

**WADAH MAKANAN DAN MINUMAN KUCING OTOMATIS
BERBASIS *INTERNET OF THINGS* (IOT)
PROYEK AKHIR**

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan
Diploma III Politeknik Manufaktur Negeri Bangka Belitung



DIUSULKAN OLEH

Dikhy Aditya Pratama 0032136

Stevy Pratiwi Andi Pacoba 0032157

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG
2024**

LEMBAR PENGESAHAN

WADAH MAKANAN DAN MINUMAN KUCING OTOMATIS BERBASIS *INTERNET OF THINGS (IOT)*

Oleh :

Dikhy Aditya Pratama 0032136

Stevy Pratiwi Andi Pacoba 0032157

Laporan akhir ini telah disetujui dan disahkan oleh sebagai salah satu syarat kelulusan

Program Diploma III Politeknik Manufaktur Negeri Bangka Belitung

Pembimbing 1



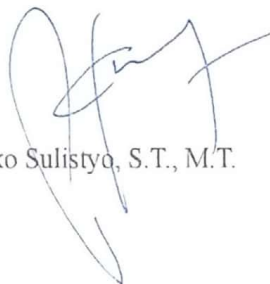
Aan Febriansyah, S.ST., M.T.

Pembimbing 2



Yudhi, S.ST., M.T.

Penguji 1



Eko Sulisty, S.T., M.T.

Penguji 2



Riki Afriansyah, S.T., M.T.

PERNYATAAN BUKAN PLAGIAT

Yang bertandatangan dibawah ini :

Dikhy Aditya Pratama 0032136

Stevy Pratiwi Andi Pacoba 0032157

Dengan judul : Wadah Makanan dan Minuman Kucing Otomatis Berbasis *Internet of Things* (IOT).

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 05 Juli 2024

Nama Mahasiswa

Tanda Tangan

Dikhy Aditya Pratama



Stevy Pratiwi Andi Pacoba



ABSTRAK

Alat ini dibuat agar mempermudah pemilik kucing dalam memberikan makan dan minum kepada kucing disaat pemilik tidak sedang berada dirumah. Pola makan kucing yang tidak teratur juga dapat membuat kucing menjadi kekurangan gizi, untuk kandungan makanan kucing juga harus diperhatikan. Tidak hanya nutrisi yang diperhatikan tetapi waktu juga harus diperhatikan agar kucing tidak akan merasa kelaparan dan kehausan dalam jangka waktu yang lama. Penulis membuat alat ini secara otomatis yang berbasis internet of things menggunakan komponen ESP 32 sebagai mikrontroler dan modul Wi-Fi yang menjadi otak dari semua program dan menggunakan komponen real time clock (RTC), sebagai pengatur waktu pemberian makan pada kucing. Tidak hanya menggunakan pengatur waktu penulis juga menggunakan firebase sebagai kendali melalui aplikasi inventor yang ada di smartphone. Jadi pemilik dapat mengontrol jumlah makanan dan minum didalam tangki. Pada alat pemberian makanan ada komponen motor servo untuk membuka dan menutup jalur makanan. Ada juga sensor load cell sebagai timbangan untuk makanan kucing agar tidak terlalu banyak disaat keluar. Dan ada sensor ultrasonic sebagai pengukur air minum didalam tangki. Jadi, pemilik kucing tidak akan khawatir lagi kucing akan kelaparan dan kehausan disaat pemilik tidak berada dirumah.

Kata kunci : Kucing, Internet Of Things, ESP32

ABSTRACT

It is made to support them in order to provide the cat to eat and drink at the cat was not be home. Irregular cat diet can also make a cat being deprived of nutrition, to the cat food also must be considered. Not only the time necessary but also must be considered that a cat would not feel famine and thirst for long periods. Author of make it automatically internet based of things using components ESP 32 as microcontroller and Wi-Fi module to the brain of all programs and use the real time clock (RTC), as officers time provision of feeding on a cat. Not only using timer writer also use firebase as control through the application of inventor in smartphone. So the owner can control the amount of food and drink in a tank. There is a provision of food on a roumanians servo motors to open and close the food. There ia also a load cell as everyone cat food so that not too much when out. And there are ultrasonic sensor for measuring drinking water in the tank. So, the cat would not worry the cat to hunger and thirst when the owner is not be home.

Keywords : Cat, Internet Of Things, ESP32

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh.

Dengan mengucapkan *Alhamdulillah Rabbil 'Alamin*, segala puji dan rasa syukur penulis panjatkan kepada Allah SWT, karena berkat rahmat dan karunia-Nya lah penulis diberi kemudahan sehingga dapat menyelesaikan Laporan Proyek Akhir yang berjudul “Wadah Makanan dan Minum Kucing Otomatis Berbasis *Internet Of Thing*” sesuai dengan waktu yang telah ditentukan.

Laporan Proyek Akhir ini di buat sebagai salah satu syarat kelulusan dan kewajiban setiap mahasiswa untuk menyelesaikan pendidikan Diploma III di Politeknik Manufaktur Negeri Bangka Belitung dengan program studi DIII Teknik Elektronika. Penyusunan laporan akhir ini sesuai dengan instruksi dan arahan dari panduan proyek akhir. Selama mengerjakan proyek akhir ini penulis di dukung dan dibimbing juga semangat. Untuk itu penulis ingin mengucapkan terimakasih dan rasa syukur kepada :

1. Allah SWT, yang telah memberikan kesehatan dan kelancaran kepada penulis selama menyelesaikan proyek akhir ini,
2. Keluarga, yang selalu memberikan dukungan kepada penulis agar selalu semangat dalam mengerjakan proyek akhir,
3. Bapak I Made Andik Setiawan, M.Eng, Ph.D selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung,
4. Bapak Aan Febriansyah, S.ST., M.T. selaku pembimbing I yang telah banyak meluangkan waktu dan memberikan solusi kepada penulis dalam mengerjakan proyek akhir,
5. Bapak Yudhi, M.T. selaku pembimbing II yang telah banyak meluangkan waktu dan memberikan solusi kepada penulis dalam mengerjakan proyek akhir,

6. Semua teman – teman yang telah memberikan bantuan dan dorongan juga semangat kepada penulis,
7. Kepada Dikhy dan Stevy yang telah mengerjakan proyek akhir ini walaupun banyak perselisihan tetapi tetap menyelesaikan kewajiban sesuai dengan waktu yang telah ditentukan.
8. Kepada Nisvina Anjelia yang telah banyak berpartisipasi dalam pembuatan proyek akhir ini dan sebagai pemberi dukungan yang paling berharga bagi kami berdua.

Penulis menyadari bahwa dalam penyusunan laporan proyek akhir ini masih jauh dari kata sempurna dan masih banyak terdapat kesalahan serta kekurangan di dalamnya, baik itu dalam sistematika penulisan laporan maupun isi laporan itu sendiri.

Oleh karna itu penulis berharap semoga laporan ini dapat bermanfaat bagi penulis sendiri dan bagi mahasiswa/mahasiswi Politeknik Manufaktur Negeri Bangka Belitung lainnya. Adapun penulis mengharapkan kritikyng bersifat membangun sebagai pembelajaran kedepannya. Apabila ada salah kata penulis meminta maaf, dan kepada Tuhan Yang Maha Esa penulis memohon ampun. Atas perhatiannya, penulis mengucapkan terima kasih.

Wassalamu'alaikum Warahmatullahi Wabaraktuh.

DAFTAR ISI

	Halaman
LEMBAR PENGESAHAN	i
PERNYATAAN BUKAN PLAGIAT	ii
ABSTRAK	iii
ABSTRACT	iv
KATA PENGANTAR.....	v
DAFTAR ISI	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
BAB I	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan Proyek Akhir	2
BAB II.....	3
DASAR TEORI	3
2.1 Tinjauan Pustaka.....	3
2.2 Kucing	4
2.3 Kandungan Didalam Makanan Kucing	5
2.4 Sistem Kontrol Otomatis	5
2.5 <i>Arduino Uno</i>	5

2.6 Mikrokontroler ESP 32.....	6
2.7 Real Time Clock (RTC).....	7
2.8 Load Cell (Sensor Berat).....	8
2.9 Ultrasonic (Sensor Jarak).....	8
2.10 Motor Servo.....	9
2.11 Liquid Crystal Display (LCD).....	10
2.12 Sistem Kontrol Manual (<i>Internet Of Things</i>).....	10
BAB III	12
METODE PELAKSANAAN.....	12
3.1 <i>Survey</i> Dan Pengumpulan Data.....	13
3.2 Perancangan <i>Hardware</i> Konstruksi Alat.....	13
3.3 Perancangan <i>Hardware Elektrik</i>	13
3.4 Pembuatan <i>Hardware</i> dan <i>Software Elektrik</i>	14
3.4.1 Pembuatan <i>Hardware Elektrik</i>	14
3.4.2 Pembuatan <i>Software Elektrik</i>	15
3.5 Pengujian <i>Hardware</i> dan <i>Software Elektrik</i>	15
3.5.1 Pengujian <i>Hardware Elektrik</i>	15
3.5.2 Pengujian <i>Software Elektrik</i>	16
3.6 Pembuatan Laporan Akhir.....	16
BAB IV	17
PEMBAHASAN	17
4.1 Deskripsi Alat.....	17

4.2 Perancangan dan Pembuatan <i>Hardware</i> Mekanik Wadah Makanan dan Minuman Kucing.	18
4.2.1 Perancangan <i>Hardware</i> Mekanik Wadah Makanan dan Minuman Kucing	18
4.2.2 Pembuatan <i>Hardware</i> Mekanik Wadah Makanan dan Minuman Kucing ..	19
4.3 Perancangan dan Pembuatan <i>Hardware</i> Elektrik Wadah Makanan dan Minuman Kucing	21
4.3.1 Perancangan <i>Hardware Elektrik</i> Wadah Makanan dan Minuman Kucing .	22
4.3.2 Pembuatan <i>Hardware Elektrikal</i> Wadah Makanan dan Minuman Kucing.	23
4.4 Pengujian <i>Hardware Elektrikal</i> Wadah Makanan dan Minuman Kucing	24
4.4.1 Perancangan Sensor <i>Ultrasonic</i>	24
4.4.2 Pembuatan Sensor <i>Ultrasonic</i>	25
4.4.3 Pengujian Sensor <i>Ultrasonic</i>	26
4.4.4 Perancangan Sensor <i>Load Cell</i>	29
4.4.5 Pembuatan Sensor <i>Load Cell</i>	30
4.4.6 Pengujian Sensor <i>Load Cell</i>	33
4.4.7 Perancangan <i>Real Time Clock</i>	34
4.4.8 Pembuatan <i>Real Time Clock</i>	35
4.4.9 Pengujian <i>Real Time Clock</i>	36
4.5 Perancangan dan Pembuatan <i>Software Internet Of Things</i>	36
4.5.1 Perancangan <i>Software Internet Of Things</i>	36
4.5.2 Pembuatan <i>Software Internet Of Things</i>	37
4.6 Pengujian Keseluruhan Berbasis <i>Internet Of Things</i>	39
4.6.1 Hasil Pengujian Berbasis <i>Internet Of Things</i>	40

BAB V.....	42
KESIMPULAN.....	42
DAFTAR PUSTAKA.....	43
LAMPIRAN I.....	45
DAFTAR RIWAYAT HIDUP.....	45
LAMPIRAN II.....	48
PROGRAM.....	48



DAFTAR TABEL

	Halaman
Tabel 2. 1 Tinjauan Pustaka	3
Tabel 4. 1 Pengujian Ketinggian Air dan Sensor Ultrasonic.....	27
Tabel 4. 2 Hasil Pengujian Sensor Ultrasonic.....	28
Tabel 4. 3 Hasil Pengujian Sensor Load Cell.....	32
Tabel 4. 4 Hasil Pengujian Berat Makanan Secara Manual dan Sensor Load Cell	33
Tabel 4. 5 Hasil Pengujian Keseluruhan Berbasis Internet Of Things.....	41



DAFTAR GAMBAR

	Halaman
Gambar 2. 1 Kucing Rumah.....	4
Gambar 2. 2 Arduino Uno.....	6
Gambar 2. 3 ESP 32.....	7
Gambar 2. 4 Real Time Clock.....	8
Gambar 2. 5 Load Cell.....	8
Gambar 2. 6 Ultrasonic.....	9
Gambar 2. 7 Motor Servo.....	9
Gambar 2. 8 Liquid Crystal Display.....	10
Gambar 3. 1 Flowchart Pembuatan Alat.....	12
Gambar 3. 2 Desain Konstruksi Alat.....	13
Gambar 3. 3 Desain Hardware Elektrik.....	14
Gambar 4. 1 Wadah Makanan dan Minuman Kucing.....	18
Gambar 4. 2 Pembuatan Tangki Makanan.....	19
Gambar 4. 3 Pembuatan Tuas Motor Servo.....	20
Gambar 4. 4 Pembuatan Jalan Turun Makanan.....	20
Gambar 4. 5 Pembuatan Tempat Peletakan Sensor Ultrasonic.....	20
Gambar 4. 6 Diagram Blok Elektrikal.....	21
Gambar 4. 7 Flowchart Elektrikal.....	22
Gambar 4. 8 Skema Perkabelan Rangkaian Elektrikal.....	23
Gambar 4. 9 Pembuatan Box Kontrol Elektrikal.....	23
Gambar 4. 10 Box Kontrol Elektrikal.....	24
Gambar 4. 11 Skema Pengkabelan ESP32, Sensor ultrasonic dan LCD.....	25
Gambar 4. 12 Pembuatan Sensor Ultrasonic.....	26
Gambar 4. 13 Blok Diagram Pengujian Sensor Ultrasonic.....	26
Gambar 4. 14 Pengujian Ketinggian Air Secara Manual.....	28

Gambar 4. 15 Hasil Pengujian Menggunakan Sensor Ultrasonic	28
Gambar 4. 17 Skema Pengkabelan ESP32, Sensor Load Cell dan LCD	30
Gambar 4. 18 Pembuatan Sensor Load Cell	31
Gambar 4. 19 Blok Diagram Pengujian Sensor Load Cell	31
Gambar 4. 20 Hasil Pengujian Menggunakan Sensor Load Cell pada LCD	33
Gambar 4. 21 Hasil Pengujian Berat Makanan Menggunakan Timbangan Digital.....	33
Gambar 4. 22 Skema Perkabelan Real Time Clock, ESP32 dan LCD	34
Gambar 4. 23 Pembuatan Real Time Clock.....	35
Gambar 4. 24 Diagram Blok Pengujian Real Time Clock.....	35
Gambar 4. 25 Tampilan Pengujian Real Time Clock.....	36
Gambar 4. 26 Tampilan Pada Serial Monitor.....	36
Gambar 4. 27 Perancangan Software MIT App Inventor	37
Gambar 4. 28 Pembuatan Software MIT App Inventor	38
Gambar 4. 29 Program Penampil Database ke MIT App Inventor	38
Gambar 4. 30 Program Penyimpan Data dari Firebase ke MIT App Inventor	39
Gambar 4. 31 Tampilan Awal Aplikasi MIT App Inventor	39
Gambar 4. 32 Hasil Pengujian Keseluruhan pada Aplikasi MIT Inventor	40
Gambar 4. 33 Hasil Pengujian Keseluruhan pada LCD.....	40
Gambar 4. 34 Hasil Pengujian Sensor Load Cell Pada Hardware	40
Gambar 4. 35 Hasil Pengujian Menggunakan Timbangan Digital	41

BAB I

PENDAHULUAN

1.1 Latar Belakang

Hewan memiliki tempat khusus di hati manusia dari semua lapisan masyarakat. Mereka sering berperan sebagai sahabat dan memberikan kesempatan buat interaksi sosial. Berbagai macam hewan yang biasa dipelihara sebagai hewan peliharaan, yakni mamalia, burung, reptil, dan ikan. Diantaranya, kucing sudah menjadi salah satu pilihan yang paling dicintai dan populer di seluruh dunia. Dikenal karena perawatannya yang rendah dan sejarah persahabatannya yang panjang dengan manusia, kucing ialah hewan peliharaan favorit bagi banyak orang.

Ialah hal yang umum bagi seseorang buat memiliki kucing tetapi kesulitan menemukan waktu buat merawatnya secara memadai dengan memberikan nutrisi dan hidrasi yang tepat. Kurangnya perhatian terhadap kebutuhan dasar hewan peliharaan bisa menimbulkan dampak negatif pada kesehatan dan kesejahteraan kucing secara keseluruhan, yang pada akhirnya mengakibatkan kucing menjadi tidak sehat dan tidak bahagia.

Dalam upaya memberikan kemudahan bagi pemilik kucing sibuk yang mungkin kesulitan memberi makan dan menghidrasi hewan peliharaannya secara konsisten, proyek ini diberi nama 'Wadah Pakan dan Minuman Kucing Otomatis Berbasis *Internet of Things* (IOT)'. Alat inovatif ini bertujuan buat meringankan beban perawatan kucing sehari-hari, memungkinkan pemilik memastikan kucing peliharaannya membisa nutrisi yang baik dan terhidrasi tanpa menambah stres atau kekhawatiran.

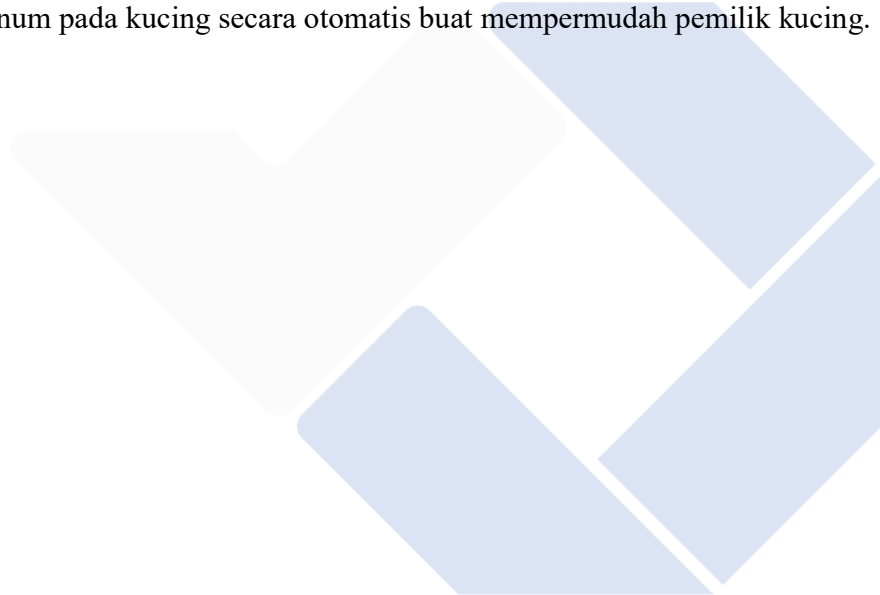
1.2 Perumusan Masalah

Berlandaskan penjelasan diawal, berikut ialah beberapa masalah yang dibisakan

1. Kapasitas tangki makanan utama 1 kg.
2. Sensor berat yang dipakai *load cell*.
3. Pada aplikasi ada pengaturan waktu dan total makanan yang tersedia, saat di kirim akan menakar secara otomatis total makanan dan minuman yang keluar.

1.3 Tujuan Proyek Akhir

Adapun tujuan pembuatan proyek akhir ini membuat alat buat memberikan makan dan minum pada kucing secara otomatis buat mempermudah pemilik kucing.



BAB II

DASAR TEORI

2.1 Tinjauan Pustaka

Tinjauan pustaka ini memiliki tujuan untuk membandingkan proyek akhir maupun penelitian yang berkaitan dengan judul proyek akhir ini. Dibawah ini beberapa penelitian yang berkaitan.

Tabel 2. 1 Tinjauan Pustaka

No	Judul	Hasil
1	Alat Pemberi Makan dan Minum Kucing Terjadwal Otomatis Berbasis <i>Arduino Mega</i> [1].	Alat pemberi pakan otomatis ini menggunakan sensor <i>RFID</i> sebagai sensor pengamanan agar jam makan kucing terjadwal dan kucing tidak menjadi obesitas karena terlalu sering makan.
2	Rancang Bangun Otomatisasi Pemberi Pakan Dan Minum Kucing Berbasis <i>Internet Of Things (IOT)</i> [2].	Alat pemberi pakan kucing ini menggunakan <i>web server thingsboard</i> buat menampilkan data pakan dan minum kucing.
3	Otomasi Wadah Pakan Kucing Berbasis <i>Mikrokontroler ESP32 WROOM</i> [3].	Alat pemberi makan dan minum kucing terjadwal secara otomatis alat ini menggunakan aplikasi blink sebagai mikrkontoler makanan yang ada diwadah. Aplikasi blink memonitor makanan per jam. Alat ini juga menggunakan kamera untuk memonitor

apakah makanan masih ada didalam wadah.

Dari hasil analisis data yang telah dilakukan sebelumnya, dapat disimpulkan bahwa prinsip kerja alat sama dengan prinsip kerja sebelumnya, dan komposisi yang dikembangkan juga sama dengan prinsip kerja sebelumnya. Pada proyek akhir ini penulis menggunakan push button dan LCD sebagai kontrol manual apabila sewaktu waktu aplikasi mengalami gangguan.

2.2 Kucing

Kucing yang biasa disebut dengan kucing domestik atau kucing rumahan ialah salah satu spesies mamalia karnivora yang termasuk dalam famili *Felidae*. Istilah "kucing" biasanya mengacu pada kucing peliharaan, namun bisa juga mencakup anggota keluarga *Felidae* yang lebih besar, seperti singa dan harimau [4].



Gambar 2. 1 Kucing Rumah

Selama lebih dari 5.000 tahun, kucing sudah terkait dengan peradaban manusia, terbukti dengan ditemukannya kerangka kucing di *Siprus* yang berasal dari zaman

kuno. Pada 3.500 SM, orang Mesir Kuno menyadari pentingnya kucing dalam melindungi tanaman mereka dari hewan pengerat di lumbung [4].

Saat ini, kucing menempati posisi penting sebagai salah satu hewan yang paling dicintai secara global. Kucing ras, termasuk *Persia*, *Siam*, *Manx*, dan *Sphinx*, sangat dicari karena silsilahnya. Ras ini biasanya dibesarkan di fasilitas penangkaran dengan campuran ras lain, seperti kucing liar atau domestik. Sikap kucing yang menawan dan menawan menjadikannya pilihan populer sebagai hewan peliharaan [4].

2.3 Kandungan Didalam Makanan Kucing

Saat memberikan makanan kepada kucing, penting buat mempertimbangkan dan memenuhi kebutuhan nutrisi spesifiknya, termasuk protein, lemak, karbohidrat, mineral, vitamin, dan hidrasi [5]. Persyaratan makanan kucing bergantung pada faktor-faktor seperti berat dan usianya. Hal ini terutama berlaku pada kucing jantan yang sedang kawin, kucing betina yang sedang hamil atau menyusui, serta anak kucing yang masih dalam masa pertumbuhan. Kucing dewasa biasanya makan beberapa kali dalam porsi kecil sepanjang hari, dengan frekuensi ideal sekitar 3-4 kali sehari [6].

2.4 Sistem Kontrol Otomatis

Sistem kontrol otomatis ialah alat canggih yang dirancang buat mengatur pemberian makan dan hidrasi kucing secara efisien. Memanfaatkan *mikrokontroler Arduino Uno* dan *ESP32* buat komunikasi serial, sistem ini mengintegrasikan komponen seperti *servo*, sensor sel beban, dan *ultrasonik* buat mengotomatisasi proses. Melewati aplikasi yang mudah dipakai, pemilik hewan peliharaan bisa memantau level makanan dan air secara *real-time*, memastikan hewan peliharaan berbulu mereka selalu membisa nutrisi dan hidrasi yang baik.

2.5 Arduino Uno

Penemunya sudah merancang sebuah sistem yang memungkinkan pengguna dengan mudah mengatur pengatur waktu agar makanan mereka dibagikan tiga kali sehari hanya dengan menekan satu tombol di perangkat keras. Perangkat lunak ini juga

memungkinkan tampilan dan penyesuaian pengatur waktu, serta memantau berat sel beban dan ketinggian air di dalam tangki agar tidak kosong. Namun, sistem ini bergantung pada koneksi *Wi-Fi* yang sudah diprogram ke dalam perangkat lunak dan tidak bisa diubah tanpa mengubah pengaturan di dalam program [7].



Gambar 2. 2 *Arduino Uno*

Arduino Uno menawarkan serangkaian fitur, termasuk 14 pin digital (6 di antaranya bisa dipakai sebagai *output PWM*), 6 input analog, *osilator Kristal 16MHz*, *port USB*, *port* sumber tegangan, *header ICSP*, dan reset. Tombol ini dirancang buat mencakup semua komponen yang diperlukan buat mendukung *mikrokontroler* secara efektif. *Arduino Uno* bisa diberi daya baik dengan menghubungkannya ke komputer melewati USB atau dengan menyuplai daya DC dari baterai atau adaptor AC-DC, menjadikannya alat serbaguna dan mudah dipakai buat berbagai proyek [7].

2.6 Mikrokontroler ESP 32

Mikrokontroler ESP32 ialah *mikrokontroler SoC* yang dilengkapi dengan *WI-FI* 802.11 b/g/n, Bluetooth versi 4.2, dan berbagai periferal. Ia menawarkan serangkaian fitur lengkap, termasuk prosesor, penyimpanan, dan akses *GPIO*. Dengan kemampuannya, *ESP32* bisa berfungsi sebagai alternatif yang cocok buat sirkuit *Arduino*, menyediakan konektivitas *WI-FI* langsung [8].



Gambar 2. 3 *ESP 32*

Board ini tersedia dalam dua versi, 30 *GPIO* dan 36 *GPIO*, keduanya memiliki tujuan yang sama. Kami sudah memilih versi 30 *GPIO* karena disertakannya dua pin *GND*. Pin diberi label dengan nyaman di bagian atas papan buat memudahkan identifikasi. Selain itu, *board* ini dilengkapi antarmuka *USB* ke *UART* yang mudah dipakai buat pemrograman yang lancar, menjadikannya ideal buat mengembangkan aplikasi seperti *Arduino IDE* dan mengakses sumber daya melewati konektor *micro USB* [8].

2.7 Real Time Clock (RTC)

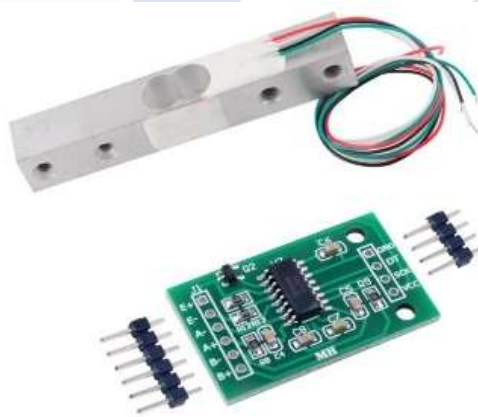
Modul *RTC* berfungsi sebagai alat penting buat menyimpan dan memelihara informasi *real-time* secara akurat. Terdiri dari kristal dan baterai, *RTC* mengandalkan fungsi kristal yang tepat buat penghitungan waktu dan pasokan daya berkelanjutan dari baterai buat mempertahankan operasinya. Data waktu yang terekam, termasuk detik, menit, jam, dan tanggal, disimpan dalam memori melewati proses ini. Namun, jika baterai rusak atau dilepas, mekanisme penghitungan kristal akan berhenti, mengakibatkan hilangnya informasi waktu yang akurat dan potensi kehilangan data jika *RTC* disetel ulang secara tidak terduga [9].



Gambar 2. 4 *Real Time Clock*

2.8 Load Cell (Sensor Berat)

Sensor berat, juga dikenal sebagai sel beban, ialah komponen penting dalam sistem penimbangan digital. Ini dirancang khusus buat mengukur tekanan atau berat suatu beban dengan presisi. Sel beban biasanya dipakai di jembatan timbang buat menentukan berat truk yang membawa bahan mentah secara akurat. Sensor ini beroperasi berlandaskan prinsip tekanan buat memberikan pengukuran yang akurat [10].



Gambar 2. 5 *Load Cell*

2.9 Ultrasonic (Sensor Jarak)

Sensor *ultrasonik* memiliki kemampuan buat mendeteksi objek pada jarak tertentu yang dilaluinya, pada akhirnya menawarkan keuntungan dalam mendeteksi objek dari

jarak yang lebih jauh. Sensor *ultrasonik HC-SR04* ialah komponen yang memanfaatkan gelombang *ultrasonik* yang terdiri dari *transmitter* dan *receiver*. Pemancar mengirimkan gelombang *ultrasonik*, sedangkan penerima menangkap gelombang pantulan. Dengan kemampuan mengukur jarak antara 2 cm hingga 400 cm dengan akurasi kurang lebih 3 mm, sensor *ultrasonik HC-SR04* menjadi alat yang handal buat mendeteksi objek [11].



Gambar 2. 6 *Ultrasonic*

2.10 Motor *Servo*

Motor *servo* ialah perangkat *aktuator* putar canggih yang dilengkapi dengan sistem kontrol umpan balik *loop* tertutup, memungkinkan penyesuaian posisi sudut poros keluaran motor secara tepat. Terdiri dari motor DC, roda gigi, rangkaian kontrol, dan *potensiometer*, motor *servo* memakai *potensiometer* buat menentukan rentang putaran. Dengan mengirimkan lebar pulsa yang bervariasi (0,5 ms - 2 ms) melewati kaki sinyal, sinyal *PWM* bisa dipakai buat menyempurnakan sudut rotasi sumbu motor *servo* [12].



Gambar 2. 7 Motor *Servo*

2.11 *Liquid Crystal Display (LCD)*

LCD memanfaatkan sifat unik kristal cair buat menghasilkan gambar visual, menjadikannya komponen penting dalam perangkat elektronik seperti laptop, ponsel, dan TV. Terdiri dari komponen lampu latar dan lapisan kristal cair, layar ini mengandalkan sumber cahaya eksternal buat memantulkan dan mentransmisikan cahaya melewatinya. Kristal cair organik, yang diapit di antara dua permukaan kaca konduktif, tidak memancarkan cahaya sendiri melainkan memanipulasi dan mengarahkan cahaya yang melewatinya [13].



Gambar 2. 8 *Liquid Crystal Display*

Liquid Crystal Display Dalam pengoperasiannya, perangkat ini mengandalkan komponen memori seperti *DDRAM*, *CGRAM*, dan *CGROM*, serta register kontrol termasuk register perintah dan register data. Dilengkapi dengan 5 pin penting: pin data, RS (pilih register), R/W (baca/tulis), E (aktifkan), dan *VLCD* [14].

2.12 *Sistem Kontrol Manual (Internet Of Things)*

Konsep *Internet of Things* mencakup perangkat elektronik yang bisa berkomunikasi dengan pengguna buat memantau dan mengontrol fungsinya melewati Internet. Ide ini pertama kali diperkenalkan oleh Kevin Ashton pada tahun 1999. *Internet of Things* berfungsi sebagai landasan pertukaran informasi global, memungkinkan layanan rumit disampaikan dengan menghubungkan objek fisik dan virtual melewati teknologi komunikasi canggih. [15].

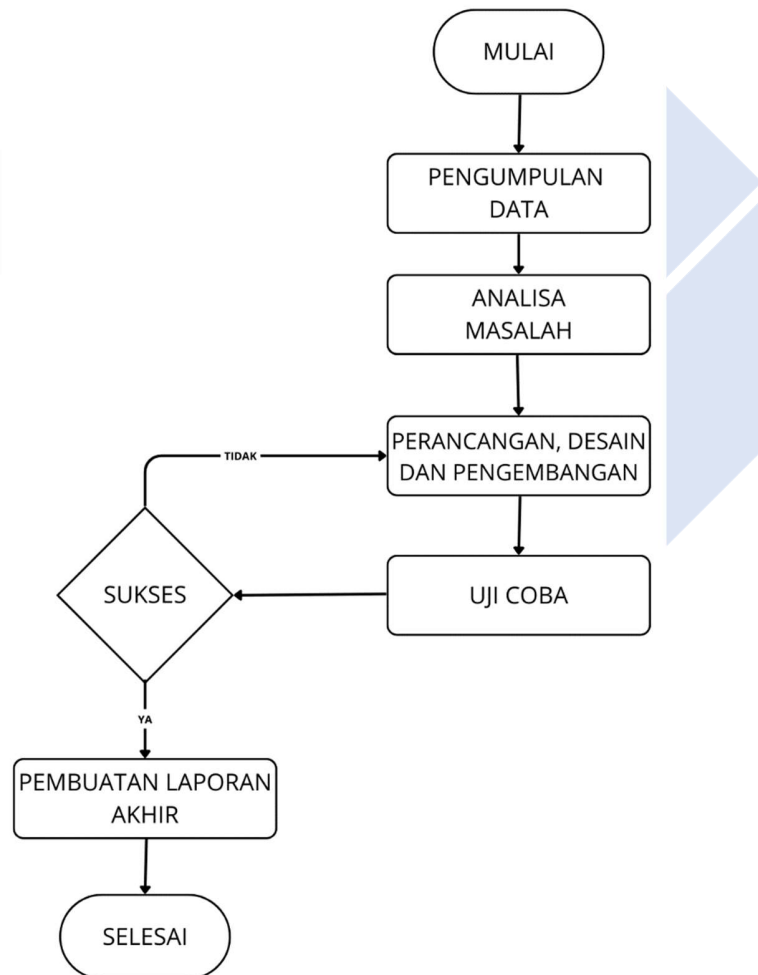
Kemajuan konektivitas internet sudah meluas melampaui perangkat tradisional seperti ponsel pintar dan komputer, kini mencakup semua perangkat elektronik. Hal ini memungkinkan terjadinya integrasi teknologi *Internet of Things*, memungkinkan perangkat berkomunikasi satu sama lain dan terhubung ke internet buat berbagai aplikasi. Contohnya ialah penggunaan teknologi *Internet of Things* pada perangkat pemberian makan kucing otomatis di rumah tangga, yang dikontrol melewati aplikasi *Inventor* dan *Firebase* buat meningkatkan efisiensi dan kenyamanan. [16].



BAB III

METODE PELAKSANAAN

Secara umum proyek akhir yang berjudul “ Wadah Makanan dan Minuman Kucing Berbasis *Internet Of Thing* (IOT) “ ini melakukan pemberian makanan dan minuman kucing secara otomatis yang bisa di monitoring, pembuatan alat ini memiliki beberapa tahapan yakni :



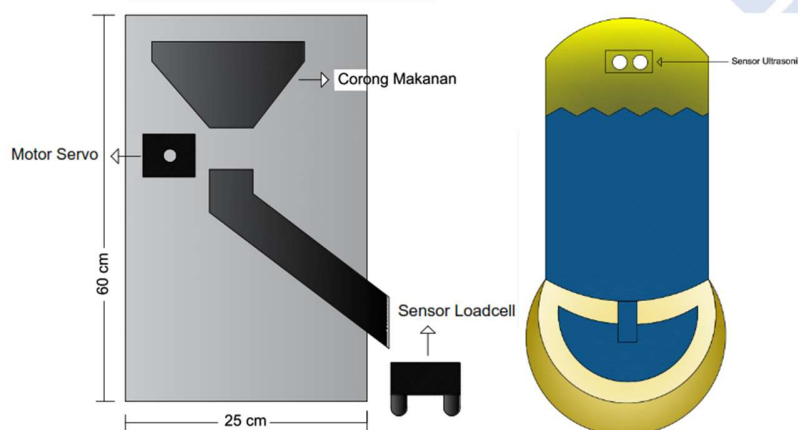
Gambar 3. 1 *Flowchart* Pembuatan Alat

3.1 Survey Dan Pengumpulan Data

Sebuah studi komprehensif dilangsungkan terhadap individu yang bertanggung jawab merawat kucing, buat memahami tantangan yang mereka hadapi dalam menyediakan makanan dan perawatan menyeluruh buat kucing peliharaan mereka. Selain mengumpulkan wawasan dari pemilik kucing, data juga dikumpulkan dari berbagai sumber seperti jurnal, supervisor, dan sumber online. Penelitian ini bertujuan buat memperoleh pemahaman lebih dalam tentang kebutuhan dan persyaratan yang terlibat dalam menjaga kehidupan kucing yang sehat dan bahagia.

3.2 Perancangan *Hardware* Konstruksi Alat

Saat merancang alat buat menyediakan makanan dan minuman kucing, pertimbangan wajib diberikan pada bentuk wadah dan kotak kontrol. Selain itu, pendekatan yang bijaksana wajib diambil ketika merancang sistem kontrol buat mengoperasikan motor *servo*.

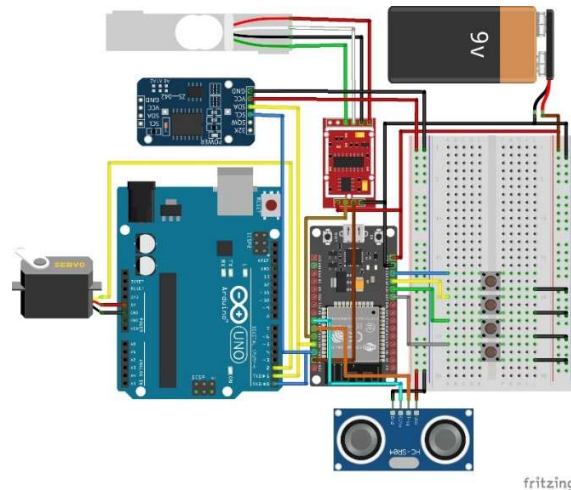


Gambar 3. 2 Desain Konstruksi Alat

3.3 Perancangan *Hardware* Elektrik

Saat membuat hardware elektrik, pertimbangan yang cermat dibutuhkan untuk memilih komponen yang khusus untuk perangkat yang dimaksud, termasuk *arduino uno*, *ESP32*, sensor *ultrasonic*, *load cell*, *real time clock*, motor *servo*, dan layar *LCD*

12 x 6. Pada gambar dibawah menjelaskan pemasangan pin – pin yang menghubungkan semua komponen.



Gambar 3. 3 *Desain Hardware Elektrik*

3.4 Pembuatan *Hardware* dan *Software Elektrik*

Pembuatan *hardware* dan *software* elektrik adalah tahapan implementasi dari rancangan yang telah dibuat sebelumnya.

3.4.1 Pembuatan *Hardware Elektrik*

Tahapan pembuatan *hardware* elektrik dimulai dari memasang komponen – komponen kedalam box kontrol seperti rancangan sebelumnya. Buat pembuatan *hardware* elektrik secara mekanik yakni :

1. Menggabungkan komponen – komponen menjadi satu kedalam box kontrol seperti *arduino uno*, *ESP32*, *real time clock*.
2. Pemasangan motor *servo* pada tutup tanki penyimpanan makanan utama.
3. Pemasangan *load cell* pada alur turunnya makanan buat mencapai pintu utama keluarnya makanan.
4. Pemasangan sensor *ultrasonik* buat monitoring persediaan air didalam tanki.

3.4.2 Pembuatan *Software Elektrik*

Tahapan pembuatan *software* elektrik ini dilangsungkan dengan menyesuaikan rancangan yang sudah dilangsungkan sebelumnya. Tahapan ini mencakup pengembangan *software* mikrokontroler.

1. Pemograman *ESP32* yang dikomunikasikan ke *arduino uno* buat memonitoring air ditanki dan mengatur waktu buat mengatur jam makan.
2. Pemograman *ESP32* buat menampilkan perubahan data dari *load cell* dan *real time clock* yang di tampilkan ke LCD 12 x 6.
3. Pemograman *arduino uno* buat mengaktifkan motor servo sebagai penggerak pintu buat turunnya makanan.

3.5 Pengujian *Hardware dan Software Elektrik*

Sesudah pengembangan komponen perangkat keras dan perangkat lunak, langkah selanjutnya melibatkan pengujian komponen tersebut dalam wadah makanan dan minuman kucing yang dilengkapi IoT. Fase pengujian ini memiliki tujuan penting buat menilai fungsionalitas sistem pengendalian dan pemantauan, memastikan keselarasan dengan konseptualisasi awal.

3.5.1 Pengujian *Hardware Elektrik*

1. Pengujian *arduino uno* dengan sensor *load cell*, *RTC*, dan LCD 16 x 2 buat mengetahui berat makanan kucing dan menampilkan keterangan berat dan waktu di LCD bisa berfungsi dengan baik.
2. Pengujian *arduino uno* dengan motor *servo* buat mengontrol pintu buka tutup wadah makanan kucing.
3. Pengujian sensor *ultrasonik* dengan *Arduino uno* buat mengetahui persediaan air ditanki.
4. Pengujian *arduino uno* yang di komunikasi serial dengan *ESP32* sebagai media yang mengatur dan mengontrol melewati *smartphone*.

3.5.2 Pengujian *Software Elektrik*

Pengujian *software* pada proyek akhir ini dilangsungkan dengan mengatur waktu pada 3 waktu, saat sudah diwaktu yang sudah diatur akan langsung menjalankan motor *servo*, dan turun ke wadah makanan yang dibawahnya ada sensor *load cell*. Sensor *ultrasonic* buat mengontrol air ditanki utama tempat minum.

3.6 Pembuatan Laporan Akhir

Berbagai tahapan dalam pembuatan laporan tugas akhir ini mencakup seluruh aspek proyek, dimulai dari pendahuluan, landasan teori, strategi pelaksanaan, pembahasan, dan pada akhirnya berujung pada kesimpulan dan rekomendasi. Sesudah pengembangan komponen perangkat keras dan perangkat lunak, langkah selanjutnya melibatkan pengujian komponen tersebut dalam wadah makanan dan minuman kucing yang dilengkapi *Internet of Things*. Fase pengujian ini memiliki tujuan penting buat menilai fungsionalitas sistem pengendalian dan pemantauan, memastikan keselarasan dengan konseptualisasi awal.

BAB IV

PEMBAHASAN

Bab 4 mendalami proses rumit dalam mengembangkan proyek akhir, menggambarkan kesejajaran dengan metode teliti yang dipakai dalam membuat dan mengevaluasi wadah makanan dan minuman kucing. Bagian selanjutnya memberikan rincian tentang bab pembahasan:

1. Deskripsi Alat
2. Perancangan dan pembuatan *hardware* wadah makanan dan minuman kucing.
3. Perancangan dan pembuatan *hardware* elektrik wadah makanan dan minuman kucing.
4. Pengujian *hardware* elektrik wadah makanan dan minuman kucing.
5. Perancangan dan pembuatan *software Internet Of Things*.
6. Pengujian keseluruhan berbasis *Internet Of Things*.

4.1 Deskripsi Alat

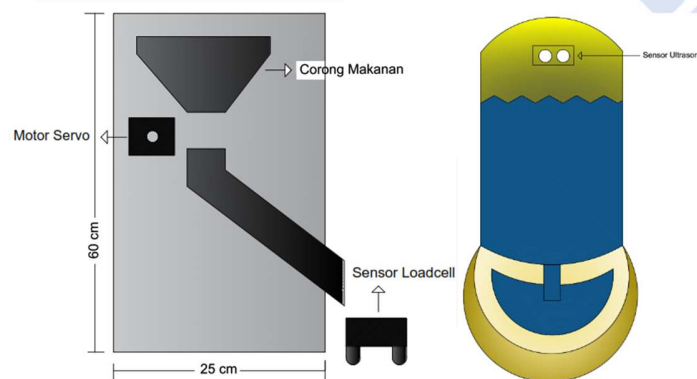
Wadah makanan dan minuman kucing inovatif ini memanfaatkan teknologi *Internet Of Things* buat memberi makan kucing secara otomatis pada waktu yang dijadwalkan. Proses pemberian makan berhenti sesudah loadcell memperlihatkan makanan sudah mencapai total yang diinginkan. Pada waktu makan ke-2 dan ke-3, jadwal pemberian makan dilanjutkan berlandaskan program yang sudah ditentukan sebelumnya. Dengan desainnya yang ergonomis, wadah ini memastikan akses mudah bagi kucing buat memperoleh makanan dan airnya tanpa rasa tidak nyaman. Selain itu, wadah ini dilengkapi sistem pengingat buat memastikan pemilik kucing tidak lupa memberi makan hewan peliharaannya, pada akhirnya airnya tidak mengering.

4.2 Perancangan dan Pembuatan *Hardware* Mekanik Wadah Makanan dan Minuman Kucing.

Dalam desain dan pembuatan perangkat keras buat wadah makanan dan minuman kucing, proses perencanaan yang cermat diterapkan buat menempatkan komponen secara strategis berlandaskan fungsi yang dimaksudkan. Perencanaan yang matang yang dipakai dalam proyek ini berfungsi sebagai cetak biru buat membuat perangkat keras wadah makanan dan minuman kucing sesuai dengan desain awal..

4.2.1 Perancangan *Hardware* Mekanik Wadah Makanan dan Minuman Kucing

Pada desain dan pembuatan *hardware* untuk wadah makanan dan minuman kucing, proses perancangan yang baik diterapkan untuk menempatkan komponen secara strategis berlandaskan fungsi yang dimaksudkan. Perencanaan matang yang dipakai dalam proyek ini berfungsi sebagai media untuk membuat *hardware* wadah makanan dan minuman kucing sesuai dengan desain awal.



Gambar 4. 1 Wadah Makanan dan Minuman Kucing

Mengacu pada gambar 4.1 yakni gambar wadah makanan dan minuman kucing. Desain tersebut memiliki tinggi yakni 60 cm dengan lebar 25cm. Dapat dilihat pada desain gambar makanannya terdapat beberapa komponen yang berada di dalamnya yaitu terdapat motor *servo*, corong makanan kucing, dan terdapat sensor *load cell*, yang mana sensor *load cell* tersebut berfungsi untuk mengukur berat dari makanan kucing

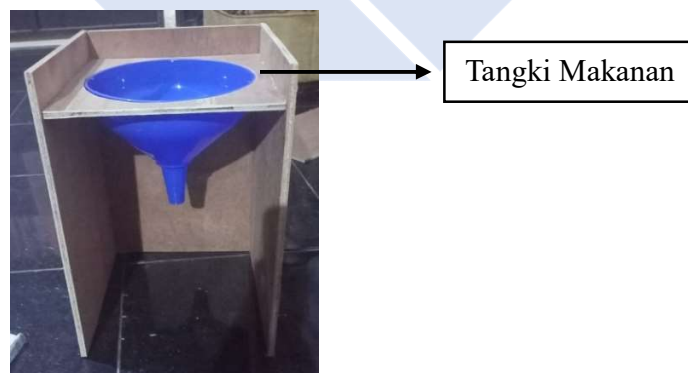
yang berada di dalam corong, pada saat berat makanan kucing tersebut sudah mencapai berat 50 gram maka secara otomatis motor *servo* akan menutup corong makanan. Sedangkan pada desain minuman terdapat sensor *ultrasonik* pada bagian atas desainnya, dimana sensor *ultrasonik* pada desain minuman tersebut berfungsi untuk mendeteksi jumlah persentasi air yang berada di dalam tangki air.

4.2.2 Pembuatan *Hardware* Mekanik Wadah Makanan dan Minuman Kucing

Kerangka konstruksi wadah makanan dan minuman ini dibuat dengan cara menggabungkan triplek yang sudah dipotong – potong sesuai dengan ukuran yang diperlukan. Dinding – dinding ini direkatkan menggunakan paku dan baut sehingga penyambungan setiap dinding menjadi lebih kokoh. Wadah makanan ini memiliki satu ruangan yang berukuran tinggi 60 cm dan lebar 25 cm. Pada ruangan tangki ini memiliki dua pintu, yang pertama pintu disamping sebagai ruangan kontrol untuk motor *servo* dan penutup corong tangki.

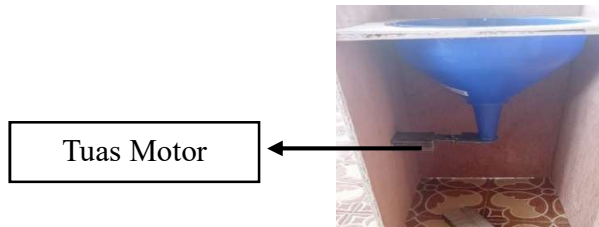
Kemudian dipasang komponen – komponen sesuai dengan tempatnya masing – masing. Pada wadah makanan dipasang 1 buah motor *servo* dan 1 buah *load cell*. Pada wadah minuman dipasangkan sensor *ultrasonik*.

- Pembuatan Wadah Makanan



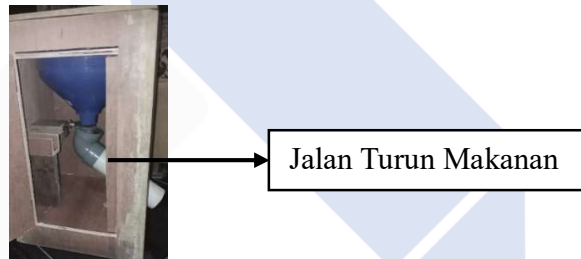
Gambar 4. 2 Pembuatan Tangki Makanan

Pada gambar 4.2 merupakan gambar dari proses pembuatan tangki makanan yang berfungsi sebagai tempat untuk menyimpan makanan kucing berkapasitas sebesar 1kg.



Gambar 4. 3 Pembuatan Tuas Motor *Servo*

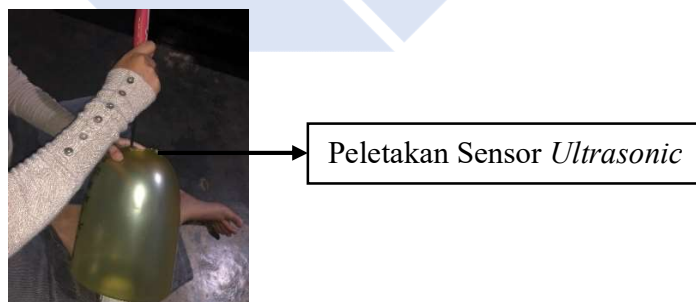
Pada gambar 4.3 merupakan gambar dari proses pembuatan tuas motor *servo* yang berfungsi buat membuka dan menutup tangki makanan kucing.



Gambar 4. 4 Pembuatan Jalan Turun Makanan

Pada gambar 4.4 merupakan proses pembuatan jalan turun makanan kucing menuju ke piring sensor *load cell*.

- Pembuatan Wadah Minuman



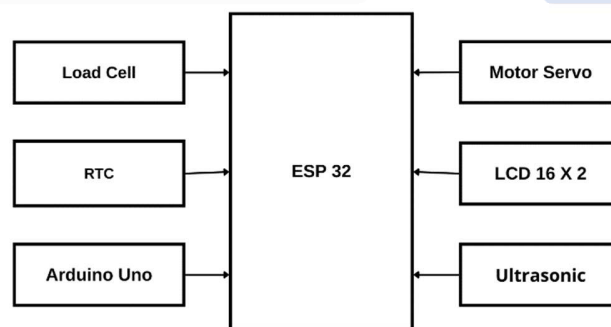
Gambar 4. 5 Pembuatan Tempat Peletakan Sensor *Ultrasonic*

Pada gambar 4.5 merupakan proses pembuatan tempat peletakan sensor *ultrasonic* dengan cara dilakukan pembolongan pada bagian atas tangki minuman kucing.

4.3 Perancangan dan Pembuatan *Hardware* Elektrik Wadah Makanan dan Minuman Kucing

Setelah merancang komponen elektrik dengan baik untuk sistem kontrol dalam wadah makanan dan minuman kucing, langkah selanjutnya adalah membuat diagram blok terperinci untuk secara visual mewakili sifat saling berhubungan dari setiap komponen.

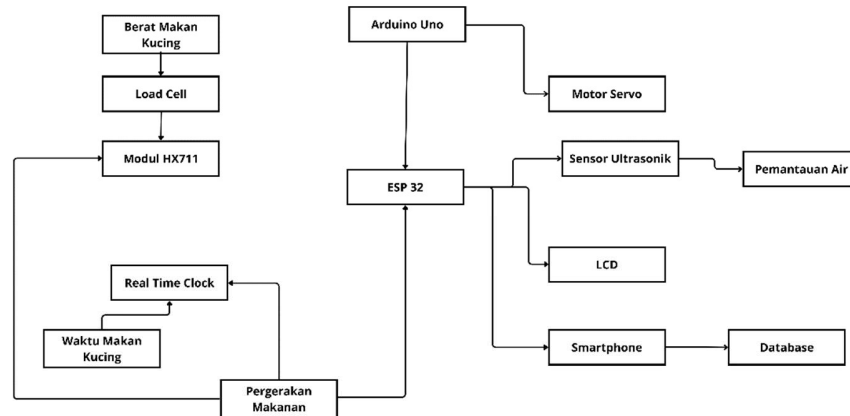
Langkah selanjutnya melibatkan pembuatan rencana diagram blok yang baik, di mana setiap blok saling berhubungan. Diagram blok memiliki berbagai tujuan, seperti menyederhanakan penjelasan fungsi suatu alat, menganalisis operasi rangkaian, dan memfasilitasi deteksi kesalahan dalam proses konstruksi. Di bawah ini adalah diagram blok yang dirancang dengan baik yang dipakai dalam tahap perencanaan proyek wadah makanan dan minuman kucing.



Gambar 4. 6 Diagram Blok *Elektrikal*

Gambar 4.6 merupakan gambar dari blok diagram *hardware* elektrik. Pada blok diagram tersebut terdapat *ESP 32*, *load cell*, *RTC*, *arduino uno* yang dijadikan sebagai inputan dari perancangan *hardware* elektrik, sedangkan motor *servo*, *LCD 16x2*, dan *ultrasonic* yang dijadikan sebagai *output* dari perancangan *hardware* elektrik.

Setelah blok diagram dibuat *flowchart* dibawah ini berfungsi sebagai pacuan jalur komponen elektrikal pada proyek wadah makanan dan minum kucing. *Flowchart* dibawah ini meliputi komponen *inputan*, proses dan *output*.



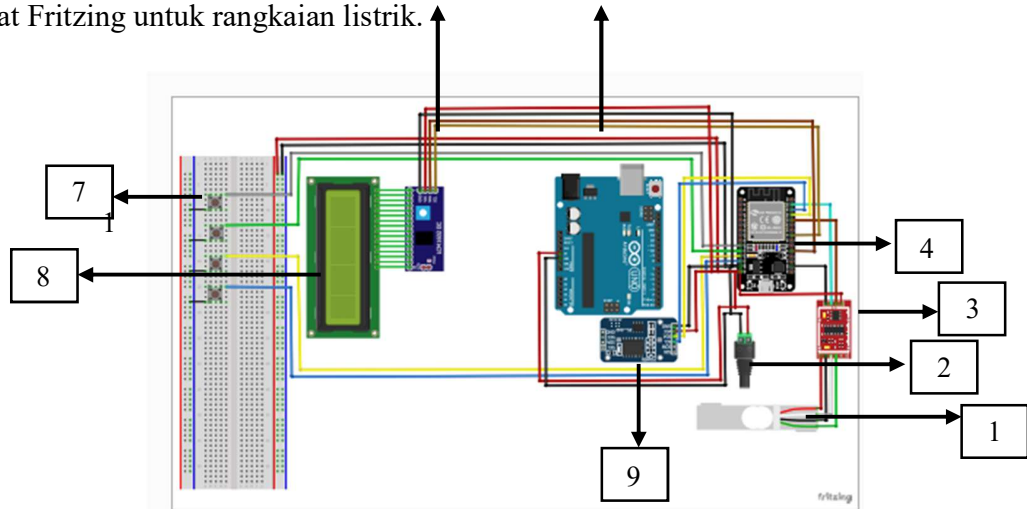
Gambar 4. 7 *Flowchart Elektrikal*

Gambar 4.3 merupakan gambar dari *flowchart* komponen. Gambar tersebut mempunyai *ESP 32* yang dijadikan sebagai otak dari segala komponen yang digunakan dan juga untuk program yang dikerjakan. Jadi pada saat sensor *load cell* mendeteksi berat makanan kucing yang sudah melebihi 50 gram maka motor *servo* akan secara otomatis menutup corong makanan tersebut. Pada bagian sebelah kanan terdapat sensor *ultrasonik* yang digunakan sebagai alat untuk mendeteksi persentase air yang berada di dalam tangki minuman kucing. Untuk *real time clock* bermaksud untuk mengatur timeline jam dari setiap kucing yang akan makan. Aplikasi yang dibuat menggunakan *MIT app inverter* yang akan di instal di *smartphone*, dimana dalam aplikasi tersebut terdapat persentase air, berat makanan kucing, dan terdapat settingan makanan kucing yang terbagi menjadi 3 bagian waktu.

4.3.1 Perancangan *Hardware Elektrik* Wadah Makanan dan Minuman Kucing

Tahap ini merupakan tahap perencanaan kerangka pengendalian yang mampu mengendalikan pengendalian terhadap pemegang makanan dan minuman. Pada perencanaan ini terdapat dua bagian penting yaitu input informasi dari sensor *load cell* dan sensor *ultrasonik*. *ESP32* kemudian akan memproses data dari ketiga sensor

tersebut untuk mengendalikan motor *servo*. Berikut ini adalah skema pengkabelan yang dibuat Fritzing untuk rangkaian listrik.

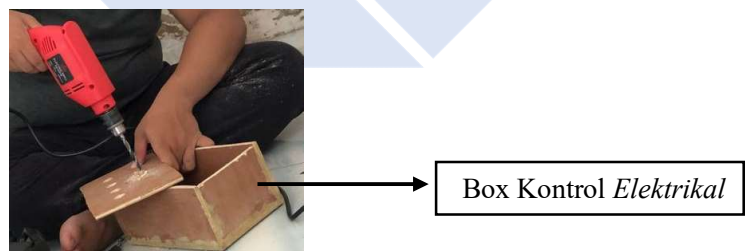


Gambar 4. 8 Skema Perkabelan Rangkaian *Elektrikal*

Keterangan :

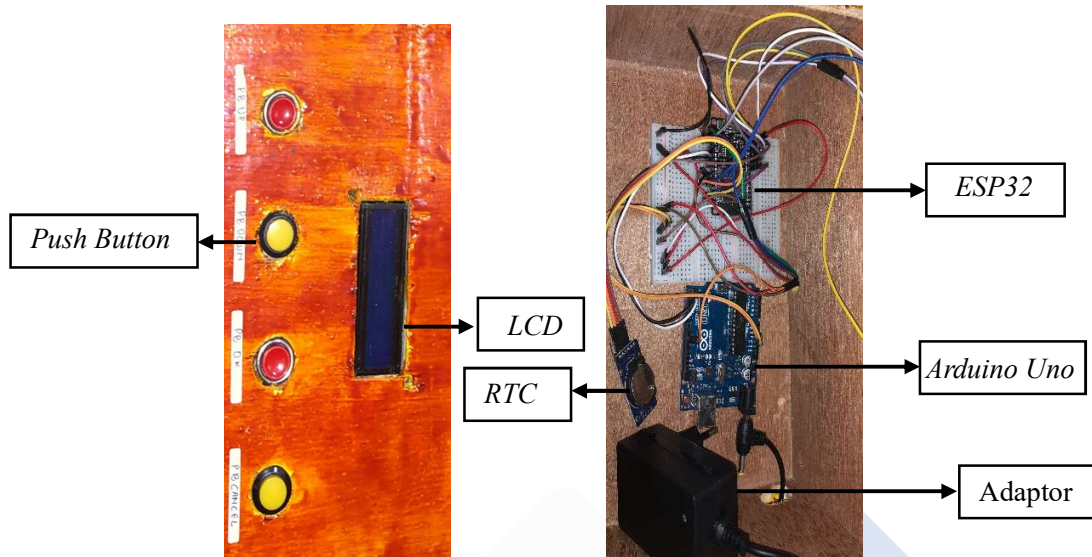
- | | | |
|-----------------------|-----------------------|-----------------------|
| 1. <i>Load Cell</i> | 4. <i>ESP32</i> | 7. <i>Push Button</i> |
| 2. <i>Adaptor</i> | 5. <i>Arduino Uno</i> | 8. <i>LCD 16X2</i> |
| 3. <i>Modul HX711</i> | 6. <i>Modul I2C</i> | 9. <i>RTC</i> |

4.3.2 Pembuatan *Hardware Elektrikal* Wadah Makanan dan Minuman Kucing



Gambar 4. 9 Pembuatan *Box Kontrol Elektrikal*

Pembuatan box kontrol ini berfungsi sebagai tempat untuk peletakan komponen dan sebagai kontrol manual.



Gambar 4. 10 Box Kontrol *Elektrikal*

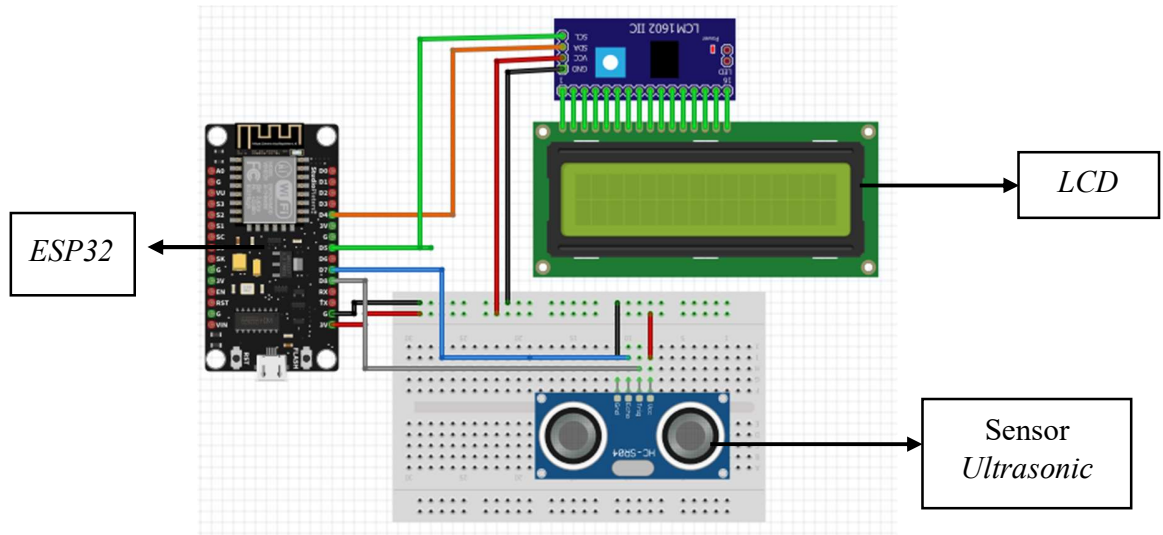
Pada gambar 4.3 terdapat box kontrol *elektrikal* yang memiliki perakitan komponen terdiri dari *ESP32*, *arduino uno*, *RTC* dan adaptor sedangkan pada tutup box kontrol terdiri dari 4 buah *push button* dan *LCD*.

4.4 Pengujian *Hardware Elektrikal* Wadah Makanan dan Minuman Kucing

Pada pengujian *hardware elektrikal* ini berfungsi untuk menguji apakah komponen yang dipakai bisa berfungsi dengan baik. Komponen yang dipakai yakni ialah sensor *load cell* dan sensor *ultrasonic*.

4.4.1 Perancangan Sensor *Ultrasonic*

Pada tahap ini dilakukan perancangan skema rangkaian dengan cara menghubungkan pin-pin pada sensor *ultrasonic*, *liquid crystal display* dan *ESP32* pada aplikasi *Fritzing*.



Gambar 4. 11 Skema Pengkabelan *ESP32*, Sensor *ultrasonic* dan *LCD*

Dapat dilihat pada gambar 4.11, penjelasan dari pin *ESP32*, *LCD* dan sensor *ultrasonic* adalah sebagai berikut :

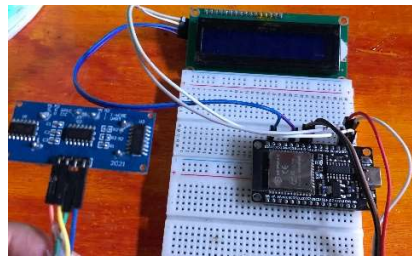
- Pin *VCC* ke pin 3.3v *ESP32*
- Pin *GnD* ke pin *GnD ESP32*
- Pin *Trig ultrasonic* ke pin D8 *ESP32*
- Pin *Echo ultrasonic* ke pin D7 *ESP32*
- Pin *SCL LCD* ke pin D5 *ESP32*
- Pin *SDA LCD* ke pin D4 *ESP32*

Ini merupakan point – point dari pin penghubung antara *ESP32*, *LCD* dan sensor *ultrasonic*.

4.4.2 Pembuatan Sensor *Ultrasonic*

Pada tahap ini dilakukan pembuatan rangkaian dengan cara menghubungkan pin-pin pada sensor *ultrasonic*, *liquid crystal display* dan *ESP32*. Pin *VCC* ke pin 3.3v *ESP32*, pin *GnD* ke pin *GnD ESP32*, pin *Trig ultrasonic* ke pin D8 *ESP32*, pin *Echo*

ultrasonic ke pin D7 *ESP32*, pin *SCL LCD* ke pin D5 *ESP32*, pin *SDA LCD* ke pin D4 *ESP32*.



Gambar 4. 12 Pembuatan Sensor *Ultrasonic*

4.4.3 Pengujian Sensor *Ultrasonic*

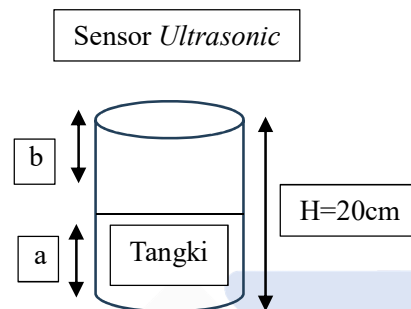
Untuk melakukan pengujian Sensor Ultrasonic, perlu menyiapkan blok diagram guna mempermudah dalam proses pengujian. Berikut merupakan Blok Diagram dari pengujian sensor Ultrasonic.



Gambar 4. 13 Blok Diagram Pengujian Sensor *Ultrasonic*

Gambar 4.13 Merupakan gambar blok diagram pengujian *ultrasonic*, mengacu pada gambar tersebut yang merupakan langkah awal dalam proses pengujian sensor dimana *ultrasonic* dijadikan sebagai *inputan*, Sedangkan *ESP32* dijadikan sebagai perangkat proses dalam pengujian, *LCD* yang dijadikan sebagai *output*. Pada saat sensor *ultrasonic* aktif maka *ESP32* sebagai perangkat proses akan memproses data

yang dideteksi dan ditangkap oleh sensor *ultrasonic* dan meneruskan hasil pemrosesan tersebut ke *output* yakni *LCD*. Dengan itu *LCD* akan menampilkan ketinggian air. Berikut cara pengujian yang dilakukan untuk menguji sensor *ultrasonic*.



Tabel 4. 1 Pengujian Ketinggian Air dan Sensor *Ultrasonic*

No	Ketinggian Tangki	Hasil Sensor <i>Ultrasonic</i>	Hasil (a)
1	20cm	17cm	3 cm
2	20cm	12cm	8 cm
3	20cm	7cm	13 cm

Rumus ketinggian tangki:

$$\text{Rumus: } a = h - b$$

- Perhitungan ketinggian tangki 1 : $a = 20 - 17 = 3\text{cm}$
- Perhitungan ketinggian tangki 2 : $a = 20 - 12 = 8\text{cm}$
- Perhitungan ketinggian tangki 3 : $a = 20 - 7 = 13\text{cm}$

Perhitungan ini bertujuan untuk menghitung ketinggian air dan sensor *ultrasonic*.

- Pengujian sensor *ultrasonic* pada tangki air yang diukur secara manual.



Gambar 4. 14 Pengujian Ketinggian Air Secara Manual

- Hasil pengujian sensor *ultrasonic* yang ditampilkan pada *LCD*.



Gambar 4. 15 Hasil Pengujian Menggunakan Sensor *Ultrasonic*

Tabel 4. 2 Hasil Pengujian Sensor *Ultrasonic*

No	Ketinggian air pada <i>ultrasonic</i>	Ketinggian air pada tangki	Selisih	% Error
1	50 %	51 %	1 %	1,9 %
2	62,5 %	62 %	0,5 %	0,8%
3	77 %	77 %	0 %	-

Rumus persentase error, dimana:

$$\text{Persentase error} = \left| \frac{\text{Alat Ukur} - \text{Sensor}}{\text{Alat Ukur}} \right| \times 100 = \dots \%$$

Alat Ukur = Hasil pengukuran pada timbangan

Sensor = Hasil pengukuran pada sensor *load cell*

Berikut ini perhitungan secara teori: Perhitungan persentase error pada tabel 4.2 pengujian sensor ultrasonic dengan alat ukur:

- Persentase error pengukuran ke-1 = $\left| \frac{51-50}{51} \right| \times 100 = 1,9\%$
- Persentase error pengukuran ke-2 = $\left| \frac{62-6,5}{62} \right| \times 100 = 0,8\%$
- Persentase error pengukuran ke-3 = $\left| \frac{77-77}{77} \right| \times 100 = 0\%$

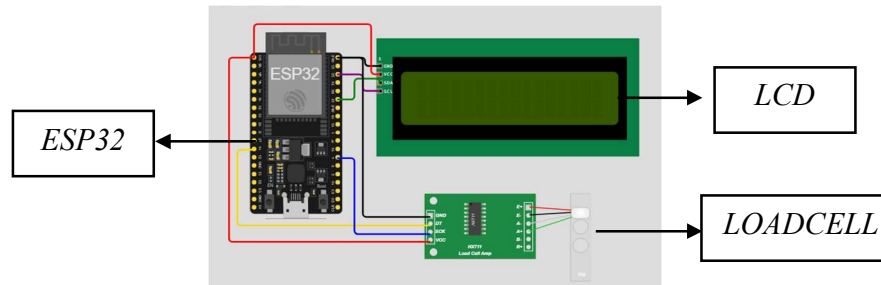
Kesimpulan dari 3 pengujian yang memiliki hasil yang berbeda beda dikarenakan air memiliki permukaan yang tidak rata. Dari hasil pengujian ini terlihat sensor mendeteksi sesuai dengan jarak yang dilangsungkan dalam percobaan. Tetapi sensor *ultrasonic* memiliki selisih yang berbeda tergantung pada permukaan air saat pengujian. Pada proses mengetahui apakah 100% itu tepat di 4 liter, penulis memiliki perhitungan yakni:

$$\begin{aligned}
 &= \frac{\text{Jumlah Bagian} \times \text{Jumlah Total}}{100} \\
 &= \frac{90 \times 4.000}{100} \\
 &= 3.600 \text{ ml}
 \end{aligned}$$

Jadi untuk presentase air di 90% itu 3.600 ml. Pada setiap 10% itu air akan berkurang sekitar 400 mili. Pada pengujian kali ini penulis menggunakan 90% menjadi patokan untuk mengetahui berapa perhitungannya. Kesimpulannya sensor *ultrasonic* dapat digunakan dengan baik karena selisih yang tidak terlalu jauh dengan ketinggian air yang diperhitungkan secara manual.

4.4.4 Perancangan Sensor *Load Cell*

Dengan menghubungkan pin pada sensor *load cell*, *liquid crystal display*, dan *ESP32* pada aplikasi Fritzing pada titik ini maka akan tercipta skema rangkaian.



Gambar 4. 16 Skema Pengkabelan *ESP32*, Sensor *Load Cell* dan *LCD*

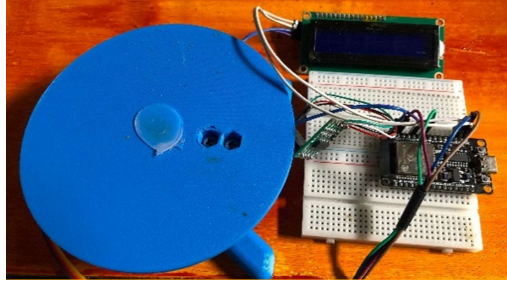
Dapat dilihat pada gambar 4.17, penjelasan dari pin *ESP32*, *LCD* dan sensor *load cell* adalah sebagai berikut :

- Pin *VCC* ke pin *VCC ESP32*
- Pin *GnD* ke pin *GnD ESP32*
- Pin *DT* ke pin 14 *ESP32*
- Pin *SCK* ke pin 4 *ESP32*
- Pin *SDA* ke pin D4 *ESP32*
- Pin *SCL* ke pin D5 *ESP32*

Ini merupakan point – point dari pin penghubung antara *ESP32*, *LCD* dan sensor *load cell*.

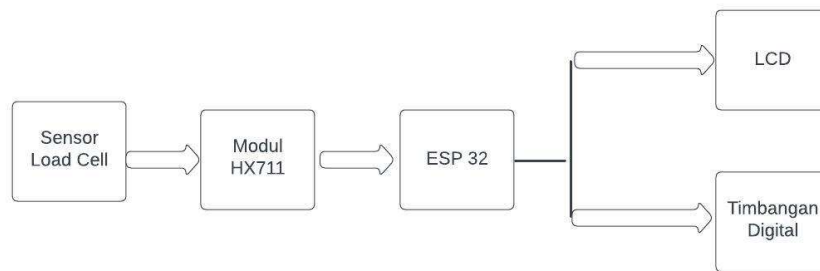
4.4.5 Pembuatan Sensor *Load Cell*

Pin sensor dihubungkan pada titik ini untuk membuat sirkuit *load cell*, *liquid crystal display* dan *ESP32*. Pin *VCC* ke pin 3.3v *ESP32*, pin *GnD* ke pin *GnD ESP32*, pin *DT load cell* ke pin D21 *ESP32*, pin *SCK load cell* ke pin D22 *ESP32*, pin *SCL LCD* ke pin D5 *ESP32*, pin *SDA LCD* ke pin D4 *ESP32*.



Gambar 4. 17 Pembuatan Sensor *Load Cell*

Untuk melakukan pengujian sensor *load cell*, perlu menyiapkan blok diagram guna mempermudah dalam proses pengujian. Berikut merupakan Blok Diagram dari pengujian sensor *load cell*.



Gambar 4. 18 Blok Diagram Pengujian Sensor *Load Cell*

Gambar 4.19 Merupakan blok diagram pengujian sensor *load cell*, mengacu pada gambar tersebut yang merupakan langkah awal dalam proses pengujian sensor dimana sensor *load cell* dan modul *HX711* yang dijadikan sebagai *inputan*, Sedangkan *ESP32* dijadikan sebagai Perangkat proses dalam pengujian, *LCD* dan *Timbangan Digital* yang dijadikan sebagai *output*. Pada saat sensor *load cell* dan modul *HX711* aktif maka *ESP32* sebagai perangkat proses akan memproses data yang dideteksi dan ditangkap oleh sensor *load cell* dan modul *HX711* dan meneruskan hasil pemrosesan tersebut ke *output* yakni *LCD* dan timbangan digital. Dengan itu *LCD* akan menampilkan berat makanan pelet kucing sedangkan timbangan digital akan memberikan perbandingan berat dari makanan kucing tersebut. Berikut merupakan tabel hasil pengujian.

Tabel 4. 3 Hasil Pengujian Sensor *Load Cell*

No	Berat makanan pada <i>load cell</i>	Berat makanan timbangan digital	Selisih	% Error
1	99 gram	100 gram	1 gram	1 %
2	100 gram	102 gram	2 gram	1,9 %
3	150 gram	153 gram	3 gram	1,9 %

Rumus persentase error, dimana:

$$\text{Persentase error} = \left| \frac{\text{Alat Ukur} - \text{Sensor}}{\text{Alat Ukur}} \right| \times 100 = \dots \%$$

Alat Ukur = Hasil pengukuran pada timbangan

Sensor = Hasil pengukuran pada sensor *load cell*

Berikut ini perhitungan secara teori: Perhitungan persentase error pada tabel 4.3 pengujian sensor *load cell* dengan timbangan digital :

- Persentase error pengukuran ke-1 = $\left| \frac{100-99}{100} \right| \times 100 = 1 \%$
- Persentase error pengukuran ke-2 = $\left| \frac{102-100}{102} \right| \times 100 = 1,9 \%$
- Persentase error pengukuran ke-3 = $\left| \frac{153-150}{153} \right| \times 100 = 1,9 \%$

Kesimpulan yang didapat adalah *load cell* dapat berfungsi dengan baik dan dapat digunakan. Antara *load cell* dan timbangan digital memiliki selisih 2 gram dan memiliki persentase error yang dihitung secara teori memiliki rata – rata 1,9%.

4.4.6 Pengujian Sensor *Load Cell*

- Hasil pengujian sensor *load cell* yang ditampilkan pada *LCD*.



Gambar 4. 19 Hasil Pengujian Menggunakan Sensor *Load Cell* pada *LCD*

- Pengujian sensor *load cell* pada makanan yang ditimbang menggunakan timbangan digital.



Gambar 4. 20 Hasil Pengujian Berat Makanan Menggunakan Timbangan Digital

- Hasil pengujian berat makanan dengan format tabel.

Tabel 4. 4 Hasil Pengujian Berat Makanan Secara Manual dan Sensor *Load Cell*

No	Berat makanan pada <i>load cell</i>	berat makanan secara manual	Selisih	% Error
1	55 gram	57 gram	2 gram	3,5 %
2	57 gram	59 gram	2 gram	3,3 %
3	58 gram	60 gram	2 gram	3,4 %

Rumus persentase error, dimana:

$$\text{Persentase error} = \left| \frac{\text{Alat Ukur} - \text{Sensor}}{\text{Alat Ukur}} \right| \times 100 = \dots \%$$

Alat Ukur = Hasil pengukuran pada timbangan

Sensor = Hasil pengukuran pada sensor *load cell*

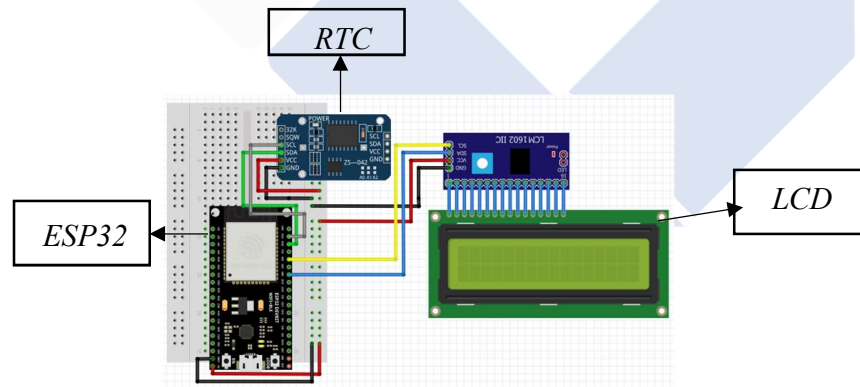
Berikut ini perhitungan secara teori: Perhitungan persentase error pada tabel 4.4 pengujian sensor *load cell* dengan timbangan digital :

- Persentase error pengukuran ke-1 = $\left| \frac{57-5}{57} \right| \times 100 = 3,5 \%$
- Persentase error pengukuran ke-2 = $\left| \frac{59-57}{59} \right| \times 100 = 3,3 \%$
- Persentase error pengukuran ke-3 = $\left| \frac{60-58}{60} \right| \times 100 = 3,4 \%$

Kesimpulan yang didapat adalah *load cell* dapat berfungsi dengan baik dan dapat digunakan. Antara *load cell* dan timbangan digital memiliki selisih 2 gram dan memiliki persentase error yang dihitung secara teori memiliki rata – rata 3,3%.

4.4.7 Perancangan *Real Time Clock*

Dengan menghubungkan pin pada titik ini, skema rangkaian dirancang dan dibuat *real time clock*, *liquid crystal display* dan *ESP32* pada aplikasi *Fritzing*.



Gambar 4. 21 Skema Perkabelan *Real Time Clock*, *ESP32* dan *LCD*

Dapat dilihat pada gambar 4.22, penjelasan dari pin *ESP32*, *LCD* dan *real time clock* adalah sebagai berikut :

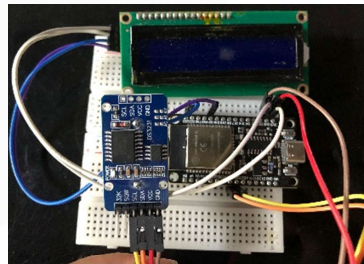
- Pin *VCC* ke pin *VCC ESP32*
- Pin *GnD* ke pin *GnD ESP32*

- Pin *SCL RTC* ke pin D9 *ESP32*
- Pin *SDA RTC* ke pin D10 *ESP32*
- Pin *SDA LCD* ke pin D4 *ESP32*
- Pin *SCL LCD* ke pin D5 *ESP32*

Ini merupakan point – point dari pin penghubung antara *ESP32*, *LCD* dan sensor *load cell*.

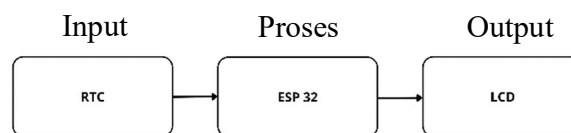
4.4.8 Pembuatan *Real Time Clock*

Menghubungkan pin pada titik ini akan menciptakan sirkuit pada *real time clock*, *liquid crystal display* dan *ESP32*. Pin *VCC* ke pin 3.3v *ESP32*, pin *GnD* ke pin *GnD* *ESP32*, pin *SCL RTC* ke pin D9 *ESP32*, pin *SDA RTC* ke pin D10 *ESP32*, pin *SCL LCD* ke pin D5 *ESP32*, pin *SDA LCD* ke pin D4 *ESP32*.



Gambar 4. 22 Pembuatan *Real Time Clock*

Untuk melakukan pengujian sensor *load cell*, perlu menyiapkan blok diagram guna mempermudah dalam proses pengujian. Berikut merupakan Blok Diagram dari pengujian *real time clock*.



Gambar 4. 23 Diagram Blok Pengujian *Real Time Clock*

Gambar 4.24 Merupakan blok diagram pengujian *real time clock*, mengacu pada gambar tersebut yang merupakan langkah awal dalam proses pengujian dimana *real*

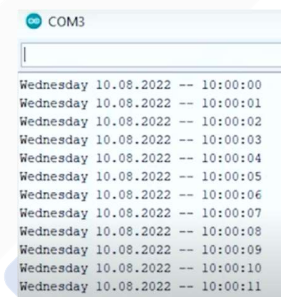
time clock yang dijadikan sebagai *inputan*, Sedangkan *ESP32* dijadikan sebagai perangkat proses dalam pengujian, dan *LCD* yang dijadikan sebagai *output*. Pada saat *real time clock* aktif maka *ESP32* sebagai perangkat proses akan memproses data yang dideteksi dan ditangkap oleh *real time clock* dan meneruskan hasil pemrosesan tersebut ke *output* yakni *LCD*. Dengan itu *LCD* akan menampilkan waktu.

4.4.9 Pengujian *Real Time Clock*

- Hasil pengujian *real time clock* yang ditampilkan pada *LCD*



Gambar 4. 24 Tampilan Pengujian *Real Time Clock*



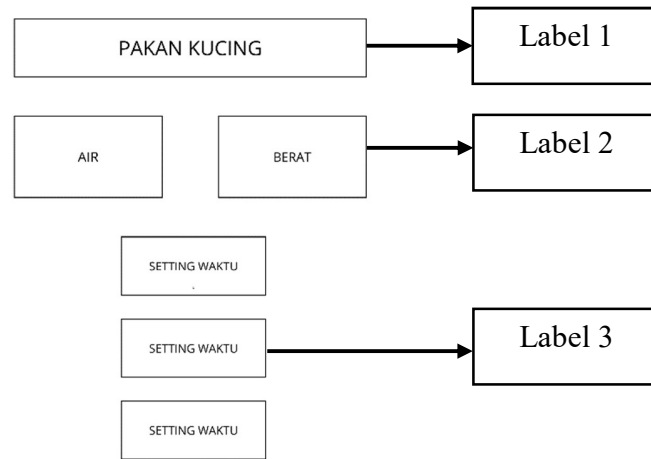
Gambar 4. 25 Tampilan Pada *Serial Monitor*

4.5 Perancangan dan Pembuatan *Software Internet Of Things*

Perancangan dan pembuatan *software internet of things* ini dibuat agar desain dari aplikasi inventor dapat digunakan dengan efisien.

4.5.1 Perancangan *Software Internet Of Things*

Saat membuat *software*, pertimbangan matang diberikan untuk menggunakan pengontrol *Arduino Uno* dan *ESP32* untuk semua komponen. *database* yang dipakai dalam proyek ini, yang disebut *firebase*, berfungsi sebagai platform untuk mengatur waktu secara efisien dan mengontrol makanan dan minuman menggunakan teknologi *Internet of Things*.



Gambar 4. 26 Perancangan *Software* MIT App Inventor

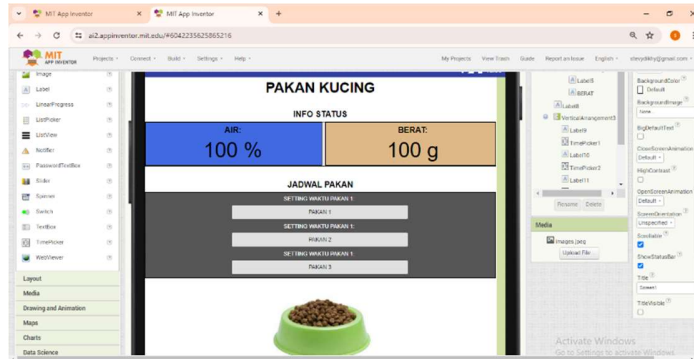
Keterangan dari perancangan aplikasi :

- Label 1 dan label 2 digunakan untuk tampilan teks. Apa pun yang ingin kita sertakan dalam desain aplikasi dapat ditulis ke dalam komponen ini.
- Label 3 digunakan buat mengatur waktu, agar dapat mengatur waktu perlu ditambahkan timepicker yang berfungsi agar waktu dapat tampil pada aplikasi.

Aplikasi ini sudah dirancang untuk sebuah sistem yang memungkinkan pengguna dengan mudah mengatur pengatur waktu agar makanan dapat keluar tiga kali sehari hanya dengan menekan tombol diaplikasi. *Software* ini juga memungkinkan tampilan dan penyesuaian pengatur waktu, serta memantau berat makanan dan ketinggian air di dalam tangki agar tidak kosong. Namun, sistem ini bergantung pada koneksi *Wi-Fi* yang sudah diprogram ke dalam *software* dan tidak bisa diubah tanpa mengubah pengaturan di dalam program. Untuk tampilannya dapat dilihat pada gambar 4.23.

4.5.2 Pembuatan *Software Internet Of Things*

Setelah mengetahui rancangan yang diinginkan selanjutnya adalah membuat dan mendesain tampilan aplikasi dengan semenarik mungkin.



Gambar 4. 27 Pembuatan *Software* MIT App Inventor

Pemrograman berikut untuk menghubungkan MIT App Inventor ke Firebase dapat dibuat pada menu blok, mengikuti pembuatan tampilan visual aplikasi pada menu designer.

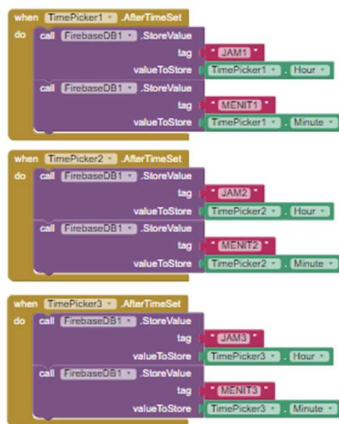


Program keluaran data *ESP32* untuk aplikasi MIT App Inventor. Aplikasi MIT App Inventor menampilkan program ini secara *real time*.

Gambar 4. 28 Program Penampil *Database* ke MIT App Inventor

Pemrograman juga dapat digunakan untuk menyimpan data dari MIT App Inventor ke *Firebase*, selain memanggil data dari *Firebase* ke MIT App Inventor. Salah satu

model pemrograman untuk menyimpan informasi dari MIT Application Innovator ke *Firebase* adalah sebagai berikut.



Program untuk mentransfer data dari MIT App Inventor ke *ESP32*. Apabila tombol setting waktu di atur waktu maka akan menyimpan nilai yang telah diatur menggunakan timepicker.

Gambar 4. 29 Program Penyimpanan Data dari *Firebase* ke MIT App Inventor

4.6 Pengujian Keseluruhan Berbasis *Internet Of Things*

Pada pengujian keseluruhan ini menggunakan *button* yang berada di aplikasi *inventor* untuk menjalankan program ini. Pada pengujian ini menggunakan jarak 10 meter agar hasil yang didapatkan dapat sesuai dengan yang diinginkan.

Disini penulis menggunakan 50 gram sebagai patokan untuk berat makanan kucing. akan tetapi kurang akurat dikarenakan adanya masalah dari *hardware*. Apabila makanan tidak berkurang maka pada waktu makan selanjutnya makanan tidak akan keluar agar makanan tidak terlalu banyak didalam piring makan kucing.



Gambar 4. 30 Tampilan Awal Aplikasi MIT App Inventor

Dapat dilihat pada gambar 4.26 merupakan tampilan awal aplikasi yang menampilkan presentasi air didalam tangki, jumlah berat makanan dan jadwal pakan yang dapat di atur sesuai dengan yang diinginkan. Pada saat jam yang di atur belum tercapai maka tampilan berat tidak akan berubah.

4.6.1 Hasil Pengujian Berbasis *Internet Of Things*

- Hasil pengujian yang ditampilkan pada aplikasi



Gambar 4. 31 Hasil Pengujian Keseluruhan pada Aplikasi MIT Inventor

- Hasil pengujian yang ditampilkan pada *LCD*



Gambar 4. 32 Hasil Pengujian Keseluruhan pada *LCD*

- Hasil pengujian sensor *loadcell* pada *hardware*



Gambar 4. 33 Hasil Pengujian Sensor *Load Cell* Pada *Hardware*

- Hasil pengujian yang ditimbang menggunakan timbangan digital



Gambar 4. 34 Hasil Pengujian Menggunakan Timbangan Digital

- Hasil pengujian keseluruhan berbasis *internet of things* dengan format tabel.

Tabel 4. 5 Hasil Pengujian Keseluruhan Berbasis *Internet Of Things*

No	Waktu makan	Persentase air	Pengujian pada sensor <i>load cell</i>	Pengujian pada timbangan digital	selisih	% error
1	08 : 10	51 %	55 gram	57 gram	2 gram	3,5 %
2	12 : 10	78 %	57 gram	59 gram	2 gram	3,3 %
3	16 : 10	60%	58 gram	60 gram	2 gram	3,4 %

Kesimpulannya hasil pengujian keseluruhan wadah makanan dan minuman kucing dapat dilihat pada tabel 4.5, pada tabel diatas terdapat 3 waktu makan sesuai dengan yang telah diuji cobakan. Pada presentase air memang menurun dengan tidak signifikan tergantung dari seberapa banyak kucing yang meminum air tersebut. Untuk berat makanan memang melebihi dari yang telah di program karena adanya masalah pada *hardware*. Pengaturan waktu pada pengujian memang diatur lebih 10 menit agar penulis dapat melihat jalannya motor servo dengan lebih teliti lagi. Untuk melihat perhitungan persentase error dapat dilihat pada halaman 26 disana penulis membuat perhitungan yang dapat mudah dipahami.

BAB V

KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil pengujian yang sudah dilangsungkan dalam proyek akhir dengan judul “Wadah Makanan dan Minuman Kucing Otomatis Berbasis *Internet Of Things*”. Bisa ditarik kesimpulan yakni:

1. Wadah makanan dan minuman kucing ini berbasis *internet of things* yang bisa dikontrol melalui aplikasi. Aplikasi ini memiliki menu buat bisa mengatur waktu, memonitoring total makanan dan total air di dalam tangki minuman.
2. Jumlah berat makanan yang ditimbang pada *load cell* belum sepenuhnya dapat bekerja dengan baik, hasil dari pengujian didapatkan persentase error sebesar 3%. Data diambil dari hasil pengujian pada tabel 4.4.
3. Sensor *ultrasonic* memiliki rata - rata persentase error sebesar 1%. Data diambil dari hasil pengujian pada tabel 4.2.

5.2 Saran

Berdasarkan hasil pengujian yang sudah dilangsungkan pada proyek akhir yang berjudul “Wadah Makan dan Minum Kucing Otomatis Berbasis *Internet Of Things*”. Didapatkan beberapa saran pada alat agar lebih baik lagi.

1. Hasil *load cell* yang dihasilkan kurang akurat karena adanya masalah pada hardware, tetapi persentase error yang didapatkan perbedaannya memang tidak terlalu jauh dari berat benda yang ditimbang.
2. Konstruksi terlalu besar untuk wadah makan kucing, sebaiknya bisa dibuat lebih kecil lagi.

DAFTAR PUSTAKA

- [1] Ayunita, R. (2016). Alat Pemberi Pakan dan Minum Kucing Terjadwal Otomatis Berbasis Mikrokontroller. Diss. Universitas Gadjah Mada, 2016.
- [2] Risanda, B. (2023). Rancang Bangun Otomatisasi Pemberi Pakan dan Minum Kucing Berbasis Internet Of Things (IoT). (Doctoral dissertation, Universitas Mercu Buana Jakarta).
- [3] Alfian, N. S. (2022). Otomasi Wadah Pakan Kucing Berbasis Mikrokontroller ESP32 WROOM. (Doctoral dissertation, Sekolah Tinggi Informatika dan Komputer Indonesia).
- [4] “Kucing,”.[Online]. Available: <https://id.wikipedia.org/wiki/Kucing>.
- [5] Levina. (2022, Mei 9). Retrieved from 6 Jenis Nutrisi Ini Wajib Ada Dalam Makanan Kucing: <https://www.ruparupa.com/blog/jenis-nutrisi-wajib-ada-dalam-makanan-kucing/>.
- [6] Sari, P. I. (2023, Oktober 2). *Takaran Makanan Kucing*. Retrieved from <https://helohealth.com/sehat/informasi-kesehatan/takaran-makan-kucing/>.
- [7] “Arduino Uno,”.[Online].Available: <https://id.wikipedia.org/wiki/Arduino>
- [8] Kali, M. M., Tarigan, J., & Louk, A. C. (2016). Sistem alarm kebakaran menggunakan sensor infra red dan sensor suhu berbasis arduino uno. *Jurnal Fisika: Fisika Sains Dan Aplikasinya*, 1(1), 25-31.
- [9] Nizam, M. N., Yuana, H., & Wulansari, Z. (2022). Mikrokontroler Esp 32 Sebagai Alat Monitoring Pintu Berbasis Web. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 6(2), 767-772.

- [10] Devitasari, R., & Kartika, K. P. (2020). Rancang bangun alat pemberi pakan kucing otomatis menggunakan mikrokontroler nodemcu berbasis internet of thing (IoT). *Antivirus: Jurnal Ilmiah Teknik Informatika*, 14(2), 152-164.
- [11] Irwan, H. (2017). Alat Ukur Berat dan Tinggi Sapi.
- [12] Dewi, C., Arthana, R., & Setemen, K. (2023). RANCANG BANGUN ALAT PAKAN KUCING RANCANG BANGUN ALAT PAKAN KUCING DENGAN MENGGUNAKAN MIKROKONTROLER BERBASIS INTERNET OF THINGS (IOT). *KARMAPATI (Kumpulan Artikel Mahasiswa Pendidikan Teknik Informatika)*, 12(3), 177-199.
- [13] Tsabit, M. L., Ismail, S. J. I., & Sularsa, A. (2020). Perancangan Sistem Otomatisasi Pemberian Pakan Kucing Menggunakan Penjadwalan Berbasis Mikrokontroler. *eProceedings of Applied Science*, 6(3).
- [14] Lisika, F. (2022). *TA: Rancang Bangun Pemberi Pakan Kucing Otomatis Berbasis Internet Of Things (IoT)* (Doctoral dissertation, Universitas Dinamika).
- [15] Surahman, A., Aditama, B., Bakri, M., & Rasna, R. (2021). Sistem Pakan Ayam Otomatis Berbasis Internet Of Things. *Jurnal Teknologi Dan Sistem Tertanam*, 2(1), 13-20.
- [16] Ahmad Ihwal, F., & Siti, R. (2023). *PEMBUATAN KONTROL DAN MONITORING PEMBERIAN PAKAN KELINCI SECARA OTOMATIS BERBASIS IOT* (Doctoral dissertation, Politeknik Manufaktur Negeri Bangka Belitung).

A decorative graphic consisting of two hands, one light blue and one light grey, positioned as if holding each other. The hands are stylized with geometric shapes.

LAMPIRAN I

DAFTAR RIWAYAT HIDUP

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Stevy Pratiwi Andi Pacoba
Tempat, Tanggal Lahir : Sungailiat, 08 September 2003
Alamat Rumah : Jl. Batin Tikal, Sri Pemandang
No. HP : 0831 7543 8001
Email : stevypratiwi08@gmail.com
Jenis Kelamin : Perempuan
Agama : Islam



2. Riwayat Pendidikan

- | | |
|---|-----------------|
| 1. SD Negeri 1 Sungailiat | 2009 - 2015 |
| 2. SMP Negeri 1 Sungailiat | 2015 - 2018 |
| 3. SMK Negeri 1 Sungailiat | 2018 - 2021 |
| 4. Politeknik Manufaktur Negeri Bangka Belitung | 2021 – Sekarang |

3. Pendidikan Non – Formal

-

Sungailiat, 05 Juni 2024

Stevy Pratiwi Andi Pacoba

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Dikhy Aditya Pratama
Tempat, Tanggal Lahir : Sungailiat, 17 Mei 2003
Alamat Rumah : JL. Raya Kenanga Bagian Timur
No. HP : 0819 9671 2459
Email : dikhyaditya6@gmail.com
Jenis Kelamin : Laki - Laki
Agama : Islam



2. Riwayat Pendidikan

1. SD Negeri 16 Sungailiat 2009 - 2015
2. SMP Negeri 2 Sungailiat 2015 - 2018
3. SMA Setia Budi Sungailiat 2018 - 2021
4. Politeknik Manufaktur Negeri Bangka Belitung 2021 – Sekarang

3. Pendidikan Non - Formal

-

Sungailiat, 05 Juni 2024

Dikhy Aditya Pratama



LAMPIRAN II

PROGRAM

LAMPIRAN PROGRAM

❖ PROGRAM ESP 32

```
//=====//  
  
#include <WiFi.h>  
#include <Firebase_ESP_Client.h>  
#include "addons/TokenHelper.h"  
#include "addons/RTDBHelper.h"  
#include <LiquidCrystal_I2C.h>  
#include "RTCLib.h"  
#include <Arduino.h>  
#include "HX711.h"  
#include "soc/rtc.h"  
#define WIFI_SSID "stevy"  
#define WIFI_PASSWORD "12345678"  
#define API_KEY "AIzaSyAbJ4aV8zDzq126FPXANEMW9EazvssIZjg"  
#define DATABASE_URL "https://pakan-kucing-6e4d9-default-rtdb.firebaseio.com/"  
FirebaseData fbdo;  
FirebaseAuth auth;  
FirebaseConfig config;  
bool signupOK = false;  
LiquidCrystal_I2C lcd(0x27, 16, 2);  
RTC_DS3231 rtc;  
#define Pb_Up 13  
#define Pb_Down 12  
#define Pb_Ok 14  
#define Pb_Cancel 27  
#define BOUNCE 100
```

```

#define Echo 18
#define Trig 19
float Jarak_Sensor=0;
int persen_air;
int JAM,MENIT,DETIK,TGL,BULAN,TAHUN,SUHU,TANGGAL,BERAT,AIR;
int JAM1,JAM2,JAM3,MENIT1,MENIT2,MENIT3;
String jam,menit,detik,tanggal,bulan,tahun,suhu;
String J1,J2,J3,M1,M2,M3;
bool blink = false;
unsigned long old = 0;
char lcdBuff[16];
const int LOADCELL_DOUT_PIN = 26;
const int LOADCELL_SCK_PIN = 4;
HX711 scale;
void setup () {
Serial.begin(9600);
Serial2.begin(9600);
rtc.begin();
lcd.begin();
lcd.backlight();
lcdPrint(0,0,"Menghubungkan ke");
lcdPrint(0,1," Wifi ");
load_cell_setup();
pinMode(Pb_Up, INPUT_PULLUP);
pinMode(Pb_Down, INPUT_PULLUP);
pinMode(Pb_Ok, INPUT_PULLUP);
pinMode(Pb_Cancel, INPUT_PULLUP);

```



```

pinMode(Trig, OUTPUT);
pinMode(Echo, INPUT);
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
while (WiFi.status() != WL_CONNECTED){
Serial.print(".");
delay(300);    }
Serial.println(WiFi.localIP());
Serial.println();
config.api_key = API_KEY;
config.database_url = DATABASE_URL;
if (Firebase.signUp(&config, &auth, "", "")){
Serial.println("ok");
signupOK = true; }
else{
Serial.printf("%s\n", config.signer.signupError.message.c_str()); }
config.token_status_callback = tokenStatusCallback;
Firebase.begin(&config, &auth);
Firebase.reconnectWiFi(true);
load_cell();
lcd.clear();    }
void loop () {
RTC();
// SERVO();
load_cell();
lcdPrint(0,1,"**Pakan Kucing**");
if(bacaButton(Pb_Ok)){
while(bacaButton(Pb_Ok)){}
}
}

```

```

delay(BOUNCE);
Serial.println("ok");
Menu_Utama(); }
Terima_Data();
// Serial.println(J1 + ":" + M1);
// Serial.println(J2 + ":" + M2);
// Serial.println(J3 + ":" + M3);
// Serial.println();    }

❖ MENU
//=====//
void Menu_Utama(){
  lcd.clear();
  int menu = 1;
  while(true){
    lcdPrint(0, 0, " *Pengaturan* ");
    switch (menu){
      case 1 : lcdPrint(0, 1, "< Seting Rtc >"); break;
      case 2 : lcdPrint(0, 1, "< Pakan 1 >"); break;
      case 3 : lcdPrint(0, 1, "< Pakan 2 >"); break;
      case 4 : lcdPrint(0, 1, "< Pakan 3 >"); break;
      case 5 : lcdPrint(0, 1, "< Status >"); break;  }
    if(bacaButton(Pb_Up)){
      while(bacaButton(Pb_Up)){}
      delay(BOUNCE);
      menu++;
      if(menu>5)menu=5;  }
    if(bacaButton(Pb_Down)){

```

```

while(bacaButton(Pb_Down)){
delay(BOUNCE);
menu--;
if(menu<1)menu=1; }
if(bacaButton(Pb_Ok)){
while(bacaButton(Pb_Ok)){
delay(BOUNCE);
switch (menu){
case 1 : Setting_RTC(); break;
case 2 : Pakan1(); break;
case 3 : Pakan2(); break;
case 4 : Pakan3(); break;
case 5 : Status(); break;  }}
if(bacaButton(Pb_Cancel)){
while(bacaButton(Pb_Cancel)){
delay(BOUNCE);
break; }}}
void Setting_RTC(){
lcd.clear();
int menu = 1;
while(true){
switch (menu){
case 1 : pengaturanTanggal(); break;
case 2 : pengaturanBulan();break;
case 3 : pengaturanTahun();break;
case 4 : pengaturanJam();break;
case 5 : pengaturanMenit();break; }

```

```

if(bacaButton(Pb_Ok)){
while(bacaButton(Pb_Ok)){}
delay(BOUNCE);
menu++;
if(menu>5)menu=1;  }
if(bacaButton(Pb_Cancel)){
while(bacaButton(Pb_Cancel)){}
delay(BOUNCE);
rtc.adjust(DateTime(TAHUN, BULAN, TANGGAL, JAM, MENIT, 30));
break;  }}}
void Pakan1(){
lcd.clear();
int menu = 1;
while(true){
switch (menu){
case 1 : Jam1(); break;
case 2 : Menit1(); break;  }
if(bacaButton(Pb_Ok)){
while(bacaButton(Pb_Ok)){}
delay(BOUNCE);
menu++;
if(menu>2)menu=1;  }
if(bacaButton(Pb_Cancel)){
while(bacaButton(Pb_Cancel)){}
delay(BOUNCE);
if (Firebase.ready() && signupOK ) {
Firebase.RTDB.setString(&fbdo, "Pakan_Kucing/JAM1",JAM1);

```

```

Firebase.RTDB.setString(&fbdo, "Pakan_Kucing/MENIT1",MENIT1);    }
break;   }}}
void Pakan2(){
lcd.clear();
int menu = 1;
while(true){
switch (menu){
case 1 : Jam2(); break;
case 2 : Menit2(); break;      }
if(bacaButton(Pb_Ok)){
while(bacaButton(Pb_Ok)){}
delay(BOUNCE);
menu++;
if(menu>2)menu=1;    }
if(bacaButton(Pb_Cancel)){
while(bacaButton(Pb_Cancel)){}
delay(BOUNCE);
if (Firebase.ready() && signupOK ) {
Firebase.RTDB.setString(&fbdo, "Pakan_Kucing/JAM2",JAM2);
Firebase.RTDB.setString(&fbdo, "Pakan_Kucing/MENIT2",MENIT2);    }
break;   }}}
void Pakan3(){
lcd.clear();
int menu = 1;
while(true){
switch (menu){
case 1 : Jam3(); break;

```

```

case 2 : Menit3(); break;      }
if(bacaButton(Pb_Ok)){
while(bacaButton(Pb_Ok)){}
delay(BOUNCE);
menu++;
if(menu>2)menu=1;    }
if(bacaButton(Pb_Cancel)){
while(bacaButton(Pb_Cancel)){}
delay(BOUNCE);
if (Firebase.ready() && signupOK ) {
Firebase.RTDB.setString(&fbdo, "Pakan_Kucing/JAM3",JAM3);
Firebase.RTDB.setString(&fbdo, "Pakan_Kucing/MENIT2",MENIT3); }
break; } } }
void Status(){
lcd.clear();
while(true){
Jarak_Sensor = Test_Jarak();
if(Jarak_Sensor >= 23){
Jarak_Sensor = 23;    }
Jarak_Sensor = 23 - Jarak_Sensor;
if(Jarak_Sensor >= 20){
Jarak_Sensor = 20;    }
load_cell();
persen_air = (Jarak_Sensor * 100)/20;
sprintf(lcdBuff, "Berat :%ig   ",BERAT);
lcdPrint(0,0,lcdBuff);
sprintf(lcdBuff, "Air  :%ipersen   ",persen_air);

```

```

lcdPrint(0,1,lcdBuff);
Serial.println(persen_air);
if(bacaButton(Pb_Cancel)){
while(bacaButton(Pb_Cancel)){}
delay(BOUNCE);
break; }
delay(250);      }}

```

❖ SUB MENU

```

//=====//
void pengaturanTanggal(){
blinking();
lcdPrint(0, 0, "* Seting Rtc *");
sprintf(lcdBuff, "%02i/%02i/%04i %02i:%02i", TANGGAL, BULAN, TAHUN,
JAM, MENIT);
if(blink == true){
lcdPrint(0,1, lcdBuff);
}else{
sprintf(lcdBuff, " /%02i/%04i %02i:%02i", BULAN, TAHUN, JAM, MENIT);
lcdPrint(0,1, lcdBuff);      }
if(bacaButton(Pb_Up)){
while(bacaButton(Pb_Up)){}
delay(BOUNCE);
TANGGAL++;
if(TANGGAL>31)TANGGAL=31;  }
if(bacaButton(Pb_Down)){
while(bacaButton(Pb_Down)){}
delay(BOUNCE);

```

```

TANGGAL--;
if(TANGGAL<1)TANGGAL=1;          }}
void pengaturanBulan(){
blinking();
lcdPrint(0, 0, "* Seting Rtc *");
sprintf(lcdBuff, "%02i/%02i/%04i %02i:%02i", TANGGAL, BULAN, TAHUN,
JAM, MENIT);
if(blink == true){
lcdPrint(0,1, lcdBuff);
}else{
sprintf(lcdBuff, "%02i/ /%04i %02i:%02i", TANGGAL, TAHUN, JAM, MENIT);
lcdPrint(0,1, lcdBuff);      }
if(bacaButton(Pb_Up)){
while(bacaButton(Pb_Up)){}
delay(BOUNCE);
BULAN++;
if(BULAN>31)BULAN=31; }
if(bacaButton(Pb_Down)){
while(bacaButton(Pb_Down)){}
delay(BOUNCE);
BULAN--;
if(BULAN<1)BULAN=1;          }}
void pengaturanTahun(){
blinking();
lcdPrint(0, 0, "* Seting Rtc *");
sprintf(lcdBuff, "%02i/%02i/%04i %02i:%02i", TANGGAL, BULAN, TAHUN,
JAM, MENIT);
if(blink == true){

```



```

lcdPrint(0,1, lcdBuff);
}else{
sprintf(lcdBuff, "%02i/%02i/ %02i:%02i", TANGGAL, BULAN, JAM, MENIT);
lcdPrint(0,1, lcdBuff);      }
if(bacaButton(Pb_Up)){
while(bacaButton(Pb_Up)){}
delay(BOUNCE);
TAHUN++;
if(TAHUN>3000)TAHUN=3000;      }
if(bacaButton(Pb_Down)){
while(bacaButton(Pb_Down)){}
delay(BOUNCE);
TAHUN--;
if(TAHUN<1)TAHUN=1;      }}
void pengaturanJam(){
blinking();
lcdPrint(0, 0, "* Seting Rtc *");
sprintf(lcdBuff, "%02i/%02i/%04i %02i:%02i", TANGGAL, BULAN, TAHUN,
JAM, MENIT);
if(blink == true){
lcdPrint(0,1, lcdBuff);
}else{
sprintf(lcdBuff, "%02i/%02i/%04i :%02i", TANGGAL, BULAN, TAHUN,
MENIT);
lcdPrint(0,1, lcdBuff);      }
if(bacaButton(Pb_Up)){
while(bacaButton(Pb_Up)){}
delay(BOUNCE);

```

```

JAM++;
if(JAM>23)JAM=23; }
if(bacaButton(Pb_Down)){
while(bacaButton(Pb_Down)){}
delay(BOUNCE);
JAM--;
if(JAM<0)JAM=0;    }}
void pengaturanMenit(){
blinking();
lcdPrint(0, 0, "* Seting Rtc *");
sprintf(lcdBuff, "%02i/%02i/%04i %02i:%02i", TANGGAL, BULAN, TAHUN,
JAM, MENIT);
if(blink == true){
lcdPrint(0,1, lcdBuff);
}else{
sprintf(lcdBuff, "%02i/%02i/%04i %02i: ", TANGGAL, BULAN, TAHUN, JAM);
lcdPrint(0,1, lcdBuff);    }
if(bacaButton(Pb_Up)){
while(bacaButton(Pb_Up)){}
delay(BOUNCE);
MENIT++;
if(MENIT>59)MENIT=59;  }
if(bacaButton(Pb_Down)){
while(bacaButton(Pb_Down)){}
delay(BOUNCE);
MENIT--;
if(MENIT<0)MENIT=0;    }}

```

```

void Jam1(){
blinking();
lcdPrint(0, 0, "* PAKAN1 *");
sprintf(lcdBuff, " %02i:%02i ",JAM1, MENIT1);
if(blink == true){
lcdPrint(0,1, lcdBuff);
}else{
sprintf(lcdBuff, " :%02i ",MENIT1);
lcdPrint(0,1, lcdBuff);      }
if(bacaButton(Pb_Up)){
while(bacaButton(Pb_Up)){}
delay(BOUNCE);
JAM1++;
if(JAM1>23)JAM1=23;      }
if(bacaButton(Pb_Down)){
while(bacaButton(Pb_Down)){}
delay(BOUNCE);
JAM1--;
if(JAM1<0)JAM1=0; }}
void Menit1(){
blinking();
lcdPrint(0, 0, "* PAKAN1 *");
sprintf(lcdBuff, " %02i:%02i ",JAM1, MENIT1);
if(blink == true){
lcdPrint(0,1, lcdBuff);
}else{
sprintf(lcdBuff, " %02i: ",JAM1);

```

```

lcdPrint(0,1, lcdBuff);      }
if(bacaButton(Pb_Up)){
while(bacaButton(Pb_Up)){}
delay(BOUNCE);
MENIT1++;
if(MENIT1>59)MENIT1=59;      }
if(bacaButton(Pb_Down)){
while(bacaButton(Pb_Down)){}
delay(BOUNCE);
MENIT1--;
if(MENIT1<0)MENIT1=0;  }}
void Jam2(){
blinking();
lcdPrint(0, 0, "*  PAKAN2  *");
sprintf(lcdBuff, "  %02i:%02i  ",JAM2, MENIT2);
if(blink == true){
lcdPrint(0,1, lcdBuff);
}else{
sprintf(lcdBuff, "  :%02i  ",MENIT2);
lcdPrint(0,1, lcdBuff);      }
if(bacaButton(Pb_Up)){
while(bacaButton(Pb_Up)){}
delay(BOUNCE);
JAM2++;
if(JAM2>23)JAM2=23;      }
if(bacaButton(Pb_Down)){
while(bacaButton(Pb_Down)){}

```

```

delay(BOUNCE);
JAM2--;
if(JAM2<0)JAM2=0;}}
void Menit2(){
blinking();
lcdPrint(0, 0, "* PAKAN2 *");
sprintf(lcdBuff, " %02i:%02i ",JAM2, MENIT2);
if(blink == true){
lcdPrint(0,1, lcdBuff);
}else{
sprintf(lcdBuff, " %02i: ",JAM2);
lcdPrint(0,1, lcdBuff); }
if(bacaButton(Pb_Up)){
while(bacaButton(Pb_Up)){}
delay(BOUNCE);
MENIT2++;
if(MENIT2>59)MENIT2=59; }
if(bacaButton(Pb_Down)){
while(bacaButton(Pb_Down)){}
delay(BOUNCE);
MENIT2--;
if(MENIT2<0)MENIT2=0; }}
void Jam3(){
blinking();
lcdPrint(0, 0, "* PAKAN3 *");
sprintf(lcdBuff, " %02i:%02i ",JAM3, MENIT3);
if(blink == true){

```

```

lcdPrint(0,1, lcdBuff);
}else{
sprintf(lcdBuff, "    :%02i    ",MENIT3);
lcdPrint(0,1, lcdBuff);    }
if(bacaButton(Pb_Up)){
while(bacaButton(Pb_Up)){}
delay(BOUNCE);
JAM3++;
if(JAM3>23)JAM3=23;    }
if(bacaButton(Pb_Down)){
while(bacaButton(Pb_Down)){}
delay(BOUNCE);
JAM3--;
if(JAM3<0)JAM3=0;}}
void Menit3(){
blinking();
lcdPrint(0, 0, "*  PAKAN3  *");
sprintf(lcdBuff, "    %02i:%02i    ",JAM3, MENIT3);
if(blink == true){
lcdPrint(0,1, lcdBuff);
}else{
sprintf(lcdBuff, "    %02i:    ",JAM3);
lcdPrint(0,1, lcdBuff);    }
if(bacaButton(Pb_Up)){
while(bacaButton(Pb_Up)){}
delay(BOUNCE);
MENIT3++;

```

```

if(MENIT3>59)MENIT3=59;      }
if(bacaButton(Pb_Down)){
while(bacaButton(Pb_Down)){}
delay(BOUNCE);
MENIT3--;
if(MENIT3<0)MENIT3=0;  }}

```

❖ PROGRAM SERVO

```

//===== //
#include <Servo.h>
Servo myservo; // create servo object to control a servo
// twelve servo objects can be created on most boards
int pos = 0; // variable to store the servo position
void setup() {
myservo.attach(9); // attaches the servo on pin 9 to the servo object}
void loop() {
if(Serial.available(>0)){
char msg = Serial.read();
if(msg == '#'){
myservo.write(90);
}else if(msg == '*'){
myservo.write(0);      }}
// myservo.write(0);
// delay(1000);
// myservo.write(90);
// delay(1000);  }

```