BAB III. PERANCANGAN SISTEM

3.1 Definisi Masalah

Penumpukan sampah plastik yang semakin mengkhawtirkan dapat memicu permasalahan lingkungan yang lebih serius. Sampah plastik diketahui memiliki rantai karbon yang panjang, sehingga menyulitkannya terurai pada tanah. Selain itu sampah plastik juga menggangu keseimbangan ekosistem pada laut yang menyebabkan banyak hewan-hewan laut yang mati atau bermutasi. Berangkat dari latar belakang tersebut dibutuhkan sebuah inovasi baru yaitu tempat sampah yang mampu menampung sekaligus mengolah sampah plastik.

Untuk menunjang sekaligus memudahkan pengoperasian dan pemantauan tempat sampah tersebut, diperlukan pengimplementasian konsep IoT yang nantinya membentuk sebuah software berbasis *mobile application* yang dapat di instal pada *smartphone* atau dioperasikan melalui web dengan *personal computer*. Agar *software* dapat terhubung dengan *hardware*, maka diperlukan sebuah modul wifi untuk menyediakan jaringan internet pada *hardware* untuk menunjang proses komunikasi keduanya.

3.2 Analisa Kebutuhan

Sistem pengoperasian dan pemantauan *hardware* tempat sampah dirancang menggunakan konsep IoT yang nantinya proses keseluruhan kerja *hardware* dapat dilakukan secara *wireless*. Karena terhubung secara *wireless* maka dibutuhkan sebuah sistem perangkat lunak berbentuk aplikasi *mobile* atau web yang mampu memenuhi kebutuhan seperti *monitoring* aktual, *controlling*, pemberian notifikasi, analisa penggunaan, serta *user interface* yang menarik untuk memberikan kemudahan dalam penggunaan.

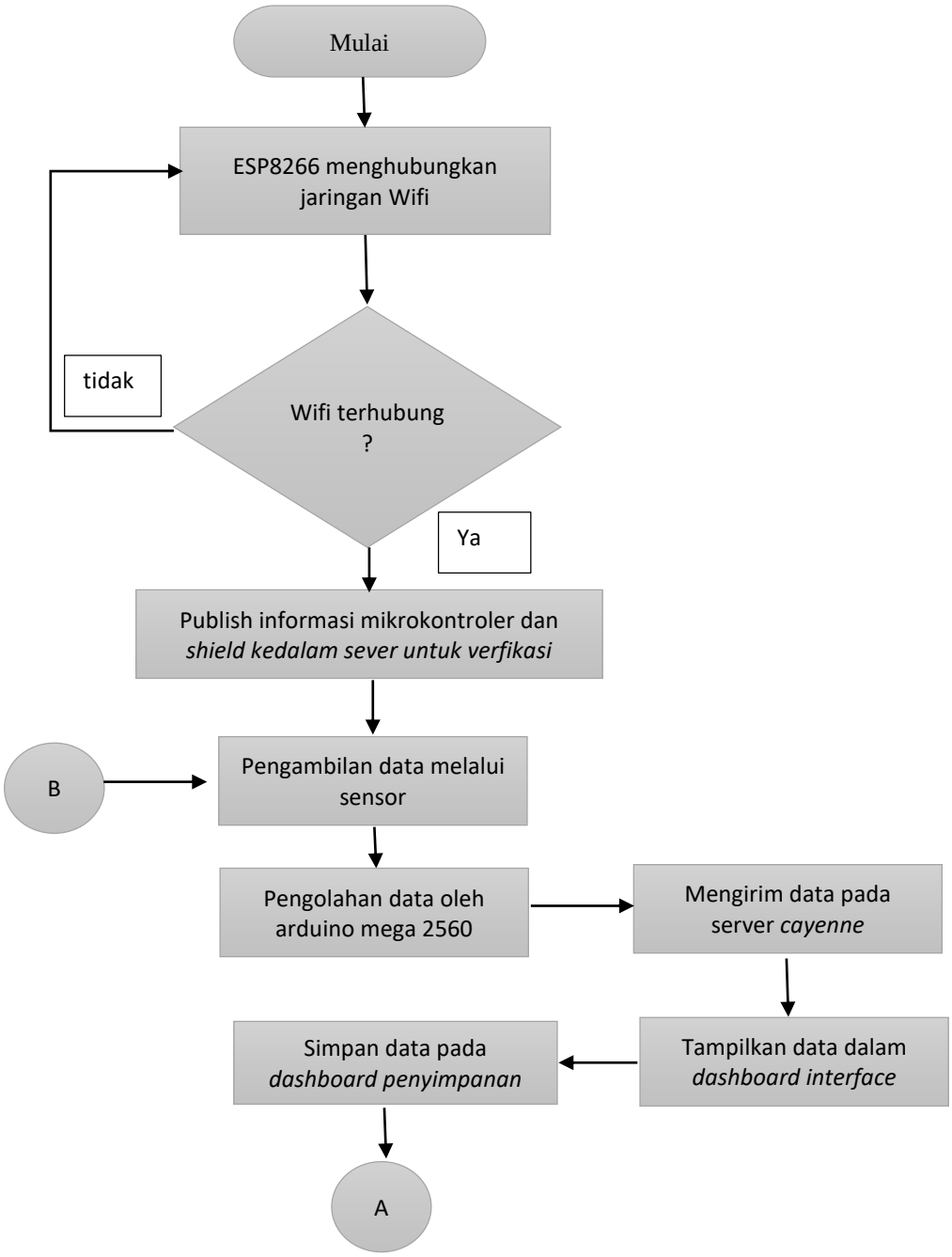
3.3 Spesifikasi Sistem Perangkat lunak

Spesifikasi sistem perangkat lunak dari penelitian ini adalah:

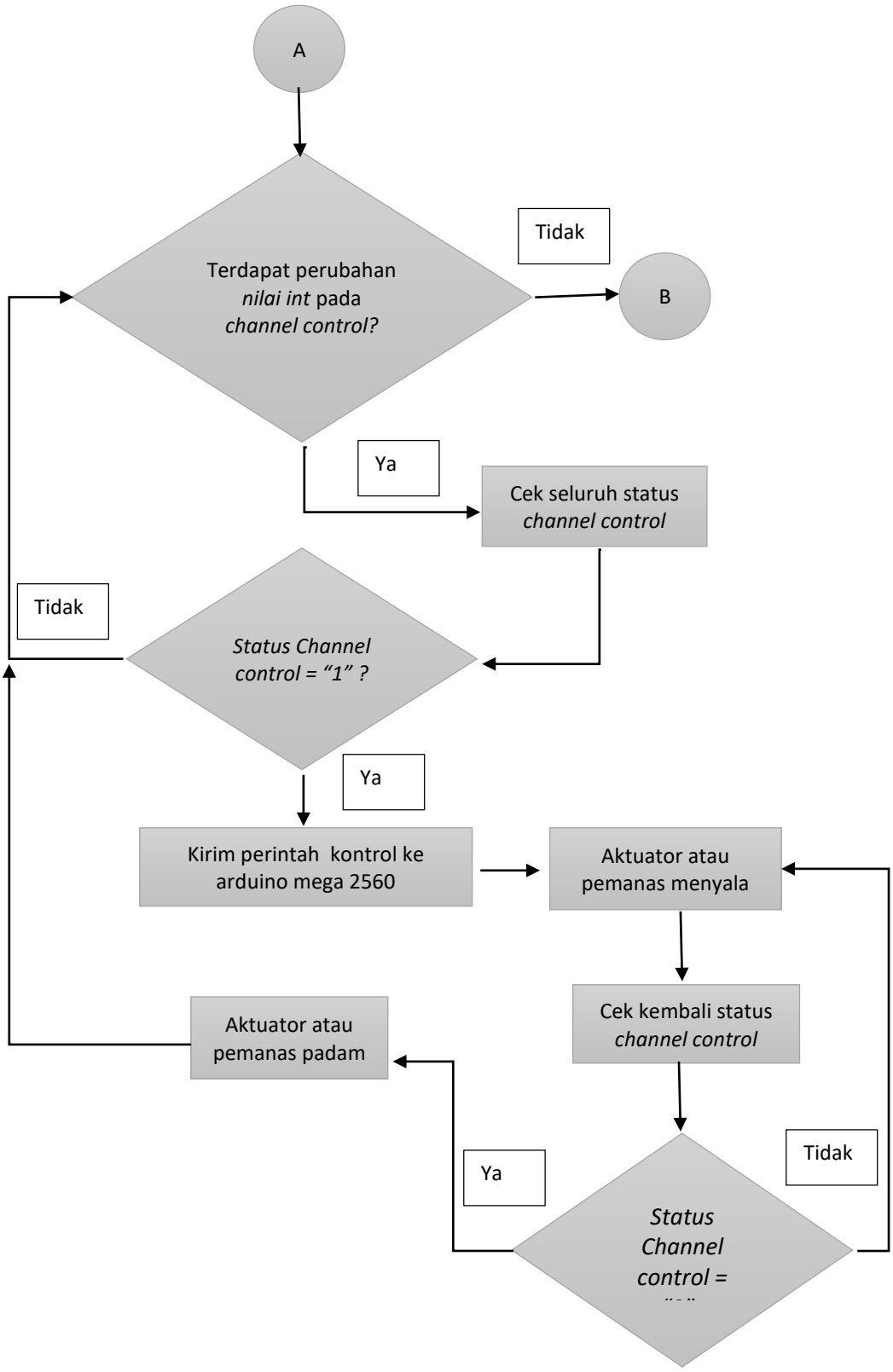
1. Aplikasi *mobile*/web menerima data sensor yang dikirimkan oleh mikrokontroler untuk nantinya ditampilkan pada layar *smartphone* atau komputer pengguna.
2. Terdapat tampilan *login user* untuk menunjang keamanan penggunaan.
3. Terdapat tampilan dashboard pada aplikasi mobile/web yang berisi parameter-parameter data pembacaan sensor ataupun parameter data yang telah diolah oleh mikrokontroler.
4. Aplikasi *mobile*/web memiliki tombol remote yang digunakan untuk mengontrol silinder kompresi, pemanas, dan kunci *lockdoor.*
5. Pada aplikasi *mobile/*web terdapat *widget* grafik untuk melihat pembacaan volume sampah bedasarkan history waktu.
6. Aplikasi dilengkapi dengan notifikasi SMS dan email untuk memperingatkan pengguna ketika suhu dari pemanas telah mencapai batas pengoperasian.

3.4 Flowchart Sistem Perangkat Lunak

Alur kerja dari *software mobile app atau* web dapat dilihat pada gambar 3.1 dan 3.2, dari gambar *flowchart* tersebut dapat diketahui proses pertama adalah dengan menyambungkan modul ESP8266 dengan jaringan Wifi. Berikutnya setelah wifi terhubung, sensor akan mengambil seluruh data yang diperlukan dan mengolahnya melalui mikrokontroler. Setelah pengolahan data selesai, maka data tersebut siap dikirimkan kedalam web server *cayenne*. Data tersebut nantinya akan disimpan kedalam *dashboard* penyimpanan sekaligus ditampilkan pada *dashboard interface,* yang nantinya dapat diakses oleh perangkat yang terkoneksi, baik itu *smartphone* dan juga personal komputer. Untuk proses pengontrolan, web server akan membaca perubahan nilai integer pada status *channel control.* Ketika salah satu *channel control* mengalami perubahan nilai integer dari “0” ke “1”, maka web server akan mengirimkan perintah kepada mikrokontroler untuk melakukan esekusi *control* kepada aktuator sesuai dengan *chnnel* yang mengalami perubahan.

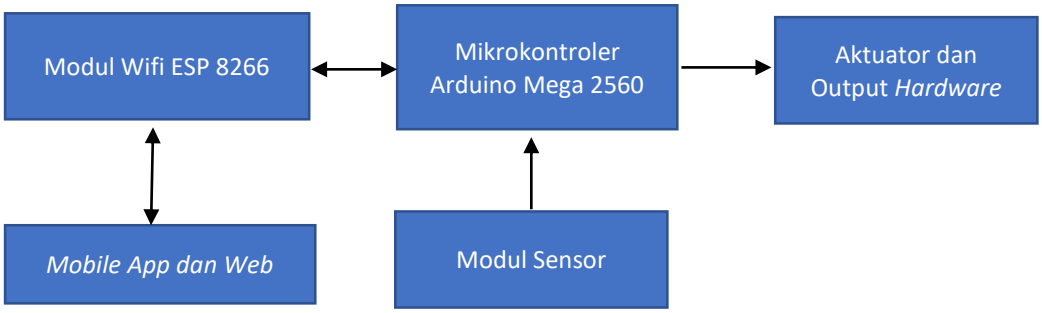


Gambar 3. 1 *Flowchart* system perangkat lunak bag.1

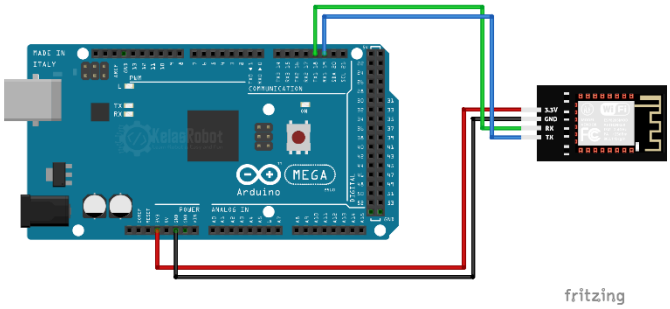


Gambar 3. 2 *Flowchart* system perangkat lunak bag.2

3.5 Perancangan *Hardware*

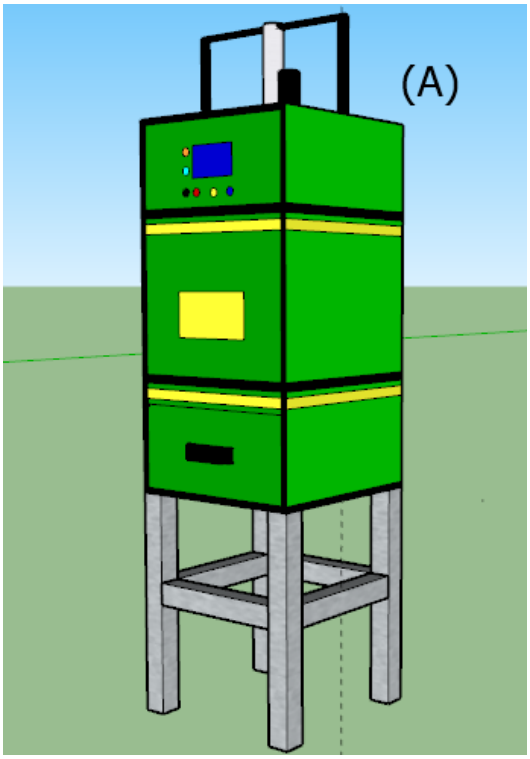


Gambar 3. 3 Blok diagram perancangan *hardware*

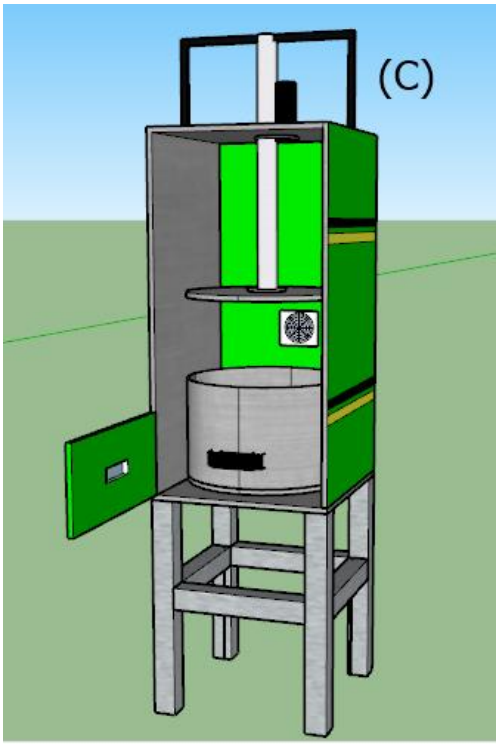


Gambar 3. 4 Konfigurasi arduino mega dengan ESP8266-01

Dapat dilihat pada gambar 3.1 perancangan *hardware* melibatkan modul sensor, komponen aktuator dan output lalu modul wifi ESP8266-01. Modul sensor digunkan sebagai pengukur besaran fisis dari sistem yang dibutuhkan, sensor-sensor yang digunakan antara lain adalah sensor suhu, sensor jarak, sensor pendeteksi logam dan non logam. Berikutnya untuk aktuator yang digunakan antara lain adalah motor servo, motor silinder, pemanas heater dan kipas *exhaust*. Sebagai pengendali utama digunakan mirkokontroler arduino mega 2560 yang dihubungkan dengan chip ESP8266-01 untuk mendukung koneksi WiFi, konfigurasi sambungan tersebut dapat dilihat pada gambar 3.2. dikarenakan pada penelitian ini berfokus pada pengimplementasian konsep IoT, maka konfigurasi hardware yang ditampilkan hanya sebatas mikrokontroler dengan modul wifi ESP8266-01 yang menunjang keberlangsungan sistem IoT. Desain keseluruhan *hardware* tempat sampah dapat dilihat pada gambar 3.3 dan gambar 3.4 dibawah ini.

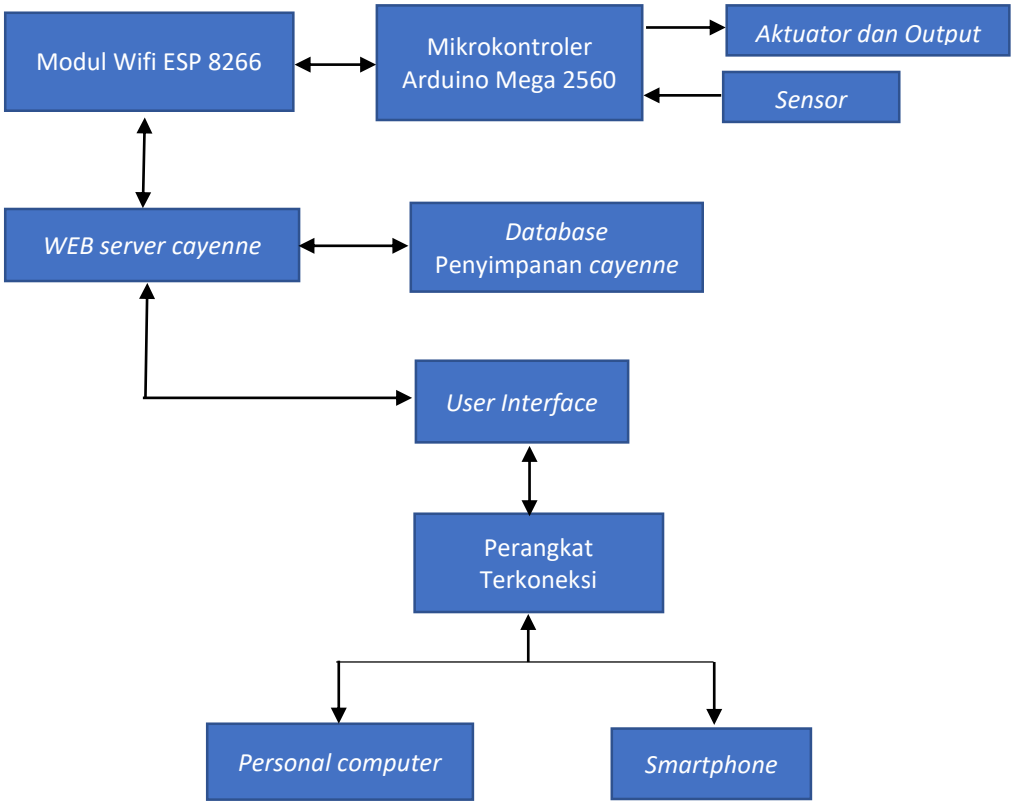


Gambar 3. 5 Desain keseluruhan *hardware* tampak depan



Gambar 3. 6 Desain keseluruhan *hardware* tampak dalam

3.6 Perancangan Perangkat Lunak



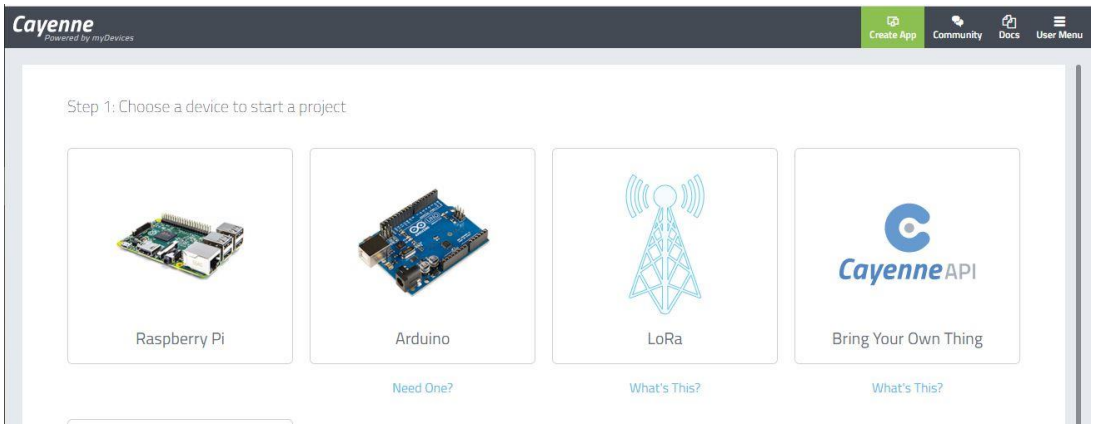
Gambar 3. 7 Blok diagram perancangan perangkat lunak

Pada penelitian ini digunakan beberapa software dan *platform IoT* dalam perancanggannya, pengunaan software pertama adalah Arduino IDE yang berfungsi untuk memprogram board mikrokontroler agar dapat memproses input dan output dari sensor yang terhubung. Selain itu, dengan bantuan software Arduino IDE nantinya mikrokontroler dapat diprogram untuk tersambung dengan jaringan internet melalui bantuan modul wifi ESP8266-01. Berikutnya untuk *platform* yang digunakan pada perancangan *user interface* sekaligus menjadi *database* penyimpanan adalah *my device cayenne*. *Platform* ini cukup dihandalkan karena penggunaannya yang mudah dan kompatibel dengan hampir seluruh *board* mikrokontroler yang beredar dipasaran. Berikut ini adalah lagkah-langkah saat melakukan perancangan system perangkat lunak.

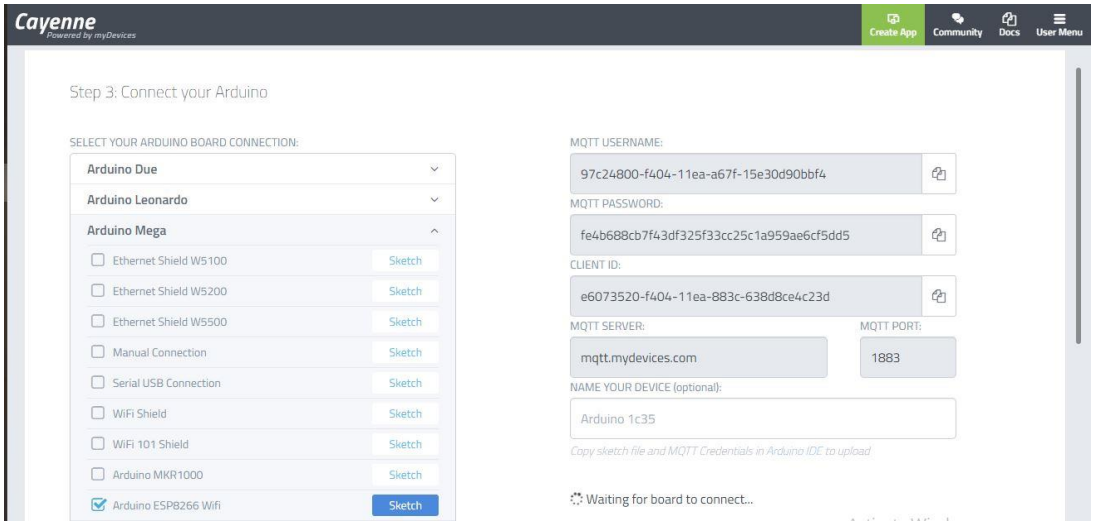
3.5.1 *Setting* perangkat mikrokontroler dengan web server *cayenne*

Sebelum melakukan konfigurasi pastikan terlebih dahulu ketersedian akun *my device cayenne* jika belum memiliknya, silahkan daftarkan terlebih dahulu akun pengguna demi keamanan pada saat pengimplementasian dan uji coba. Langkah awal yang diperlukan adalah dengan membuka situs *my device*

cayenne lalu lakukan login menggunakan akun yang telah didaftarkan sebelumnya. Setelah itu terdapat tampilan menu untuk memimilih jenis *board* mikrokontroler beserta shield yang ingin dipakai. Pada perancangan ini digunakan mikrokontroler arduino mega 2560 dengan tambahan *shield* ESP8266 untuk menunjang konektivitas.



Gambar 3. 8 Tampilan menu pemilihan mikrokontroler pada *platform cayenne.*

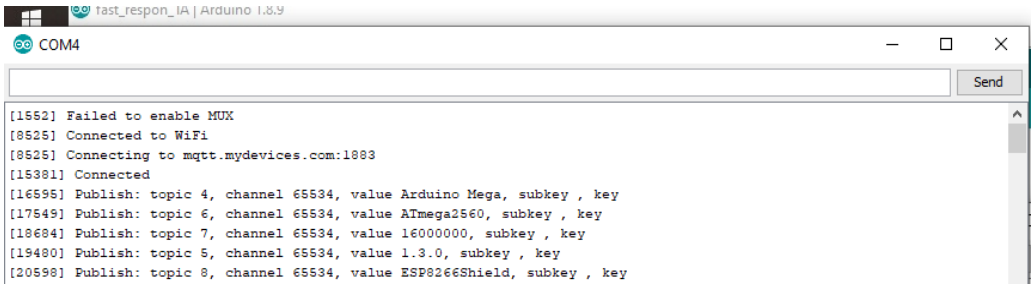


Gambar 3. 9 Tampilan konfigurasi mikrokontroler pada *platform cayenne.*

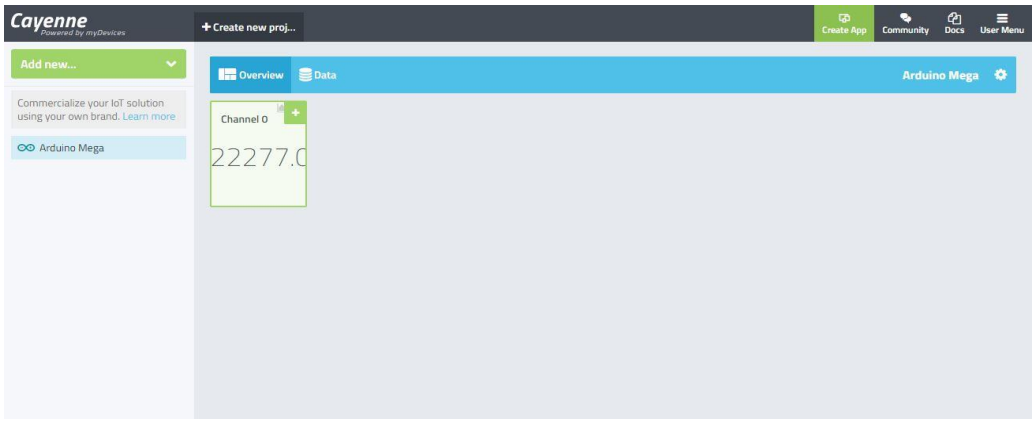
3.5.2 *Connect* ESP8266 dengan *cloud server cayenne*

Langkah awal yang diperlukan untuk Berikutnya silahkan copy kode MQQT berupa *username*, *password*, dan *ClientID* yang tertera pada gambar 3.7 kedalam program yang telah dibuat sebelumnya, untuk tampilan *source program* dapat dilihat pada lampiran. Upload program yang telah dibuat pada *software* arduino IDE kedalam *board* mikrokontroler lalu buka serial monitor untuk melihat respon konektivitas apabila sudah terhubung maka respon tampilan serial monitor akan seperti gambar 3.8, setelahnya buka kembali *platform my device cayenne* lallu tunggu hingga tampilan berganti menjadi

dashboard user interface. Tampilan dari *dashboard interface* dapat dilihat pada gambar 3.9.



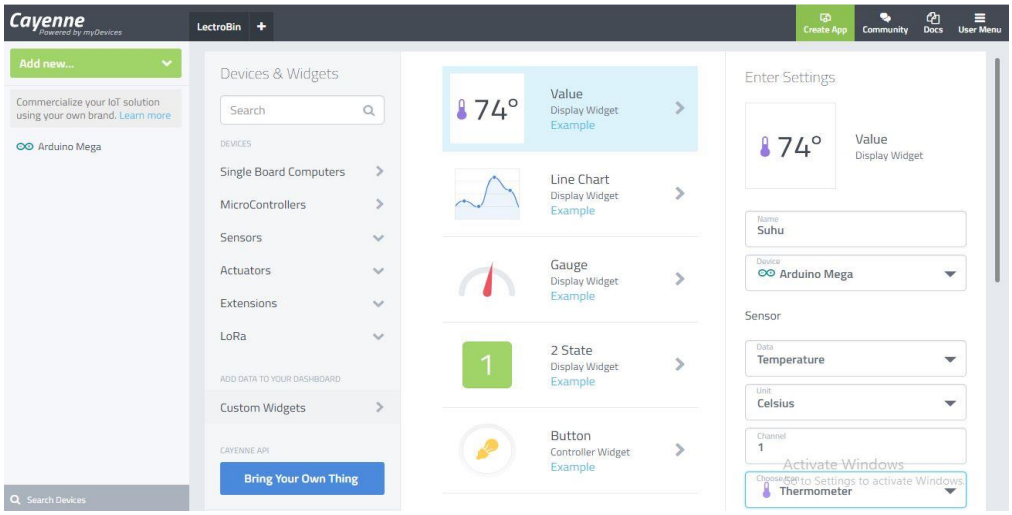
Gambar 3. 10 Tampilan respon ESP8266 pada serial monitor



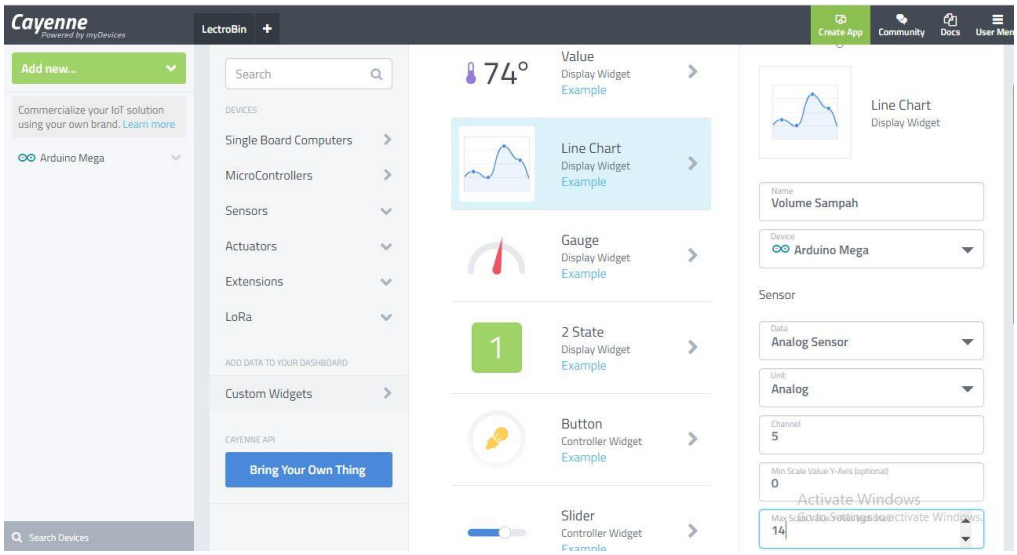
Gambar 3. 11 Tampilan awal *dashboard user interface*

3.5.3 Perancangan tampilan *dashboard interface*

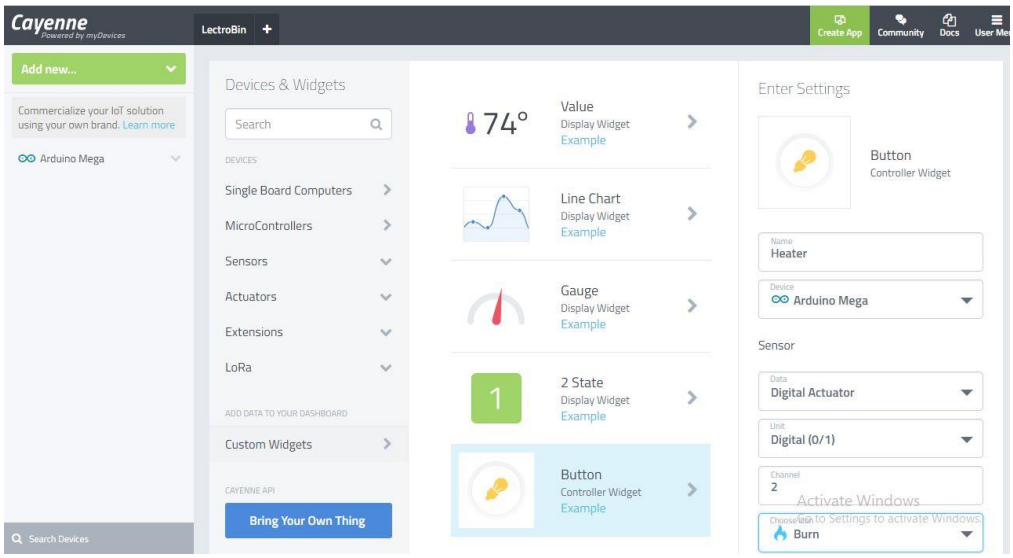
Untuk merancang tampilan dari *dashboard interface*, langkah awal yang harus dikerjakan adalah dengan pergi ke menu *device and widget* lalu pilih *custom widget* setelah itu sesuaikan tampilan *interface* yang diinginkan dengan memilih *widget* yang sesuai dengan selera anda. Konfigurasi yang perlu dilakukan pada saat memilih *widget*, diantaranya adalah masukan nama suhu, atur variable jenis data, satuan unit, dan terakhir yang penting adalah nomor *channel*. Nomor channel ini penting agar data dapat tampil sesuai dengan settingan pengguna dan digunakan untuk mengindentfikasi pada saat pengiriman data. Perlu diperhatikan juga untuk tidak memberi nomor yang sama antara parameter data satu dengan data yang lain untuk menghindari tertukarnya nilai. Berikut ini adalah contoh gambar petunjuk ketika mengatur masing-masing *widget* seperti konfigurasi menampilkan *value*, konfigurasi menampilkan chart/grafik, dan konfigurasi *button* untuk keperluan kontrol



Gambar 3. 12 Konfigurasi tampilan *value* pada *custom widget*



Gambar 3. 13 Konfigurasi tampilan grafik/chart pada *custom widget*

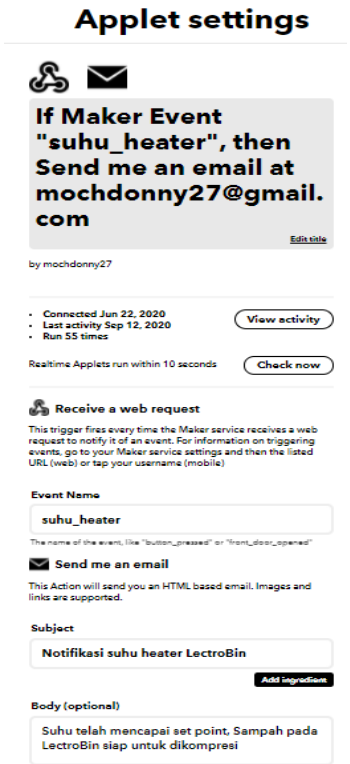


Gambar 3. 14 Konfigurasi funsgi *button* pada *custom widget*

3.5.4 Perancangan fungsi notifikasi

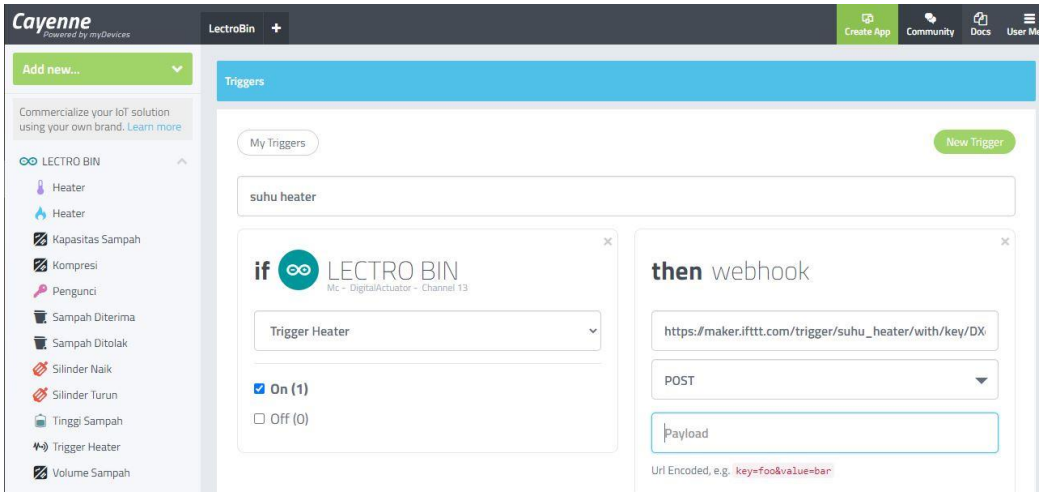
Proses terakhir pada perancangan sistem perangkat lunak adalah dengan memasang notifikasi pada sistem. Notifikasi yang dimaksud adalah notifikasi terkait salah satu keadaan pada *hardware* yang memiliki tingkat urgensi yang tinggi, pada kasus tempat sampah ini notifikasi digunakan untuk memberikan peringatan akan suhu pemanas, suhu pemanas tersebut perlu diperhatikan mengingat penggunaan daya listrik yang besar sekaligus bahaya komponen *hardware* lainnya yang sensitif terhada suhu panas.

Notifikasi yang akan digunakan berupa e-mail dan SMS (*Short Message Service).* Untuk notifikasi email perancangan, dilakukan dengan melibatkan satu *platform* web lagi yaitu adalah IFTT (*If This Then That*). 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* kalimat pada pesan yang akan dikirimkan ke pengguna, hal ini dipilih dengan alasan *platform my device cayenne* tidak memberikan fasilitas untuk memodifikasi pesan itu sendiri. Sedangkan untuk notifikasi SMS akan menggunakan layanan yang disediakan oleh *my device cayenne* dengan batasan pengiriman 90 buah pesan dalam 30 hari. Berikut ini adalah tampilan applet yang telah dibuat pada *platform IFFT.*

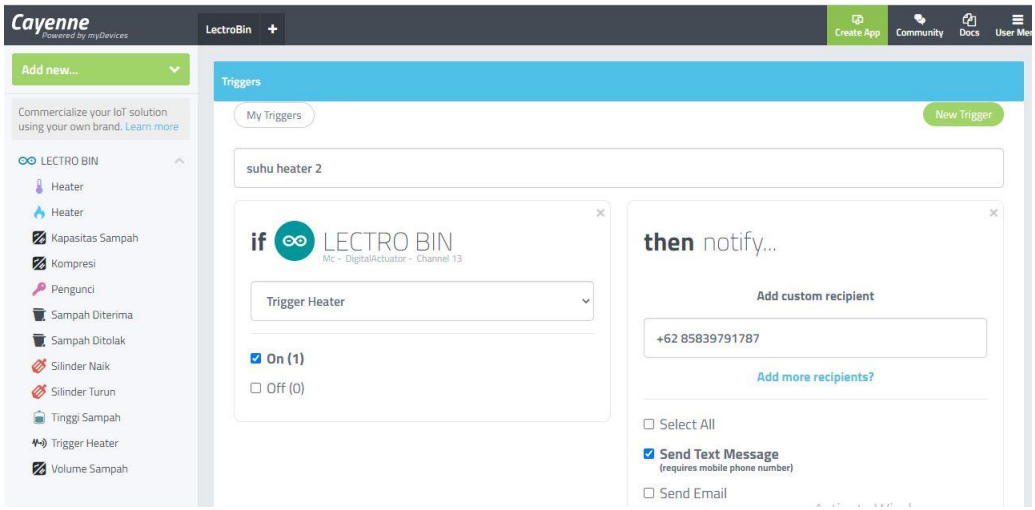


Gambar 3. 15 Konfigurasi applet notifikasi suhu pada *platform IFFT*

Konfigurasi yang perlu dilakukan untuk mengaktifkan fitur ini adalah dengan pergi ke menu *triggers* lalu pada bagian *if* pilih *channel triger* yang akan digunakan, pada kasus ini *channel triger* diberikan nama *Trigger heater*. *Channel triger heater* telah diprogram sebelumnya pada *software* arduino IDE dengan fungsi memberikan nilai *true* ketika suhu pada pemanas telah mencapai *set point.* Berikutnya pada fungsi *then* gunakan fitur *cayenne webhook trigger* untuk menjembatani pengiriman pesan custom yang sebelumnya dibuat pada *platform* IFFT. Untuk notifikasi layanan SMS, konfigurasi menu *triggers* yang dilakukan sama saja seperti pada notifikasi email yang membedakannya adalah pada bagian *then* cukup diganti dengan fitur *notify* lalu masukan nomer *handphone* pengguna dan lakukan *cheklist box* pada *Send Text Massage.* Konfigurasi kedua notifikasi tersebut bisa dilihat pada gambar 3.14 dan 3.15



Gambar 3. 16 Konfigurasi untuk mengaktifkan fitur notifikasi email



Gambar 3. 17 Konfigurasi untuk mengaktifkan fitur notifikasi SMS