

LAPORAN TUGAS AKHIR 2

Dispenser Pintar Pengharum Ruangan Terintegrasi



Penyusun:

Wildan Firdaus (20524189)

Faris Fatha Fallahian (20524203)

Program Studi Teknik Elektro

Fakultas Teknologi Industri

Universitas Islam Indonesia

Yogyakarta

2024

HALAMAN PENGESAHAN

Dispenser Pintar Pengharum Ruangan Terintegrasi

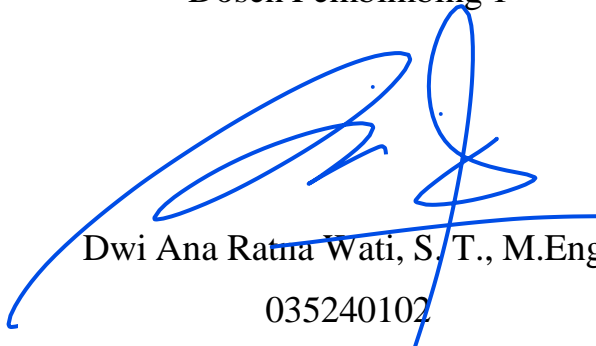
Penyusun:

Wildan Firdaus (20524189)

Faris Fatha Fallahian (20524203)

Yogyakarta, 6 Juli 2024

Dosen Pembimbing 1



Dwi Ana Ratna Wati, S. T., M.Eng.
035240102

Dosen Pembimbing 2



Tito Yuwono, S. T., M.Sc., Ph.D.
005240102

Program Studi Teknik Elektro

Fakultas Teknologi Industri

Universitas Islam Indonesia

Yogyakarta

2024

LEMBAR PENGESAHAN TUGAS AKHIR

Dispenser Pintar Pengharum Ruangan Terintegrasi



Disusun oleh:

Wildan Firdaus 20524189

Faris Fatha Fallahian 20524203

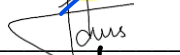
Telah dipertahankan di depan dewan penguji
pada tanggal: 16 Juli 2024

Susunan Dewan Penguji

Ketua Penguji

: Dwi Ana Ratna Wati, S.T., M.Eng. 

Anggota Penguji 1

: Firdaus, S.T., M.T., Ph.D. 

Anggota Penguji 2

: Ilham Unggara, S.Kom., M.Cs. 

Tugas akhir ini telah disahkan sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana Teknik

Tanggal: 2 Agustus 2024

Ketua Program Studi Teknik Elektro



Dwi Ana Ratna Wati, S.T., M.Eng. 

035240102

PERNYATAAN

Dengan ini kami menyatakan bahwa:

1. Tugas Akhir ini tidak mengandung karya yang diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi lainnya, dan sepanjang pengetahuan kami juga tidak mengandung karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.
2. Informasi dan materi Tugas Akhir yang terkait hak milik, hak intelektual, dan paten merupakan milik bersama antara tiga pihak, yaitu penulis, dosen pembimbing, dan Universitas Islam Indonesia. Dalam hal ini, penggunaan informasi dan materi Tugas Akhir terkait paten maka akan didiskusikan lebih lanjut untuk mendapatkan persetujuan dari ketiga pihak tersebut di atas.

Yogyakarta, 1 Agustus 2024



Wildan Firdaus (20524189)



Faris Fatha Fallahian (20524203)

DAFTAR ISI

DAFTAR ISI.....	5
DAFTAR TABEL.....	8
DAFTAR GAMBAR.....	9
RINGKASAN.....	11
BAB 1. PENDAHULUAN.....	12
1.1 Latar belakang dan Identifikasi Masalah.....	12
1.2 Rumusan Masalah.....	16
1.3 Tujuan.....	16
1.4 Batasan Masalah.....	16
1.5 Batasan Realistis Aspek Keteknikan.....	16
BAB 2. IDENTIFIKASI KEBUTUHAN SISTEM.....	17
2.1 Studi Literatur dan Observasi.....	17
2.2 Dasar Teori.....	24
2.2.1 Dispenser Pengharum Ruangan Otomatis.....	24
2.2.2 <i>Internet Of Things</i>	24
2.2.3 ESP32.....	25
2.2.4 <i>Sensor Load Cell</i>	25
2.2.5 <i>Real Time Clock</i>	26
2.2.6 Modul LM2596.....	27
2.2.7 Modul TP4056.....	28
2.2.8 Baterai Lithium-Ion.....	28
2.2.9 <i>Relay 5V Single Channel</i>	30
2.2.10 BMS 3S.....	31
2.2.11 Modul RGB LED.....	31
2.2.12 Step Down 5V MP1584.....	32
2.2.13 Blynk.....	32
2.3 Analisis <i>Stakeholder</i>	33
2.3.1 Penghuni Rumah.....	33

2.3.2	Petugas <i>Maintenance</i>	33
2.4	Analisis Aspek yang Mempengaruhi Sistem.....	33
2.4.1	Aspek Ekonomi.....	33
2.4.2	Aspek Sosial.....	33
2.4.3	Aspek Kesehatan.....	34
2.4.4	Aspek Pengoperasian.....	34
2.5	Spesifikasi Sistem.....	34
BAB 3. USULAN SOLUSI		36
3.1	Usulan Solusi 1.....	36
3.1.1	Desain Sistem 1.....	36
3.1.2.	Rencana Anggaran Desain Sistem 1.....	41
3.1.3	Analisis Risiko Desain 1.....	41
3.1.4	Pengukuran Performa Desain 1.....	41
3.2	Usulan Solusi 2.....	42
3.2.1	Desain Sistem 2.....	43
3.2.2	Rencana Anggaran Desain 2.....	48
3.2.3	Analisis Risiko Desain 2.....	49
3.2.4	Pengukuran Performa Desain 2.....	49
3.3	Analisis dan Penentuan Usulan Solusi/Desain Terbaik.....	50
3.4	Gantt <i>Chart</i>	51
3.5	Realisasi Pelaksanaan Tugas Akhir 1.....	52
BAB 4. HASIL RANCANGAN DAN METODE PENGUKURAN		55
4.1	Hasil Rancangan Sistem.....	55
4.1.1	Flowchart Coding Smart AFRESH IO.....	55
4.1.2	Wiring Diagram.....	57
4.1.3	Gambar Desain Tiga Dimensi (3D).....	58
4.1.4	User Interface.....	59
4.1.5	Hasil Perancangan Akhir.....	60
4.2	Metode Pengukuran Kinerja Hasil Perancangan.....	61
4.2.1	Pengukuran <i>Monitoring</i> Baterai.....	61
4.2.2	Pengukuran <i>Monitoring</i> Cairan Pengharum Ruangan.....	62

4.2.3	Pengukuran Delay IoT	63
4.2.4	Pengukuran Daya Tahan Baterai SMART AFRESH IO	63
4.2.5	Pengukuran <i>Charging</i> Baterai SMART AFRESH IO	64
BAB 5. HASIL PENGUKURAN DAN ANALISIS		65
5.1.	Analisis Hasil	65
5.1.1	Hasil dan Analisis Pengujian Parameter	65
5.1.2	Perbandingan Performa	78
5.1.3	Pemenuhan Spesifikasi Sistem.....	79
5.1.4	Pengalaman Pengguna.....	80
5.1.5	Kesesuaian Perencanaan dalam Manajemen Tim dan Realisasinya	81
5.2	Dampak Implementasi Sistem	85
BAB 6. KESIMPULAN DAN SARAN.....		86
6.1	Kesimpulan.....	86
6.2	Saran.....	87
DAFTAR PUSTAKA		88
LAMPIRAN – LAMPIRAN		91

DAFTAR TABEL

Tabel 1.1. Hasil survei antara pengembang dan pengguna	15
Tabel 2.1 Hasil Studi Literatur Solusi Sejenis	19
Tabel 3.1. Inventarisasi kebutuhan usulan sistem perangkat keras sistem 1.....	40
Tabel 3.2. Rencana anggaran pengembangan sistem 1.....	41
Tabel 3.3. Inventarisasi kebutuhan usulan sistem perangkat keras sistem 2.....	47
Tabel 3.4. Rencana anggaran pengembangan sistem 2.....	48
Tabel 3.5 <i>Gantt chart</i> pelaksanaan <i>Capstone Project</i> sistem <i>Smart AFRESH IO</i>	51
Tabel 3.6 Realisasi aktivitas pelaksanaan tugas akhir 1.....	52
Tabel 5.1 Hasil Pengukuran Fitur <i>Monitoring Baterai</i>	66
Tabel 5.2 Hasil Pengukuran Fitur <i>Monitoring Cairan Pengharum Ruangan</i>	68
Tabel 5.3 Hasil Pengukuran <i>Delay</i> pada Fitur Pengaturan Interval Waktu Penyemprotan	71
Tabel 5.4 Hasil Pengukuran <i>Delay</i> pada Fitur Penjadwalan Otomatis	73
Tabel 5.5 Perbandingan performa antara sistem yang dibuat dengan sistem lain.....	78
Tabel 5.6 Perbandingan usulan dan hasil perancangan sistem.....	79
Tabel 5.7 Pengalaman Pengguna	80
Tabel 5.8 Kesesuaian antara usulan dan realisasi <i>timeline</i> pengerjaan Tugas Akhir 2.....	81
Tabel 5.9 Kesesuaian RAB Tugas Akhir antara usulan dan realisasi	82
Tabel 5.10 Realisasi aktivitas pelaksanaan tugas akhir 2.....	84

DAFTAR GAMBAR

Gambar 1.1. Hasil survei antara pengembang dan pengguna (pertanyaan 1).....	14
Gambar 1.2 Hasil survei antara pengembang dan pengguna (pertanyaan 2).....	14
Gambar 1.3 Hasil survei antara pengembang dan pengguna (pertanyaan 3).....	14
Gambar 1.4 Hasil survei antara pengembang dan pengguna (pertanyaan 4).....	15
Gambar 1.5 Hasil survei antara pengembang dan pengguna (pertanyaan 5).....	15
Gambar 2.1 Dispenser pengharum ruangan otomatis	24
Gambar 2.2 <i>Internet of things</i>	25
Gambar 2.3 ESP32.....	25
Gambar 2.4 Sensor <i>load cell</i>	26
Gambar 2.5 <i>Real Time Clock</i> (RTC).....	27
Gambar 2.6 Modul <i>LM2596</i>	28
Gambar 2.7 Modul <i>TP4056</i>	28
Gambar 2.8 Baterai Li-ion	29
Gambar 2.9 Koneksi baterai secara seri dan paralel	30
Gambar 2.10 Relay single channel	30
Gambar 2.11 BMS	31
Gambar 2.12 Modul RGB LED	32
Gambar 2.13 Modul <i>step down</i> 5V MP1584	32
Gambar 3.1 Diagram blok sistem <i>SMART AFRESH IO</i>	37
Gambar 3.2 Proses cara kerja sistem usulan solusi 1	37
Gambar 3.3 <i>Wiring diagram</i> sistem usulan solusi 1	38
Gambar 3.4 Desain model alat usulan 1 yang akan direalisasikan.....	38
Gambar 3.5 Desain UI alat yang akan digunakan oleh <i>user</i>	39
Gambar 3.6 Diagram blok sistem <i>SMART AFRESH IO</i>	44
Gambar 3.7 Proses cara kerja sistem usulan solusi 2.....	45
Gambar 3.8 <i>Wiring diagram</i> sistem usulan solusi 2	45
Gambar 3.9 Desain model alat usulan 2 yang akan direalisasikan.....	46
Gambar 3.10 Desain UI alat yang akan digunakan oleh <i>user</i>	46

Gambar 4.1 Flowchart coding SMART AFRESH IO	55
Gambar 4.2 Wiring diagram SMART AFRESH IO	57
Gambar 4.3 Desain model alat usulan 2 yang direalisasikan	58
Gambar 4.4 Desain UI alat yang akan digunakan oleh user	59
Gambar 4.5 Tampilan <i>Datalogging</i> pada <i>google sheets</i>	59
Gambar 4.6 Hasil perancangan akhir SMART AFRESH IO.....	60
Gambar 4.7 <i>Coding</i> kalibrasi tegangan untuk fitur <i>monitoring</i> baterai	62
Gambar 4.8 <i>Coding</i> kalibrasi <i>load cell</i>	62
Gambar 5.1 Pengujian <i>error monitoring</i> baterai.....	67
Gambar 5.2 Pengujian <i>error monitoring</i> cairan pengharum ruangan	69
Gambar 5.3 Pengujian <i>delay</i> waktu antara saat alat dinyalakan menggunakan saklar dengan terhubung ke <i>Blynk</i>	71
Gambar 5.4 Pengujian interval waktu penyemprotan melalui <i>Blynk</i>	72
Gambar 5.5 Pengujian notifikasi melalui <i>Blynk</i> dan email	73
Gambar 5.6 Pengujian daya tahan baterai SMART AFRESH IO.....	74
Gambar 5.7 Tegangan <i>cut off</i> bawah dari BMS pada baterai.....	76
Gambar 5.8 Pengujian lama <i>charging</i> baterai yang digunakan dalam alat SMART AFRESH IO	77

RINGKASAN

Pengharum ruangan merupakan salah satu produk yang berperan penting dalam menciptakan suasana ruangan yang menyenangkan dan menyegarkan dengan memberikan bau harum di dalam suatu ruangan. Pada masa kini, masyarakat lebih memilih untuk menggunakan pengharum ruangan otomatis karena tipe tersebut dapat menyemprotkan cairan pengharum ruangan secara otomatis dalam kurun waktu tertentu. Akan tetapi alat pengharum ruangan otomatis ini masih memiliki beberapa kekurangan seperti masalah yang paling utama adalah kesulitan pengguna dalam melakukan pengecekan baterai dan jumlah cairan pengharum pada alat tersebut karena posisi penempatannya yang biasanya diletakkan di tempat tinggi sehingga tidak dapat dilakukan pengecekan setiap saat.

Solusi dari permasalahan tersebut adalah dengan membuat sistem yang dapat melakukan *monitoring* baterai dan jumlah cairan pengharum ruangan secara *real time*, yaitu dengan cara mengintegrasikan dispenser pengharum ruangan tersebut dengan IoT (*Internet Of Things*). Dengan begitu, pemantauan kondisi baterai dan jumlah cairan pengharum ruangan dapat dilakukan secara mudah melalui *web server* atau aplikasi yang dibuat untuk sistem ini. Maka dari itu terciptalah produk SMART AFRESH IO, dispenser pengharum ruangan terintegrasi IoT yang menyediakan berbagai macam fitur guna membantu dan mempermudah pengguna untuk melakukan perawatan terhadap dispenser pengharum ruangan pada tempat mereka masing-masing. Fitur-fitur yang disediakan oleh alat ini diantaranya adalah *monitoring* baterai, *monitoring* isi kaleng cairan pengharum ruangan, pengaturan interval waktu penyemprotan secara virtual dengan 6 opsi yang berbeda, penjadwalan otomatis dan notifikasi ketika kondisi baterai dan jumlah cairan pengharum ruangan sudah mendekati habis.

Produk SMART AFRESH IO sudah melewati beberapa pengujian untuk memastikan sistem berjalan dengan baik dan mencapai hasil yang diinginkan. Dari hasil berbagai uji coba yang telah dilakukan, didapati bahwa produk SMART AFRESH IO sudah dapat melakukan *monitoring* baterai, *monitoring* cairan pengharum ruangan, mengatur interval waktu penyemprotan secara virtual, penjadwalan otomatis pada jam kerja yaitu dari jam 8 pagi sampai jam 5 sore, serta dapat memberikan notifikasi ketika kondisi baterai dan jumlah cairan pengharum ruangan sudah mendekati habis atau kritis. Sehingga berdasarkan hasil perancangan SMART AFRESH IO dapat disimpulkan bahwa sebagian besar alat sudah sesuai dengan spesifikasi yang telah diusulkan, serta sudah memenuhi tujuan dari projek ini.

BAB 1. PENDAHULUAN

1.1 Latar belakang dan Identifikasi Masalah

Pengharum ruangan merupakan salah satu produk yang berperan penting dalam menciptakan suasana ruangan yang menyenangkan dan menyegarkan dengan memberikan bau harum di dalam suatu ruangan. Banyak sekali manfaat dari pengharum ruangan, yaitu untuk menghilangkan bau tidak sedap pada suatu ruangan, meningkatkan *mood* pada suatu ruangan, dan tentunya menciptakan suasana lingkungan atau ruangan yang menyenangkan dan menyegarkan[1]. Produk ini dapat digunakan di berbagai jenis tempat seperti rumah, ruang kelas, kantor, toko, restoran, dan lain sebagainya. Bentuk dari pengharum ruangan ini bermacam-macam, beberapa diantaranya yang cukup umum adalah semprotan pengharum ruangan, yaitu pengharum ruangan dalam botol kaleng yang digunakan dengan cara menekan *nozzle* untuk mengeluarkan semprotan aroma wangi, dan pengharum ruangan otomatis, yaitu alat yang akan menyemprotkan pengharum ruangan secara otomatis pada interval waktu tertentu.

Pada masa kini, masyarakat lebih memilih untuk menggunakan pengharum ruangan otomatis untuk berbagai jenis ruangan seperti ruang tamu, kelas, ruang kerja, kamar mandi, ruangan-ruangan besar, dan lain sebagainya. Hal tersebut karena pengharum ruangan otomatis dapat menyemprotkan cairan pengharum ruangan secara otomatis dalam kurun waktu tertentu, seperti misalkan setiap 5, 15, atau 30 menit sekali, sehingga memudahkan kita dalam memperoleh aroma yang wangi pada suatu ruangan. Pengguna hanya perlu meletakkan pengharum ruangan otomatis ini pada suatu tempat, umumnya digantungkan atau ditempel di tembok, lalu jika cairan pengharum ruangnya sudah habis maka pengguna dapat mengisi ulang cairan pengharum ruangnya tersebut[2]. Akan tetapi alat pengharum ruangan otomatis ini belum sepenuhnya sempurna, berdasarkan survey yang kami lakukan dengan memberikan kuesioner kepada berbagai narasumber yaitu kerabat, rekan-rekan kami, serta sejumlah petugas *cleaning service* dan staf-staf bagian akademik Fakultas Teknologi Industri Universitas Islam Indonesia, pengharum ruangan otomatis ini masih memiliki beberapa kekurangan. Dari hasil survey yang kami lakukan, masalah yang paling utama dari pengharum ruangan otomatis ini adalah kesulitan pengguna dalam melakukan pengecekan baterai dan jumlah cairan pengharum pada alat tersebut karena posisi penempatannya yang biasanya diletakkan di tempat tinggi sehingga tidak dapat dilakukan pengecekan setiap saat, sebanyak 75,9% narasumber setuju bahwa proses pengecekan kondisi

baterai dan jumlah cairan dalam dispenser pengharum ruangan saat ini belum efisien. Lalu disusul oleh permasalahan mengenai kondisi dimana alat sudah tidak menyemprot karena baterai habis padahal cairan pengharum ruangnya masih ada, kemudian kondisi dimana alat masih menyemprot tetapi tidak wangi karena cairan pengharum ruangan sudah habis, lalu terakhir adalah masalah mengenai baterai dan cairan pengharum yang boros karena penyemprotan yang terlalu sering.

Solusi awal dari beberapa masalah tersebut adalah memberikan notifikasi berupa suara seperti dengan *buzzer* atau berupa indikator LED seperti pada meteran listrik. Akan tetapi, ketika terdapat lebih dari satu pengharum ruangan seperti pada perkantoran, perhotelan, dan lain sebagainya, maka solusi tersebut akan tidak terlalu efektif. Tiap ruang tetap memerlukan petugas atau operator untuk memantau notifikasi berupa *buzzer* atau indikator LED pada dispenser pengharum ruangan tersebut. Sehingga solusi lain untuk mempermudah dan melakukan proses *monitoring* baterai dan cairan pengharum ruangan tersebut secara efisien adalah dengan mengintegrasikan alat dengan IoT (*Internet Of Things*). Dengan begitu, pemantauan kondisi baterai dan jumlah cairan pengharum ruangan dapat dilakukan secara mudah melalui *web server* atau aplikasi yang dibuat untuk sistem ini, serta operator dapat mengetahui ketika kondisi baterai dan cairan pengharum ruangan yang sudah mau habis melalui notifikasi yang dikirimkan oleh sistem.

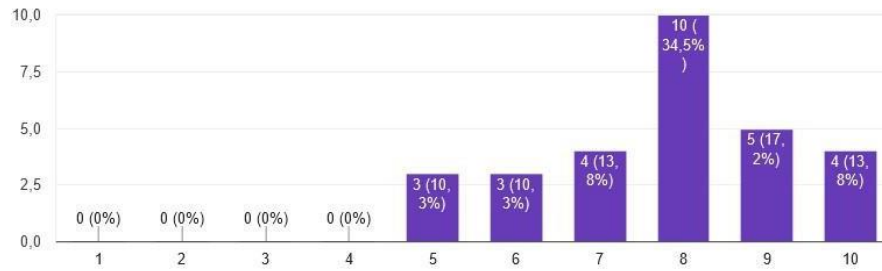
Sehingga berdasarkan permasalahan-permasalahan di atas, pada Tugas Akhir yang berjudul “Dispenser Pintar Pengharum Ruangan Terintegrasi” ini akan dibuat sebuah Dispenser Pintar Pengharum Ruangan yang terintegrasi dengan IoT. Sebanyak 93% narasumber menyatakan bahwa pengintegrasian Dispenser Pengharum Ruangan dengan IoT ini perlu dilakukan. Kemudian berdasarkan hasil survei yang telah kami lakukan, fitur utama dispenser pintar pengharum ruangan terintegrasi yang diinginkan oleh pengguna adalah notifikasi atau pemberitahuan kepada *user* ketika kondisi baterai dan jumlah cairan pengharum ruangan mendekati habis, kemudian disusul oleh fitur untuk memantau atau *monitoring* kondisi baterai dan jumlah cairan pengharum ruangan, lalu fitur ID pada dispenser pengharum ruangan untuk identifikasi setiap alat jika alat terletak di berbagai ruang (lebih dari 1).

Berikut beberapa gambar yang menunjukkan hasil survey yang telah kami lakukan :

Seberapa penting penggunaan Dispenser Pengharum Ruangan dalam suatu ruangan?



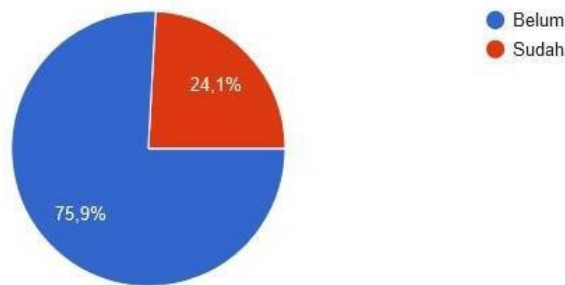
29 jawaban



Gambar 1.1. Hasil survei antara pengembang dan pengguna (pertanyaan 1)

Apakah proses pengecekan kondisi baterai dan jumlah cairan dalam Dispenser Pengharum Ruangan saat ini sudah efisien?

29 jawaban

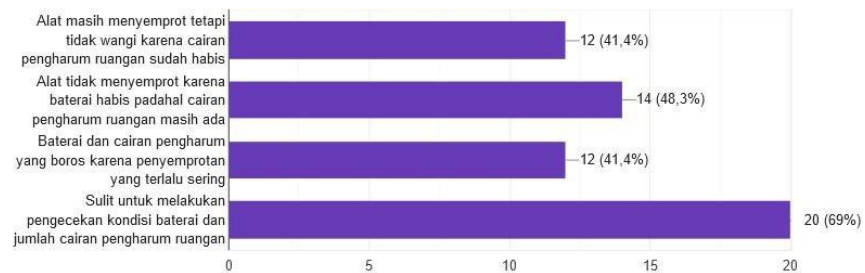


Gambar 1.2 Hasil survei antara pengembang dan pengguna (pertanyaan 2)

Apa kekurangan dari Dispenser Pengharum Ruangan saat ini? (pilih satu atau lebih jawaban)



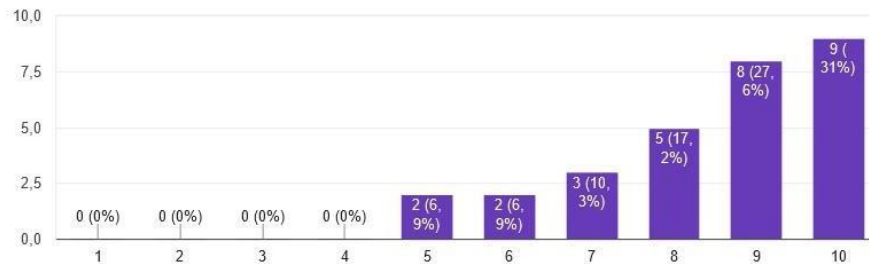
29 jawaban



Gambar 1.3 Hasil survei antara pengembang dan pengguna (pertanyaan 3)

Dari permasalahan di atas, seberapa penting pembuatan Dispenser Pintar Pengharum Ruangan Terintegrasi? Salin

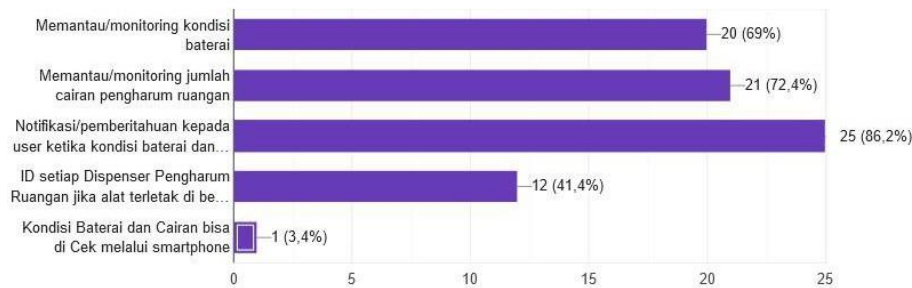
29 jawaban



Gambar 1.4 Hasil survei antara pengembang dan pengguna (pertanyaan 4)

Fitur apa saja yang diinginkan untuk Dispenser Pintar Pengharum Ruangan Terintegrasi (pilih satu atau lebih jawaban) Salin

29 jawaban



Gambar 1.5 Hasil survei antara pengembang dan pengguna (pertanyaan 5)

Tabel 1.1. Hasil survei antara pengembang dan pengguna

Pertanyaan	Jawaban/tanggapan
Seberapa penting penggunaan dispenser pengharum ruangan dalam suatu ruangan?	Sebanyak 89,6% responden menjawab sangat penting
Apakah proses pengecekan kondisi baterai dan jumlah cairan dalam dispenser pengharum ruangan saat ini sudah efisien?	Sebanyak 75,9% responden menjawab belum efisien
Apa kekurangan dari dispenser pengharum ruangan saat ini? (pilih satu atau lebih jawaban)	Sulit melakukan pengecekan kondisi baterai dan jumlah cairan pengharum, alat tidak menyemprot karena baterai habis, alat menyemprot tetapi tidak wangi karena cairan habis, baterai dan cairan

	pengharum yang boros karena penyemprotan yang terlalu sering
Dari permasalahan di atas, seberapa penting pembuatan dispenser pintar pengharum ruangan terintegrasi?	Sebanyak 93% responden menjawab bahwa pembuatan Dispenser Pintar Pengharum Ruangan Terintegrasi perlu dilakukan
Fitur apa saja yang diinginkan untuk dispenser pintar pengharum ruangan terintegrasi (pilih satu atau lebih jawaban)	Notifikasi kepada <i>user</i> ketika kondisi baterai dan cairan pengharum mendekati habis, <i>monitoring</i> jumlah cairan pengharum ruangan, <i>monitoring</i> kondisi baterai, ID pada dispenser pengharum ruangan untuk identifikasi setiap alat jika alat terletak di berbagai ruang

1.2 Rumusan Masalah

1. Bagaimana perancangan dispenser pintar pengharum ruangan terintegrasi?
2. Bagaimana kinerja dari dispenser pintar pengharum ruangan terintegrasi?

1.3 Tujuan

Berdasarkan rumusan masalah yang telah diuraikan di atas, tujuan dari penelitian ini adalah untuk merancang dispenser pintar pengharum ruangan terintegrasi serta menganalisis kinerjanya.

1.4 Batasan Masalah

1. Menggunakan jenis *refill* yang sudah *ready*
2. Tidak mempertimbangkan lama waktu proses *charging* baterai
3. Volume penyemprotan akan disesuaikan dengan standar yang digunakan secara umum
4. Tidak mempertimbangkan kekentalan atau keenceran cairan pengharum ruangnya

1.5 Batasan Realistis Aspek Keteknikan

1. Berat alat diperkirakan sebesar kurang dari 1 kg
2. Harus memenuhi standar keselamatan dan keamanan di dalam ruangan
3. Diperlukan konektivitas internet menggunakan *Wifi*

BAB 2. IDENTIFIKASI KEBUTUHAN SISTEM

2.1 Studi Literatur dan Observasi

Teknologi yang semakin berkembang memungkinkan manusia untuk ingin terus meningkatkan atau memperbaiki suatu permasalahan dengan menciptakan suatu alat tertentu sebagai solusi dari permasalahan tersebut. Untuk memperdalam dan menambah wawasan serta informasi yang diperlukan dalam penelitian ini, dan juga komponen yang dibutuhkan untuk menentukan spesifikasi dari alat yang akan dibuat, maka perlu dilakukan studi literatur dan observasi. Didapati bahwa terdapat beberapa penelitian-penelitian yang sebelumnya sudah dilakukan untuk permasalahan mengenai dispenser pengharum ruangan ini. Bagian selanjutnya merupakan penjelasan untuk mengetahui relevansi penelitian ini dengan penelitian-penelitian yang sebelumnya telah dilakukan tersebut.

Pada penelitian berjudul “Sistem Kontrol Alat Pengharum Ruangan Otomatis Berbasis Mikrokontroler Berdasarkan Tingkat Kepadatan Jumlah Pengguna Ruangan” yang telah dilakukan oleh Gusrio Tendra dan Muhammad ini digunakan sensor optik dan mikrokontroler yang fungsinya membaca pergerakan pengguna ruangan yang masuk ataupun keluar, sehingga sensor optic dipasang pada pintu masuk dan pintu keluar. Alat akan aktif apabila terdapat orang yang melewati sensor optik tersebut. Ketika ada pengguna yang memasuki ruangan maka angka pada layar LCD *seven segment* akan ditambahkan dengan jumlah pengguna yang sedang berada di dalam, lalu ketika ada pengguna yang keluar ruangan maka sistem akan mengurangi jumlah pengguna yang ada di ruangan. Alat ini memiliki kelebihan antara lain interval penyemprotan pengharum ruangan yang berubah sesuai dengan jumlah orang dalam suatu ruangan, semakin banyak jumlah orangnya maka interval penyemprotan akan semakin kecil, yang berarti penyemprotan menjadi lebih sering. Adapun kekurangannya ialah tidak dapat memantau kondisi baterai dan jumlah cairan pengharum ruangan [3].

Pada penelitian berjudul “Rancang Bangun Pengharum Ruangan Otomatis Menggunakan RTC (*Real Time O’Clock*) Berbasis Arduino UNO” yang telah dilakukan oleh Feby Mira Trisna, Adi Sutrsiman, dan M. Miftakul Amin, digunakan RTC (*Real Time O’Clock*) sebagai pengaturan waktu agar dapat menghemat penggunaan cairan pengharum ruangan. Sistem ini diawali dengan aktifnya RTC, kemudian sensor akan menampilkan waktu saat ini pada LCD, lalu sensor akan memberikan data *high* kepada mikrokontroler yang akan mengaktifkan *driver relay*, lalu penyemprot akan aktif sesuai dengan waktu yang telah di atur pada program. Dengan tambahan

RTC tersebut menjadikan alat pengharum ruangan otomatis akan hidup dan mati berdasarkan waktu yang telah diatur dalam program. Adapun kelebihan dari alat ini antara lain alat dapat menyemprot pada jam kerja saja dan pada saat bukan jam kerja bisa diatur menjadi tidak menyala secara otomatis. Namun adapun kekurangan dari alat ini antara lain alat tidak dapat memberikan informasi kepada pengguna ketika cairan pengharum ruangan sudah habis [4].

Lalu pada penelitian berjudul “Perancangan dan Pembuatan Alat Pengatur Perioda Penyemprotan pada Room Deodorizer Dispenser Berbasis Mikrokontroler” yang telah dilakukan oleh Dody Nugroho, Lanny Augustine, dan Alber Gunadhi ini digunakan sensor cahaya LDR (*Light Dependent Resistor*) untuk mendeteksi penerangan lampu di dalam ruangan agar jika lampu dipadamkan maka mikrokontroler akan mengontrol supaya pengharum tidak disemprotkan, serta sensor *Infra Red* untuk mendeteksi isi dari kaleng pengharum berdasarkan hasil semprotan, sehingga pengguna dapat langsung mengganti kaleng pengharum saat isi pengharum sudah habis. Hasil yang didapatkan adalah alat akan menyemprot cairan pengharum sesuai dengan *input* waktu yang diinginkan oleh operator, alat tidak akan menyemprot ketika sensor LDR tidak terkena cahaya penerangan dan pada saat sensor *infra red* tidak mendeteksi adanya semprotan dari isi pengharum, maka *buzzer* akan berbunyi sebagai indikator bahwa isi pengharum telah habis. Adapun kekurangan dari alat ini adalah notifikasi mengenai jumlah sisa cairan pengharum ruangan yang sudah habis masih menggunakan *buzzer* dan operator perlu menekan tombol *enter* secara manual untuk memberhentikan bunyinya [5].

Kemudian pada penelitian yang dilakukan oleh Adila Nadia, Rahmat Rasyid, dan Harmadi yang berjudul “Sistem *Monitoring* Ketinggian Cairan Infus Berbasis Sensor Serat Optik *Evanescent*” digunakan sensor optik yang bekerja berdasarkan perubahan tegangan keluaran fotodetektor OPT101 terhadap ketinggian cairan infus untuk mengatasi permasalahan mengenai keterlambatan dalam penggantian tabung infus. Sistem mampu *me-monitoring* ketinggian cairan infus secara *realtime* pada saat ketinggian 1 cm di atas leher tabung infus. Hasil *monitoring* ketinggian cairan infus diamati dengan 2 kondisi yaitu aman dan awas yang ditampilkan pada LCD, LED sebagai lampu indikator dan bunyi pada *buzzer*. Ketika ketinggian cairan infus lebih besar dari 1 cm maka *output* menampilkan indikator LED2 yang artinya cairan infus masih dalam keadaan aman, namun ketika cairan infus mencapai ketinggian besar atau sama dengan 1 cm maka LED akan menampilkan indikator LED1 yang berarti pasien atau tim medis harus waspada dalam penggantian cairan infus serta *buzzer* akan berbunyi. Kelebihan dari alat yang dibuat ini adalah

Persentase ketepatan rata-rata pengukuran adalah 92,73%. Namun alat ini juga masih memiliki kekurangan yaitu sistem belum terintegrasi dengan IoT, kemudian penggunaan sensor fotodiode dan infrared menyebabkan sistem mudah terpengaruh oleh interferensi gelombang elektromagnetik, sehingga menyebabkan *delay* pada pengiriman data [6].

Selanjutnya pada penelitian berjudul “IoT-Based Infusion Fluid *Monitoring* System Using the MQTT Protocol” yang dilakukan oleh Nur Afiyat, Raizly Helmi Navilla, dan Mohammad Hariyadi, telah diajukan suatu solusi yaitu sistem *monitoring* cairan infus *Ringer’s lactate* sebanyak 500 ml menggunakan sensor *load cell* HX711 dengan protokol IoT MQTT untuk mengatasi permasalahan *monitoring* cairan infus secara manual yang tidak efisien. Hasil dari penelitian tersebut adalah sistem *monitoring* untuk cairan infus *Ringer’s lactate* menggunakan protokol MQTT dengan mengintegrasikan beberapa *hardware* seperti NodeMCU, sensor *load cell* HX711 untuk menimbang berat cairan infus, dan LCD 16x2 untuk menampilkan berat kantong infus secara lokal. Lalu *software* yang digunakan untuk memonitor berat cairan infus adalah MQTT Dash. Keunggulan atau kelebihan sistem ini adalah akurasi pembacaan berat dari sensor *load cell* HX711 yang sangat bagus. Hal tersebut dibuktikan dengan persentase *error* pembacaan berat cairan infus antara 0,08% sampai 0,64%. Kemudian diperoleh *Delay* sistem sebesar 15,15 ms sampai 23,31 ms dan rata-rata *packet loss* sebesar 0% sampai 0,01% dari sistem IoT yang dirancang, yang mana hal-hal tersebut sudah memenuhi standar TIPHON [7].

Tabel 2.1 Hasil Studi Literatur Solusi Sejenis

Judul	Usulan solusi	Hasil/Evaluasi (Kelebihan/Kekurangan)
Sistem Kontrol Alat Pengharum Ruangan Otomatis Berbasis Mikrokontroler Berdasarkan Tingkat Kepadatan Jumlah Pengguna Ruangan [3]	Sistem pengharum ruangan otomatis dengan menggunakan sensor optik dan mikrokontroler yang dapat membaca pergerakan pengguna ruangan yang masuk maupun keluar.	Hasil: Pengharum ruangan otomatis yang hanya akan aktif atau hidup ketika sensor optik yang digunakan mendeteksi seseorang melewati sensor tersebut. Kelebihan: interval penyemprotan pengharum ruangan yang berubah sesuai dengan jumlah orang dalam suatu ruangan, semakin banyak

Judul	Usulan solusi	Hasil/Evaluasi (Kelebihan/Kekurangan)
		<p>jumlah orangnya maka interval penyemprotan akan semakin kecil, yang berarti penyemprotan menjadi lebih sering.</p> <p>Kekurangan: Tidak dapat memantau kondisi baterai dan jumlah cairan pengharum ruangan</p>
<p>Rancang Bangun Pengharum Ruangan Otomatis Menggunakan RTC (<i>Real Time O'Clock</i>) Berbasis Arduino UNO [4]</p>	<p>Sistem pengharum ruangan otomatis yang dimodifikasi dengan menambahkan RTC (<i>Real Time O'Clock</i>) sebagai pengaturan waktu agar dapat menghemat penggunaan cairan pengharum ruangan.</p>	<p>Hasil : Alat pengharum ruangan otomatis akan hidup dan mati berdasarkan waktu yang telah diatur dalam program</p> <p>Kelebihan : Alat pengharum ruangan akan hidup pada saat jam kerja yaitu hari senin-jum'at pada jam 8 pagi sampai jam 5 sore dan akan mati pada saat bukan jam kerja yaitu hari sabtu dan minggu, serta pada malam hari.</p> <p>Kekurangan : Tidak dapat memberikan informasi kepada pengguna ketika cairan pengharum ruangan sudah habis</p>
<p>Perancangan dan Pembuatan Alat Pengatur Periode Penyemprotan Pada Room</p>	<p>Sistem pengharum ruangan otomatis yang dilengkapi dengan</p>	<p>Hasil : Alat akan menyemprot cairan pengharum sesuai dengan <i>input</i></p>

Judul	Usulan solusi	Hasil/Evaluasi (Kelebihan/Kekurangan)
Deodorizer Dispenser Berbasis Mikrokontroler [5]	sensor cahaya LDR (<i>Light Dependent Resistor</i>) untuk mendeteksi penerangan lampu di dalam ruangan, serta sensor Infra Red untuk mendeteksi isi dari kaleng pengharum berdasarkan hasil semprotan.	<p>waktu yang diinginkan oleh operator, alat tidak akan menyemprot ketika sensor LDR tidak terkena cahaya penerangan dan pada saat sensor <i>infra red</i> tidak mendeteksi adanya semprotan dari isi pengharum, maka <i>buzzer</i> akan berbunyi sebagai indikator bahwa isi pengharum telah habis.</p> <p>Kelebihan : Penggunaan sensor LDR sebagai penentuan alat akan menyemprot atau tidak berdasarkan kondisi lampu penerangan di dalam ruangan tersebut dapat menghemat cairan pengharum ruangan.</p> <p>Kekurangan : Notifikasi mengenai jumlah sisa cairan pengharum ruangan yang sudah habis masih menggunakan <i>Buzzer</i> dan operator perlu menekan tombol <i>enter</i> secara manual untuk memberhentikan bunyinya.</p>
Sistem <i>Monitoring</i> Ketinggian Cairan Infus Berbasis Sensor Serat Optik <i>Evanescent</i> [6]	Sistem <i>monitoring</i> ketinggian cairan infus berbasis sensor serat optik <i>evanescent</i> ,	Hasil : Sistem mampu <i>me-monitoring</i> ketinggian cairan infus secara <i>realtime</i> pada saat ketinggian 1 cm di atas leher tabung infus. Hasil

Judul	Usulan solusi	Hasil/Evaluasi (Kelebihan/Kekurangan)
	dimana sensor optik yang digunakan bekerja berdasarkan perubahan tegangan keluaran fotodetektor OPT101 terhadap ketinggian cairan infus.	<p><i>monitoring</i> ketinggian cairan infus diamati dengan 2 kondisi yaitu aman dan awas yang diampilkan pada LCD, LED sebagai lampu indikator dan bunyi pada <i>buzzer</i>.</p> <p>Kelebihan : Persentase ketepatan rata-rata pengukuran adalah 92,73%</p> <p>Kekurangan : Belum terintegrasi dengan IoT. Lalu penggunaan sensor fotodioda dan infrared menyebabkan sistem mudah terpengaruh oleh interferensi gelombang elektromagnetik, sehingga menyebabkan <i>delay</i> pada pengiriman data.</p>
<i>IoT-Based Infusion Fluid Monitoring System Using the MQTT Protocol</i> [7]	Sistem <i>monitoring</i> cairan infus <i>Ringer's lactate</i> sebanyak 500 ml menggunakan sensor <i>load cell</i> HX711.	Hasil : Sistem <i>monitoring</i> untuk cairan infus <i>Ringer's lactate</i> menggunakan protokol MQTT dengan mengintegrasikan beberapa <i>hardware</i> seperti NodeMCU, sensor <i>load cell</i> HX711 untuk menimbang berat cairan infus, dan LCD 16x2 untuk menampilkan berat kantong infus secara lokal. Lalu <i>software</i> yang digunakan untuk memonitor

Judul	Usulan solusi	Hasil/Evaluasi (Kelebihan/Kekurangan)
		berat cairan infus adalah MQTT Dash. Kelebihan : Akurasi pembacaan berat dari sensor <i>load cell</i> HX711 sangat bagus. Hal tersebut dibuktikan dengan persentase <i>error</i> pembacaan berat cairan infus antara 0,08% sampai 0,64%. Kemudian diperoleh <i>Delay</i> sistem sebesar 15,15 ms sampai 23,31 ms dan rata-rata <i>packet loss</i> sebesar 0% sampai 0,01% dari sistem IoT yang dirancang, yang mana hal-hal tersebut sudah memenuhi standar TIPHON. Kekurangan : -

Berdasarkan studi literatur dan observasi yang telah dilakukan, penelitian-penelitian mengenai dispenser pengharum ruangan yang sebelumnya telah dilakukan tersebut belum terintegrasi IoT, contohnya seperti pada referensi [5], dimana sistem notifikasi mengenai jumlah sisa cairan pengharum ruangan yang sudah habis masih menggunakan *Buzzer*, tidak memiliki fitur-fitur seperti untuk *monitoring* kondisi baterai dan sisa cairan pengharum ruangan serta notifikasi saat kondisi baterai dan sisa cairan pengharum ruangnya mendekati habis seperti yang rencananya akan diterapkan pada alat kami. Kemudian berdasarkan studi literatur dari penelitian-penelitian sebelumnya mengenai *monitoring* ketinggian cairan infus, didapati bahwa untuk dispenser pintar pengharum ruangan terintegrasi yang akan dirancang, penggunaan sensor *load cell* akan lebih ideal untuk *monitoring* jumlah cairan pengharum pada alat tersebut.

2.2 Dasar Teori

2.2.1 Dispenser Pengharum Ruangan Otomatis

Seperti namanya, dispenser pengharum ruangan otomatis ini sudah pasti tidak butuh dioperasikan untuk bisa menyemprotkan cairan pengharum ruangan. Banyak manfaat yang bisa diambil dari dispenser otomatis ini, seiring dengan berbagai alasan kenapa ruangan wajib wangi dan memiliki aroma yang semerbak [8]. Beberapa manfaat yang bisa didapatkan antara lain adalah pengguna tidak perlu repot untuk menyemprotkan cairan pengharum ruangan secara manual.



Gambar 2.1 Dispenser pengharum ruangan otomatis [8].

2.2.2 *Internet Of Things*

Internet of Things atau IoT merupakan jaringan dimana berbagai perangkat saling terkait yang menghubungkan dan menukar data dengan *cloud* dan perangkat IoT lain. Dalam beberapa tahun terakhir, IoT menjadi salah satu teknologi paling penting pada abad ke-21. Dengan IoT, kita dapat menghubungkan objek-objek atau perangkat yang digunakan sehari-hari dengan internet melalui perangkat tertanam. Sistem IoT terdiri dari perangkat-perangkat *smart* yang menggunakan sistem tertanam seperti prosesor, sensor, dan komunikasi *hardware* untuk mengumpulkan, mengirim, dan memproses data yang diterima atau diperoleh dari lingkungannya. Perangkat IoT membagikan data sensor yang telah dikumpulkan dengan menghubungkan ke *gateway* IoT yang bertindak sebagai hub pusat tempat dimana perangkat IoT dapat mengirim data. Perangkat-perangkat ini bekerja tanpa interaksi manusia, walaupun manusia tetap bisa berinteraksi dengan *device-device* tersebut. IoT merupakan salah satu teknologi penting dalam kehidupan sehari-hari, dan perkembangannya patut untuk diikuti[9].



Gambar 2.2 *Internet of things* [9].

2.2.3 ESP32

ESP32 merupakan modul mikrokontroler terintegrasi yang memiliki fitur lengkap dan kinerja tinggi. Modul ini merupakan pengembangan dari ESP8266, yang juga merupakan modul WiFi populer. ESP32 memiliki 2 prosesor komputasi, satu prosesor untuk mengelola jaringan WiFi dan *Bluetooth*, serta satu prosesor lagi untuk menjalankan aplikasi. ESP32 sangat cocok digunakan untuk proyek-proyek IoT karena mampu menyambungkan ke jaringan internet dengan mudah. Modul ini dapat terhubung ke jaringan WiFi dengan menggunakan protokol TCP/IP yang memungkinkan. ESP32 memiliki banyak keunggulan seperti kemampuan *multitasking* yang luar biasa, konsumsi daya rendah, dan harga yang terjangkau. Dengan demikian, ESP32 merupakan pilihan tepat bagi yang ingin membuat proyek IoT dengan biaya yang terjangkau[10].

ESP32



Gambar 2.3 ESP32 [10].

2.2.4 *Sensor Load Cell*

Sensor load cell merupakan sensor yang mengkonversi *input* gaya mekanik seperti beban, berat, tensi, kompresi atau tekanan menjadi *variable* fisik lain, dalam hal ini menjadi sinyal

keluaran listrik yang dapat diukur, diubah, dan distandarisasi. Saat gaya yang diterapkan pada sensor meningkat, sinyal listrik berubah secara proporsional. Secara definisi, *load cell* merupakan sejenis transduser gaya. Transduser gaya menjadi elemen penting dalam banyak industri, mulai dari otomotif, manufaktur presisi tinggi, *aerospace* dan pertahanan, otomasi industri, medis & farmatika, dan robotika dimana pengukuran gaya yang dapat diandalkan dan presisi tinggi menjadi tujuan utama.

Secara struktural, sensor gaya atau transduser ini terbuat dari badan logam (juga disebut *flexure*) yang diikatkan pada pengukur regangan *foil* (*foil strain gauges*). Ketika gaya (tensi atau kompresi) diterapkan, badan logam bertindak sebagai “pegas” dan mengalami sedikit perubahan bentuk, kecuali jika diberi beban berlebih, ia akan kembali ke bentuk aslinya. Ketika *flexure* berubah bentuk, *strain gage* juga mengubah bentuknya dan mengakibatkan hambatan listrik, yang menciptakan variasi tegangan diferensial melalui rangkaian jembatan *Wheatstone*. Sehingga, perubahan tegangan yang terjadi sebanding dengan gaya fisik yang diterapkan pada *flexure* [11].

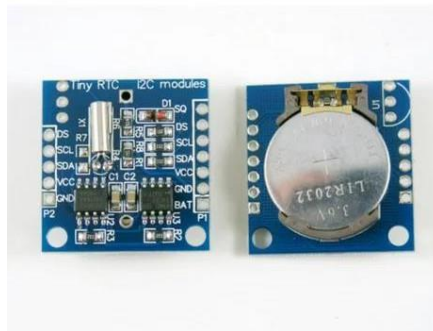


Gambar 2.4 Sensor *load cell* [11].

2.2.5 Real Time Clock

Real Time Clock atau yang biasa disebut RTC merupakan sebuah perangkat keras yang digunakan untuk melacak waktu secara kontinu dan akurat dalam sistem komputer dan perangkat elektronik lainnya. Komponen elektronik ini umumnya terdapat dalam mikrokontroler atau *chip* khusus. Fungsi utama dari RTC adalah untuk menghitung waktu dengan akurat, termasuk tahun, bulan, hari, jam, menit, dan detik. RTC juga bisa menghitung hari dalam seminggu, fitur tambahan seperti alarm atau kalender juga tersedia pada beberapa versi lainnya. Selain itu, fungsi utama lainnya adalah pemantauan waktu secara *real time*, yaitu menjaga waktu yang tepat dan terus

berjalan sepanjang waktu. Lalu RTC digunakan dalam sistem pencatatan data (*logging*) untuk memberikan *timestamp* yang akurat pada data yang dihasilkan atau disimpan. RTC juga dapat digunakan untuk mengatur alarm dan pengingat pada perangkat, seperti alarm jam dinding atau alarm pada perangkat seluler. Kemudian RTC juga dapat membantu dalam manajemen daya, terutama pada perangkat bergerak seperti laptop dan ponsel. RTC memiliki berbagai penerapan di berbagai bidang seperti komputer dan server, kendaraan, peralatan medis, IoT, dan kontrol industri [12].



Gambar 2.5 Real Time Clock (RTC) [12].

2.2.6 Modul LM2596

Modul *Step Down* LM2596 merupakan Modul DC *Buck Converter* yang pada rangkaiannya menggunakan IC LM2596S yang berfungsi untuk mengubah tingkatan tegangan DC menjadi lebih rendah dibanding tegangan masukannya. Tegangan masukan dapat dialiri tegangan berapa pun antara 3 Volt hingga 40 Volt DC. Lalu outputnya akan diubah menjadi tegangan yang lebih rendah di antara 1,5 Volt hingga 35 Volt DC.

IC LM2596S ini dirangkaikan dengan komponen elektronika seperti kapasitor menggunakan SMD *Solid Capacitor* merk Sanyo yang terkenal dengan kualitasnya yang prima, induktor berintikan *ferrite-drum* induktansi tinggi (*high-Q inductance*) dengan pelindung magnetik, *multi-turn* potentiometer dengan resolusi dan akurasi hambatan yang tinggi (bukan potentiometer biasa yang resolusinya rendah), dan dioda SMD tipe Schottky SS54 yang bersifat *low dropout* (LDO) *voltage* [13].



Gambar 2.6 Modul LM2596 [13].

2.2.7 Modul TP4056

TP4056 merupakan modul yang dibuat khusus untuk proses *charging* baterai Li-ion. Gambar 2.7 menunjukkan modul TP4056. Tegangan keluaran charger ini berkisar di 4,2 Volt. Modul ini dilengkapi dengan fitur proteksi tegangan untuk menjaga masa kerja baterai tetap optimum dan juga mampu melakukan recharge secara otomatis [14].



Gambar 2.7 Modul TP4056 [14].

2.2.8 Baterai Lithium-Ion

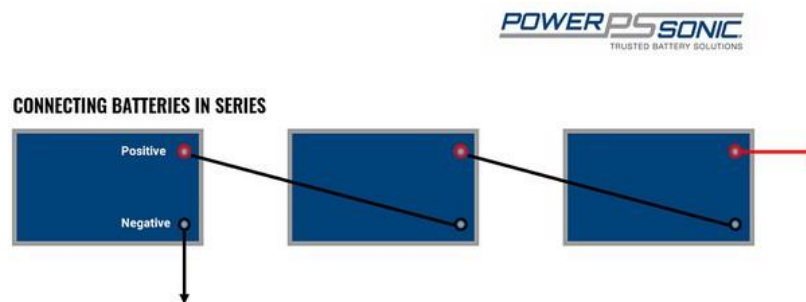
Baterai li-ion (lithium-ion) merupakan baterai *rechargeable* yang paling umum digunakan, baterai ini banyak digunakan dalam elektronik portabel seperti *smartphone* dan transportasi listrik. Baterai li-ion terdiri dari satu atau lebih lithium-ion *cells*, mereka disebut sebagai baterai ketika sel-sel tersebut dipasang di dalam perangkat dengan *protective circuit board*. Dalam baterai li-ion, ion litium (Li⁺) bergerak antara katoda dan anoda secara internal. Sedangkan pada rangkaian eksternal, elektron bergerak berlawanan arah. Migrasi ini lah yang membuat baterai memberi daya pada perangkat, karena baterai menciptakan arus listrik. Saat baterai melakukan pelepasan atau *discharging*, anoda melepaskan ion-ion litium ke katoda, menghasilkan aliran elektron yang memberikan daya ke perangkat yang digunakan. Saat baterai sedang diisi hal sebaliknya terjadi,

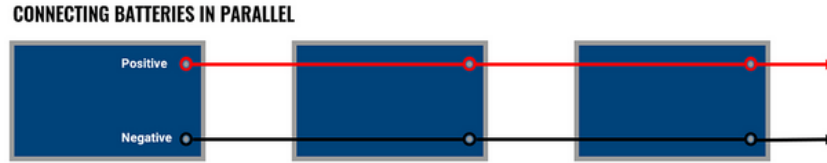
ion-ion litium dilepaskan oleh katoda dan diterima oleh anoda [15]. Beberapa keuntungan dari baterai li-ion adalah lebih ringan dan *compact* dibandingkan dengan beberapa tipe baterai *rechargeable* lainnya, lalu tingkat *auto discharge* yang relatif rendah sehingga dapat mempertahankan kapasitas baterai untuk jangka waktu yang lebih lama ketika tidak digunakan dibandingkan dengan beberapa tipe baterai lain.



Gambar 2.8 Baterai Li-ion [15].

Seperti halnya baterai pada umumnya, baterai li-ion dapat dihubungkan secara seri, paralel, atau bahkan gabungan antara keduanya. Penghubungan baterai secara seri akan meningkatkan tegangan yang digunakan sementara kapasitasnya tetap, sedangkan penghubungan baterai secara paralel akan meningkatkan kapasitasnya sementara tegangannya tetap sama [16]. Ilustrasi penghubungan baterai secara seri dan paralel tersebut dapat diperhatikan pada gambar di bawah.



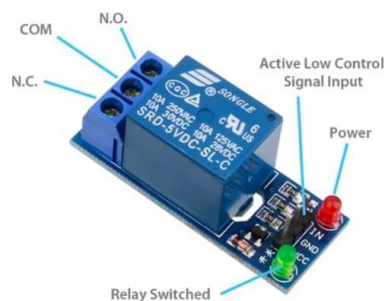


Gambar 2.9 Koneksi baterai secara seri dan paralel [16].

2.2.9 Relay 5V Single Channel

Modul *relay* merupakan suatu perangkat yang beroperasi berdasarkan prinsip elektromagnetik dan menggunakan energi listrik untuk menggerakkan kontaktor dari posisi hidup ke posisi mati dan sebaliknya. Buka tutup kontaktor ini terjadi karena adanya efek induksi magnet yang dihasilkan oleh kumparan induksi listrik. Perbedaan paling mendasar antara *relay* dan sakelar adalah perubahannya dari posisi ON ke posisi OFF. *Relay* menggunakan arus listrik untuk bergerak secara otomatis, namun saklar bersifat manual.

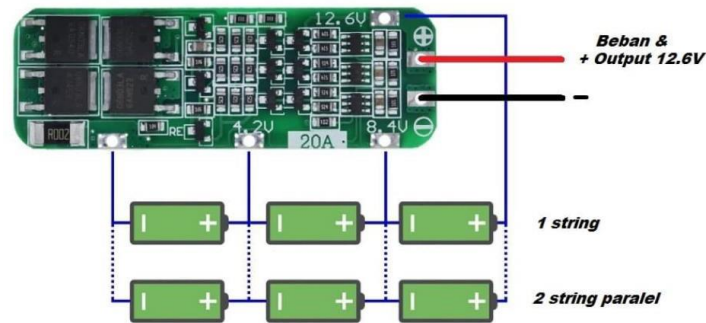
Dapat dipahami bahwa *relay* beroperasi dengan gaya elektromagnetik, yang mana terbuat dari inti besi yang dililitkan pada gulungan kawat dan diberi energi. Ketika kumparan diberi energi, inti besi secara otomatis menjadi magnet dan menarik pembawa, berubah dari keadaan semula tertutup menjadi terbuka. Jika gaya kumparan hilang selama waktu tersebut, maka pegas pada ujung penahan akan tertarik sehingga menyebabkan menutup dari keadaan terbuka. Secara umum kondisi atau posisi pada *relay* terbagi menjadi dua, yaitu NC atau *Normally Close*, yaitu kondisi awal atau kondisi dimana *relay* dalam posisi tertutup karena tidak menerima arus listrik. Lalu NO atau *Normally Open*, yaitu kondisi Dimana *relay* berada dalam posisi terbuka karena menerima arus listrik [17].



Gambar 2.10 Relay single channel [17].

2.2.10 BMS 3S

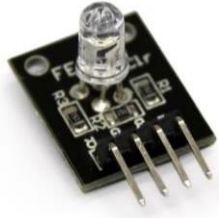
Battery Management System (BMS) dirancang untuk memantau, mengontrol, dan melindungi baterai dalam berbagai aplikasi yang memerlukan penyimpanan dan pemanfaatan energi secara simultan, seperti pada baterai laptop, kendaraan listrik, perangkat penyimpanan energi, dan perangkat yang menggunakan baterai, khususnya baterai litium. Fungsi utama BMS adalah memantau dan mengelola berbagai parameter baterai guna memastikan masa penggunaan, kinerja baterai dan keamanan baterai yang tetap optimal. Salah satu fungsi utamanya adalah memantau tegangan dan arus baterai selama *charging* dan *discharging* sehingga umur baterai terjaga, jadi saat terdapat salah satu sel baterai yang penuh terlebih dahulu atau *overcharge* maka BMS akan memberhentikan proses *charging* untuk baterai tersebut dan tetap melakukan pengisian untuk sel baterai lainnya hingga semua baterai seimbang [18].



Gambar 2.11 BMS [18].

2.2.11 Modul RGB LED

LED RGB adalah jenis LED yang mampu menghasilkan kombinasi warna merah (*red*), hijau (*green*), dan biru (*blue*) untuk menghasilkan banyak warna. Mirip dengan LED biasa, LED ini memiliki anoda dan katoda, namun terdapat 3 anoda yang masing-masing mewakili warna merah, hijau, dan biru. Warna yang dihasilkan oleh LED RGB bergantung pada tegangan yang diberikan pada masing-masing anoda. Modul ini memiliki 4 kaki yaitu R, G, B dan *Common* [19].



Gambar 2.12 Modul RGB LED.

2.2.12 Step Down 5V MP1584

Modul *step down* 5V ini dapat menurunkan tegangan dari 7V sampai 28V menjadi tegangan *output* yang tetap yaitu sebesar 5V dengan arus maksimum 3A. Fungsi utama dari modul ini adalah untuk menurunkan tegangan dari yang lebih tinggi menjadi tegangan yang lebih rendah dan stabil yaitu 5V. Lalu dapat menyediakan arus maksimum mencapai 3A untuk *device* yang perlu daya pada tegangan 5V. Modul ini banyak digunakan pada berbagai aplikasi seperti sistem mikrokontroler dan perangkat-perangkat yang membutuhkan tegangan stabil 5V [20].



Gambar 2.13 Modul *step down* 5V MP1584.

2.2.13 Blynk

Blynk merupakan platform *Internet of Things* (IoT) yang memungkinkan *user* untuk membuat UI (*User Interface*) untuk mengendalikan perangkat keras dari jarak jauh melalui *desktop web* ataupun aplikasi *mobile*. Dengan Blynk, pengguna dapat menghubungkan berbagai *device* seperti mikrokontroler Raspberry Pi, Arduino, ESP8266, ESP32 dan lain sebagainya ke internet dan mengendalikan serta memantau *device* tersebut dari jarak jauh. Platform ini mempunyai tiga komponen utama, yaitu *Blynk App* yang memungkinkan *user* untuk membuat UI untuk proyek *user* menggunakan berbagai macam *widget* yang telah disediakan. Lalu ada *Blynk Server* yang bertanggung jawab atas semua komunikasi antara *hardware* dengan *smartphone*. Kemudian yang ketiga adalah *Blynk Libraries* yang memungkinkan komunikasi dengan server dan memproses semua perintah yang masuk dan keluar [21].

2.3 Analisis Stakeholder

2.3.1 Penghuni Rumah

Dalam suatu rumah tangga, dispenser pengharum ruangan umumnya diletakkan di ruang tamu. Dengan pengintegrasian dispenser pengharum ruangan dengan IoT, penghuni rumah seperti ibu rumah tangga, kepala keluarga, ART, atau siapapun dalam rumah yang sudah mengerti cara menggunakan dan memanfaatkan fitur-fitur yang disediakan oleh dispenser pengharum ruangan terintegrasi ini dapat melakukan pengendalian dan *monitoring* alat dari jarak jauh dengan mudah.

2.3.2 Petugas Maintenance

Dispenser pengharum ruangan juga banyak digunakan pada suatu kantor, ruang kelas, hotel, toko, restoran, dan lain sebagainya untuk suasana lingkungan atau ruangan yang menyenangkan dan menyegarkan. Pada tempat-tempat tersebut, umumnya ada petugas yang ditugaskan untuk melakukan *maintenance* pada dispenser pengharum ruangan tersebut. Dengan dilakukannya pengintegrasian dispenser pengharum ruangan dengan IoT, petugas *maintenance* dapat dengan mudah melakukan *monitoring* kondisi baterai dan jumlah cairan pengharum ruangan secara *real time*. Fitur notifikasi yang diberikan juga dapat meningkatkan ketepatan waktu dalam proses *maintenance* dispenser pengharum ruangan yang digunakan.

2.4 Analisis Aspek yang Mempengaruhi Sistem

2.4.1 Aspek Ekonomi

Dispenser pengharum ruangan terintegrasi akan memiliki berbagai komponen di dalamnya, proses *prototyping* yang dilakukan kemungkinan akan dilakukan lebih dari sekali yang berarti akan ada beberapa versi atau iterasi dari alat ini sebelum akhirnya didapatkan versi akhir yang diharapkan dan dapat memenuhi kebutuhan *user* sesuai dengan yang telah ditetapkan. Jika komponen-komponen yang digunakan cukup mahal, maka biaya yang akan dikeluarkan dalam perancangan dispenser pengharum ruangan akan cukup tinggi. Oleh karena itu perlu dilakukan survei mengenai komponen-komponen apa saja yang akan digunakan, yang memiliki kualitas bagus serta harga yang wajar atau sesuai dengan *budget*.

2.4.2 Aspek Sosial

Dispenser pengharum ruangan terintegrasi akan diletakkan atau ditempelkan pada tembok, sehingga walaupun desainnya sederhana tetapi peneliti harus mempertimbangkan faktor tampilan yang *aesthetic* agar dispenser pengharum ruangan terintegrasi tetap terlihat menarik dan serasi

dengan ruang dimana alat tersebut diletakkan, sehingga penghuni ruangan memiliki kesan yang baik terhadap alat tersebut. Diharapkan bahwa seluruh komponen yang digunakan dalam alat ini dapat tertutup oleh *casing* dengan sempurna sehingga alat memiliki desain dan tampilan yang rapi.

2.4.3 Aspek Kesehatan

Dalam perancangan dispenser pengharum ruangan terintegrasi, aspek kesehatan perlu dipertimbangkan karena cairan pengharum ruangan yang digunakan mengandung bahan kimia. Sehingga alat ini harus memenuhi standar keselamatan dan keamanan di dalam ruangan, seperti memiliki berat yang ringan agar tidak jatuh dan berpotensi menimpa seseorang, atau penempatannya yang diletakkan tidak terlalu dekat dengan penghuni ruangan agar penghuni ruangan tidak terkena langsung dengan semprotan ketika alat menyemprot cairan pengharum ruangnya.

2.4.4 Aspek Pengoperasian

Sebagian besar fitur-fitur dari dispenser pengharum ruangan terintegrasi ini dapat dirasakan oleh pengguna atau *user* melalui aplikasi IoT yang digunakan. Sehingga pengguna harus mengunduh aplikasi yang dibutuhkan dan tahu cara menggunakan aplikasi tersebut agar dapat menikmati fitur-fitur yang disediakan pada dispenser pengharum ruangan terintegrasi dengan baik seperti memantau kondisi baterai dan jumlah cairan pengharum ruangan yang masih tersisa pada dispenser pengharum ruangan terintegrasi, dapat melihat notifikasi yang muncul, dan pengaturan-pengaturan lainnya yang tersedia.

2.5 Spesifikasi Sistem

Berdasarkan kajian literatur pada sub bab 2.1, dasar teori dan informasi yang diperoleh, spesifikasi dan kriteria dispenser pengharum ruangan terintegrasi yang akan dirancang adalah sebagai berikut :

- Alat memiliki berat maksimal 2 kg
- Memiliki sensor berat untuk memonitor jumlah cairan pengharum ruangan
- Menggunakan RTC (*Real Time Clock*) untuk penerapan fitur penjadwalan otomatis
- Memiliki fitur pengaturan volume maupun frekuensi penyemprotan
- Memiliki fitur *monitoring* persentase baterai dan jumlah cairan pengharum ruangan
- Menggunakan catu daya baterai untuk dispenser pengharum ruangnya
- Sistem terhubung dengan jaringan internet (Wifi)

- Daya tahan baterai yang diperkirakan dapat bertahan hingga 1 bulan
- Pengecasan baterai yang diperkirakan berdurasi 2 jam
- Aplikasi Android untuk mengoperasikan alat

BAB 3. USULAN SOLUSI

Berdasarkan masalah-masalah dan kebutuhan pengguna mengenai dispenser pengharum ruangan otomatis saat ini, maka pada Tugas Akhir yang berjudul *Dispenser Pintar Pengharum Ruangan Terintegrasi* ini kami mengusulkan 2 solusi. Solusi pertama adalah dispenser pengharum ruangan yang terintegrasi IoT dengan sumber daya berupa baterai AAA, sehingga pengguna hanya perlu mengganti baterai jika baterai sudah habis. Sedangkan pada solusi kedua adalah dispenser pengharum ruangan yang terintegrasi IoT dengan sumber daya berupa baterai *rechargeable* Li-ion dan tambahan beberapa komponen lainnya. Berikut penjelasan lebih detail mengenai kedua usulan solusi yang kami ajukan tersebut :

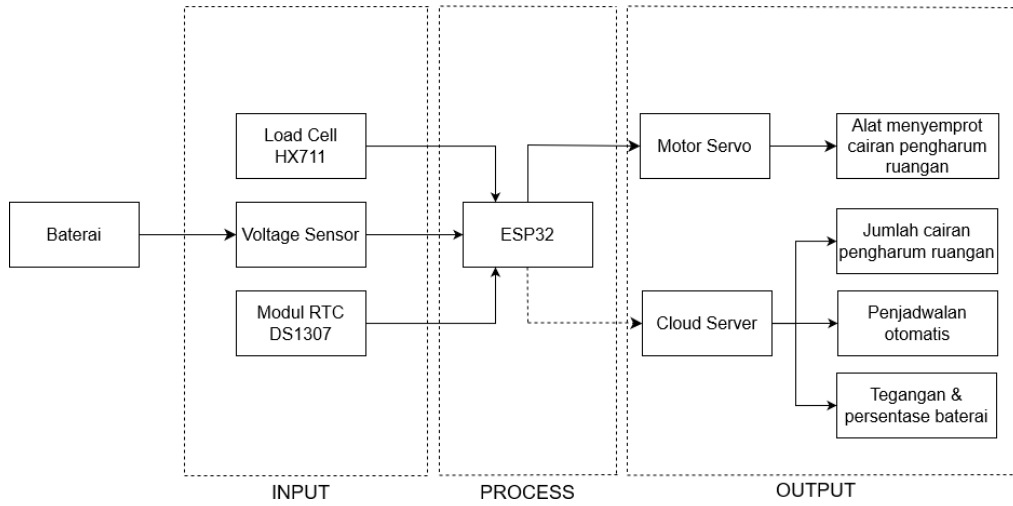
3.1 Usulan Solusi 1

Usulan solusi 1 adalah sistem bernama *Smart AFRESH IO*. Alat ini merupakan dispenser pengharum ruangan yang sudah terintegrasi dengan *internet of things* (IoT) yang dibekali dengan berbagai macam fitur untuk membantu dan memudahkan pengguna dalam *monitoring* dan *maintenance* dispenser pengharum ruangan secara jarak jauh. Dengan diintegrasikannya alat ini dengan IoT, alat ini mampu memberikan berbagai macam fitur yang sangat berguna bagi pengguna seperti *monitoring* kondisi baterai, jumlah cairan pengharum ruangan, notifikasi ketika kondisi baterai dan jumlah cairan pengharum ruangan sudah mendekati habis, penjadwalan otomatis, serta pengaturan volume maupun frekuensi penyemprotan. Pada desain 1 ini, baterai yang digunakan adalah baterai AAA.

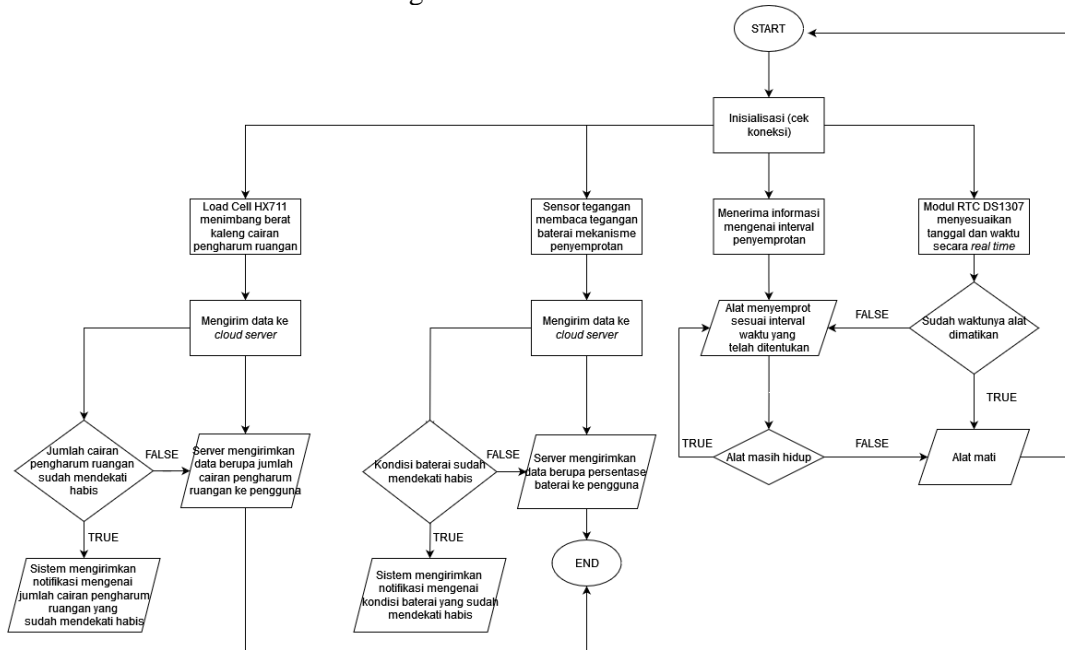
3.1.1 Desain Sistem 1

Gambar 3.1 sampai 3.5 adalah ilustrasi gambaran keseluruhan sistem yang mencakup proses kerja sistem, *wiring* diagram, dan desain alat yang akan dirancang. Pada desain sistem 1 ini, catu daya untuk mikrokontroler ESP32 akan berasal dari 3 buah baterai AAA yang di seri dan akan terhubung dengan pin Vin milik ESP32. Secara umum cara kerja sistem adalah saat sistem pertama dinyalakan, sistem yang sudah terhubung ke WiFi akan menyesuaikan interval penyemprotan cairan pengharum ruangan sesuai dengan yang telah ditentukan oleh pengguna pada aplikasi. Sensor tegangan (*voltage sensor*) akan membaca tegangan dari tiga buah baterai AAA yang dan mengirimkan hasil pembacaannya ke *cloud server* agar kondisi baterai dapat tertampil pada aplikasi. Sensor berat atau *load cell* akan mengirimkan hasil pembacaan berat kaleng cairan pengharum ruangan ke *cloud server* agar jumlah cairan pengharum ruangan dapat tertampil pada aplikasi. Modul *Real Time Clock* (RTC) DS1307 akan menyesuaikan tanggal dan waktu sesuai

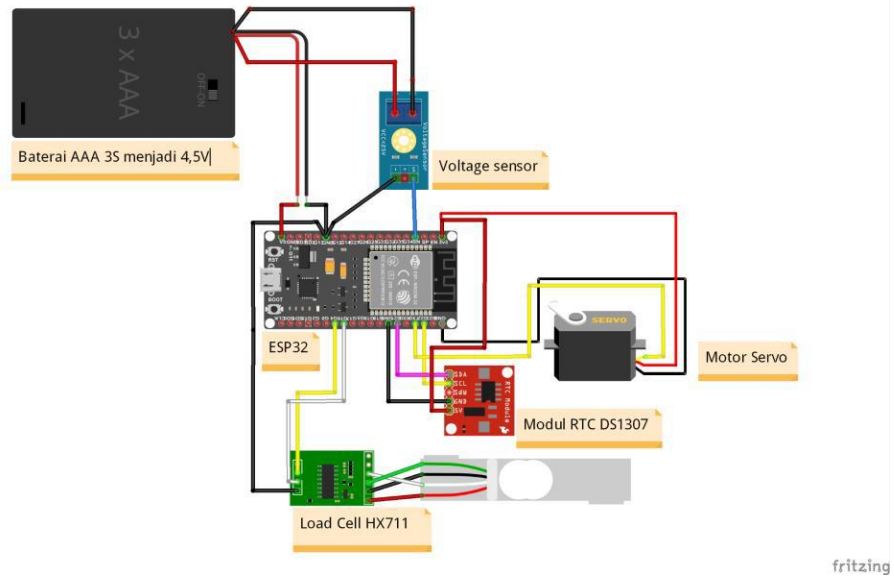
dengan waktu saat itu juga untuk melakukan penjadwalan otomatis dimana alat dapat diatur untuk mati ketika hari sudah menjelang sore atau malam.



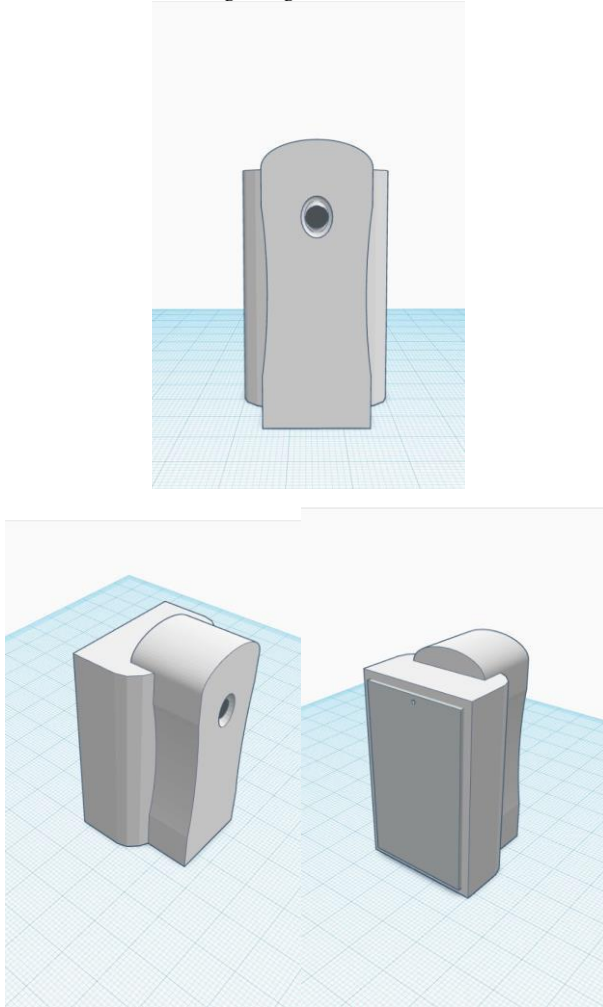
Gambar 3.1 Diagram blok sistem SMART AFRESH IO



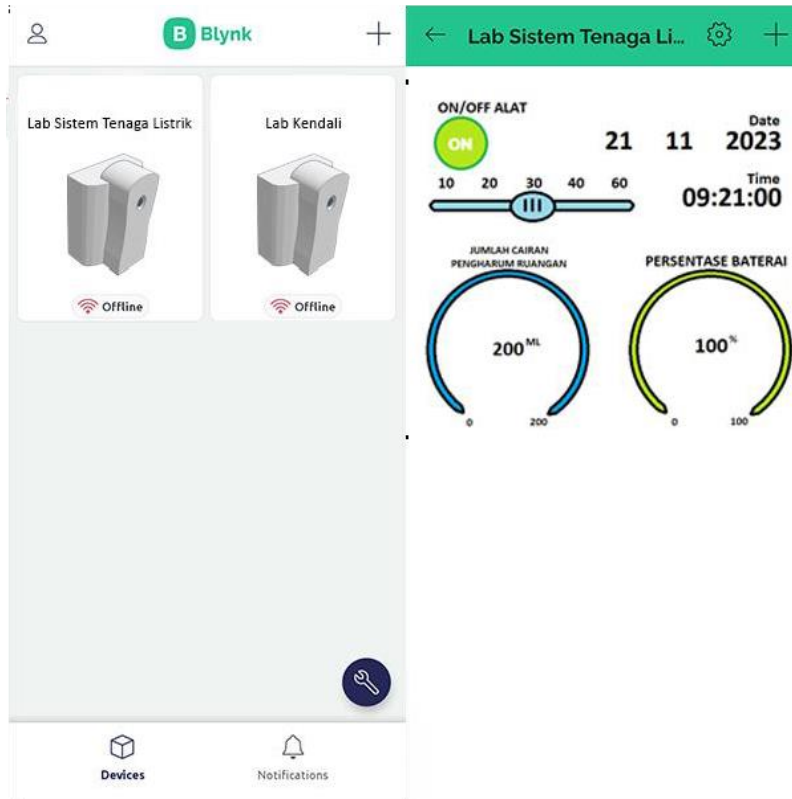
Gambar 3.2 Proses cara kerja sistem usulan solusi 1



Gambar 3.3 *Wiring diagram* sistem usulan solusi 1



Gambar 3.4 Desain model alat usulan 1 yang akan direalisasikan



Gambar 3.5 Desain UI alat yang akan digunakan oleh *user*

Desain ini menggunakan baterai AAA *replaceable* karena kemudahan dalam penggantian baterai, pengguna hanya perlu mengganti baterai yang sudah habis dengan yang baru, kemudian alat akan dapat segera digunakan kembali. Ukuran baterai yang cukup kecil juga akan menghemat tempat pada bagian *internal* alat. Pada gambar 3.5 di atas dapat diperhatikan ilustrasi mengenai UI aplikasi dari alat yang dirancang, dimana pengguna dapat melakukan *monitoring* baterai, jumlah cairan pengharum ruangan, mengatur frekuensi atau interval waktu penyemprotan, dan penjadwalan otomatis alat. Untuk dapat memenuhi usulan sistem tersebut, maka diperlukan inventarisasi kebutuhan sistem perangkat keras. Tabel 3.1 memperlihatkan kebutuhan sistem sesuai usulan dan spesifikasi yang dibutuhkan.

Tabel 3.1. Inventarisasi kebutuhan usulan sistem perangkat keras sistem 1

No	Nama Alat	Keterangan
1	Casing Dispenser Pengharum Ruang Terintegrasi	Dibuat untuk menjadi tempat <i>mounting</i> alat yang telah didesain agar alat terlihat <i>aesthetic</i> dan dapat melindungi dari suhu lingkungan. Perangkat ini dapat dibuat dari bahan filamen 3D <i>printing</i> maupun bahan sejenis.
2	Mikrokontroler ESP 32	Bagian utama dari sistem untuk kendali dan interkoneksi modul-modul dan komponen-komponen yang digunakan dengan WiFi.
3	Baterai	Baterai yang dapat dengan mudah diganti dan terpisah dari modul utama. Dalam hal ini kami cenderung mencari tipe baterai AAA dengan tegangan kerja 1,5 V. Ini diperkirakan dapat bertahan dalam jangka waktu 1 bulan.
4	Motor servo	Digunakan untuk mekanisme penyemprotan, dimana ketika sudah waktunya untuk menyemprot maka servo akan aktif dan menekan katup semprotan kaleng untuk menyemprot pengharum ruangan.
5	Voltage Sensor	Sensor tegangan atau voltage sensor pada desain ini digunakan untuk mengetahui tegangan dari 2 buah baterai AAA yang digunakan untuk mekanisme penyemprotan. Hasil pembacaan tegangan juga akan dikonversi ke persentase, sehingga pengguna dapat melihat tegangan dan persentase kedua baterai secara <i>real time</i> .
6	Load Cell HX711	Sensor berat yang digunakan adalah <i>load cell</i> HX711, ia akan menimbang berat dari kaleng cairan pengharum ruangan dan hasilnya dikonversi menjadi besaran tertentu (seperti persentase) agar pengguna dapat mengetahui jumlah cairan pengharum ruangan yang tersisa secara <i>real time</i> .
7	Modul RTC DS1307	Modul RTC DS1307 ini digunakan untuk memperoleh data mengenai waktu dengan satuan detik, menit, jam, hari, tanggal, bulan, dan tahun secara <i>real time</i> . Pada desain sistem 1, modul ini akan berguna untuk penerapan fitur penjadwalan otomatis, dimana alat dapat diatur untuk mati pada waktu tertentu seperti saat menjelang sore atau malam.

3.1.2. Rencana Anggaran Desain Sistem 1

Tabel 3.2. Rencana anggaran pengembangan sistem 1

No.	Item/Pengeluaran	Satuan	Harga satuan	Jumlah	Harga Total
1	Desain <i>Casing</i> Dispenser Pengharum Ruangan Terintegrasi	Pcs	Rp. 275.000, -	1	Rp. 275.000, -
2	Mikrokontroler ESP 32	Pcs	Rp. 80.000, -	1	Rp. 80.000, -
3	Baterai AAA	Pcs	Rp. 15.000, -	1	Rp. 15.000, -
4	Motor servo	Pcs	Rp. 20.000, -	1	Rp. 20.000, -
5	Voltage Sensor	Pcs	Rp. 25.000, -	1	Rp. 25.000, -
6	<i>Load Cell</i> HX711	Pcs	Rp. 30.000, -	1	Rp. 30.000, -
7	Modul RTC DS1307	Pcs	Rp. 10.000, -	1	Rp. 10.000, -
Total Belanja					Rp. 455.000, -

3.1.3 Analisis Risiko Desain 1

Usulan desain 1 memiliki risiko desain yang dapat mempengaruhi proses pembuatan alat sebagai berikut :

1. Aspek Engineering

Berdasarkan percobaan yang sebelumnya pernah dilakukan pada mata kuliah lain, proses pengkalibrasian *voltage sensor* diperkirakan akan cukup rumit. Sehingga dikhawatirkan dapat menghambat proses penyelesaian pembuatan alat. Kemudian penggunaan baterai jenis AAA akan memiliki beberapa kekurangan dibandingkan baterai Li-ion seperti kapasitas yang lebih kecil serta memiliki kemampuan *self-discharge* yang lebih tinggi, sehingga akan mempengaruhi umur dari baterai tersebut jika digunakan secara intensif.

3.1.4 Pengukuran Performa Desain 1

Pengukuran performa pada alat perlu dilakukan untuk mengetahui apakah sistem pada alat dapat bekerja dengan baik dan sudah sesuai dengan yang diharapkan atau belum. Adapun pengukuran performa pada *Smart AFRESH IO* yang memiliki berbagai macam fitur ini adalah sebagai berikut.

1. Konektivitas Internet

Agar alat dapat berfungsi dengan baik, tentu diperlukan konektivitas internet yang memadai. Analisa performa pada bagian ini adalah apakah alat sudah dapat terkoneksi ke internet dengan baik serta seberapa cepat informasi yang terkirim ditampilkan pada aplikasi untuk *user*. Akan dianalisa apakah terdapat *delay* yang cukup signifikan yang dapat mengganggu proses *monitoring* oleh *user* secara *real time* atau tidak.

2. Monitoring baterai

Untuk fitur *monitoring* baterai dapat dilakukan pengukuran mengenai penurunan tegangan baterai ketika alat digunakan dalam jangka waktu tertentu seperti misal dalam seminggu. Kemudian setelah memperoleh informasi hal tersebut, dapat dilakukan estimasi mengenai berapa lama baterai AAA 4,5V yang digunakan dapat bertahan sesuai dengan interval waktu penyemprotan yang dipilih. Lalu pengkonversian tegangan baterai menjadi persentase juga dapat menjadi analisis bahwa fitur *monitoring* baterai pada alat ini sudah sesuai yang diharapkan atau belum, sebelum dikonversi menjadi persentase tegangan baterai yang tertampil pada aplikasi akan dibandingkan dengan tegangan baterai yang diukur menggunakan multimeter agar didapatkan hasil pembacaan tegangan baterai yang memiliki persentase *error* pembacaan rendah.

3. Monitoring jumlah cairan pengharum ruangan

Pada fitur ini, hasil pembacaan jumlah cairan pengharum ruangan dari sensor *load cell* akan ditampilkan dalam bentuk seperti bar atau dalam satuan ml untuk tiga kondisi yaitu jika kaleng cairan pengharum ruangan penuh, sisa setengah, dan mendekati habis. Untuk mengetahui akurasi, hasil pembacaan dari sensor *load cell* pada ketiga kondisi tersebut akan dibandingkan dengan berat kaleng yang sesungguhnya dengan menggunakan timbangan agar didapatkan hasil pembacaan jumlah cairan pengharum ruangan yang memiliki persentase *error* pembacaan rendah.

4. Penjadwalan otomatis menggunakan RTC

Pada fitur penjadwalan otomatis parameter pengukurannya adalah *delay*, apakah modul RTC (*Real Time Clock*) yang digunakan dapat berfungsi dengan akurat sesuai dengan pengaturan waktu yang telah ditentukan seperti dalam hal ketepatan untuk mematikan alat dan interval waktu penyemprotan. Waktu yang telah dijadwalkan akan dibandingkan dengan waktu sebenarnya.

3.2 Usulan Solusi 2

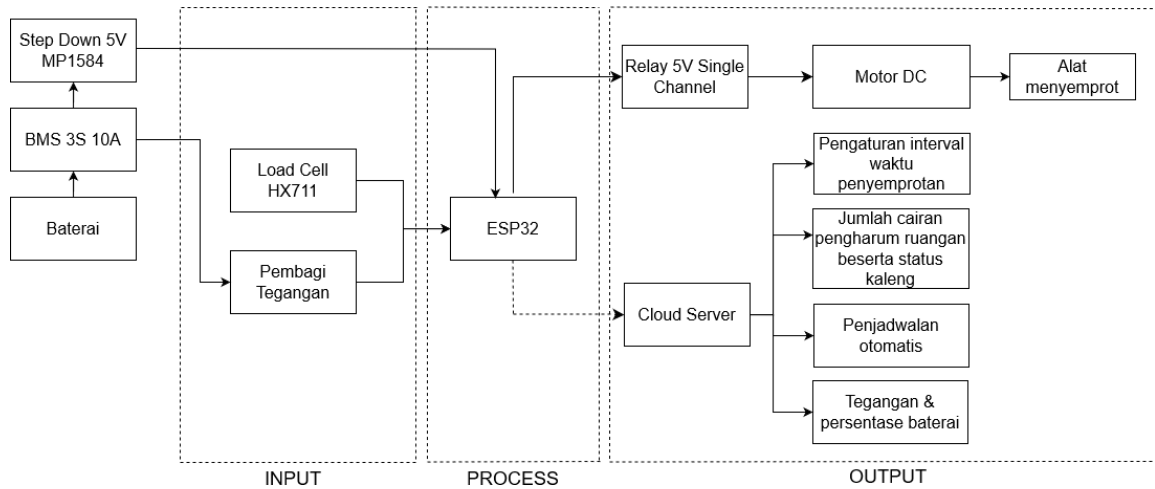
Usulan solusi 2 adalah sistem bernama *Smart AFRESH IO*. Alat ini merupakan dispenser pengharum ruangan yang sudah terintegrasi dengan *internet of things* (IoT) yang dibekali dengan

berbagai macam fitur untuk membantu dan memudahkan pengguna dalam *monitoring* dan *maintenance* dispenser pengharum ruangan secara jarak jauh. Dengan diintegrasikannya alat ini dengan IoT, alat ini mampu memberikan berbagai macam fitur yang sangat berguna bagi pengguna seperti *monitoring* kondisi baterai, jumlah cairan pengharum ruangan, notifikasi ketika kondisi baterai dan jumlah cairan pengharum ruangan sudah mendekati habis, penjadwalan otomatis, serta pengaturan interval waktu penyemprotan. Pada desain 2 ini, baterai yang digunakan adalah 3 buah baterai Li-ion yang bisa di *charge*, menggunakan modul BMS 3S untuk fitur proteksi ketiga baterai tersebut, fitur *monitoring* baterai akan dilakukan menggunakan pembagi tegangan, serta akan digunakan *relay 5V single channel* untuk pengaktifan motor untuk mekanisme penyemprotan.

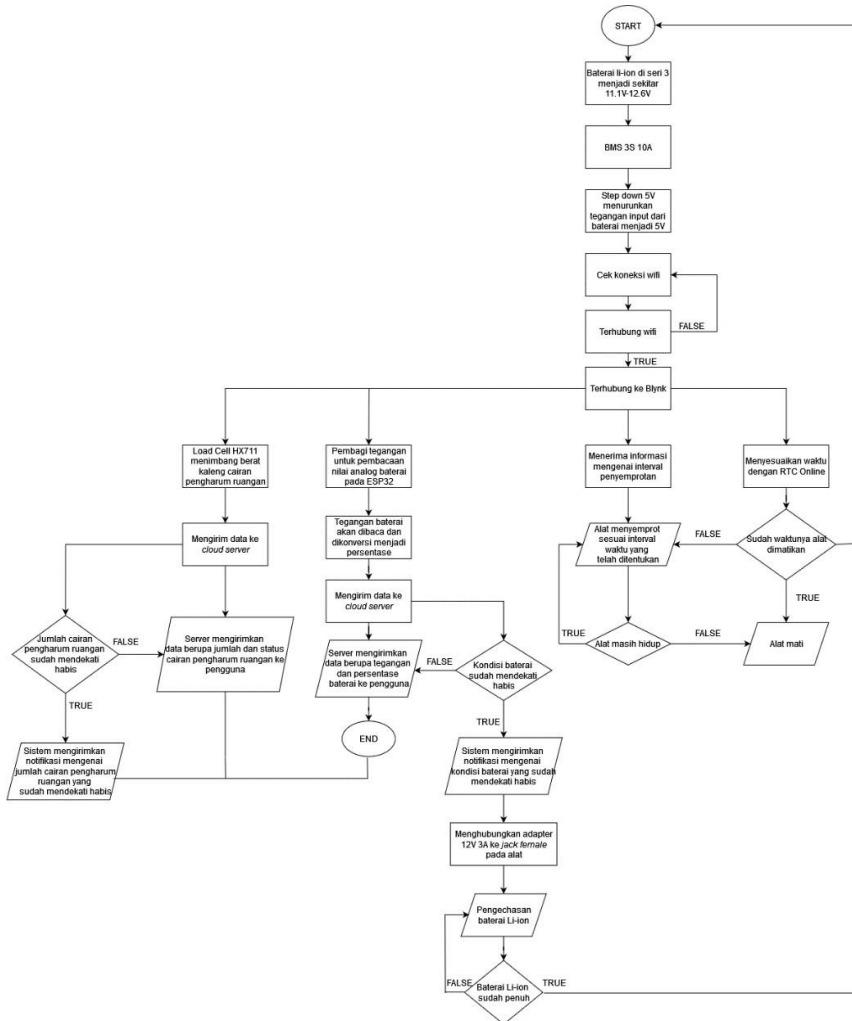
3.2.1 Desain Sistem 2

Gambar 3.6 sampai 3.10 adalah ilustrasi gambaran keseluruhan sistem yang mencakup proses kerja sistem, *wiring* diagram, dan desain alat yang akan dirancang. Pada desain sistem 2 ini daya untuk mikrokontroler ESP32 akan berasal dari 3 buah baterai Li-ion yang di seri menjadi maksimal 12.6 V dan akan terhubung dengan pin Vin milik ESP32. Baterai tersebut akan terhubung dengan modul BMS 3S yang dapat melindungi baterai dari *short circuit*, *overcharge* tegangan dan *overdischarge* tegangan. Kemudian sebelum baterai terhubung ke ESP32, akan terdapat rangkaian pembagi tegangan yang menggunakan dua buah resistor dengan ketentuan $R1 = 3.3 \text{ k}\Omega$ dan $R2 = 1.1 \text{ k}\Omega$ untuk menurunkan tegangan dari sumber tegangan baterai li-ion agar dapat bekerja pada tegangan pin analog milik ESP32 yaitu maksimal 3.3 V untuk fitur *monitoring* tegangan baterai. Baterai tersebut juga terhubung ke modul *step down 5V* yang berfungsi untuk menurunkan tegangan dari baterai menjadi tegangan tetap sebesar 5V agar dapat memberi *power* ke ESP32 sesuai dengan tegangan yang dibutuhkan. Kemudian terdapat *relay single channel 5V* untuk mekanisme penyemprotan. Secara umum cara kerja sistem adalah saat sistem pertama dinyalakan, sistem yang sudah terhubung ke WiFi akan menyesuaikan interval penyemprotan cairan pengharum ruangan sesuai dengan yang telah ditentukan oleh pengguna pada aplikasi. Baterai Li-ion yang sebelumnya sudah terhubung modul BMS lalu ke pembagi tegangan akan terhubung ke pin ADC (GPIO 34) milik ESP32 dan akan mengirimkan hasil pembacaan tegangan serta pembacaan yang telah dikonversi ke persentase ke *cloud server* agar kondisi baterai dapat tertampil pada aplikasi *Blynk*. Sensor berat atau *load cell* akan mengirimkan hasil pembacaan berat kaleng cairan pengharum ruangan ke *cloud server* agar jumlah dan persentase cairan pengharum ruangan dapat tertampil pada aplikasi *Blynk*. Penggunaan *relay single channel 5V*

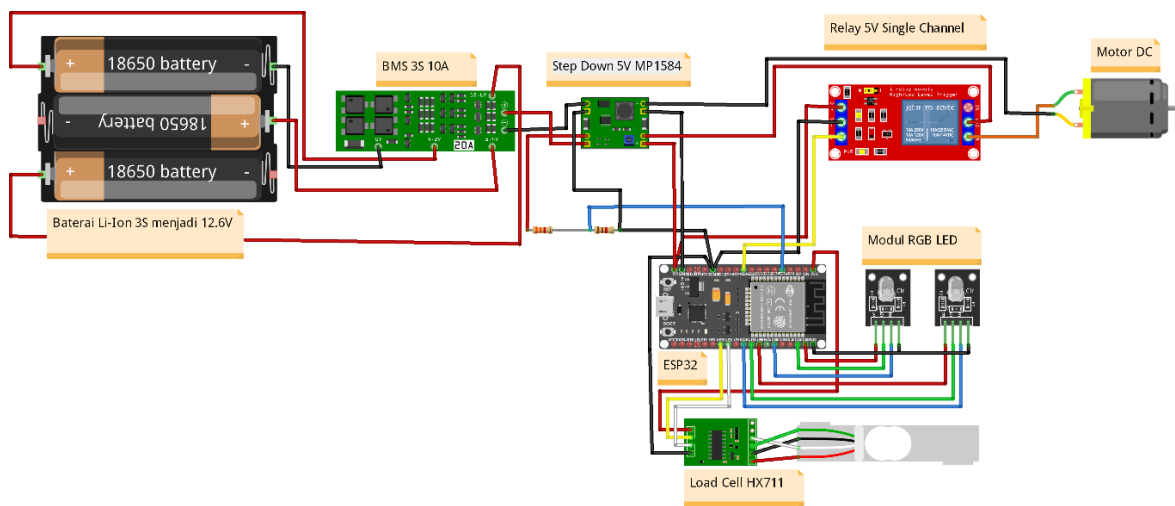
adalah untuk mengaktifkan motor DC pada mekanisme penyemprotan sesuai dengan interval waktu penyemprotan yang telah dipilih oleh *user* pada aplikasi Blynk. Dua buah modul RGB LED yang digunakan berfungsi sebagai indikator visual pada fitur *monitoring* baterai dan *monitoring* cairan pengharum ruangan. Lalu pemrograman pada ESP32 akan menyesuaikan tanggal dan waktu sesuai dengan waktu saat itu juga dari server waktu *real time* yang tersedia di internet untuk melakukan penjadwalan otomatis dimana alat dapat diatur untuk mati atau tidak menyemprot ketika hari sudah menjelang sore atau malam.



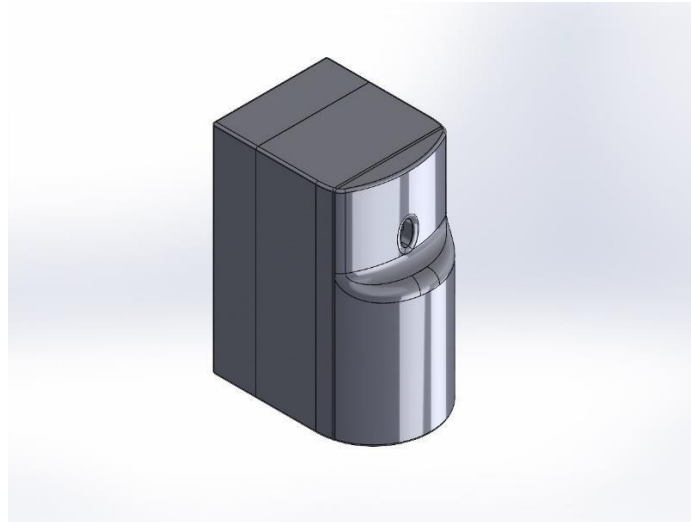
Gambar 3.6 Diagram blok sistem SMART AFRESH IO



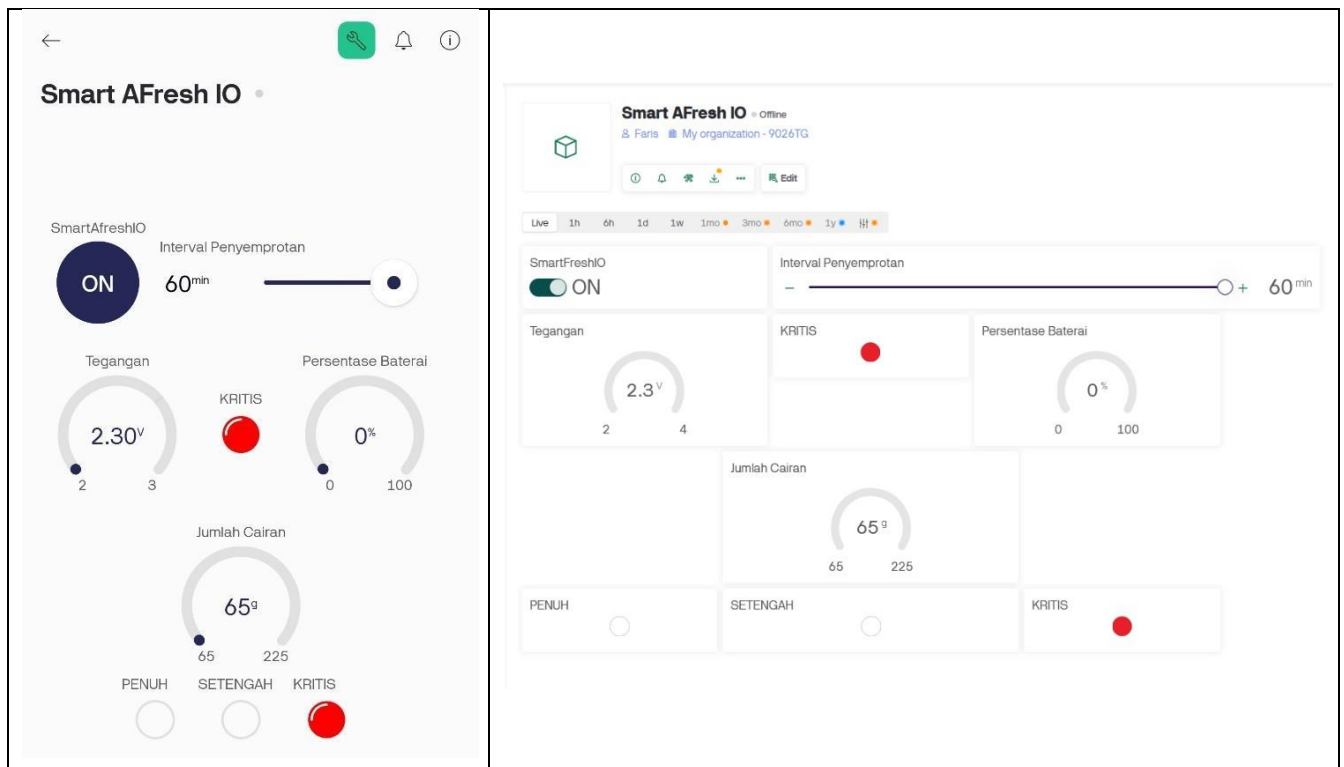
Gambar 3.7 Proses cara kerja sistem usulan solusi 2



Gambar 3.8 Wiring diagram sistem usulan solusi 2



Gambar 3.9 Desain model alat usulan 2 yang akan direalisasikan



Gambar 3.10 Desain UI alat yang akan digunakan oleh user

Desain ini menggunakan baterai Li-ion yang *rechargeable* agar alat memiliki daya tahan yang lebih lama karena kapasitasnya lebih besar dibandingkan baterai AAA. Kalibrasi untuk *monitoring* kondisi baterai juga diperkirakan akan lebih mudah dibandingkan dengan jika menggunakan sensor tegangan. Penggunaan BMS sebagai proteksi tambahan untuk baterai juga

dapat menjaga kualitas dan umur baterai li-ion yang digunakan. Pada gambar 3.10 di atas, dapat diperhatikan ilustrasi mengenai UI aplikasi dari alat yang dirancang, dimana pengguna dapat melakukan *monitoring* baterai, jumlah cairan pengharum ruangan, mengatur frekuensi atau interval waktu penyemprotan. Untuk dapat memenuhi usulan sistem tersebut, maka diperlukan inventarisasi kebutuhan sistem perangkat keras. Tabel 3.3 memperlihatkan kebutuhan sistem sesuai usulan dan spesifikasi yang dibutuhkan.

Tabel 3.3. Inventarisasi kebutuhan usulan sistem perangkat keras sistem 2

No	Nama Alat	Keterangan
1	Casing Dispenser Pengharum Ruang Terintegrasi	Dibuat untuk menjadi tempat <i>mounting</i> alat yang telah didesain agar alat terlihat <i>aesthetic</i> dan dapat melindungi dari suhu lingkungan. Perangkat ini dapat dibuat dari bahan filamen 3D <i>printing</i> maupun bahan sejenis.
2	Mikrokontroler ESP 32	Bagian utama dari sistem untuk kendali dan interkoneksi modul-modul dan komponen-komponen yang digunakan dengan WiFi.
3	Baterai	Baterai yang dapat dengan mudah diganti dan terpisah dari modul utama. Dalam hal ini kami cenderung mencari tipe baterai Li-ion <i>cell</i> dengan tegangan kerja 3,7 V dan kapasitas 1500 mAh. Ini diperkirakan dapat bertahan dalam jangka waktu yang cukup lama.
4	BMS 3S 10A	Modul ini merupakan <i>Battery Management System</i> untuk baterai yang dirangkai seri 3, ia berfungsi untuk melindungi baterai dari <i>short circuit</i> , <i>overcharge</i> tegangan dan <i>overdischarge</i> tegangan.
5	Step down 5V MP1584	Step down 5V MP1584 pada desain ini digunakan untuk menurunkan tegangan dari 3 buah baterai Li-ion yang digunakan pada alat agar dapat men-supply daya ESP32 sesuai dengan tegangan yang dibutuhkan mikrokontroler tersebut.
6	Relay 5V single channel	Komponen ini digunakan untuk mengaktifkan motor DC pada mekanisme penyemprotan sesuai dengan interval waktu penyemprotan yang telah dipilih oleh <i>user</i> pada aplikasi Blynk.
7	Motor DC	Digunakan untuk mekanisme penyemprotan, dimana ketika sudah waktunya untuk menyemprot maka motor DC akan aktif dan menekan katup semprotan kalen untuk menyemprot pengharum ruangan.
8	Pembagi tegangan	Pembagi tegangan pada desain ini digunakan untuk <i>charging</i> baterai Li-ion yang digunakan pada alat, modul ini juga sudah dilengkapi sistem BMS di dalamnya untuk melindungi terjadinya <i>overcharging</i> atau <i>overdischarging</i> serta memiliki indikator pengechasan.
9	Load Cell HX711	Sensor berat yang digunakan adalah <i>load cell</i> HX711, ia akan menimbang berat dari kaleng cairan pengharum ruangan dan hasilnya dikonversi menjadi besaran tertentu (seperti persentase) agar pengguna dapat mengetahui jumlah cairan pengharum ruangan yang tersisa secara <i>real time</i> .

3.2.2 Rencana Anggaran Desain 2

Tabel 3.4. Rencana anggaran pengembangan sistem 2

No.	Item/Pengeluaran	Satuan	Harga satuan	Jumlah	Harga Total
1	Casing Dispenser Pengharum Ruangan Terintegrasi	Pcs	Rp. 275.000, -	1	Rp. 275.000, -
2	Mikrokontroler ESP 32	Pcs	Rp. 80.000, -	1	Rp. 80.000, -
3	Baterai Li-Ion 3,7V	Pcs	Rp. 43.000, -	3	Rp. 129.000, -
4	BMS 3S 10A	Pcs	Rp. 17.000, -	1	Rp. 17.000, -
5	Motor DC	Pcs	Rp. 20.000, -	1	Rp. 20.000, -
6	Relay single channel 5V	Pcs	Rp. 7.000, -	1	Rp. 7.000, -
7	Load Cell HX711	Pcs	Rp. 30.000, -	1	Rp. 30.000, -
8	Step down 5V MP1584	Pcs	Rp. 8.000, -	1	Rp. 8.000, -
9	Resistor ¼ watt metal film 1K Ohm 1%	Pcs	Rp 100,00	4	Rp. 400, -
10	Resistor ¼ watt metal film 100 Ohm 1%	Pcs	Rp 100,00	2	Rp. 200, -
Total Belanja					Rp. 566.600, -

3.2.3 Analisis Risiko Desain 2

Usulan desain 2 memiliki risiko desain yang dapat mempengaruhi proses pembuatan alat sebagai berikut :

1. Aspek Engineering

Pada desain 2 ini modul atau komponen elektronik yang digunakan lebih banyak dibandingkan dengan desain 1, sehingga dikhawatirkan desain casing alat atau bentuk alat secara keseluruhan menjadi lebih besar karena banyaknya komponen yang digunakan tersebut. Selain itu, penggunaan baterai Li-ion yang memiliki ukuran lebih besar daripada baterai AAA juga dapat berkontribusi terhadap resiko tersebut. Kemudian rangkaian elektronis juga akan menjadi sedikit lebih kompleks karena tambahan beberapa modul yang digunakan tersebut.

2. Aspek Ekonomi

Resiko pada aspek ekonomi untuk desain 2 ini adalah biaya pembuatan alat. Biaya pembuatan alat untuk desain 2 ini diperkirakan akan lebih tinggi dibandingkan desain 1 dikarenakan modul atau komponen elektronis yang perlu dibeli lebih banyak. Selain itu, ukuran casing yang diperkirakan lebih besar atau kompleks juga akan menambah biaya pembuatan karena akan menggunakan lebih banyak material dalam pembuatan casing-nya.

3.2.4 Pengukuran Performa Desain 2

Pengukuran performa pada alat perlu dilakukan untuk mengetahui apakah sistem pada alat dapat bekerja dengan baik dan sudah sesuai dengan yang diharapkan atau belum. Adapun pengukuran performa pada *Smart AFRESH IO* yang memiliki berbagai macam fitur ini adalah sebagai berikut.

1. Konektivitas internet

Agar alat dapat berfungsi dengan baik, tentu diperlukan konektivitas internet yang memadai. Analisa performa pada bagian ini adalah apakah alat sudah dapat terkoneksi ke internet dengan baik serta seberapa cepat informasi yang terkirim ditampilkan pada aplikasi untuk *user*. Akan dianalisa apakah terdapat *delay* yang cukup signifikan yang dapat mengganggu proses *monitoring* oleh *user* secara *real time* atau tidak.

2. Monitoring baterai

Untuk fitur *monitoring* baterai dapat dilakukan pengukuran mengenai penurunan tegangan baterai ketika alat digunakan dalam jangka waktu tertentu seperti misal setiap 1 jam. Lalu setelah memperoleh informasi hal tersebut, dapat dilakukan estimasi mengenai berapa lama baterai Li-ion yang di seri 3 tersebut dapat bertahan sesuai dengan interval waktu penyemprotan yang dipilih. Lalu pengkonversian tegangan baterai menjadi persentase juga dapat menjadi analisis bahwa fitur *monitoring* baterai pada alat ini sudah sesuai yang diharapkan atau belum, sebelum dikonversi menjadi persentase tegangan baterai yang tertampil pada aplikasi akan dibandingkan dengan tegangan baterai yang diukur menggunakan multimeter agar didapatkan hasil pembacaan tegangan baterai yang memiliki persentase *error* pembacaan rendah.

3. Monitoring jumlah cairan pengharum ruangan

Pada fitur ini, hasil pembacaan jumlah cairan pengharum ruangan dari sensor *load cell* akan ditampilkan dalam bentuk seperti bar atau dalam satuan ml untuk tiga kondisi yaitu jika kaleng cairan pengharum ruangan penuh, sisa setengah, dan mendekati habis. Untuk mengetahui akurasi, hasil pembacaan dari sensor *load cell* pada ketiga kondisi tersebut akan dibandingkan dengan berat kaleng yang sesungguhnya dengan menggunakan timbangan agar didapatkan hasil pembacaan jumlah cairan pengharum ruangan yang memiliki persentase *error* pembacaan rendah.

4. Penjadwalan otomatis menggunakan RTC *online*

Pada fitur penjadwalan otomatis parameter pengukurannya adalah *delay*, apakah RTC (*Real Time Clock*) *online* yang digunakan dapat berfungsi dengan akurat sesuai dengan pengaturan waktu yang telah ditentukan seperti dalam hal ketepatan untuk mematikan mekanisme penyemprotan alat. Waktu yang telah dijadwalkan akan dibandingkan dengan waktu sebenarnya.

3.3 Analisis dan Penentuan Usulan Solusi/Desain Terbaik

Berdasarkan kedua usulan solusi yang telah dijelaskan sebelumnya, penulis memutuskan bahwa desain sistem 2 merupakan usulan atau solusi terbaik karena beberapa alasan, diantaranya adalah *Smart AFRESH IO* akan memiliki daya tahan yang lebih lama karena menggunakan baterai li-ion yang kapasitasnya lebih besar dibandingkan baterai AAA. Adapun penulis lebih memilih untuk menggunakan baterai sebagai catu daya dari alat adalah karena alat umumnya akan diletakkan pada tempat tinggi seperti tembok, sehingga penting untuk membuat alat ini bersifat *portable* dan tidak menggunakan sumber daya dari stop kontak. Penggunaan RTC *online* memungkinkan alat untuk mematikan mode penyemprotan pada saat bukan jam kerja seperti

malam hari, sehingga dapat menghemat konsumsi daya baterai. Kemudian kalibrasi untuk *monitoring* kondisi baterai juga diperkirakan akan lebih mudah jika dibandingkan dengan menggunakan sensor tegangan. Lalu penggunaan BMS 3S 10A yang merupakan (*Battery Management System*) akan memberikan fitur proteksi tegangan untuk menjaga kualitas baterai serta agar masa kerja baterai tetap optimum.

3.4 Gantt Chart

Pada laporan Tugas Akhir ini kami memvisualisasikan beberapa hal, yaitu tanggal dimulainya suatu proyek, berbagai tugas yang harus dikerjakan, anggota yang mengerjakan tugas, waktu dimulai dan diselesaikan tugas, lamanya waktu yang diperlukan, pengelompokan tugas antar anggota, dan tanggal diselesaikan proyek. Berikut *ganttt chart* untuk proyek *Smart AFRESH IO* pada Tugas Akhir ini :

Tabel 3.5 *Gantt chart* pelaksanaan *Capstone Project* sistem *Smart AFRESH IO*

No.	Kegiatan/Capaian	Bulan ke -											
		9	10	11	12	1	2	3	4	5	6	7	8
1	Survei dan identifikasi permasalahan	W, F											
2	Mencari literatur dan informasi untuk kebutuhan dan spesifikasi sistem	W, F	W, F										
3	Mengumpulkan seluruh ide solusi dan finalisasi usulan perancangan sistem beserta manajemen dan rancangan belanja			W, F	W, F								
4	Pengumpulan proposal Tugas Akhir 1/ <i>Capstone Project</i> dan seminar				W, F								
5	Pembelian alat dan bahan				W, F	W, F	W, F						
6	Perancangan sistem sesuai proposal						W, F	W, F	W, F				

No.	Kegiatan/Capaian	Bulan ke -											
		9	10	11	12	1	2	3	4	5	6	7	8
7	Testing, validasi dan penulisan laporan akhir								W, F	W, F			
8	Expo dan pengumpulan laporan akhir										W, F		

Ket. : PIC – *Person in Charge* (Pihak yang bertanggung untuk kegiatan tersebut) W : Wildan, F : Faris

3.5 Realisasi Pelaksanaan Tugas Akhir 1

Tabel 3.6 Realisasi aktivitas pelaksanaan tugas akhir 1

No	Hari, Tanggal, Durasi (jam atau hari)	Aktivitas	Pelaksana
1	Jumat, 15 September 2023, 3 jam	Memahami dan mencari informasi seputar kebutuhan-kebutuhan pengguna dan problem-problem yang ada mengenai dispenser pengharum ruangan otomatis saat ini	Faris Wildan
2	Sabtu, 23 September 2023	Mulai membuat latar belakang proposal	Faris Wildan
3	Minggu, 24 September 2023, 1 jam	Membuat form kuesioner untuk survey perihal permasalahan dispenser pengharum ruangan otomatis	Faris Wildan
4	Senin, 25 September 2023	Menyebarkan form kuesioner ke kerabat, rekan-rekan kami, serta sejumlah petugas <i>cleaning service</i> dan staf-staf bagian akademik Fakultas Teknologi Industri UII	Faris Wildan

5	Senin, 25 September 2023	Membuat rumusan masalah dan menyelesaikan latar belakang	Faris Wildan
6	Senin, 2 Oktober 2023	Membuat tujuan, batasan masalah, dan batasan realistik aspek keteknikan, serta mulai membuat studi literatur dan observasi (BAB 2)	Faris Wildan
7	9-16 Oktober 2023	Menyelesaikan studi literatur dan observasi, serta mulai mengerjakan dasar teori	Faris Wildan
8	22-23 Oktober 2023	Membuat analisis <i>stakeholder</i> , analisis aspek yang mempengaruhi sistem, dan spesifikasi sistem	Faris Wildan
9	Sabtu & Minggu, 4-5 November 2023	Membuat penjelasan, proses kerja, <i>wiring</i> diagram usulan desain sistem 1	Faris Wildan
10	Selasa, 7 November 2023	Membuat RAB usulan desain sistem 1	Faris Wildan
11	11-13 November 2023	Membuat penjelasan, proses kerja, <i>wiring</i> diagram usulan desain sistem 2, serta membuat analisis resiko desain untuk usulan solusi desain 1 dan desain 2	Faris Wildan
12	Sabtu, 18 November 2023	Membuat desain 3D dan desain UI untuk usulan solusi desain 1 dan 2	Faris Wildan
13	19-20 November 2023	Membuat pengukuran performa untuk usulan solusi desain 1 dan desain 2	Faris Wildan

14	Senin, 27 November 2023	Menentukan usulan solusi desain terbaik dan membuat diagram blok untuk kedua usulan solusi desain	Faris Wildan
15	Selasa, 28 November 2023	Membuat <i>Gantt chart</i>	Faris Wildan
16	Minggu, 3 December 2023	Mengisi tabel realisasi aktivitas pelaksanaan tugas akhir 1	Faris Wildan

BAB 4. HASIL RANCANGAN DAN METODE PENGUKURAN

4.1 Hasil Rancangan Sistem

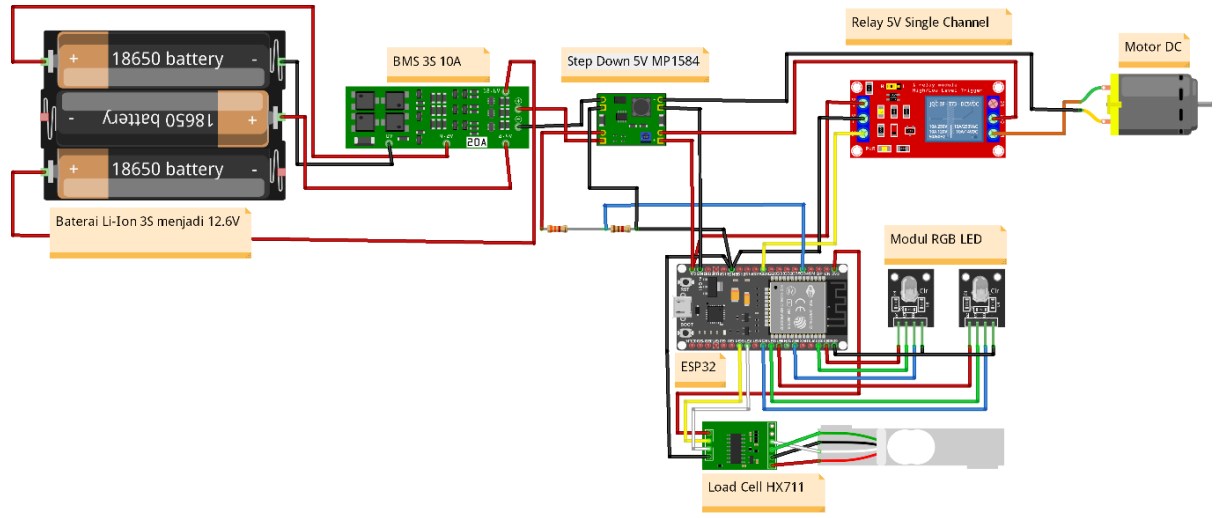
4.1.1 Flowchart Coding Smart AFRESH IO



Gambar 4.1 Flowchart coding SMART AFRESH IO

Gambar 4.1 di atas merupakan *flowchart* dari SMART AFRESH IO. Pada *flowchart* tersebut terdapat berbagai macam fungsi *void*, beberapa diantaranya yang utama adalah untuk membuat fitur-fitur yang disediakan oleh alat ini, yaitu pengukuran berat kaleng cairan pengharum ruangan menggunakan sensor load cell HX711, *memonitoring* tegangan baterai, dan mengontrol *relay* untuk mekanisme penyemprotan. Untuk fitur *monitoring* tegangan baterai terdapat komputasi untuk mengolah tegangan dengan metode rata-rata setiap 20 detik agar didapatkan hasil pembacaan tegangan pada Blynk yang sesuai atau mendekati dengan pengukuran tegangan pada multimeter. Data yang dikumpulkan kemudian dikirim ke platform Blynk untuk *monitoring* dari jarak jauh serta dicatat ke *Google Sheets* untuk pencatatan data secara *real time*. Pada bagian *void setup* terdapat inisialisasi koneksi Wi-Fi yang dikelola oleh WiFiManager dan memanfaatkan EEPROM yang memiliki fungsi menyimpan pengaturan durasi aktif *relay* untuk fitur pengaturan interval waktu penyemprotan dan status *relay*. Selain itu mekanisme pengaturan *relay* juga diatur berdasarkan waktu kerja yang telah ditentukan yaitu dari jam 8 pagi sampai jam 5 sore, fitur penjadwalan otomatis tersebut memanfaatkan NTP server untuk menyinkronkan waktu agar didapatkan data yang akurat. Pada bagian *void setup* juga dideklarasikan bahwa dua pin LED RGB yang digunakan pada alat merupakan komponen *output* yang berfungsi untuk menunjukkan status baterai dan berat kaleng cairan pengharum ruangan secara visual pada alat. Lalu dilakukan penghubungan ke Blynk, *setup* fungsi Timer untuk inisialisasi fungsi-fungsi yang akan dipanggil pada bagian utama *coding* dan kalibrasi *load cell*. Kemudian pada bagian utama *coding* yaitu *void loop* hanya dijalankan fungsi Blynk dan Timer, dimana semua *coding* yang berhubungan dengan Blynk dan Timer tersebut akan dipanggil secara terus menerus, sehingga alat dapat mendapatkan dan memberikan data-data yang telah diperoleh lalu diolah menjadi fitur-fitur yang disediakan oleh SMART AFRESH IO ini.

4.1.2 Wiring Diagram

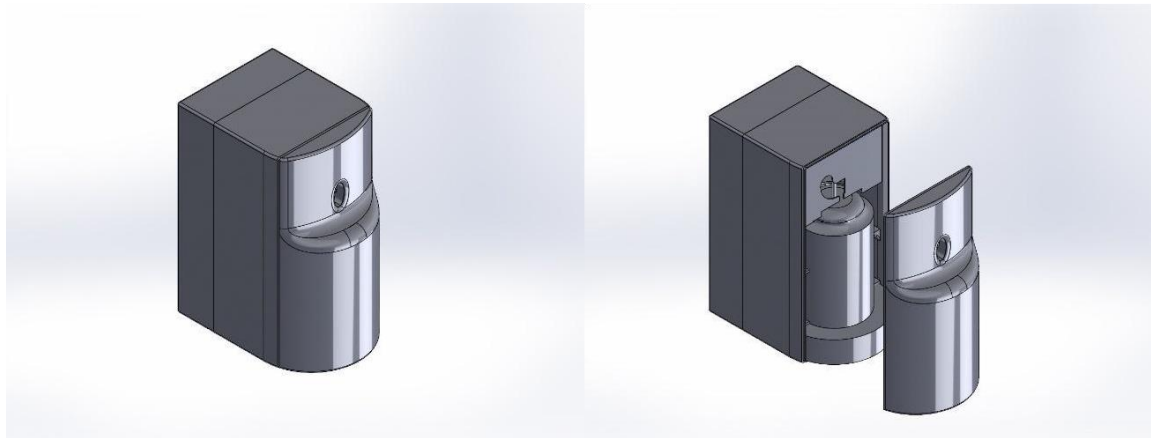


Gambar 4.2 Wiring diagram SMART AFRESH IO

Pada gambar 4.2 di atas yang merupakan rangkaian dari SMART AFRESH IO, daya untuk mikrokontroler ESP32 akan berasal dari 3 buah baterai Li-ion yang di seri menjadi maksimal 12.6 V dan akan terhubung dengan pin Vin milik ESP32. Baterai tersebut akan terhubung dengan modul BMS 3S yang dapat melindungi baterai dari *short circuit*, *overcharge* tegangan dan *overdischarge* tegangan. Lalu sebelum baterai terhubung ke ESP32, akan terdapat rangkaian pembagi tegangan yang menggunakan dua buah resistor dengan ketentuan $R1 = 3.3 \text{ k}\Omega$ dan $R2 = 1.1 \text{ k}\Omega$ untuk menurunkan tegangan dari sumber tegangan baterai li-ion agar dapat bekerja pada tegangan pin analog milik ESP32 yaitu maksimal 3.3 V untuk fitur *monitoring* tegangan baterai. Baterai tersebut juga terhubung ke modul *step down 5V* yang berfungsi untuk menurunkan tegangan dari baterai menjadi tegangan tetap sebesar 5V agar dapat memberi *power* ke ESP32 sesuai dengan tegangan yang dibutuhkan. Kemudian terdapat *relay single channel 5V* untuk mekanisme penyemprotan. Secara umum cara kerja sistem adalah saat sistem pertama dinyalakan, sistem yang sudah terhubung ke WiFi akan menyesuaikan interval penyemprotan cairan pengharum ruangan sesuai dengan yang telah ditentukan oleh pengguna pada aplikasi. Baterai Li-ion yang sebelumnya sudah terhubung modul BMS lalu ke pembagi tegangan akan terhubung ke pin ADC (GPIO 34) milik ESP32 dan akan mengirimkan hasil pembacaan tegangan serta pembacaan yang telah dikonversi ke persentase ke *cloud server* agar kondisi baterai dapat tertampil pada aplikasi *Blynk*. Sensor berat atau *load cell* akan mengirimkan hasil pembacaan berat kaleng cairan pengharum ruangan ke *cloud*

server agar jumlah dan persentase cairan pengharum ruangan dapat tertampil pada aplikasi *Blynk*. Penggunaan *relay single channel 5V* adalah untuk mengaktifkan motor DC pada mekanisme penyemprotan sesuai dengan interval waktu penyemprotan yang telah dipilih oleh *user* pada aplikasi *Blynk*. Dua buah modul RGB LED yang digunakan berfungsi sebagai indikator visual pada fitur *monitoring* baterai dan *monitoring* cairan pengharum ruangan. Lalu pemrograman pada ESP32 akan menyesuaikan tanggal dan waktu sesuai dengan waktu saat itu juga dari server waktu *real time* yang tersedia di internet untuk melakukan penjadwalan otomatis dimana alat dapat diatur untuk mati atau tidak menyemprot ketika hari sudah menjelang sore atau malam.

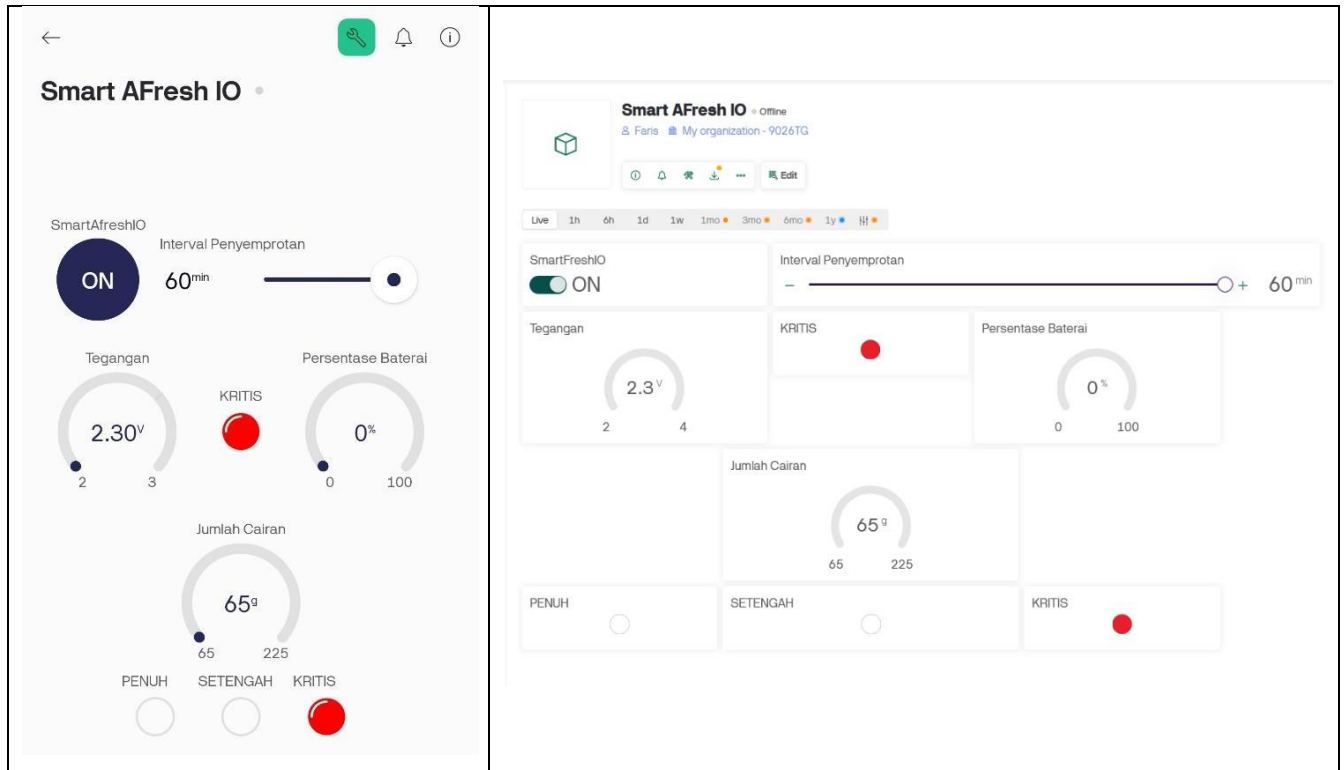
4.1.3 Gambar Desain Tiga Dimensi (3D)



Gambar 4.3 Desain model alat usulan 2 yang direalisasikan

Gambar 4.3 di atas merupakan desain 3D dari SMART AFRESH IO. Sensor *load cell* berada di bagian bawah alat dimana kaleng akan diletakkan sebagai penimbang berat kaleng cairan pengharum ruangan. Kemudian pada bagian samping kanan dari alat Smart AFRESH IO ini terdapat indikator berupa LED untuk menunjukkan kondisi baterai dan cairan pengharum ruangan yang terbagi ke dalam tiga warna yaitu hijau, kuning dan merah. Apabila LED berwarna hijau berarti kapasitas baterai maupun cairan pengharum ruangan masih penuh, jika LED berwarna kuning berarti kapasitas sudah setengah, lalu jika LED berwarna merah berarti kapasitas kedua hal tersebut sudah mendekati habis atau sudah habis. Selain itu pada bagian samping ini juga terdapat saklar untuk mematikan sistem dari alat secara keseluruhan agar pengecasan baterai li-ion pada alat dapat dilakukan dengan lebih aman. Komponen lainnya akan diletakkan pada bagian belakang dari alat.

4.1.4 User Interface



Gambar 4.4 Desain UI alat yang akan digunakan oleh user

Gambar 4.4 di atas merupakan UI aplikasi *Blynk* dari alat yang dirancang, dimana pengguna dapat melakukan *monitoring* tegangan, persentase dan kondisi kritis baterai, status kondisi jumlah cairan pengharum ruangan yang terbagi menjadi 3 yaitu penuh, setengah dan kritis, mengatur frekuensi atau interval waktu penyemprotan, dan menyalakan serta mematikan alat dengan tombol ON atau OFF. Interval waktu penyemprotan yang dapat dipilih oleh pengguna adalah antara setiap 10, 20, 30, 40, 50, dan 60 menit sekali.

PENUH	190,7	49	2,79	12/07/2024 15:32:57
PENUH	189,8	49	2,79	12/07/2024 15:33:58
PENUH	187,8	49	2,79	12/07/2024 15:34:58
PENUH	186,9	49	2,79	12/07/2024 15:35:59
PENUH	186,8	49	2,79	12/07/2024 15:36:59
SETENGAH	169,9	32	2,62	12/07/2024 16:20:27
SETENGAH	169,8	32	2,62	12/07/2024 16:21:27
SETENGAH	169,6	32	2,62	12/07/2024 16:22:28
SETENGAH	169,5	32	2,62	12/07/2024 16:23:29
HABIS	54,1	0	0,98	06/07/2024 3:53:44
HABIS	54,2	0	0,98	06/07/2024 3:54:45
HABIS	54,3	0	0,98	06/07/2024 3:55:46
HABIS	54,4	0	0,98	06/07/2024 3:56:46

Gambar 4.5 Tampilan *Datalogging* pada *google sheets*

Gambar 4.5 di atas merupakan tampilan fitur tambahan *datalogging* menggunakan *google sheets* pada alat ini dengan keterangan kolom dari kiri ke kanan secara berurutan yaitu status isi kaleng cairan pengharum ruangan, berat kaleng cairan pengharum ruangan dalam satuan gram, persentase baterai, tegangan baterai dan tanggal waktu ketika data tersimpan di *google sheets*. Dengan fitur ini, *user* dapat menyimpan dan mengolah data-data yang diberikan atau diperoleh dari alat ini dengan mudah.

4.1.5 Hasil Perancangan Akhir



Gambar 4.6 Hasil perancangan akhir SMART AFRESH IO

Dapat diperhatikan pada gambar 4.6 di atas yang merupakan hasil perancangan dari SMART AFRESH IO. *Load cell* menopang kaleng cairan pengharum ruangan, bagian belakang alat merupakan box dari berbagai sensor dan komponen yang digunakan pada alat ini. Led sebanyak 2 buah akan digunakan sebagai indikator dari persentase baterai dan kaleng cairan pengharum ruangan. *Cover* dari alat ini serta bagian belakang yaitu box komponennya menggunakan engsel dan memiliki sistem penguncian.

4.2 Metode Pengukuran Kinerja Hasil Perancangan

4.2.1 Pengukuran *Monitoring* Baterai

Parameter pertama yang diukur adalah tegangan yang terbaca yaitu seberapa akurat tegangan baterai dan pengkonversiannya menjadi persentase baterai yang tertampil pada aplikasi *Blynk*.

1. Mengukur tegangan baterai yang melewati pembagi tegangan menggunakan multimeter terlebih dahulu
2. Pada *coding*, pin ADC ESP32 yang terhubung dengan pembagi tegangan akan membaca nilai analog yang didapatkan dari pembagi tegangan tersebut
3. Kemudian nilai analog tersebut dikonversi ke voltase
4. Setelah itu dilakukan kalibrasi agar nilai voltase yang dihasilkan *coding* sesuai atau mendekati dengan nilai voltase dari pengukuran menggunakan multimeter, dapat dilihat pada gambar 4.7 di bawah yang merupakan *coding* bagian kalibrasi tegangan, angka 4095 merupakan nilai *input* analog yang dapat dibaca oleh ESP32 karena pin ADC mikrokontroler ini memiliki resolusi sebesar 12 bit. Angka 3.3 merupakan tegangan kerja dari ESP32 tersebut. Kemudian variabel “*calibrationFactor*” sebesar 1.06 didapatkan dari komputasi rata-rata dari 3 *sample* nilai mentah atau nilai analog saat *coding* pertama kali dijalankan sebelum proses kalibrasi. Variabel “*offset*” merupakan komputasi untuk mengurangi atau menambah nilai tegangan tanpa harus mengganti nilai kalibrasi pada variabel “*calibrationFactor*”. Setelah itu akan didapatkan nilai tegangan pada variabel “*voltage*” dengan mengkalikan nilai tegangan pada proses *looping* sebelumnya dengan variabel “*calibrationFactor*” dan menjumlahkannya dengan nilai variabel “*offset*”.

```
//float voltage = (averageRawValue / 4095.0) * 3.3;
voltage = (averageRawValue / 4095.0) * 3.3;
const float calibrationFactor = 1.06; // Calibration factor
const float offset = -0.18 + 0.23; // Additional offset, 3 July 2024 : +0.33 (multi 2.50, blynk 2.60)
voltage = (voltage * calibrationFactor) + offset;
```

Gambar 4.7 *Coding* kalibrasi tegangan untuk fitur *monitoring* baterai

5. Jika nilai voltase sudah sesuai maka dilakukan pengkonversian nilai voltase menjadi persentase, jika nilai voltase masih tidak sesuai dengan hasil pengukuran multimeter maka dilakukan kalibrasi ulang dengan mengganti nilai kalibrasi pada *coding* agar diperoleh pembacaan tegangan baterai yang memiliki persentase *error* pembacaan rendah
6. Nilai voltase dan persentase untuk fitur *monitoring* baterai yang telah didapatkan akan ditampilkan pada aplikasi Blynk

4.2.2 Pengukuran *Monitoring* Cairan Pengharum Ruangan

Parameter berikutnya adalah berat kaleng cairan pengharum ruangan, dimana hasil pembacaan dari sensor *load cell* akan ditampilkan dalam bentuk satuan ml atau gram dan persentase bar.

1. Langkah pertama adalah melakukan kalibrasi dari sensor *load cell* yang digunakan
2. Lalu melakukan pengukuran berat dari hasil kalibrasi dengan meletakkan kaleng cairan pengharum ruangan di atas *load cell*. Perhatikan gambar 4.8 di bawah, nilai “998.2111” diperoleh dari pembagian nilai mentah sebesar 179678 saat menjalankan *coding* kalibrasi *load cell* secara terpisah dari *coding* utama dengan berat kaleng kondisi setengah yang diketahui yaitu 180 gram.

```
scale.set_scale(998.2111); //998.2111 //2 July 2024 : 9
//scale.set_scale(-471.497); // thi
scale.tare(); // reset the scale to 0
```

Gambar 4.8 *Coding* kalibrasi *load cell*

3. Jika berat sudah sesuai dengan hasil pengukuran berat menggunakan timbangan asli maka konversi hasil pembacaan berat ke tiga status kaleng yang berbeda yaitu penuh, setengah dan kosong, jika berat belum sesuai dengan hasil pengukuran menggunakan timbangan asli maka lakukan kalibrasi ulang agar didapatkan hasil pembacaan jumlah cairan pengharum ruangan yang memiliki persentase *error* pembacaan rendah
4. Nilai berat kaleng cairan pengharum ruangan beserta status kalengnya untuk fitur *monitoring* cairan pengharum ruangan tersebut akan ditampilkan pada aplikasi Blynk

4.2.3 Pengukuran Delay IoT

Kemudian parameter selanjutnya adalah *delay*, apakah alat dapat berfungsi dengan akurat serta responsif sesuai dengan pengaturan waktu yang telah ditentukan seperti dalam hal ketepatan untuk mematikan mekanisme penyemprotan dan pemilihan interval waktu penyemprotan. Waktu yang telah dijadwalkan dibandingkan dengan waktu sebenarnya.

1. Menyalakan Smart AFRESH IO dengan menekan saklar ON pada bagian samping alat dan hidupkan *stopwatch* secara bersamaan
2. Perhatikan tampilan Blynk untuk *device* Smart AFRESH IO, jika indikator lampu hijau di sebelah nama alat sudah menyala berarti sistem sudah terhubung dengan Blynk dan dapat digunakan, hentikan *stopwatch* untuk mengetahui berapa *delay* yang diperoleh untuk menyinkronkan status bahwa alat sudah menyala pada Blynk
3. Kemudian untuk fitur pengaturan interval waktu penyemprotan, pilih interval waktu penyemprotan yang diinginkan pada Blynk antara 10, 20, 30, 40, 50 dan 60 menit dan nyalakan *timer* dengan waktu yang sama
4. Apabila alat menyemprot saat waktu pada *timer* sudah habis, berarti pengguna sudah dapat mengatur interval waktu penyemprotan pada Blynk sesuai dengan yang diinginkan
5. Kemudian untuk fitur penjadwalan otomatis menggunakan RTC *online*, nyalakan alat dan biarkan alat menyemprot dengan interval waktu penyemprotan yang telah dipilih pada Blynk
6. Perhatikan waktu pada saat itu juga, apabila alat menyemprot di rentang jam yang telah ditentukan yaitu antara jam 9 pagi sampai jam 5 sore berarti fitur penjadwalan otomatis sudah sesuai dengan yang direncanakan yaitu hanya bekerja pada saat jam kerja

4.2.4 Pengukuran Daya Tahan Baterai SMART AFRESH IO

Parameter berikutnya adalah berapa lama daya tahan baterai yang digunakan dalam alat SMART AFRESH IO ini, berikut adalah langkah-langkah pengujiannya:

1. Nyalakan SMART AFRESH IO dan hidupkan *stopwatch* secara bersamaan, ukur tegangan awal saat alat dinyalakan
2. Tunggu hingga alat mati, jika alat sudah mati berarti BMS sudah memutus tegangan dari baterai ke alat
3. Perhatikan data tegangan yang tercatat pada *Google Sheets* dan hentikan *stopwatch*

4.2.5 Pengukuran *Charging* Baterai SMART AFRESH IO

Parameter selanjutnya adalah berapa lama waktu *charging* baterai yang digunakan dalam alat SMART AFRESH IO ini, berikut langkah-langkahnya:

1. Habiskan baterai alat terlebih dahulu
2. Lalu colokkan adaptor 12V 3A ke *jack female* pada bagian samping kiri bawah alat dan nyalakan *stopwatch*
3. Tunggu hingga baterai terisi penuh, jika baterai sudah terisi penuh matikan *stopwatch* dan perhatikan dan catat berapa lama waktu yang dibutuhkan untuk *charging*

BAB 5. HASIL PENGUKURAN DAN ANALISIS

Pengujian yang telah dilakukan mencakupi pengukuran akurasi pada alat SMART AFRESH IO yang telah dibuat. Kinerja alat akan dianalisa berdasarkan parameter-parameter yang telah ditentukan sebelumnya, dari aspek sensor atau komponen seperti baterai dan sensor *load cell*, serta dari aspek seperti *delay* pada beberapa fitur yang disediakan oleh alat ini yaitu *monitoring* baterai, *monitoring* isi kaleng cairan pengharum ruangan, pengaturan interval waktu penyemprotan, penjadwalan otomatis dan notifikasi. Analisis ini diharapkan dapat memberikan wawasan yang jelas mengenai efektivitas dan efisiensi prototipe alat SMART AFRESH IO dalam memenuhi tujuan desain yang telah direncanakan.

5.1. Analisis Hasil

5.1.1 Hasil dan Analisis Pengujian Parameter

Berikut merupakan hasil dan analisis dari pengujian beberapa parameter pada alat yang sudah kami buat, parameter yang diuji dari alat ini yaitu seberapa akurat fitur *monitoring* baterai, *monitoring* isi kaleng cairan pengharum ruangan, lama waktu *charging* baterai yang digunakan dalam alat dan *delay* yang meliputi tombol ON/OFF mekanisme penyemprotan, *slider* interval waktu penyemprotan, penjadwalan otomatis, notifikasi dan data *logging*.

5.1.1.1 Hasil dan Analisis *Monitoring* Baterai

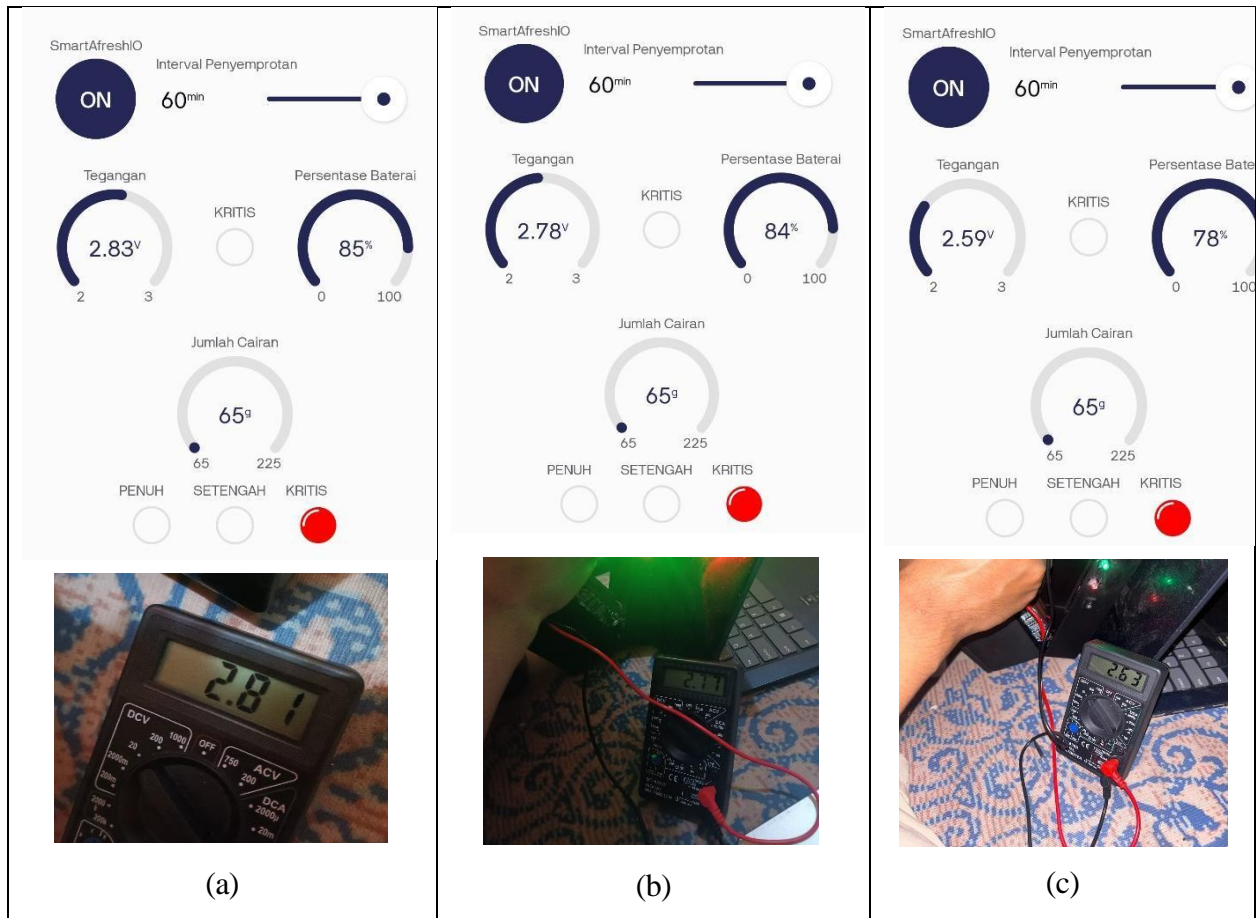
Pada bagian ini dilakukan pengujian pada fitur *monitoring* baterai yang digunakan dalam alat. Perlu diketahui bahwa pengujian dilakukan pada tegangan baterai yang melewati pembagi tegangan, berikut hasil pengujian kami pada fitur *monitoring* baterai:

$$\text{Persentase error} = \left| \frac{(\text{nilai aktual} - \text{nilai pembacaan})}{\text{nilai aktual}} \right| \times 100\% \quad 1)$$

$$\text{Akurasi} = 100\% - \frac{(\text{total nilai error})}{\text{banyaknya sampel}} \quad 2)$$

Tabel 5.1 Hasil Pengukuran Fitur *Monitoring* Baterai

No	Komponen yang Digunakan	Tegangan Aktual Pembagi Tegangan (V)	Hasil Tegangan pada Blynk (V)	Persentase <i>Error</i>	Akurasi
1	Pembagi Tegangan	2.81	2.83	0.71%	98.94%
2		2.77	2.78	0.36%	
3		2.75	2.74	0.36%	
4		2.72	2.66	2.20%	
5		2.63	2.59	1.52%	
6		2.50	2.47	1.2%	
Rata-rata persentase <i>error</i>				1.06%	



Gambar 5.1 Pengujian *error monitoring* baterai, (a) Pengujian *error monitoring* baterai pertama, (b) Pengujian *error monitoring* baterai kedua, (c) Pengujian *error monitoring* baterai kelima

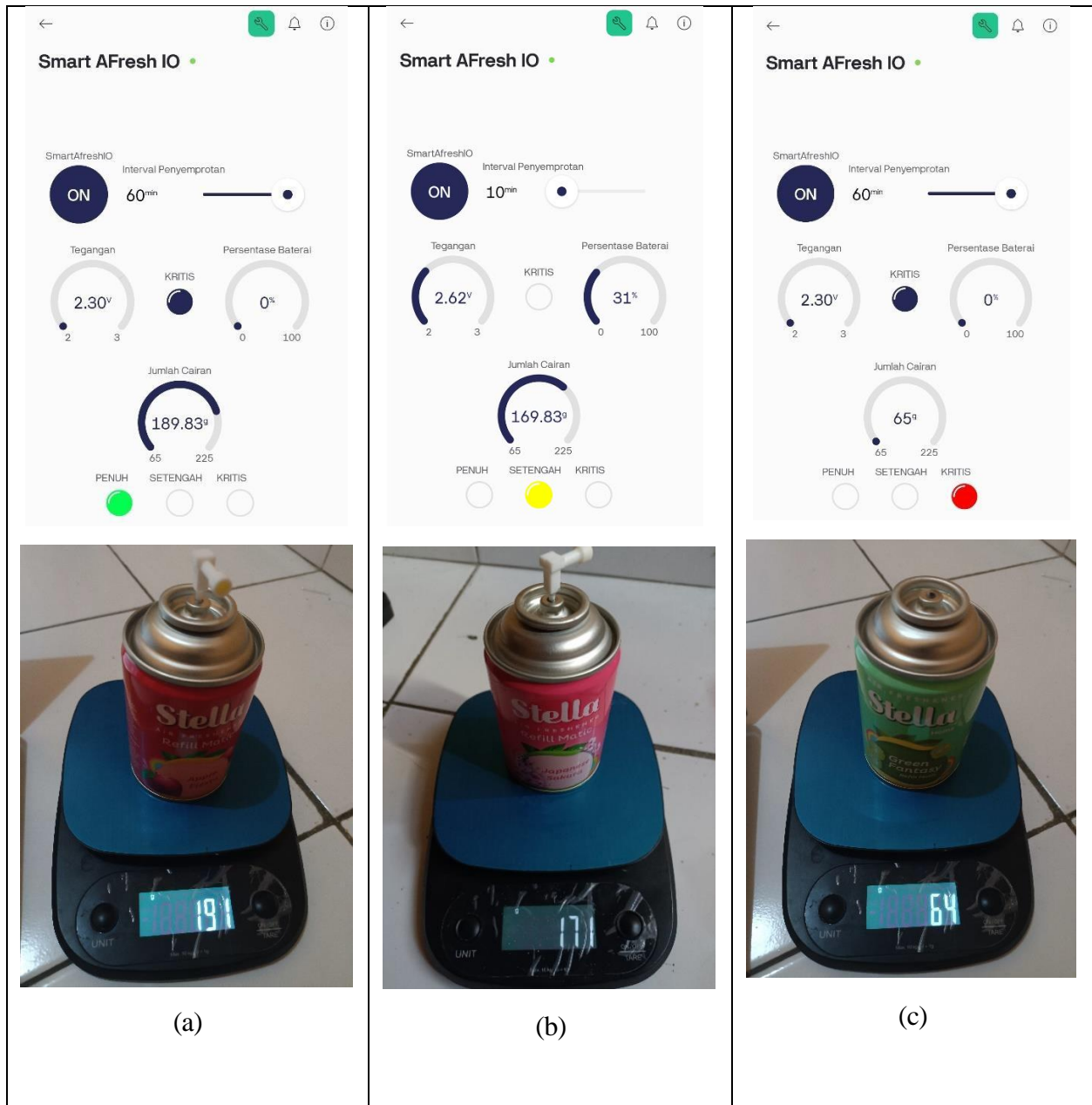
Dapat diperhatikan pada tabel 5.1 di atas, setelah dilakukan pengujian sebanyak 6 kali dengan variasi tegangan yang berbeda-beda, rata-rata persentase *error* yang diperoleh adalah 1.06%. Kemudian nilai rata-rata persentase *error* tersebut dijadikan angka pengurangan dari nilai 100, lalu didapati bahwa hasil akurasi fitur *monitoring* baterai mencapai 98.94%, sehingga dapat dikatakan bahwa akurasi fitur *monitoring* baterai ini cukup akurat karena memiliki tingkat akurasi yang cukup tinggi, dimana tegangan yang tertampil pada aplikasi Blynk untuk fitur *monitoring* baterai sudah cukup akurat dengan tegangan sebenarnya yang diukur menggunakan multimeter.

5.1.1.2 Hasil dan Analisis *Monitoring* Cairan Pengharum Ruangan

Pada pengujian fitur *monitoring* cairan pengharum ruangan pada alat, parameter yang dianalisis adalah berat kaleng cairan pengharum ruangan, di mana hasil pembacaan dari sensor *load cell* akan ditampilkan dalam satuan gram dan tiga kondisi yaitu penuh, setengah dan kritis. Adapun perhitungan persentase *error* dari pembacaan *load cell* menggunakan persamaan (1) dan (2) pada poin sebelumnya adalah sebagai berikut:

Tabel 5.2 Hasil Pengukuran Fitur *Monitoring* Cairan Pengharum Ruangan

No	Komponen yang Digunakan	Berat Aktual Kaleng (g)	Hasil Berat pada Blynk (g)	Status pada Blynk (Penuh, setengah, kritis)	Persentase <i>Error</i>	Akurasi
1	<i>Load cell 5 kg</i>	191 (Penuh)	190,7	PENUH	0.16%	94.28%
2			189,8	PENUH	0.63%	
3			187,8	PENUH	1.67%	
4			186,9	PENUH	2.15%	
5		171 (Setengah)	169,9	SETENGAH	0.64%	
6			169,8	SETENGAH	0.70%	
7			169,6	SETENGAH	0.82%	
8			169,5	SETENGAH	0.88%	
9		64 (Kosong)	54,1	KRITIS	15.47%	
10			54,2	KRITIS	15.31%	
11			54,3	KRITIS	15.16%	
12			54,4	KRITIS	15%	
Rata-rata persentase <i>error</i>					5.71%	

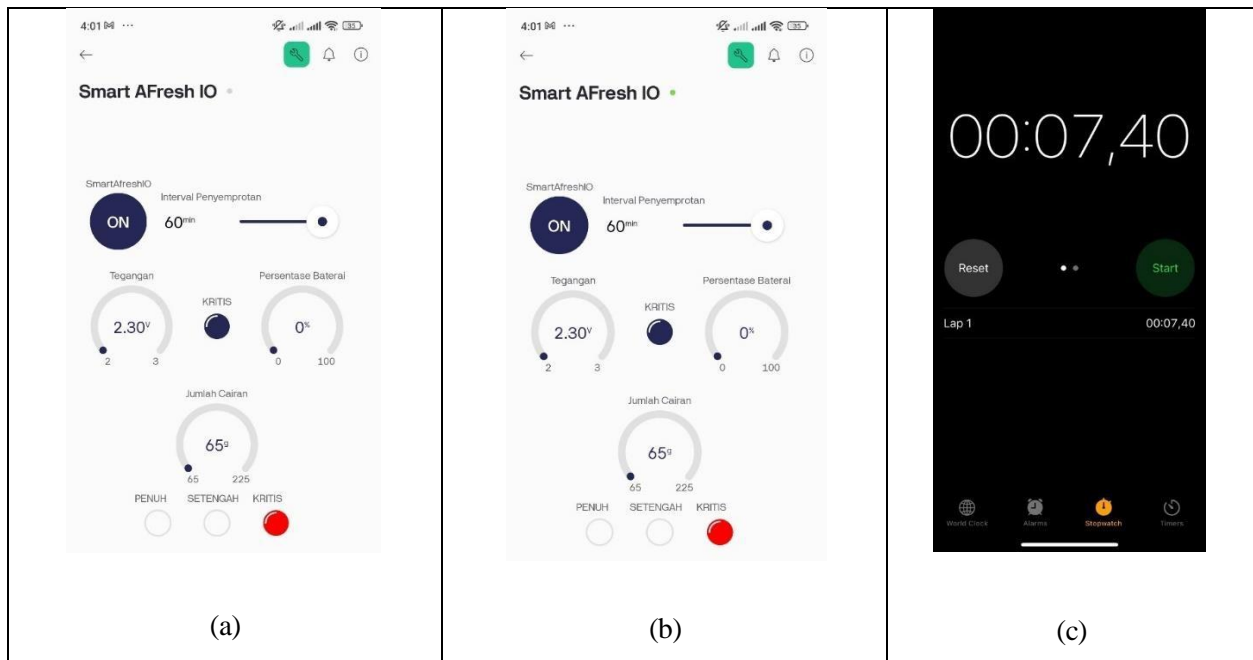


Gambar 5.2 Pengujian *error monitoring* cairan pengharum ruangan, (a) Pengujian *error monitoring* pada kaleng penuh (191 gram), (b) Pengujian *error monitoring* pada kaleng setengah (171 gram), (c) Pengujian *error monitoring* pada kaleng penuh (64 gram)

Dapat diperhatikan pada tabel 5.2 di atas, setelah dilakukan pengujian sebanyak 12 kali dengan variasi berat kaleng yang berbeda-beda yaitu menggunakan kaleng penuh, kaleng setengah dan kaleng kosong, dapat dianalisis bahwa sensor *load cell* yang digunakan terkadang memiliki sedikit perbedaan dalam pembacaan berat kaleng, ketidakakuratan tersebut paling signifikan ketika menggunakan kaleng kosong, sehingga *error* pembacaan yang diperoleh untuk kaleng tersebut cukup tinggi. Hal tersebut dapat disebabkan oleh penggunaan *load cell* 5 kg pada alat, ketidakakuratan tersebut mungkin dapat diatasi dengan lebih baik jika menggunakan *load cell* 1 kg. Adapun rata-rata persentase *error* yang diperoleh adalah 5.71%, kemudian nilai rata-rata persentase *error* tersebut dijadikan angka pengurangan dari nilai 100, lalu didapati bahwa hasil akurasi fitur *monitoring* kaleng cairan pengharum ruangan mencapai 94.28%, sehingga dapat dikatakan bahwa secara umum akurasi fitur *monitoring* kaleng cairan pengharum ruangan ini cukup akurat karena memiliki tingkat akurasi yang cukup tinggi, dimana berat yang tertampil pada aplikasi Blynk untuk fitur *monitoring* kaleng cairan pengharum ruangan sudah cukup akurat dengan berat sebenarnya yang diukur menggunakan timbangan untuk kondisi kaleng penuh dan setengah.

5.1.1.3 Hasil dan Analisis Delay

Pada pengujian ini parameter yang dianalisis adalah delay agar memastikan alat dapat berfungsi dengan akurat serta responsif sesuai dengan pengaturan waktu yang telah ditentukan. Pengujian pertama untuk parameter *delay* adalah berapa *delay* waktu antara saat alat dinyalakan menggunakan saklar sampai terhubung ke Blynk. Berdasarkan pengujian yang telah dilakukan dapat dianalisis bahwa *delay* waktu antara saat alat dinyalakan menggunakan saklar sampai terhubung ke Blynk (*Blynk online*) adalah selama 7.40 detik. Perhatikan gambar 5.3 di bawah, pada gambar (a) sistem belum terhubung ke Blynk, perhatikan indikator lampu kecil disamping nama SMART AFRESH IO, lalu pada (b) sistem sudah terhubung ke Blynk. Gambar (c) merupakan waktu yang dibutuhkan agar sistem terhubung ke Blynk, yaitu 7.40 detik.

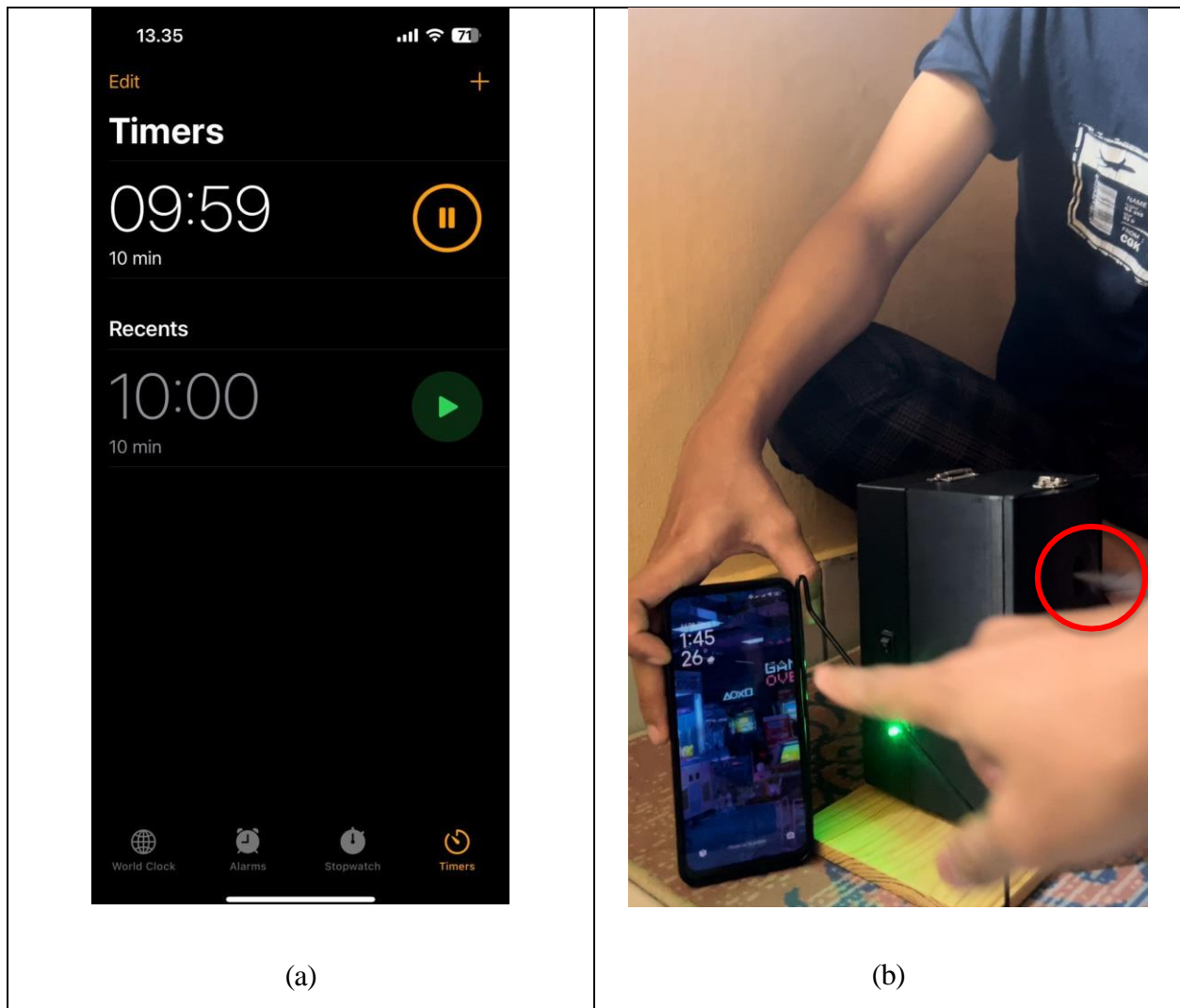


Gambar 5.3 Pengujian *delay* waktu antara saat alat dinyalakan menggunakan saklar dengan terhubung ke *Blynk*

Selanjutnya pada pengujian yang dilakukan untuk fitur pengaturan interval waktu penyemprotan, kaki memilih interval waktu penyemprotan 10, 20 dan 30 menit. Hasil dapat diperhatikan pada tabel 5.3 dan gambar 5.4 di bawah.

Tabel 5.3 Hasil Pengukuran *Delay* pada Fitur Pengaturan Interval Waktu Penyemprotan

No	Interval Waktu Penyemprotan pada Blynk (menit)	Waktu saat Pemilihan Interval	Waktu saat Menyemprot	Interval Waktu Penyemprotan Aktual (menit)
1	10	13:35	13:45	10
2	20	16:04	16:24	20
3	30	16:25	16:55	30



Gambar 5.4 Pengujian interval waktu penyemprotan melalui Blynk, (a) *Timer* 10 menit dimulai pada pukul 13:35, (b) Alat menyemprot pada pukul 13:45, tepat saat 10 menit telah berlalu (perhatikan lingkaran merah)

Dari tabel 5.3 dan gambar 5.4 di atas dapat dianalisis bahwa *timer* selama 10 menit mulai dinyalakan pada pukul 13:35 siang, lalu ketika 10 menit sudah berlalu yaitu pukul 13:45 alat menyemprot. Kemudian pada pukul 16:04 kami memilih opsi 20 menit, setelah 20 menit telah berlalu yaitu pada pukul 16:24 alat menyemprot. Lalu pada pukul 16:25 dipilih opsi 30 menit, setelah 30 menit telah berlalu yaitu saat waktu menunjukkan pukul 16:55 alat menyemprot. Sehingga didapati bahwa fitur pengaturan interval waktu penyemprotan sudah berfungsi dengan baik.

Kemudian, untuk fitur penjadwalan otomatis menggunakan NTP server, nyalakan alat dan biarkan alat menyemprot dengan interval waktu yang telah dipilih pada Blynk. Perhatikan waktu saat itu juga, apabila alat menyemprot di rentang jam yang telah ditentukan, yaitu antara

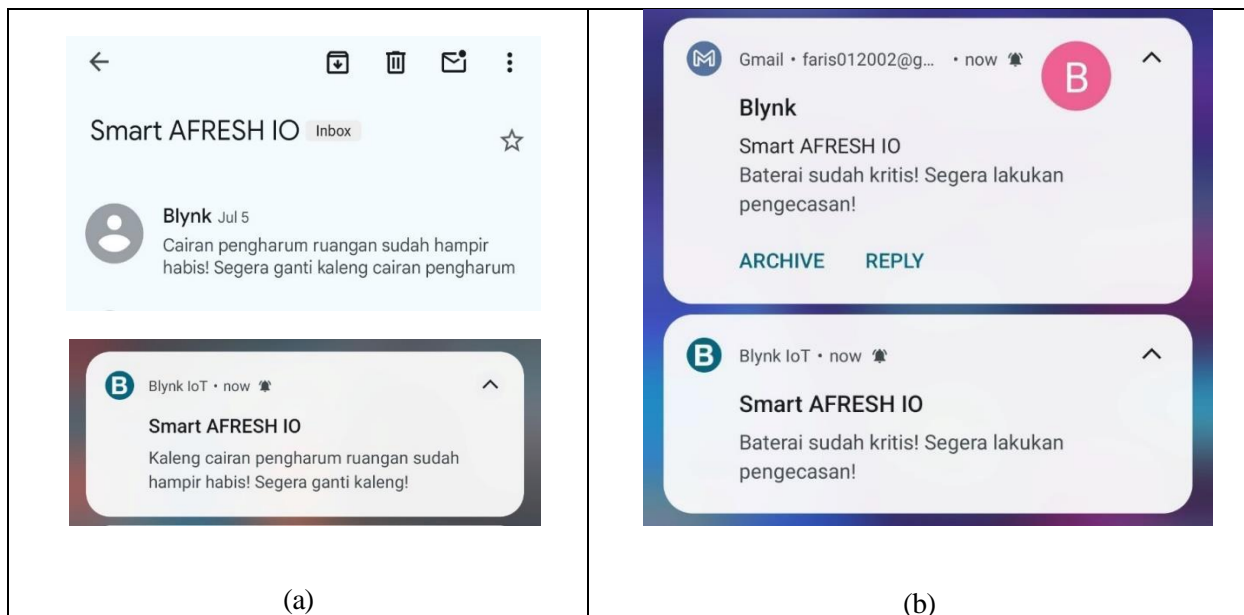
jam 8 pagi sampai jam 5 sore, berarti fitur penjadwalan otomatis sudah sesuai dengan yang direncanakan dan hanya bekerja pada saat jam kerja. Perhatikan tabel 5.4 dan di bawah ini:

Tabel 5.4 Hasil Pengukuran *Delay* pada Fitur Penjadwalan Otomatis

No	Interval Waktu Penyemprotan pada Blynk (menit)	Waktu saat Pemilihan Interval	Waktu saat Menyemprot	Mekanisme Penyemprotan
1	30	16:25	16:55	Menyemprot
2	10	16:56	17:06	Tidak Menyemprot
3	10	17:08	17:18	Tidak Menyemprot

Kemudian, Dari tabel 5.4 di atas dapat dianalisis bahwa ketika waktu sudah menunjukkan lebih dari jam 5 sore (17:00) alat sudah tidak akan menyemprot walaupun alat masih dinyalakan. Sehingga berdasarkan pengujian yang telah dilakukan tersebut fitur penjadwalan otomatis dapat dikatakan sudah berfungsi dengan baik.

Selanjutnya untuk fitur notifikasi, Blynk sudah dapat mengirim notifikasi ketika kondisi baterai dan kaleng cairan pengharum ruangan sudah kritis seperti pada gambar 5.5 di bawah, notifikasi tersebut diberi jeda waktu 5 menit, jika *user* tidak segera mengecas ataupun mengganti kaleng cairan pengharum ruangan ketika salah satu atau keduanya sudah mencapai kondisi kritis maka Blynk akan mengirimkan notifikasi setiap 5 menit kepada pengguna melalui *in app notification* atau email yang terhubung dengan *user* tersebut.



Gambar 5.5 Pengujian notifikasi melalui Blynk dan email, (a) Notifikasi ketika kondisi kaleng cairan pengharum ruangan kritis, (b) Notifikasi ketika kondisi baterai kritis

5.1.1.4 Hasil dan Analisis Daya Tahan Baterai SMART AFRESH IO

Kemudian dilakukan pengujian daya tahan baterai yang digunakan dalam alat SMART AFRESH IO. Langkah awal yang dilakukan adalah menyalakan SMART AFRESH IO dan hidupkan *stopwatch* serta ukur tegangan awal menggunakan multimeter secara bersamaan saat alat dinyalakan. Kemudian tunggu hingga alat mati, jika alat sudah mati, berarti BMS telah memutus tegangan dari baterai ke alat. Lalu perhatikan data tegangan yang tercatat pada Google Sheets dan hentikan *stopwatch* serta ukur tegangan akhir dari baterai setelah alat mati.

$$P \text{ (Watt)} = V \times i = 11.1 \times 0.3916 = 4.3468 \text{ Watt} \quad 3)$$

$$\text{Kapasitas baterai (Wh)} = 11.1 \times 3 = 33.3 \text{ Wh} \quad 4)$$

$$\text{Daya tahan baterai} = \frac{\text{Kapasitas baterai (Wh)}}{\text{Konsumsi daya (W)}} = \frac{33.3}{4.3468} = 7 \text{ jam } 40 \text{ menit} \quad 5)$$

Dari persamaan (3) di atas dapat diperkirakan bahwa konsumsi arus dari seluruh komponen yang kami gunakan pada alat ini berdasarkan *datasheet* serta pengukuran menggunakan multimeter dari komponen-komponen tersebut adalah sekitar 0.3916 A, kemudian jika nilai arus tersebut dikalikan dengan nilai tegangan sumber yang digunakan yaitu sekitar 11.1V, didapati bahwa Daya Beban (P) pada alat ini adalah sebesar 4.3468 Watt. Lalu dilakukan perhitungan daya tahan baterai menggunakan persamaan (5) seperti di atas, dimana parameter kapasitas baterai sebesar 33.3 Wh diperoleh dari perhitungan pada persamaan (4) sebelumnya, perhitungan pada persamaan (5) menghasilkan durasi daya tahan baterai selama sekitar 7 jam 40 menit. Akan tetapi pada aktualnya lebih sebentar dari perhitungan tersebut, perhatikan gambar 5.6 di bawah ini:

2,82	05/07/2024 18:39:18
2,30	05/07/2024 23:15:54

Gambar 5.6 Pengujian daya tahan baterai SMART AFRESH IO

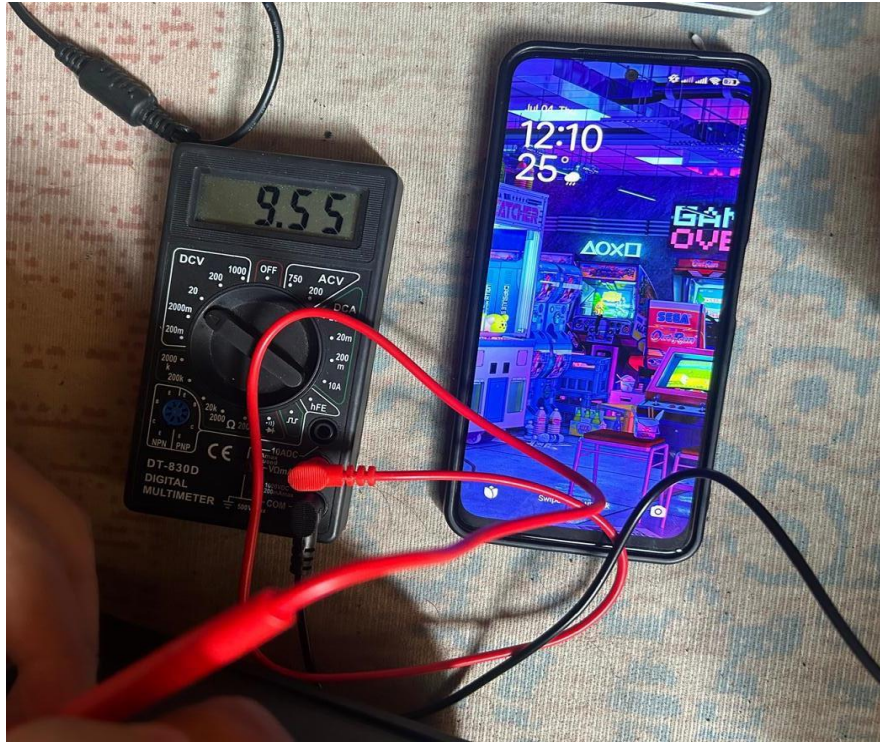
Dari gambar 5.6 di atas, pukul 18:39 merupakan saat alat baru dinyalakan dan kondisi baterai masih penuh yaitu 2.82V pada pembagi tegangan dan sekitar 11.1V pada baterai. Kemudian alat dibiarkan menyala hingga akhirnya pada pukul 23:15 BMS atau *Battery Management System* memutus tegangan dari baterai ke alat karena sudah mencapai tegangan *cut off* dari baterai tersebut yaitu 2.30V pada pembagi tegangan dan sekitar 9.55V pada baterai. Jika dihitung berdasarkan *datasheets* dan *stopwatch*, waktu yang dibutuhkan oleh baterai dari kondisi penuh sampai habis karena di *cut off* oleh BMS adalah sekitar 4 jam 35 menit. Sehingga dapat dikatakan

bahwa daya tahan baterai yang digunakan dalam SMART AFRESH IO ini adalah 4 jam 35 menit, lebih sebentar dibandingkan hasil perhitungan (7 jam 40 menit) dan spesifikasi yang diusulkan (1 bulan). Durasi daya tahan aktual baterai yang jauh lebih sebentar dari yang diperhitungkan dan spesifikasi yang diusulkan ini dapat disebabkan oleh beberapa hal, seperti konsumsi arus keseluruhan sistem yang lebih tinggi dari yang diperkirakan saat sebelum mulai melakukan perancangan *hardware* karena BMS dan beberapa komponen lain yang digunakan menimbulkan inefisiensi, sehingga konsumsi daya bebannya juga menjadi lebih tinggi yang menyebabkan daya tahan aktual baterai lebih sebentar dari perhitungan, serta kapasitas baterai tidak sebanyak yang direncanakan.

Untuk mengatasi keterbatasan baterai pada alat yang habis dalam waktu sekitar 4 jam 35 menit tersebut, penulis menyarankan untuk menambah kapasitas dari baterai yang digunakan secara signifikan dengan cara menghubungkan baterai secara paralel, akan tetapi hal ini juga akan memakan ruang lebih banyak lagi pada box komponen dari alat ini. Untuk mengatasi keterbatasan baterai pada alat yang habis dalam waktu sekitar 4 jam 35 menit tersebut, penulis menyarankan untuk menambah kapasitas dari baterai yang digunakan secara signifikan dengan cara menghubungkan baterai secara paralel, sehingga konfigurasi baterai li-ion yang digunakan menjadi seri-paralel, misalkan menjadi 3s3p (3 seri 3 paralel), dengan begitu tegangan maksimum baterai akan tetap berada di *range* 11.1V tetapi kapasitas baterai akan bertambah menjadi 9000 mAh. Dengan begitu baterai pada alat ini akan memiliki kapasitas baterai 3 kali lebih besar sehingga daya tahan baterai juga menjadi lebih lama. Akan tetapi hal ini juga akan memakan ruang lebih banyak lagi pada box komponen dari alat ini dan tentunya menambah biaya produksi alat ini karena komponen baterai yang digunakan semakin banyak.

5.1.1.5 Hasil dan Analisis *Charging* alat SMART AFRESH IO

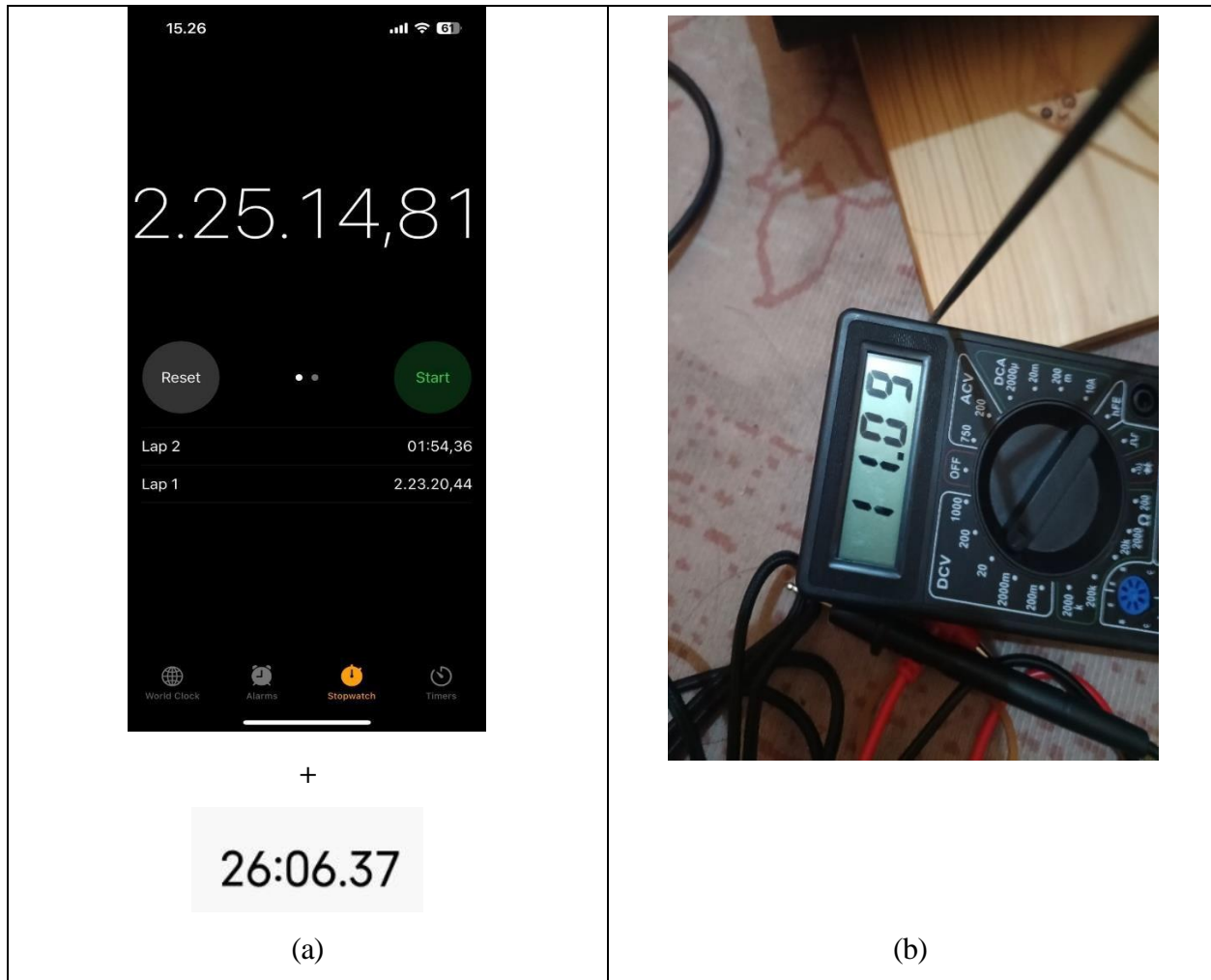
Selanjutnya untuk pengujian lama *charging* baterai yang digunakan dalam alat SMART AFRESH IO ini, langkah pertama adalah menghabiskan baterai alat terlebih dahulu. Seperti yang telah dibahas sebelumnya bahwa pada alat kami, baterai akan *cut off* pada sekitar 9.55V (sekitar 2.30V pada pembagi tegangan), dapat diperhatikan pada gambar 5.7 di bawah.



Gambar 5.7 Tegangan *cut off* bawah dari BMS pada baterai

Kemudian, colokkan adaptor 12V 3A ke *jack female* di bagian samping kiri bawah alat dan nyalakan *stopwatch*. Dari pengujian yang telah dilakukan, dapat dianalisis bahwa waktu yang diperlukan agar baterai terisi penuh lagi yaitu saat sekitar 11.1V adalah selama 2 jam 50 menit, dapat perhatikan gambar 5.8 (a) dan (b) di bawah. Kemudian dilakukan perhitungan waktu *charging* baterai berdasarkan spesifikasi baterai Li-ion 3.7 V 3000 mAh dengan spesifikasi *charger* 3A. Adapun perhitungannya adalah sebagai berikut:

$$\text{Durasi charging} = \frac{3000 \text{ mAh}}{3 \text{ A}} = 1 \text{ jam} = 60 \text{ menit}$$



Gambar 5.8 Pengujian lama *charging* baterai yang digunakan dalam alat SMART AFRESH IO (a) Waktu yang dibutuhkan agar baterai penuh kembali (2 jam 25 menit + 26 menit = 2 jam 50 menit), (b) Tegangan baterai ketika baterai sudah penuh

Berdasarkan perhitungan di atas, didapati bahwa durasi *charging* adalah selama 1 jam atau 60 menit, sedangkan pada pengujian *charging* baterai yang dilakukan didapatkan bahwa proses *charging* memakan waktu sekitar 2 jam 50 menit. Ketidakesesuaian antara durasi *charging* hasil perhitungan dengan aktualnya yang menyebabkan proses *charging* aktual menjadi lebih lama tersebut dapat dipengaruhi oleh beberapa faktor seperti efisiensi *charging* baterai yang kurang tinggi, arus yang tidak konstan saat proses *charging* aktualnya dan efisiensi rendah dari adaptor yang digunakan tersebut.

5.1.2 Perbandingan Performa

Tabel 5.5 Perbandingan performa antara sistem yang dibuat dengan sistem lain

No	Fitur/Komponen	Sistem yang dibuat	Stella Automatic Spray
1	Catu daya yang digunakan	Tiga buah baterai Li-ion 18650 yang disusun secara seri	Dua buah baterai AAA
2	Tombol ON/OFF	Memiliki tombol ON/OFF secara fisik pada bagian samping alat dan secara virtual pada Blynk	Memiliki tombol OFF secara fisik didalam alat pada pengaturan analog interval waktu penyemprotan
3	Pengaturan interval waktu penyemprotan	Dapat diatur secara virtual melalui Blynk dan mencakup 6 opsi yaitu tiap 10, 20, 30, 40, 50 dan 60 menit	Harus membuka alat untuk mengatur interval waktu penyemprotan pada tombol fisik di dalam alat, terdapa 4 opsi yaitu OFF, 10, 20 dan 40 menit
4	Fitur	<ul style="list-style-type: none"> • Indikator baterai dan isi kaleng cairan pengharum ruangan secara fisik maupun virtual <ul style="list-style-type: none"> • <i>Monitoring</i> baterai • <i>Monitoring</i> isi kaleng cairan pengharum ruangan • Dapat mematikan mekanisme penyemprotan dengan manual secara virtual melalui Blynk <ul style="list-style-type: none"> • Pengaturan interval waktu penyemprotan dilakukan secara virtual melalui Blynk • Penjadwalan otomatis untuk menyalakan dan mematikan mekanisme penyemprotan pada jam tertentu 	<ul style="list-style-type: none"> • LED hijau yang menandakan bahwa alat sudah menyala dan user sudah memilih interval waktu penyemprotan tertentu
5	Daya tahan baterai	4 jam 35 menit	1 bulan

6	Waktu pengisian baterai	2 jam 50 menit	Baterai diganti dengan yang baru
7	Berat alat (tanpa kaleng)	717 gram	225 gram
8	Harga	Rp 700.000,00	Rp 75.000,00

5.1.3 Pemenuhan Spesifikasi Sistem

Tabel 5.6 Perbandingan usulan dan hasil perancangan sistem

No	Spesifikasi	Usulan	Realisasi
1	Dimensi (panjang x lebar x tinggi)	8 x 2.5 x 19.5 cm	10 x 14.5 x 20.5 cm
2	Berat (gram)	Maksimal 2 kg	717 gram
3	Fitur <i>monitoring</i> baterai	Menampilkan tegangan dan persentase	Menampilkan tegangan, persentase dan indikator ketika sudah kritis
4	Fitur <i>monitoring</i> cairan pengharum ruangan	Menampilkan berat kaleng dalam gram atau ml dan persentasenya	Menampilkan berat kaleng dalam gram beserta status kondisi saat kaleng masih penuh, setengah atau sudah kritis
5	Fitur penjadwalan otomatis	Memiliki pengaturan waktu pada UI di Blynk	Penjadwalan otomatis sudah ditetapkan dari <i>coding</i> yaitu dari jam 8 sampai jam 5 sore
6	Daya tahan baterai	1 bulan	4 jam 35 menit
7	Mekanisme tutup cover dan box komponen pada alat	Buka tutup biasa	Sistem penguncian
8	Lama <i>charging</i> baterai alat	1 jam 30 menit	2 jam 50 menit

Pada spesifikasi dimensi, terdapat ketidaksesuaian antara usulan dengan realisasi karena terdapat perubahan komponen-komponen yang digunakan pada alat, sehingga saat dilakukan perangkaian awal dan *fitting* komponen-komponen tersebut pada desain yang diusulkan, didapati bahwa ukuran alat perlu diperbesar. Pada spesifikasi penjadwalan otomatis, kami memutuskan untuk menetapkan jam kerja dari penjadwalan otomatis pada *coding* yaitu dari jam 8 sampai jam 5 sore karena keterbatasan waktu untuk membuat fitur pengaturan jam kerja oleh user tersebut dan lebih fokus ke fitur-fitur utama dari alat ini seperti *monitoring* baterai, *monitoring* cairan pengharum ruangan dan pengaturan interval waktu penyemprotan. Untuk spesifikasi daya tahan baterai terjadi ketidaksesuaian antara daya tahan baterai usulan dengan realisasinya karena kami belum mulai merancang alat kami, sehingga kami belum mengetahui berapa besar daya beban yang digunakan oleh sensor dan komponen-komponen kami ketika alat ini dinyalakan secara terus menerus. Selanjutnya didapati bahwa durasi *charging* baterai yang lebih lama dari yang diusulkan karena pada saat perancangan kami memutuskan untuk menggunakan baterai li-ion dengan kapasitas yang lebih besar yaitu 3000 mAh, sehingga waktu pengecasan lebih lama.

5.1.4 Pengalaman Pengguna

Tabel 5.7 Pengalaman Pengguna

No	Fitur/Komponen	Capaian	Aksi/Perbaikan
1	Fungsi	Fungsi <i>monitoring</i> tegangan baterai serta persentasenya yang ditampilkan pada Blynk sudah berjalan dengan baik	Dipertahankan
2	Fungsi	Fungsi <i>monitoring</i> kaleng cairan pengharum ruangan sebagian besar akurat tetapi terkadang memiliki persentase <i>error</i> pembacaan yang tinggi	Buat kalibrasi <i>load cell</i> yang lebih akurat dan presisi terhadap berbagai berat kaleng yang berbeda (penuh, setengah dan kosong)
3	Fungsi	Indikator visual menggunakan LED untuk fitur <i>monitoring</i> baterai perlu waktu beberapa menit sebelum nyala karena <i>coding</i> pengukuran tegangan yang perlu melalui proses rata-rata terlebih dahulu	Buat kalibrasi tegangan yang lebih akurat dan tidak membutuhkan proses rata-rata tegangan

4	Fungsi	Interval waktu penyemprotan yang cukup bervariasi dan dapat diatur secara virtual	Dipertahankan
5	Fungsi	Penjadwalan otomatis sudah ditetapkan dari jam 8 pagi sampai jam 5 sore	Buat penjadwalan otomatis yang dapat diatur oleh <i>user</i> melalui UI Blynk
6	Kemudahan	Kaleng cairan pengharum ruangan tidak bisa langsung diletakkan pada <i>load cell</i> ketika alat baru dinyalakan karena <i>load cell</i> butuh waktu untuk kalibrasi	Buat agar <i>load cell</i> bisa langsung menampilkan berat kaleng yang diletakkan pada <i>load cell</i> saat kaleng sudah berada di dalam alat ketika alat dinyalakan
7	Kemudahan	Pengguna dapat dengan mudah menyimpan dan mengolah data <i>monitoring</i> baterai dan <i>monitoring</i> kaleng cairan pengharum ruangan pada <i>googlesheet</i>	Dipertahankan
8	Keamanan	Sistem penguncian pada <i>cover</i> dan bagian box komponen alat yang cukup Kuat	Dipertahankan

5.1.5 Kesesuaian Perencanaan dalam Manajemen Tim dan Realisasinya

Tabel 5.8 Kesesuaian antara usulan dan realisasi *timeline* pengerjaan Tugas Akhir 2

No	Kegiatan	Usulan waktu	Realisasi Pelaksanaan
1	Pembelian alat dan bahan	Desember - Februari	Januari - Februari
2	Perancangan sistem dengan usulan	Februari - April	Februari - Juni
3	Testing dan Validasi	April - Mei	Juni - Juli
4	Expo dan pengumpulan laporan akhir	Juni	Juli

Tabel 5.9 Kesesuaian RAB Tugas Akhir antara usulan dan realisasi

No	Jenis Pengeluaran	Usulan Biaya		Realisasi Biaya	
		Kuantitas	Total Harga	Kuantitas	Total Harga
1	ESP32	1 pc	Rp 70.000,00	1 pcs	Rp 70.000,00
2	Baterai Li-ion LG 18650 3000 mAh	3 pcs	Rp 24.000,00	3 pcs	Rp 153.000,00
3	Saklar ON-OFF AC	-	-	1 pc	Rp 2.000,00
4	BMS 3S 10A	1 pc	Rp 17.000,00	1 pc	Rp 17.000,00
5	Modul MP1584EN DC-DC <i>step down</i> 5V	1 pc	Rp 7.000,00	1 pc	Rp 7.000,00
6	<i>Relay 5V single channel</i>	1 pc	Rp 8.000,00	1 pc	Rp 8.000,00
7	Sensor HX711	1 pc	Rp 9.000,00	1 pc	Rp 9.000,00
8	Sensor <i>load cell</i> 5kg	1 pc	Rp 25.000,00	1 pc	Rp 25.000,00
9	<i>Mounting bracket holder load cell</i>	1 pc	Rp 25.000,00	1 pc	Rp 25.000,00
10	Resistor ¼ watt metal film 1K Ohm 1%	4 pcs	Rp 400,00	4 pcs	Rp 400,00
11	Resistor ¼ watt metal film 100 Ohm 1%	2 pcs	Rp 200,00	2 pcs	Rp 200,00
12	Modul LED RGB KY-016 (3 <i>Color Full Color LED</i>)	-	-	2 pcs	Rp 7000,00
13	Socket kabel 4 pin	-	-	2 pcs	Rp 1000,00
14	Jack DC <i>Female</i>	-	-	1 pc	Rp 2500,00
15	Jack DC <i>Male</i>	-	-	1 pc	Rp 2500,00
16	PCB Bolong 9x7 cm	1 pc	Rp 11.000,00	1 pc	Rp 11.000,00

No	Jenis Pengeluaran	Usulan Biaya		Realisasi Biaya	
		Kuantitas	Total Harga	Kuantitas	Total Harga
17	<i>Spacer</i> kuningan M3 10mm+6mm	-	-	10 pcs	Rp 13.000,00
18	Cetak <i>casing</i> (3D)	3 pcs	Rp 250.000,00	3 pcs	Rp 250.000,00
19	<i>Stella automatic</i> <i>spray refill</i>	1 pc	Rp 75.000,00	2 pcs	Rp 150.000,00
20	Kaleng <i>stella</i> <i>automatic spray</i>	-	-	1 pc	Rp 45.000,00
Total Harga					Rp 798.600,00

Untuk menekan biaya produksi, beberapa saran dari penulis adalah dengan menggunakan baterai li-ion 18650 non-*brand* yang lebih murah, contohnya yang tersedia di pasaran adalah baterai li-ion yang memiliki *range* harga sekitar Rp 9.000 hingga Rp 25.000. Dari segi kelebihan biaya yang dikeluarkan akan menjadi lebih murah, tetapi kekurangannya adalah baterai non-*brand* kualitasnya tidak sebgus baterai *brand* yang lebih mahal tersebut, sehingga dikhawatirkan performa baterainya menurun seperti kapasitas aktualnya tidak sesuai dengan yang tertera pada spesifikasi, degradasi yang lebih cepat dan umur baterai yang lebih pendek. Kemudian untuk pencetakan 3D *casing* dapat dibuat lebih murah lagi dengan membuat desain 3D yang lebih *compact* atau kecil sehingga tidak menambah berat pencetakan 3D *casing*, yang mana umumnya harga 3D *printing* ditetapkan berdasarkan berat per gram-nya, lalu mencari jasa desain 3D *printing* yang menawarkan harga lebih ramah di kantong, atau menggunakan material yang lebih murah seperti PLA+. Pada tugas akhir ini, penulis menggunakan bahan ABS dengan kualitas pencetakan halus dari 4 opsi pencetakan yang diberikan oleh jasa 3D *printing* yang digunakan (kualitas standar, halus, lebih halus, dan paling halus) dengan berat total dari desain sebesar 417 gram, adapun harga dari pencetakan halus tersebut adalah sekitar Rp 600/Gram, sehingga biaya yang dikeluarkan penulis untuk mencetak *casing* alat ini adalah Rp 250.000 (417 gram x Rp 600,00). Tetapi jika menggunakan bahan cetak PLA+ dengan kualitas standar, yang mana kualitas standar memiliki harga pencetakan Rp 400/Gram, biaya yang perlu dikeluarkan untuk pencetakan *casing* akan menjadi sekitar Rp 166.800 (417 x Rp 400,00), lebih murah Rp

83.200. Perlu diketahui bahwa walau biaya pencetakan akan menjadi lebih murah, tetapi kualitas *casing* yang dicetak tidak akan sehalus dan serapi pencetakan dengan kualitas halus. Kemudian penggunaan bahan PLA+ juga akan membuat *casing* menjadi lebih rapuh dan mudah retak, ini menimbulkan resiko baru jika terdapat beberapa part *casing* yang tipis.

Tabel 5.10 Realisasi aktivitas pelaksanaan tugas akhir 2

No	Hari, Tanggal, Durasi (jam atau hari)	Aktivitas	Pelaksana
1	Rabu, 10 Januari 2024	Pembelian komponen untuk pembuatan alat	Wildan, Faris
2	Senin, 4 Maret 2024 sampai Selasa, 23 April 2024	Pengecekan dan uji coba pada komponen yang akan digunakan, serta melakukan perakitan awal	Wildan, Faris
3	Jumat, 26 April 2024 sampai Selasa, 14 Mei 2024	Membuat konfigurasi pada Blynk dan membuat desain elektronis	Faris
4	Rabu, 15 Mei 2024	Membuat sketsa desain 3D dari alat	Wildan
5	Kamis, 16 Mei 2024 sampai Selasa 21 Mei 2024	Membuat desain 3D pada SOLIDWORK	Wildan, Faris
6	Rabu, 22 Mei 2024 sampai Sabtu, 25 Mei 2024	Mencetak desain 3D alat secara keseluruhan	Wildan, Faris
7	Senin, 27 Mei 2024 sampai Jumat, 21 Juni 2024	Melakukan perangkaian alat	Wildan, Faris
8	Sabtu, 22 Juni 2024 sampai Jumat, 28 Juni 2024	Melakukan uji coba dan pengambilan data	Wildan, Faris

9	Senin, 1 Juli 2024 sampai Jumat, 19 Juli 2024	Menyelesaikan laporan TA2	Wildan, Faris
---	--	---------------------------	---------------

5.2 Dampak Implementasi Sistem

1. Teknologi

Dalam bidang teknologi, pengintegrasian dispenser pengharum ruangan yang ada dipasaran dengan IoT ini dapat mendorong inovasi khususnya dalam aspek elektronika dan IoT, serta dapat meningkatkan efisiensi dalam manajemen atau *maintenance* perangkat *indoor* (rumah tangga, ruang kelas, kantor, dsb) dengan menciptakan perangkat cerdas yang lebih mudah diakses. Dengan SMART AFRESH IO, pengguna dapat melakukan *monitoring* baterai dan cairan pengharum ruangan, mengatur interval waktu penyemprotan secara virtual dan mekanisme penyemprotan alat hanya bekerja pada rentang jam kerja tertentu.

2. Sosial

Dalam bidang sosial, alat ini dapat meningkatkan kualitas hidup dengan memberikan bau harum pada suatu ruangan, menciptakan suasana lingkungan atau ruangan yang menyenangkan dan menyegarkan. Fitur-fitur yang disediakan oleh alat ini juga dapat mengurangi pekerjaan manual yang harus dilakukan pada dispenser pengharum ruangan dipasaran, yang mana belum terintegrasi dengan IoT.

3. Ekonomi

Dalam bidang ekonomi, berhubungan dengan bidang sosial sebelumnya, pengintegrasian dispenser pengharum ruangan dengan IoT ini dapat mengurangi biaya operasional karena petugas yang bertanggung jawab untuk melakukan *maintenance* pada alat akan dapat melakukan tugasnya dengan lebih efisien. Lalu, jika alat ini dikembangkan lebih lanjut lagi maka dapat berpeluang untuk membuka bisnis baru dalam bidang perangkat cerdas.

4. Lingkungan

Dalam bidang lingkungan, penggunaan baterai AAA sekali pakai pada dispenser pengharum ruangan yang ada dipasaran memiliki dampak berbahaya kepada lingkungan jika tidak dibuang dengan benar, sedangkan penggunaan baterai li-ion *rechargeable* pada SMART AFRESH IO lebih ramah lingkungan dalam jangka panjang karena dapat diisi ulang dan tidak berkontribusi kepada jumlah limbah baterai.

BAB 6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Sistem bernama SMART AFRESH IO ini merupakan dispenser pengharum ruangan yang sudah terintegrasi dengan *internet of things* (IoT), dibuat untuk meningkatkan efisiensi dari dispenser pengharum ruangan otomatis yang ada dipasaran, alat ini dibekali dengan berbagai macam fitur untuk membantu dan memudahkan pengguna dalam *monitoring* dan maintenance dispenser pengharum ruangan secara jarak jauh. Alat yang berbasis IoT ini mampu memberikan berbagai macam fitur yang sangat berguna bagi pengguna seperti *monitoring* baterai, *monitoring* isi kaleng cairan pengharum ruangan, pengaturan interval waktu penyemprotan secara virtual dengan 6 opsi yang berbeda, penjadwalan otomatis pada jam kerja yaitu dari jam 8 pagi hingga 5 sore dan notifikasi ketika kondisi baterai dan jumlah cairan pengharum ruangan sudah mendekati habis.

Berdasarkan hasil perancangan alat, dapat disimpulkan bahwa beberapa spesifikasi yang terealisasi pada alat ini sudah sesuai dengan yang sebelumnya diusulkan, dimana dari segi fisik alat memiliki berat kurang dari 1 kg, menggunakan catu daya baterai *rechargeable* dan perlu terhubung ke jaringan internet (wifi) agar pengguna dapat melakukan *monitoring* baterai, *monitoring* isi kaleng cairan pengharum ruangan, pengaturan interval waktu penyemprotan, penjadwalan otomatis, mendapatkan notifikasi ketika kondisi baterai dan jumlah cairan pengharum ruangan sudah mendekati habis pada aplikasi Blynk dan *datalogging* pada *google sheets* agar pengguna dapat menyimpan dan mengolah data-data dari alat ini dengan mudah. Perbedaan spesifikasi dari yang diusulkan dengan realisasinya seperti dimensi alat, fitur penjadwalan otomatis, daya tahan baterai maupun durasi *charging* alat, tidak menghambat tercapainya tujuan utama pembuatan alat ini, yaitu *monitoring* dispenser pengharum ruangan otomatis dari jarak jauh. Dengan begitu, tujuan dari projek ini sudah terpenuhi karena alat dapat membantu dan memudahkan pengguna dalam *monitoring* dan maintenance dispenser pengharum ruangan otomatis secara jarak jauh sesuai yang direncanakan dengan cukup baik. Sehingga berdasarkan hasil perancangan alat SMART AFRESH IO yang telah dilakukan ini dapat disimpulkan bahwa sebagian besar alat sudah sesuai dengan spesifikasi yang telah diusulkan, serta sudah memenuhi tujuan dari projek ini.

6.2 Saran

Berdasarkan hasil perancangan produk SMART AFRESH IO, alat ini masih memiliki beberapa kekurangan yang diharapkan dapat dijadikan bahan evaluasi atau perbaikan untuk kedepannya, berikut ini merupakan beberapa saran untuk pengembangan SMART AFRESH IO selanjutnya:

1. Menggunakan sensor *load cell* yang lebih bagus dan akurat, disarankan *load cell* dengan spesifikasi maksimal berat 1 kg, serta membuat kalibrasi *load cell* yang lebih akurat dan presisi terhadap berbagai berat kaleng yang berbeda (penuh, setengah dan kosong)
2. Membuat kalibrasi tegangan yang lebih akurat dan tidak membutuhkan proses rata-rata tegangan
3. Buat agar *load cell* bisa langsung menampilkan berat kaleng yang diletakkan pada *load cell* saat kaleng sudah berada di dalam alat ketika alat dinyalakan
4. Membuat fitur penjadwalan otomatis yang dapat diatur oleh *user* menggunakan semacam *slider* melalui UI Blynk
5. *Casing* alat yang lebih kecil dan tidak terlalu besar, lalu mencari jasa desain 3D *printing* yang menawarkan harga lebih ramah di kantong, atau menggunakan material yang lebih murah
6. Menambah kapasitas dari baterai yang digunakan secara signifikan dengan cara menghubungkan baterai secara paralel, sehingga konfigurasi baterai menjadi seri paralel

DAFTAR PUSTAKA

- [1] “Air Freshener, Your Front-line warrior for Bad Odours.” Accessed: Sept. 23, 2023. [Online]. Available: <https://www.refreshyourlife.in/blog/how-air-fresheners-are-useful-for-various-spaces>
- [2] “Ini Lho Manfaat Pakai Pengharum Ruangan Otomatis.” Accessed: Sept. 23, 2023. [Online]. Available: <https://www.ecocare.id/post/ini-lho-manfaat-pakai-pengharum-ruangan-otomatis>
- [3] G. Tendra and Muhammad, “SISTEM KONTROL ALAT PENGHARUM RUANGAN OTOMATIS BERBASIS MIKROKONTROLER BERDASARKAN TINGKAT KEPADATAN JUMLAH PENGGUNA RUANGAN,” 2020.
- [4] F. Mira Trisna, A. Sutrisman, M. Miftakul Amin, J. Teknik Komputer Politeknik Negeri Sriwijaya, and J. Sriwijaya Negara Bukit Besar, “Rancang Bangun Pengharum Ruangan Otomatis Menggunakan RTC (Real Time O’Clok) Berbasis Arduino UNO,” *IJCCS*, vol. 13, no. 1, pp. 87–94, 2019.
- [5] D. Nugroho, L. Agustine, and A. Gunandhi, “PERANCANGAN DAN PEMBUATAN ALAT PENGATUR PERIODA PENYEMPROTAN PADA ROOM DEODORIZER DISPENSERBERBASIS MIKROKONTROLER,” *WIDYA TEKNIK*, vol. 8, no. 1, pp. 55–64, 2009.
- [6] A. Nadia, R. Rasyid, and Harmadi, “Sistem *Monitoring* Ketinggian Cairan Infus Berbasis Sensor Serat Optik Evanescent,” *Jurnal Fisika Unand*, vol. 8, no. 4, 2019.
- [7] N. Afiyat, R. Helmi Navilla, and M. Hariyadi, “IoT-Based Infusion Fluid *Monitoring* System Using the MQTT Protocol,” *JURNAL NASIONAL TEKNIK ELEKTRO DAN TEKNOLOGI INFORMASI*, vol. 12, no. 1, 2022.
- [8] tbidjakartadua, “Ini Lho Manfaat Pakai Pengharum Ruangan Otomatis.” Accessed: Oct. 15, 2023. [Online]. Available: <https://www.ecocare.id/post/ini-lho-manfaat-pakai-pengharum-ruangan-otomatis>
- [9] A. Gillis, “What is IoT (Internet of Things) and How Does it Work? | Definition from TechTarget.” Accessed: Oct. 15, 2023. [Online]. Available: <https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT>

- [10] K. Y. Maulana, “Apa Itu ESP32, Salah Satu Modul Wi-Fi Poppuler - Krysna Yudha Maulana - anakteknik.co.id.” Accessed: Oct. 15, 2023. [Online]. Available: <https://www.anakteknik.co.id/krysnayudhamaulana/articles/apa-itu-esp32-salah-satu-modul-wi-fi-poppuler>
- [11] futek.com, “What is a Load Cell Sensor.” Accessed: Oct. 15, 2023. [Online]. Available: <https://www.futek.com/what-is-a-load-cell-sensor>
- [12] Warstek Media, “Apa Itu Real Time Clock (RTC), Fungsinya, dan Contoh Penerapannya - Warung Sains Teknologi.” Accessed: Oct. 15, 2023. [Online]. Available: <https://warstek.com/apa-itu-real-time-clock-rtc-fungsinya-dan-contoh-penerapannya/>
- [13] Irpah, “Modul lm2596 : Pengertian, Skema Rangkaian & Datasheet.” Accessed: Nov. 25, 2023. [Online]. Available: <https://praktekotodidak.com/modul-stepdown-lm2596-skema-diagram-datasheet-vout-dan-penerapan/>
- [14] A. A. Putu, I. W. Sukadana, I. W. Utama, and I. W. Sugarayasa, “Datalogger Portabel Online Untuk Remote *Monitoring* Menggunakan Arduino Mikrokontroler,” *Prosiding Seminar Nasional Teknik Elektro (FORTEI 2017)*, 2017, Accessed: Nov. 25, 2023. [Online]. Available: www.wampserver.com.
- [15] UL Research Institutes, “What Are Lithium-Ion Batteries? | UL Research Institutes.” Accessed: Nov. 25, 2023. [Online]. Available: <https://ul.org/research/electrochemical-safety/getting-started-electrochemical-safety/what-are-lithium-ion>
- [16] power-sonic, “How To Connect Batteries In Series & Parallel - Power Sonic.” Accessed: Nov. 25, 2023. [Online]. Available: <https://www.power-sonic.com/blog/how-to-connect-batteries-in-series-and-parallel/>
- [17] A. Razor, “Modul Relay Arduino: Pengertian, Gambar, Skema, dan Lainnya” Accessed: July. 2, 2024. [Online]. Available: <https://www.aldyrazor.com/2020/05/modul-relay-arduino.html>
- [18] Supri, “Fungsi dan Cara Kerja BMS Baterai 3s, 4s dan Lain-Lain” Accessed: July. 2, 2024. [Online]. Available: <https://www.spiderbeat.com/fungsi-dan-cara-kerja-bms/>
- [19] R. Muh, “Program RGB LED” Accessed: July. 2, 2024. [Online]. Available: <https://mikrokontroler.mipa.ugm.ac.id/2018/10/02/program-led-rgb/>

[20] CNCStore, “Review Beberapa Jenis Module Step Down Terkecil dan Paling Direkomendasikan” Accessed: July. 2, 2024. [Online]. Available: <https://cncstorebandunggo.blogspot.com/2019/01/review-beberapa-jenis-module-step-down.html>

[21] Osoyoo, “What is Blynk and How does it work?” Accessed: July. 2, 2024. [Online]. Available: <https://osoyoo.com/2021/10/24/what-is-blynk-and-how-does-it-work/>

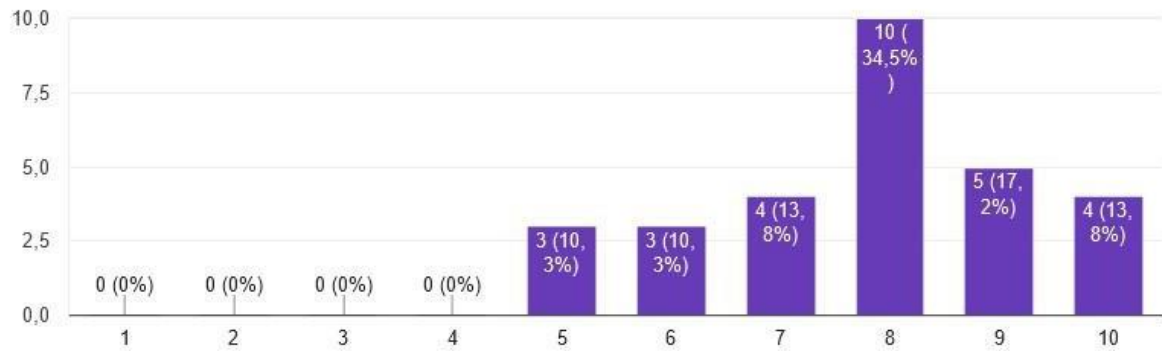
LAMPIRAN – LAMPIRAN

1. Hasil survey

Seberapa penting penggunaan Dispenser Pengharum Ruangan dalam suatu ruangan?

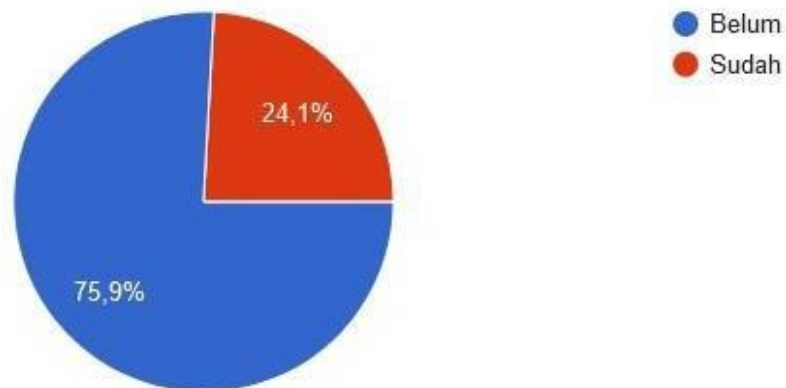


29 jawaban



Apakah proses pengecekan kondisi baterai dan jumlah cairan dalam Dispenser Pengharum Ruangan saat ini sudah efisien?

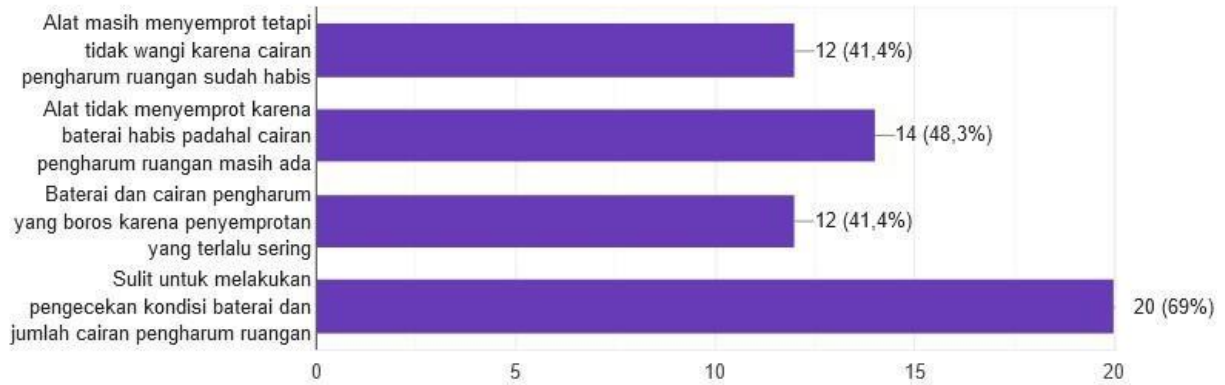
29 jawaban



Apa kekurangan dari Dispenser Pengharum Ruangan saat ini? (pilih satu atau lebih jawaban)

Salin

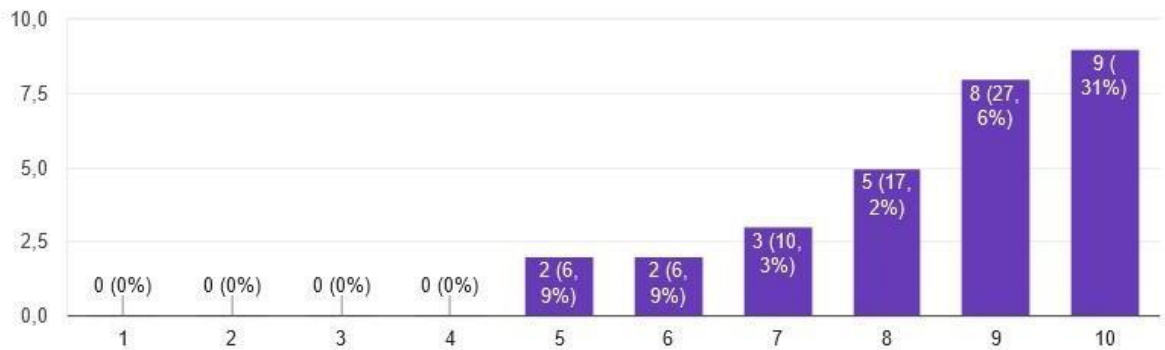
29 jawaban



Dari permasalahan di atas, seberapa penting pembuatan Dispenser Pintar Pengharum Ruangan Terintegrasi?

Salin

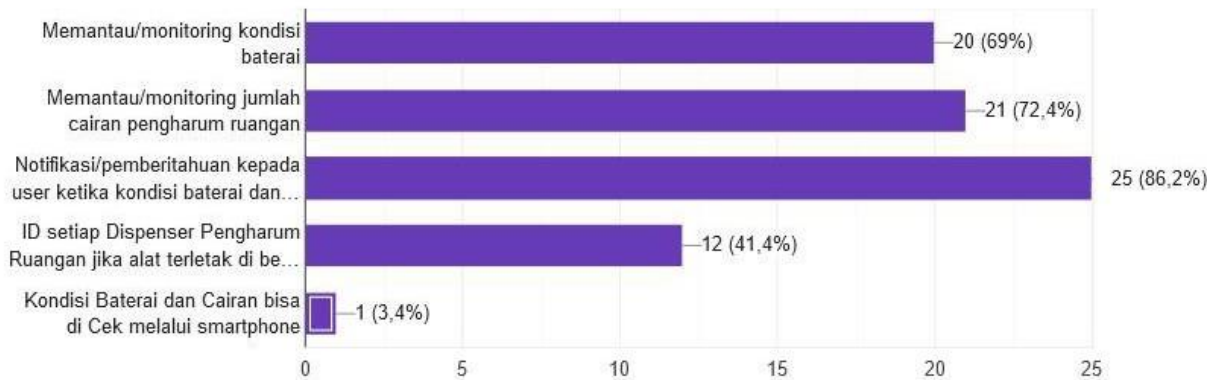
29 jawaban



Fitur apa saja yang diinginkan untuk Dispenser Pintar Pengharum Ruangan Terintegrasi (pilih satu atau lebih jawaban)

Salin

29 jawaban



2. Coding Keseluruhan dari SMART AFRESH IO

```
// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud
// See the Device Info tab, or Template settings
#define BLYNK_TEMPLATE_ID "TMPL6X0RgOGiK"
#define BLYNK_TEMPLATE_NAME "Smart Afresh IO"
char auth[] = "Q-04BbbUdX62kWi0PQ2PwwEhPkO4htnU";

//Libraries
#define BLYNK_PRINT Serial
#include <WiFiManager.h> // https://github.com/tzapu/WiFiManager
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <TimeLib.h>
#include <WidgetRTC.h>
#include <EEPROM.h>
//
#include <Arduino.h>
#include "HX711.h"
// #include "soc/rtc.h"
//
#include <Wire.h>
//
#include "time.h"
#include <ESP_Google_Sheet_Client.h>

//~~ESP32 Datalogging to Google Sheets
// Google Project ID
#define PROJECT_ID "smart-afresh-io-datalogging"
// Service Account's client email
#define CLIENT_EMAIL "smart-afresh-io-datalogging@smart-afresh-io-datalogging.iam.gserviceaccount.com"
// Service Account's private key
const char PRIVATE_KEY[] PROGMEM = "-----BEGIN PRIVATE KEY-----\nMIIEvQIBADANBgkqhkiG9w0BAQEFAASCBCwggSjAgEAAoIBAQCca3Iagc
// The ID of the spreadsheet where you'll publish the data
const char spreadsheetId[] = "lwpWGDRadDPiE9IO0gDeNF-d-m0JEcz1aP2aL4Pr6Syk";
```

```

// Timer variables
unsigned long lastTime = 0;
unsigned long timerDelay = 60000; //defaultnya 30000 utk 30 detik
// Token Callback function
void tokenStatusCallback(TokenInfo info);
// NTP server to request epoch time
const char* ntpServer = "pool.ntp.org";
// Variable to save current epoch time
unsigned long epochTime;
unsigned long RealTime;

// Function that gets current epoch time
unsigned long getTime() {
    time_t now;
    struct tm timeinfo;
    if (!getLocalTime(&timeinfo)) {
        //Serial.println("Failed to obtain time");
        return(0);
    }
    time(&now);
    return now;
}

//~~~Status Kaleng PR di Blynk
String statusKaleng;
const int NUM_ITEMS1 = 3;
const int MAX_ITEM_LENGTH1 = 15;
char menu_items1 [NUM_ITEMS1] [MAX_ITEM_LENGTH1] = {
    { "PENUH" },
    { "SETENGAH" },
    { "HABIS" }
};

//~~~Mekanisme Penyemprotan~~~
BlynkTimer timer;
WidgetRTC rtc;
int relayPin = 26;
int relayDuration = 0; // Duration in minutes
bool relayActive = true; // Relay status controlled by the button

//EEPROM addresses
const int relayDurationAddress = 0;
const int relayActiveAddress = sizeof(int);

//Timers for non-blocking delay management
unsigned long relayStartMillis = 0;
bool relayOn = false;

//~~~HX711 circuit wiring~~~
float weight;
const int LOADCELL_DOUT_PIN = 16;
const int LOADCELL_SCK_PIN = 4;
HX711 scale;

//~~~Battery Monitoring circuit wiring~~~
const int voltagePin = 34; //OG
float voltage;
float batPercentage;

// Pin untuk RGB LED 1 (Baterai)
const int redPin1 = 19; // Pin merah
const int greenPin1 = 18; // Pin hijau
const int bluePin1 = 5; // Pin biru

// Pin untuk RGB LED 2 (Berat)
const int redPin2 = 23; // Pin merah
const int greenPin2 = 22; // Pin hijau
const int bluePin2 = 21; // Pin biru

```

```

// Fungsi untuk mengatur warna RGB LED untuk baterai
void setColorBattery(int red, int green, int blue) {
  analogWrite(redPin1, red);
  analogWrite(greenPin1, green);
  analogWrite(bluePin1, blue);
}

// Fungsi untuk mengatur warna RGB LED untuk berat
void setColorWeight(int red, int green, int blue) {
  analogWrite(redPin2, red);
  analogWrite(greenPin2, green);
  analogWrite(bluePin2, blue);
}

//Blynk Setup
BLYNK_CONNECTED() {
  rtc.begin();
  Blynk.virtualWrite(V4, relayDuration);
  Blynk.virtualWrite(V5, relayActive);
}

BLYNK_WRITE(V4) {
  relayDuration = param.asInt(); // Get value from slider on virtual pin V4
  EEPROM.writeInt(relayDurationAddress, relayDuration); // Save relayDuration to EEPROM
  EEPROM.commit();
}

BLYNK_WRITE(V5) { // Virtual pin V5 for the button
  relayActive = param.asInt(); // Get value from button on virtual pin V5
  digitalWrite(relayPin, relayActive ? LOW : HIGH); // Update relay status based on button
  EEPROM.writeBool(relayActiveAddress, relayActive); // Save relayActive to EEPROM
  EEPROM.commit();
}

void sampleBattery() {
  float rawValue = analogRead(voltagePin);
  voltageSum += rawValue;
  sampleCount++;

  if (sampleCount >= numSamples) {
    float averageRawValue = voltageSum / numSamples;
    //float voltage = (averageRawValue / 4095.0) * 3.3;
    voltage = (averageRawValue / 4095.0) * 3.3;
    const float calibrationFactor = 1.06; // Calibration factor
    const float offset = -0.18 + 0.23; // Additional offset, 3 July 2024 : +0.33 (multi 2.50, blynk 2.60)
    voltage = (voltage * calibrationFactor) + offset;

    //batPercentage = (voltage / 3.3) * 100;
    const float batMin = 2.3; //tegangan max baterai dari pembagi tegangan
    const float batMax = 3.3; //tegangan min baterai dari pembagi tegangan
    batPercentage = ((voltage - batMin) / (batMax - batMin)) * 100;
    if (batPercentage < 0) batPercentage = 0; // Limit minimum percentage to 0
    if (batPercentage > 100) batPercentage = 100; // Limit maximum percentage to 100

    Serial.print("Average Voltage (20s): ");
    Serial.print(voltage, 2); // Display voltage with two decimal places
    Serial.println(" V");

    Serial.print("Percentage: ");
    Serial.print(batPercentage, 2); // Display percentage with two decimal places
    Serial.println(" %");

    // Send voltage to Blynk
    Blynk.virtualWrite(V0, voltage); // For battery voltage
    // Send percentage voltage to Blynk
    Blynk.virtualWrite(V1, batPercentage); // For battery percentage
    // Send status kritis baterai to Blynk
    Blynk.virtualWrite(V8, 0); // LED for status baterai kritis

    if (batPercentage < 30) {
      Blynk.virtualWrite(V8, 255); // Turn on LED for baterai kritis
    }
  }
}

```

```

// Atur warna RGB LED berdasarkan persentase baterai
if (batPercentage >= 70) {
  setColorBattery(0, 255, 0); // Hijau
} else if (batPercentage >= 20) {
  setColorBattery(255, 255, 0); // Kuning
} else {
  setColorBattery(255, 0, 0); // Merah
}

// Reset for next averaging cycle
voltageSum = 0;
sampleCount = 0;
}
}

void weightMonitoring() {
  // Read weight
  weight = scale.get_units(10); // Change the argument to adjust the number of readings averaged (default 1)

  // Print weight to Serial
  Serial.print("Weight: ");
  Serial.print(weight);
  Serial.println(" g");

  // Send weight to Blynk
  Blynk.virtualWrite(V2, weight);
  // Send percentage weight to Blynk

  Blynk.virtualWrite(V3, 0); // LED for Penuh
  Blynk.virtualWrite(V6, 0); // LED for Setengah
  Blynk.virtualWrite(V7, 0); // LED for Habis
  Blynk.virtualWrite(V8, 0); // LED for Kosong

  if (weight > 185) {
    statusKaleng = menu_items1[0];
    Serial.println(statusKaleng);
    Blynk.virtualWrite(V3, 255); // Turn on LED for Penuh
  } else if (weight > 70 && weight < 185) {
    statusKaleng = menu_items1[1];
    Serial.println(statusKaleng);
    Blynk.virtualWrite(V6, 255); // Turn on LED for Setengah
  } else if (weight < 70) {
    statusKaleng = menu_items1[2];
    Serial.println(statusKaleng);
    Blynk.virtualWrite(V7, 255); // Turn on LED for Habis
  } else {
    Blynk.virtualWrite(V8, 255); // Turn on LED for Kosong
  }

  // Atur warna RGB LED berdasarkan persentase berat
  if (weight > 185) {
    setColorWeight(0, 255, 0); // Hijau
  } else if (weight > 70 && weight < 185) {
    setColorWeight(255, 255, 0); // Kuning
  } else {
    setColorWeight(255, 0, 0); // Merah
  }
}
}

```

```

void checkRelay() {
  // Get current time
  int currentHour = hour();
  int currentMinute = minute();

  // Define work hours
  int startHour = 5;
  int startMinute = 0;
  int endHour = 24; // This should be 17
  int endMinute = 0;

  // Check if current time is within work hours
  if (relayActive && currentHour >= startHour && currentHour < endHour) {
    // Turn on the relay for the specified duration if not already on
    if (!relayOn) { //!relayOn true kalau relayOn = false atau 0
      relayStartMillis = millis();
      relayOn = true;
      digitalWrite(relayPin, LOW); //HIGH
    } else {
      // Check if the relay duration has elapsed
      if (millis() - relayStartMillis >= (relayDuration * 1000UL)) {
        digitalWrite(relayPin, HIGH); //LOW
        relayOn = false;
      }
    }
  } else {
    digitalWrite(relayPin, LOW); // Ensure relay is off outside work hours (tadinya HIGH)
    relayOn = false; // Reset relay status
  }
}

void tokenStatusCallback(TokenInfo info){
  if (info.status == token_status_error){
    GSheet.printf("Token info: type = %s, status = %s\n", GSheet.getTokenType(info).c_str(), GSheet.getTokenStatus(info).c_str());
    GSheet.printf("Token error: %s\n", GSheet.getTokenError(info).c_str());
  }
  else{
    GSheet.printf("Token info: type = %s, status = %s\n", GSheet.getTokenType(info).c_str(), GSheet.getTokenStatus(info).c_str());
  }
}

```

```

void publishtoGoogleSheets() {
  // Call ready() repeatedly in loop for authentication checking and processing
  bool ready = GSheet.ready();

  if (ready && millis() - lastTime > timerDelay) {
    lastTime = millis();

    FirebaseJson response;

    Serial.println("\nAppend spreadsheet values...");
    Serial.println("-----");

    FirebaseJson valueRange;
    // Get timestamp
    epochTime = getTime();
    RealTime = epochTime + 25200;

    valueRange.add("majorDimension", "COLUMNS");
    valueRange.set("values/[0]/[0]", RealTime);
    valueRange.set("values/[1]/[0]", statusKaleng);
    valueRange.set("values/[2]/[0]", weight);
    valueRange.set("values/[3]/[0]", batPercentage);
    valueRange.set("values/[4]/[0]", voltage);

    // For Google Sheet API ref doc, go to https://developers.google.com/sheets/api/reference/rest/v4/spreadsheets.values/append
    // Append values to the spreadsheet
    bool success = GSheet.values.append(&response /* returned response */, spreadsheetId /* spreadsheet Id to append */, "Sheet1");
    if (success) {
      response.toString(Serial, true);
      valueRange.clear();
    }
    else {
      Serial.println(GSheet.errorReason());
    }
    Serial.println();
    Serial.println(ESP.getFreeHeap());
  }
}

void setup() {
  Blynk.config(auth);
  // Debug console
  Serial.begin(115200);

  // Initialize EEPROM
  EEPROM.begin(512); // Initialize EEPROM with 512 bytes of space
  // Load saved values from EEPROM
  relayDuration = EEPROM.readInt(relayDurationAddress);
  relayActive = EEPROM.readBool(relayActiveAddress);

  //WiFi.mode(WIFI_STA); // explicitly set mode, esp defaults to STA+AP
  // it is a good practice to make sure your code sets wifi mode how you want it.

  WiFiManager wm;
  wm.resetSettings(); // reset settings - wipe stored credentials for testing, these are stored by the esp library

  bool res;
  res = wm.autoConnect("Smart AFRESH IO", "87654321"); // password protected ap

  if(!res) {
    Serial.println("Failed to connect");
    // ESP.restart();+
  }
  else {
    //if you get here you have connected to the WiFi
    Serial.println("Connected...yeey :)");
  }
}

```

```

//~~~Setup ESP32 Datalogging to Google Sheets
//Configure time
configTime(0, 0, ntpServer);

// Set the callback for Google API access token generation status (for debug only)
GSheet.setTokenCallback(tokenStatusCallback);
// Set the seconds to refresh the auth token before expire (60 to 3540, default is 300 seconds)
GSheet.setPrerefreshSeconds(10 * 60);
// Begin the access token generation for Google API authentication
GSheet.begin(CLIENT_EMAIL, PROJECT_ID, PRIVATE_KEY);

//~~~Setup Mekanisme Penyemprotan (relay)~
pinMode(relayPin, OUTPUT);
digitalWrite(relayPin, LOW); // Set relay off initially (tadinya HIGH with the 4 channel relay)

// Setup RGB LED pins
pinMode(redPin1, OUTPUT);
pinMode(greenPin1, OUTPUT);
pinMode(bluePin1, OUTPUT);
pinMode(redPin2, OUTPUT);
pinMode(greenPin2, OUTPUT);
pinMode(bluePin2, OUTPUT);

// Synchronize time on connection
setSyncInterval(10 * 60); // Sync interval in seconds

// Setup timers
timer.setInterval(1000L, checkRelay); // Check relay status every second
timer.setInterval(100L, sampleBattery); // Sample battery voltage every 100 ms
timer.setInterval(1000L, weightMonitoring); // Check weight every second
timer.setInterval(1000L, publishToGoogleSheets);

Serial.println("HX711 Demo");

Serial.println("Initializing the scale");

scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);

Serial.println("Before setting up the scale:");
Serial.print("read: \t\t");
Serial.println(scale.read()); // print a raw reading from the ADC

Serial.print("read average: \t\t");
Serial.println(scale.read_average(20)); // print the average of 20 readings from the ADC

Serial.print("get value: \t\t");
Serial.println(scale.get_value(5)); // print the average of 5 readings from the ADC minus the tare weight (not set yet)

Serial.print("get units: \t\t");
Serial.println(scale.get_units(5, 1)); // print the average of 5 readings from the ADC minus tare weight (not set) divided
// by the SCALE parameter (not set yet)

scale.set_scale(998.2111); //998.2111 //2 July 2024 : 996.9497 //3 July 2024 : 1110.8802, 1065.9551 (with 178 g), 1087.0471 (191 g), 1021.6911 (191 g)
//scale.set_scale(-471.497); // this value is obtained by calibrating the scale with known weights; see the README for details
scale.tare(); // reset the scale to 0

Serial.println("After setting up the scale:");

Serial.print("read: \t\t");
Serial.println(scale.read()); // print a raw reading from the ADC

Serial.print("read average: \t\t");
Serial.println(scale.read_average(20)); // print the average of 20 readings from the ADC

Serial.print("get value: \t\t");
Serial.println(scale.get_value(5)); // print the average of 5 readings from the ADC minus the tare weight, set with tare()

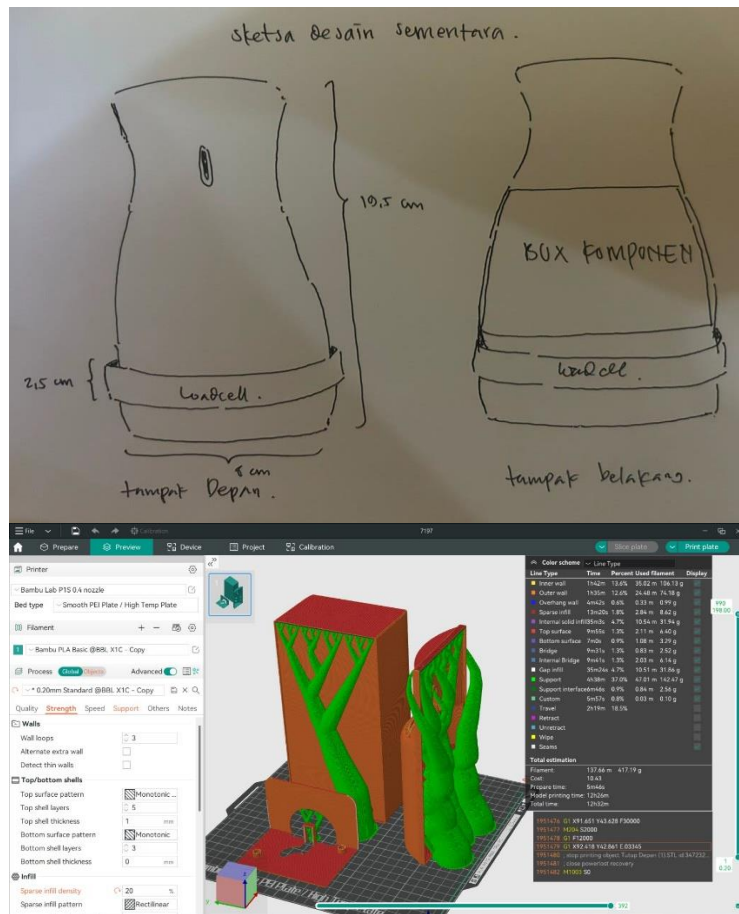
Serial.print("get units: \t\t");
Serial.println(scale.get_units(5, 1)); // print the average of 5 readings from the ADC minus tare weight, divided
// by the SCALE parameter set with set_scale

Serial.println("Readings:");
}

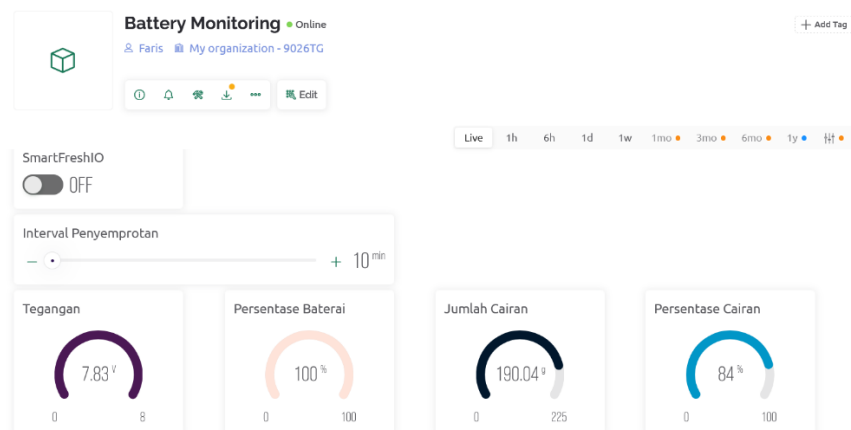
void loop() {
  Blynk.run();
  timer.run();
}

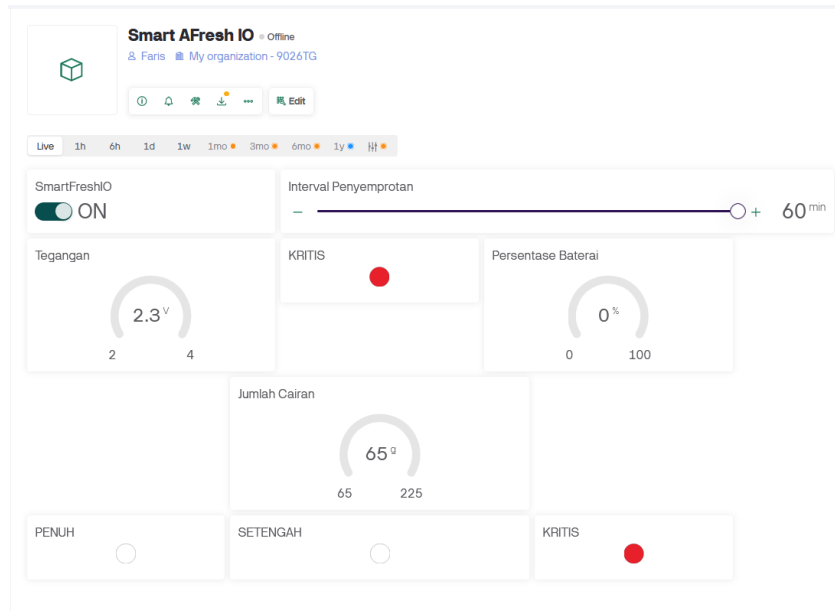
```

3. Desain 3D Sementara (Sketsa) dan Final



4. Tampilan UI Lama dan Baru pada Blynk





5. Tampilan Datalogging Menggunakan *Google Sheets*

HABIS	54,0	0	0,98	06/07/2024 3:52:43
SETENGAH	176,8	0	1,00	06/07/2024 3:38:34
PENUH	189,4	0	0,99	06/07/2024 3:42:37

6. Proses Perancangan dan Hasil Akhir Alat



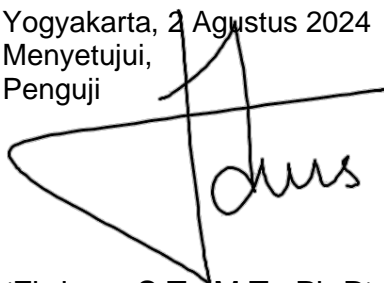
7. Lembar Saran untuk Revisi Laporan Capstone_A5

**TABEL PERBAIKAN
LAPORAN AKHIR CAPSTONE**

MAHASISWA #1 : 20524189 Wildan Firdaus
MAHASISWA #2 : 20524203 Faris Fatha Fallahian
JUDUL/TOPIK : Dispenser Pintar Pengharum Ruang Terintegrasi

No	Saran penguji	Perbaikan oleh mahasiswa	Halaman	Status
1	Lakukan tambahan analisis pada bagian konsumsi daya, dan berikan usulan solusi terkait keterbatasan baterai untuk mencukupi supply (habis dalam 4 jam)	Sudah ditambahkan analisis tambahan mengenai konsumsi daya berdasarkan konsumsi arus dari komponen-komponen yang digunakan pada alat. Kemudian kami juga memberi usulan tambahan mengenai keterbatasan daya tahan baterai dari alat ini yaitu dengan menambah kapasitas dari baterai yang digunakan secara signifikan dengan cara menghubungkan baterai secara paralel, sehingga konfigurasi baterai li-ion yang digunakan menjadi seri-paralel. Akan tetapi hal ini juga akan memakan ruang lebih banyak lagi pada box komponen dari alat ini dan tentunya menambah biaya produksi alat ini karena komponen baterai yang digunakan semakin banyak.	74-75	Approved
2	Lakukan tambahan analisis terkait dengan harga produksi, apakah ada peluang pengurangan biaya produksi? jika ada jelaskan detailnya	Sudah ditambahkan analisis mengenai peluang pengurangan biaya produksi yaitu dengan menggunakan baterai li-ion menggunakan baterai li-ion 18650 non-brand yang lebih murah, contohnya yang tersedia di pasaran adalah baterai li-ion yang memiliki range harga sekitar Rp 9.000 hingga Rp 25.000. Dari segi kelebihan biaya yang dikeluarkan akan menjadi lebih murah, tetapi kekurangannya adalah baterai non-brand kualitasnya tidak sebagus baterai brand yang lebih mahal tersebut, sehingga dikhawatirkan performa baterainya menurun seperti kapasitas aktualnya tidak sesuai dengan yang tertera pada spesifikasi, degradasi yang lebih cepat dan umur baterai yang lebih pendek. Kemudian untuk pencetakan 3D casing dapat dibuat lebih murah lagi dengan membuat desain 3D yang lebih compact atau kecil sehingga tidak menambah berat pencetakan 3D casing, lalu dapat mencari jasa desain 3D printing yang menawarkan harga lebih ramah di kantong, atau menggunakan material cetak yang lebih murah seperti dengan kualitas standar. Perlu diketahui bahwa walau biaya pencetakan akan menjadi lebih murah, tetapi kualitas casing yang dicetak tidak akan sehalus dan serapi pencetakan dengan kualitas halus. Kemudian penggunaan bahan PLA+ juga akan membuat casing menjadi lebih rapuh dan mudah retak, ini menimbulkan resiko baru jika terdapat beberapa part casing yang tipis.	83-84	Approved

Yogyakarta, 2 Agustus 2024
 Menyetujui,
 Penguji



(Firdaus, S.T., M.T., Ph.D)