada beberapa cara yang coba dilakukan untuk mengatasi penumpukan sampah, salah satunya adalah pelabelan tempat sampah organik dan sampah anorganik. Cara ini memang sangat membantu mengingat sampah tumpukan sampah organik

lebih gampang untuk diatasi dan bahkan menguntungkan jika diolah dengan baik. Yang menjadi masalah adalah tumpukan sampah plastik, penumpukan sampah plastik sejauh ini masih terlalu susah untuk diolah sehingga tumpukan sampah yang paling banyak disebabkan oleh sampah plastik. Didasari masalah tersebut kami membuat suatu rancangan alat yang mampu mengurangi bahkan mengatasi permasalahan tempat sampah. Alat ini kami sebut dengan *Lectro-Bin* (*Elektronik Dustbin For Plastic Waste*). *Lectrobin* dilengkapi dengan modul Wifi sehingga membuatnya dapat terhubung dengan jaringan internet. Dengan adanya jaringan internet maka perangkat dapat kami aplikasikan melalui *Internet of Things (IoT)*. *Internet of Things* merupakan sebuah pengembangan komunikasi jaringan dari benda yang saling terkait, terhubung satu dengan yang lain lewat komunikasi internet serta untuk saling bertukar data yang kemudian dapat mengubahnya menjadi informasi.

My Device Cayenne adalah *platform* IoT yang sekaligus berfungsi sebagai server yang mampu menyimpan project yang sedang dibuat. *Cayenne* mendukung dan sangat support terhadap berbagai jenis mikrokontroler. Pada *cayenne* selain berbagai kelebihan tersebut, masih ada fitur yang membuat *Cayenne* lebih *user-friendly* yaitu adanya aplikasi berbasis *smartphone* dengan OS Android, IOS, maupun Windows Phone sehingga memudahkan dalam membuat berbagai jenis perangkat elektronik dengan kendali jarak jauh melalui internet.

II. TEORI DASAR

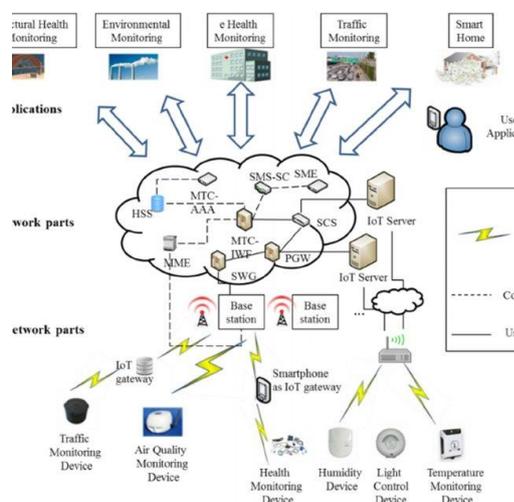
1. *Internet of Things (IoT)*

Dituliskan dalam sebuah karya ilmiah dalam McKinsey Global Institute, bahwa *internet of things* adalah sebuah teknologi yang memungkinkan kita untuk menghubungkan mesin, peralatan, dan benda fisik lainnya dengan sensor jaringan dan aktuator untuk memperoleh data dan mengelola kinerjanya sendiri, sehingga dimungkinkan adanya mesin untuk saling berkolaborasi dan bahkan bertindak berdasarkan informasi baru yang diperoleh secara independen.

Dengan perkembangan teknologi khususnya mikrokontroler dan jaringan komputer serta internet of thing tersebut dapat memudahkan segala bentuk aktifitas kehidupan manusia. Semua sebenarnya bermula pada saat awal tahun 90-an orang mulai mengenal *teknologi Internet of*

Things. Mark Weiser seorang staf Xerox Palo Alto Research Center pada tahun 1991 menerbitkan paper tentang *ubiquitous computing* dengan judul "*The Computer of the 21st Century*" yang menjelaskan tentang visi kontemporer yang berkaitan langsung dengan perkembangan teknologi internet of thing saat ini [1]. *Internet of Things* muncul dan diperkenalkan sebenarnya oleh Kevin Ashton, seorang *entrepreneur* yang fokus terhadap teknologi asal UK. Istilah *Internet of Things* pertama kali dipakai sebagai judul presentasi dalam seminarnya di depan perusahaan penyedia produk harian dari Amerika, Procter & Gamble (P&G). Kevin menjelaskan bahwa IoT sebagai sistem dimana bendabenda fisik dapat saling terhubung ke internet melalui sensor yang ada di mana-mana [2].

Internet of Things dapat digambarkan sebagai sebuah jaringan yang mampu untuk melakukan konfigurasi sendiri, serta adaptif, sehingga membentuk sebuah jaringan kompleks yang saling berhubungan dengan berbagai perangkat ke Internet melalui penggunaan protokol komunikasi standar". [3]



Gambar 2.1 Fitur dan ruang lingkup Iot [6]

2. My Device Cayenne

Merupakan sebuah *platform open source* penyedia layanan IoT yang menyediakan akses untuk berbagai macam mikrokontroler dan modul komunikasi seperti arduino, raspberry, dan Lora dengan berbagai macam shield termasuk di dalamnya terdapat Wi-fi, BLE, IR, NFC, dan lain sebagainya. *Platform* ini dilengkapi dengan berbagai macam fitur diantaranya adalah:

- Customizable Dashboard
- Remote Monitoring Control
- Alerts
- Triggers
- Scheduling
- Data
- Asset Tracking
- Custom code
- MQTT API
- LoraWan

Diantara fitur-fitur tersebut, fitur yang umum digunakan adalah *customizable dashboard* yang berguna untuk menyesuaikan tampilan *interface* dengan keinginan user, lalu *remote monitoring control*, sebagai fungsi eksekusi dan

perintah oleh user, serta *Alerts* atau biasa disebut dengan notifikasi yang digunakan sebagai peringatan dari suatu keadaan.

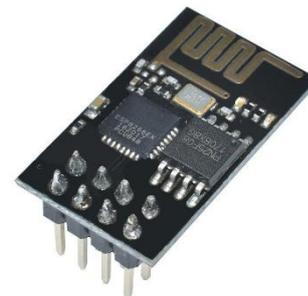
3. Modul Wi-fi ESP 8266

Modul wireless ESP8266 merupakan modul *low-cost Wi-Fi* dengan dukungan penuh untuk penggunaan TCP/IP. Modul ini di produksi oleh Espressif Chinese manufacturer. Pada tahun 2014, dan AI-Thinker manufaktur pihak ketiga dari modul ini mengeluarkan modul ESP-01 [4]. Modul ini membutuhkan daya sekitar 3.3v dengan memiliki tiga mode *wifi* yaitu *Station*, *Access Point* dan *Both*. Modul ini juga dilengkapi dengan prosesor, memori dan GPIO dimana jumlah pin bergantung dengan jenis ESP8266 yang kita gunakan. Sehingga modul ini bisa berdiri sendiri tanpa menggunakan mikrokontroler apapun karena sudah memiliki perlengkapan layaknya mikrokontroler.

Firmware default yang digunakan oleh perangkat ini menggunakan AT Command, selain itu ada beberapa Firmware SDK yang digunakan oleh perangkat ini berbasis *opensource* yang diantaranya adalah sebagai berikut :

- NodeMCU dengan menggunakan basic programming lua
- MicroPython dengan menggunakan basic programming python
- AT Command dengan menggunakan perintah perintah AT command

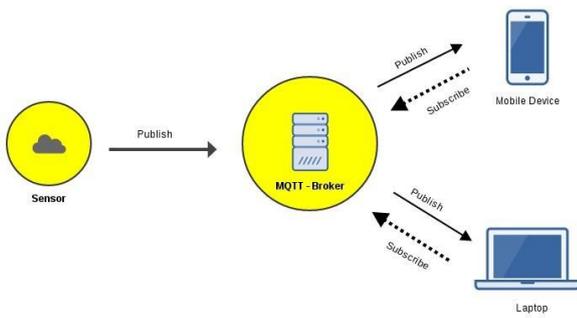
Untuk pemrogramannya sendiri kita bisa menggunakan ESPlorer untuk *Firmware* berbasis NodeMCU dan menggunakan putty sebagai *terminal control* untuk AT Command.



Gambar 2.2 Bentuk fisik modul Wifi ESP8266-01

4. Protokol MQTT

Message Queue Telemetry Transport atau yang biasa disebut MQTT yaitu protokol untuk komunikasi yang bersifat machine to machine atau M2M dan bekerja di layer ketujuh atau aplikasi dan bersifat *lightweight message*. Meskipun koneksi dalam keadaan terputus, semua pesan yang dikirim akan terjamin oleh protokol MQTT. Metode komunikasi *publish/subscribe* merupakan metode pengiriman yang digunakan oleh protokol MQTT. Pesan pada MQTT dikirim ke broker dan berisi topik yang dikirimkan oleh publisher. Kemudian topik tadi diolah untuk diteruskan ke subscriber berdasarkan dari permintaan pengguna.[5]



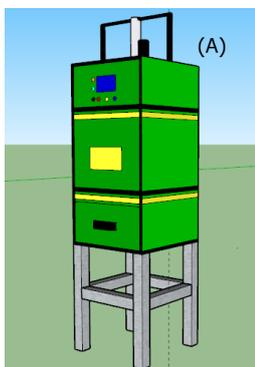
Gambar 2.3 Alur kerja dari protocol MQTT

5. Tempat Sampah Otomatis *Lectrobin*

Lectrobin merupakan sebuah tempat sampah otomatis yang dikhususkan untuk menampung sampah plastik. Tempat sampah ini dirancang dengan fitur pemilah sampah otomatis untuk bisa membedakan antara sampah anorganik dengan sampah-sampah organik ataupun logam. Selain itu terdapat juga fitur pendeteksian kapasitas untuk memberitahu kapasitas tempat sampah tersebut. Tempat sampah ini juga dilengkapi dengan pemanasan dan motor silinder untuk keperluan kompresi sampah plastik.



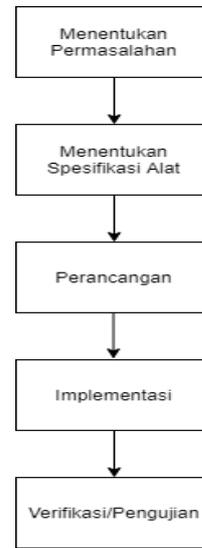
Gambar 2.3 Tampilan fisik *Lectrobin* tampak depan



Gambar 2.4 Tampilan fisik *Lectrobin* tampak samping

III. METODE PENELITIAN

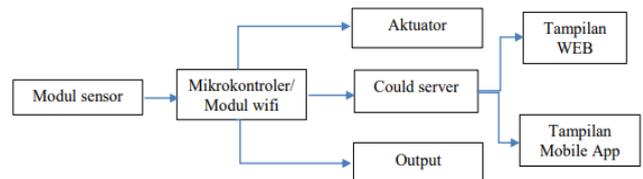
Metode penelitian yang digunakan dalam penerapan konsep IoT pada tempat sampah ini memiliki lima alur berikut adalah alurnya pada Gambar 3.1.



Gambar 3. 1 Metode Penelitian Yang Digunakan

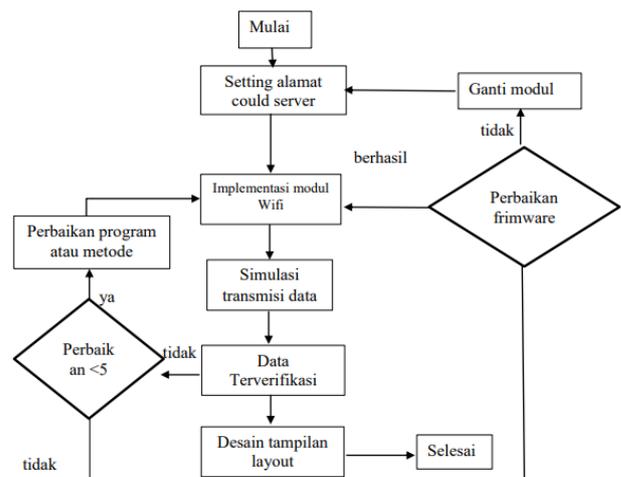
IV. PERANCANGAN SISTEM

A. Diagram Blok



Gambar 4. 1 Diagram Blok Sistem

B. Flowchart Implementasi



Gambar 4. 2 Flowchart Perancangan

V. IMPLEMENTASI DAN PENGUJIAN

1. Modul Wifi ESP8266

Pada implementasi modul Wifi ESP8266, modul dipasang secara serial dengan mikrokontroler arduino mega. Komunikasi data serial menggunakan dua buah pin yaitu pin RX untuk menerima data dan pin TX untuk mengirimkan data.

```
COM4

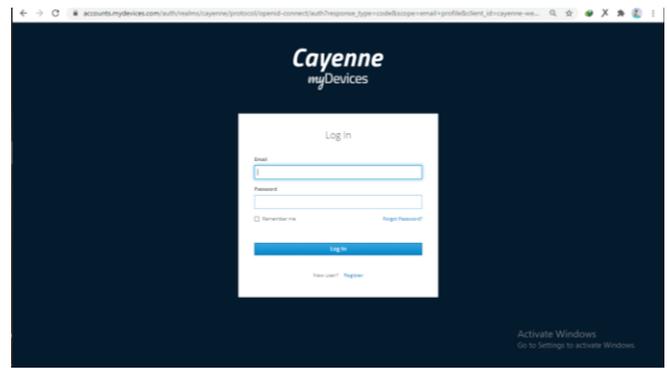
[1552] Failed to enable MUX
[8525] Connected to WiFi
[8525] Connecting to mqtt.mydevices.com:1883
[15381] Connected
[16595] Publish: topic 4, channel 65534, value Arduino Mega, subkey , key
[17549] Publish: topic 6, channel 65534, value ATmega2560, subkey , key
[18684] Publish: topic 7, channel 65534, value 16000000, subkey , key
[19480] Publish: topic 5, channel 65534, value 1.3.0, subkey , key
[20598] Publish: topic 8, channel 65534, value ESP8266Shield, subkey , key
```

Gambar 5.1 Hasil pengujian modul wifi pada arduino mega

Pada pengujian ini, memerlukan bantuan komunikasi serial untuk melihat kondisi koneksi esp266-01 dapat terhubung dengan jaringan internet yang telah diinisialisasikan dalam source code. Berikut adalah hasil pengujannya, WIFI esp8266-01 diinisialisasikan akan terkoneksi dengan jaringan hotspot dengan nama akses poin “vivo 1901” beserta passwordnya yang telah di deklarasikan pada *hardcode*. Pada gambar 5.1, terlihat bahwa perintah untuk menghubungkan esp8266-01 dengan jaringan hotspot telah berhasil dengan status “*connected to wifi*”. Selanjutnya modul wifi akan menghubungkan dengan *could server cayenne* lalu melakukan publish data.

2. Dashboard Cayenne

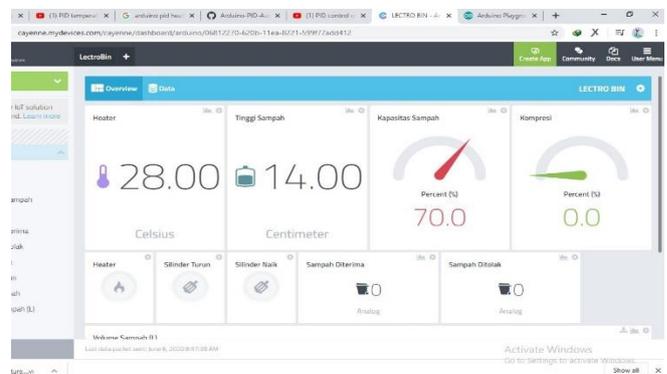
Langkah awal untuk implementasi *dashboard cayenne* adalah dengan melakukan login pada aplikasi atau web *my device cayenne*. Selanjutnya dilakukan konfigurasi untuk menyesuaikan *board mikrokontroler* dan *shield* yang digunakan, pada percobaan ini digunakan *board arduino mega* dan *shield ESP8266*. Setelah konfigurasi tersebut salin kode autentikasi MQTT untuk keperluan keamanan untuk mengamankan *dashboard* yang telah dibuat sebelumnya.



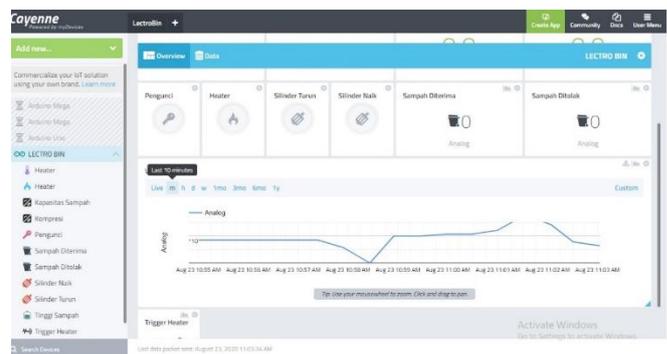
Gambar 5.2 Tampilan login pada web my device cayenne

MQTT Username	841490a0-0e0f-11ea-84bb-8f71124cfdff
MQTT Password	95ec26f94abef35924008fe292c5b43b54fdb932
Client ID	06812270-620b-11ea-8221-599f77add412

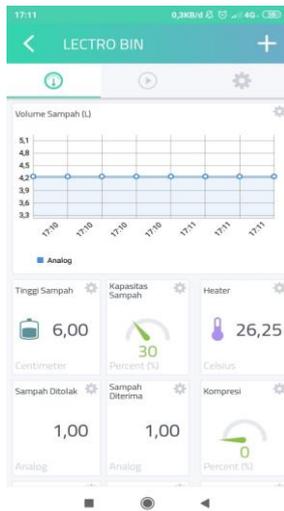
Gambar 5.3 Kode autentikasi MQTT



Gambar 5.4 Tampilan dashboard pada web cayenne



Gambar 5.5 Tampilan grafik pada dashboard web cayenne



Gambar 5. 6 Tampilan *dashboard cayenne* aplikasi android

Setelah semua konfigurasi selesai dilakukan maka proses selanjutnya adalah dengan melakukan pengujian, pengujian dilakukan dengan melakukan pembacaan hasil masing-masing sensor yang dikirimkan oleh mikrokontroler kedalam *cloud server cayenne*. Selain melakukan pengiriman data seperti yang nampak pada serial monitor pada gambar 5.7, pengujian juga dilakukan dengan melakukan penerimaan data hasil inputan user pada aplikasi *cayenne* untuk keperluan control kedalam mikrokontroler, hasil dari proses penerimaan data dapat dilihat pada gambar 5.8. Hasil pengiriman data oleh mikrokontroler akan tampil pada dashboard sesuai channel dan parameter masing-masing yang telah dibuat sebelumnya, sedangkan proses pengiriman data oleh *cayenne* dapat terverifikasi apabila mikrokontroler melakukan proses eksekusi berupa *controlling*.

```
[60626] Publish: topic 1, channel 1, value 1, subkey null, key null
[60895] Publish: topic 1, channel 2, value 1, subkey null, key null
[61134] Publish: topic 1, channel 4, value 27.000, subkey c, key temp
[61319] Publish: topic 1, channel 5, value 0.000, subkey null, key null
[61544] Publish: topic 1, channel 6, value 0, subkey cm, key prox
[61783] Publish: topic 1, channel 7, value 0, subkey p, key prox
[61962] Publish: topic 1, channel 12, value 0, subkey p, key prox
[62132] Publish: topic 1, channel 13, value 0, subkey d, key digital_sensor
```

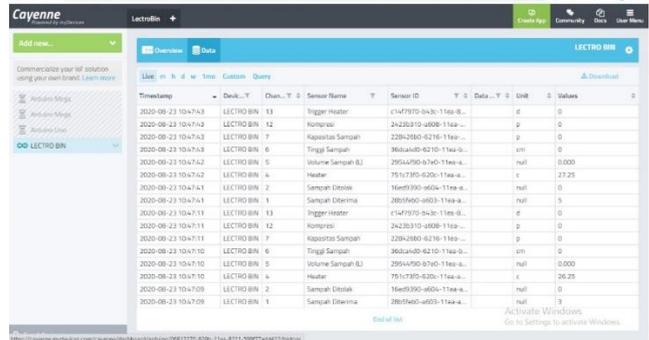
Gambar 5. 7 Tampilan serial monitor saat proses pengiriman data kedalam server *cayenne*

```
[63812] Message received: topic 2, channel 10
[63813] In: value 0, channel 10
[63830] publishState: topic 1 channel 10
[63873] Publish: topic 1, channel 10, value 0, subkey , key
[64186] Send response: xFryHG26HsxJqqx
```

Gambar 5. 8 Tampilan serial monitor saat proses penerimaan data dari server *cayenne*

3. Penyimpanan *cayenne*

Pada bagian penyimpanan, pengujian dilakukan dengan melihat apakah data hasil pengiriman oleh mikrokontroler dapat ditampung dan tersimpan pada bagian penyimpanan. Selain itu keberhasilan dari pengujian ini terlihat apabila hasil pembacaan data pada mikrokontroler sama dengan data hasil pengiriman yang tersimpan pada server *cayenne*.



Gambar 5. 10 Tampilan *dashboard penyimpanan* pada web *cayenne*

```
Suhu Heater = 27.00C || Tinggi = 0cm || Kapasitas Penampungan = 0% || Jarak Kompresi = 4cm || button = 0***
Suhu Heater = 27.00C || Tinggi = 0cm || Kapasitas Penampungan = 0% || Jarak Kompresi = 4cm || button = 0***
[66841] Connection ok
Suhu Heater = 27.00C || Tinggi = 0cm || Kapasitas Penampungan = 0% || Jarak Kompresi = 4cm || button = 0***
Suhu Heater = 27.00C || Tinggi = 0cm || Kapasitas Penampungan = 0% || Jarak Kompresi = 4cm || button = 0***
```

Gambar 5. 11 Tampilan serial monitor hasil pembacaan

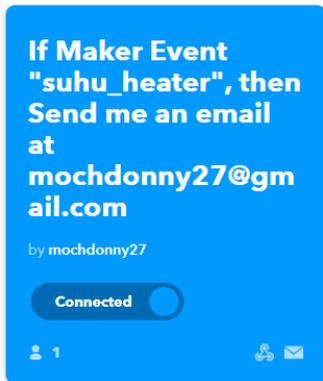
Pada serial monitor hasil pembacaan sensor pada gambar 5.11 dapat dilihat pada parameter yang ada pada serial monitor, status dari suhu adalah berkisar 27°C , tinggi sampah adalah 0 dan kapasitasnya masih dalam keadaan kosong yaitu 0%.Berikutnya pada gambar 5.10 dapat terlihat bahwasanya data hasil penyimpanan pada *cloud server cayenne* sama dengan pembacaan pada serial monitor, hal ini menandakan bahwa data yang dikirimkan berhasil sampai ke *cloud server* dan tersimpan dengan baik.

4. Notifikasi SMS dan *Email*

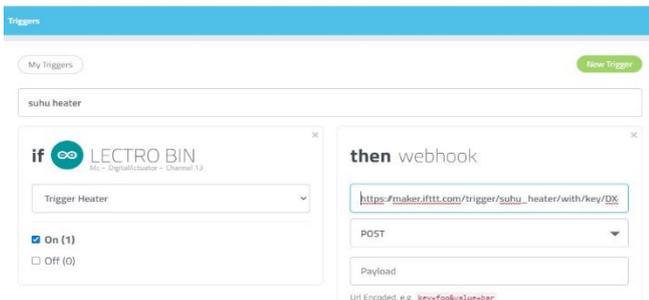
Notifikasi di desain untuk memberikan peringatan kepada user bahwasanya suhu pemanas yang telah di set pada tempat sampah telah mencapai set poinnya, hal ini penting mengingat pemanas pada tempat sampah merupakan hal yang perlu diperhatikan mengingat penggunaan dayanya yang besar sehingga membutuhkan notifikasi yang akurat untuk meminimalisir pemborosan daya tersebut. Implementasi pada notifikasi dilakukan dengan cara menghubungkan antara webhook dan IFFT (*If This Then That*) dengan web *cayenne* pada bagian *trigger*, proses tersebut dimulai dengan mengkonfigurasi pada webhook untuk menyiapkan fungsi *callback* agar data *trigger* yang ditentukan pada web *cayenne* dapat diterima dan dieksekusi oleh URL Email pada IFFT. Setelah konfigurasi webhook dan IFFT untuk mendapatkan URL layanan email, selanjutnya URL tersebut kita salin dan *copy* kedalam *dashboard trigger* dengan mengganti pada kolom setup terlebih dahulu dari *notify* menjadi *webhook*. Penggunaan

webhook dan IFFT pada notifikasi bertujuan untuk memberikan pesan dengan kalimat custom yang diinginkan. Selanjutnya untuk notifikasi SMS cukup dengan mengkonfigurasi pada *dashboard trigger cayenne* dengan memilih setup *notify* dan memasukkan nomer *handphone* yang diperlukan, Penggunaan notifikasi SMS terbatas pada penggunaan kata-kata dan jumlah pesan yang dikirimkan. Layanan ini memiliki *default* pengiriman pesan sebanyak 90 buah selama satu bulan.

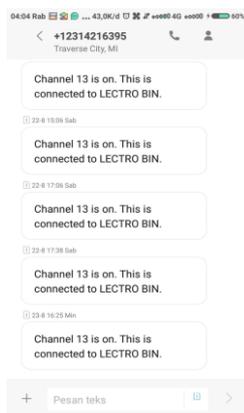
My Applets



Gambar 5. 12 Tampilan Applets pada webhook



Gambar 5. 13 Tampilan *dashboard trigger cayenne*



Gambar 5. 14 Notifikasi SMS yang diterima oleh pengguna



Gambar 5. 15 Notifikasi Email yang diterima oleh pengguna

Berikutnya untuk memverifikasi pengujian yang dilakukan telah berhasil, maka dilakukan pengetesan dengan menyalakan pemanas dan menunggunya mencapai nilai setpoint suhu yang ditentukan yaitu 80°C. Berikutnya dapat dilihat pada gambar 5.14 dan 5.15 notifikasi berupa SMS dan Email telah muncul pada *handphone* pengguna, hal ini menandakan bahwa pengujian untuk sub notifikasi telah berhasil dijalankan.

VI. PEMBAHASAN

Penerapan sitem IoT pada tempat sampah otomatis sudah berjalan dengan baik, hal itu dibuktikan dengan berfungsi *dashboard* utama, penyimpanan, dan *trigger* pada aplikasi dan web *my device cayenne*. Pada konfigurasi modul wifi ESP8266 berjalan dengan normal dengan terhubungnya jaringan hotspot yang telah disediakan dengan modul tersebut, selanjutnya *hardware* akan disambungkan kedalam situs *mydevicecayenne.com* sebagai *could server* utama.

Berikutnya pengujian dilakukan untuk melihat bagaimana respon dari *dashboard* utama *cayenne*, pengujian ini bertujuan untuk melihat secepat apa respon yang ditampilkan pada *dashboard* utama ketika menerima data hasil pembacaan mikrokontroler dan sebaliknya respon pada mikrokontroler ketika menerima data hasil pengiriman dari *cloud server cayenne*. Berikut ini adalah respon waktu yang diperlukan untuk masing-masing pengujian diatas.

Tabel 6.1 Pengukuran Respon Waktu Saat Mengirimkan Data

No	Parameter Data	Waktu yang dibutuhkan untuk menampilkan data dalam aplikasi (detik)
1	suhu	2,41
2	suhu	2,99
3	suhu	1,37
4	suhu	2,15
5	suhu	1,51

6	tinggi sampah	2,42
7	tinggi sampah	1,81
8	tinggi sampah	2,92
9	tinggi sampah	1,32
10	tinggi sampah	1,67
11	Kapasitas sampah	2,25
12	Kapasitas sampah	2,47
13	Kapasitas sampah	2,58
14	Kapasitas sampah	1,94
15	Kapasitas sampah	2,72
16	Volume sampah	3,22
17	Volume sampah	1,82
18	Volume sampah	2,73
19	Volume sampah	2,58
20	Volume sampah	1,88
Rata-Rata		2,37

Tabel 6.2 Pengukuran Respon Waktu Saat Menerima Inputan Data

No	Tombol Perintah	Waktu respon yang dibutuhkan untuk eksekusi pada hardware (detik)
1	Heater	1,92
2	Heater	1,61
3	Heater	3,06
4	Heater	1,68
5	Heater	1,39
6	Lockdoor	1,11
7	Lockdoor	1,45
8	Lockdoor	1,37
9	Lockdoor	3,21
10	Lockdoor	1,33
11	Silinder Turun	2,03
12	Silinder Turun	2,42
13	Silinder Turun	2,81
14	Silinder Turun	2,25
15	Silinder Turun	2,57
16	Silinder Naik	2,82
17	Silinder Naik	2,63
18	Silinder Naik	3,31
19	Silinder Naik	2,92
20	Silinder Naik	3,12
Rata-rata		2,20

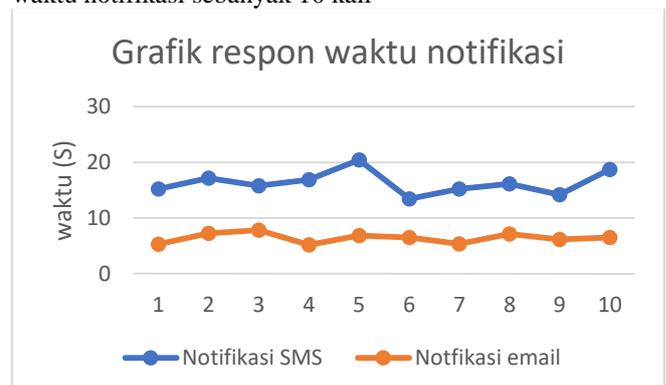
Dalam sistem *lectrobin* proses pengiriman data pembacaan dilakukan sebanyak 30 detik sekali sedangkan untuk proses penerimaan data dari aplikasi akan dapat direspon dengan waktu yang leluasa. Dilihat dari tabel 6.1

dilakukan percobaan waktu respon untuk menampilkan data sebanyak 20 kali yang berisi beberapa parameter data. Percobaan tersebut dilakukan untuk mengetahui waktu yang di perlukan aplikasi *cayenne* untuk menampilkan data hasil pengiriman mikrokontroler. Percobaan dilakukan dengan mencatat waktu yang di perlukan aplikasi *cayenne* untuk menampilkan data menggunakan stopwatch. Dari percobaan tersebut di dapatkan rata – rata untuk menampilkan pada aplikasi *cayenne* yaitu 2,37 detik. Terdapat beberapa faktor yang memengaruhi waktu penerimaan data tersebut salah satu yang paling berpengaruh adalah kecepatan jaringan koneksi internet.

Pada tabel 6.2 dapat dilihat terdapat 20 data percobaan pengaktifkan tombol pada aplikasi *cayenne* yang digunakan untuk tujuan *controlling*. Percobaan ini dilakukan untuk melihat sejauh mana waktu yang dibutuhkan *hardware* untuk melakukan eksekusi perintah sesuai kegunaan tombol, ketika *user* menekan tombol pada aplikasi *cayenne*. Dari percobaan tersebut didapatkan nilai waktu rata-rata respon sebesar 2,20 detik, yang tidak jauh berbeda nilainya dengan respon waktu untuk menampilkan data yang telah diuji coba sebelumnya. Respon waktu ini tidak selamanya konstan mengingat yang sudah dijelaskan sebelumnya koneksi jaringan internet akan sangat mempengaruhi hal ini.

Berikutnya untuk sub penyimpanan, *dashboard* penyimpanan telah berfungsi dengan normal dan mampu menyimpan data secara detail dari parameter-parameter yang telah disiapkan sebelumnya. Data tersebut mencakup tinggi sampah, kapasitas sampah, volume sampah, jumlah sampah diterima dan ditolak, serta suhu pemanas. Data-data tersebut nantinya akan disimpan selama perangkat tempat sampah otomatis (*Lectrobin*) dalam keadaan menyala/*standby*, data parameter tersebut juga dapat kita akses kembali sewaktu waktu dengan mengatur *history* tanggal, bulan, bahkan tahun.

Dari hasil pengujian sub notifikasi yang dilakukan pada aplikasi *cayenne* berhasil memberikan notifikasi apabila suhu heater telah mencapai set point yang telah ditentukan. Notifikasi akan di berikan dalam bentuk email dan sms, untuk bentuk notifikasi email akan memanfaatkan web IFFT (*If This Then That*) juga dikenal sebagai IFTTT, adalah layanan berbasis *web freeware* yang menciptakan rantai pernyataan kondisional sederhana, yang disebut applet. Applet dipicu oleh perubahan yang terjadi dalam layanan web lain, Penggunaan *platform* IFFT ditujukan untuk membuat *custom* pesan kalimat yang akan dikirimkan ke pengguna, hal ini dipilih dengan alasan *platform cayenne* tidak memberikan fasilitas untuk memodifikasi pesan itu sendiri. Berikut hasil pengujian tampilan notifikasi Dilakukan percobaan respon waktu notifikasi sebanyak 10 kali



Gambar 6.1 Grafik Perbandingan Waktu Notifikasi Email dan SMS

Percobaan ini bertujuan untuk mengetahui waktu yang dibutuhkan aplikasi untuk memberikan peringatan ke pengguna sehingga dapat di dapatkan informasi efektifitas fitur ini. Percobaan tersebut dilakukan dengan mencatat waktu yang di perlukan aplikasi cayenne untuk memberikan notifikasi setelah parameter ambang batas terpenuhi. Rata – rata waktu respon peringatan berupa notifikasi email yang dikirimkan oleh platform IFFT lebih cepat di bandingkan dengan peringatan berupa sms yang di kirimkan oleh platform cayenne, sehingga peringatan pada berupa notifikasi email lebih efektif.

VII. KESIMPULAN

Penerapan konsep Internet of Things Pada tempat sampah otomatis (lectrobin) sudah berjalan dengan baik dan sesuai dengan perancangan yang di buat sebelumnya. Modul wifi ESP8266 berperan penting sebagai jalur komunikasi sekaligus penyedia akses internet terhadap hardware, proses pengiriman data yang dilakukan oleh hardware kedalam server cayenne menggunakan protokol MQTT dengan metode kerja publish and subscribe. Tampilan dashboard utama yang telah di desain dan di implementasikan sejauh percobaan sudah sesuai dan mampu menampilkan data-data sensor yang telah dikirim oleh mikrokontroler sebanyak 30 detik sekali, selain itu pengujian pada akses remote juga telah memenuhi standar yang diinginkan. Pada dashboard penyimpanan fungsi histori telah tampil dengan data yang masuk sesuai dengan pembacaan pada serial monitor, selain itu pada dashboard tersebut user dapat mengakses kemabali data-data yang lalu dengan mensetting parameter tanggal, bulan, dan tahun. Kecepatan respon yang didapat untuk menampilkan data pada aplikasi maupun web berniali rata-rata 2,37 detik, sedangkan pada akses remote respon waktu dari user menekan tombol hingga perintah esekusi dijalankan adalah sekitar 2,20 detik. Selanjutnya untuk kerja dari notifikasi sudah sesuai dengan keinginan yang dapat dilihat pada bagian pengujian. Penggunaan IFFT dan webhook dimaksudkan untuk memeberikan pesan custom yang tidak disediakan oleh my device cayenne. Respon waktu untuk menampilkan notifikasi saat trigger bekerja dan sampai pada pengguna memiliki perbedaan antara notifikasi sms dan email. Pada pengujian notifikasi email jauh lebih cepat terkirim dibandingkan dengan notifikasi sms yang ditampilkan.

REFERENCES

- [1] Weiser, M.: The Computer for the 21st Century. Scientific American 265(9):66–75 (1991)
- [2] IEEE, Issue 1 MAY 2015, "Towards Definition Internet of Things (IoT)", iot.ieee.org.
- [3] Mattern, F., Floerkemeier, C.: From the Internet of Computers to the Internet of Things.
- [4] Harry Yuliansyah, “Uji Kinerja Pengiriman Data Secara Wireless Menggunakan Modul ESP8266 Berbasis Rest Architecture”, Jurnal Rekayasa dan Teknologi Elektro, Vol. 10, No. 2, 2016
- [5] Zavero Brillianata Abilovani , Widhi Yahya , Fariz Andri Bakhtiar, “Implementasi Protokol MQTT Untuk Sistem Monitoring Perangkat IoT”, Jurnal Pengembangan Teknologi Komunikasi dan Komputer, Vol. 2, No.12, pp. 7521-7527, 2018
- [6] Zeeshan Abbas,Wonyong Yoon, (2015) . “A Survey on Energy Conserving Mechanisms for the Internet of Things: Wireless Networking Aspects”, Busan: Department of Electronics Engineering, Dong-A University