

**PERANCANGAN KANDANG KUCING PINTAR BERBASIS
*INTERNET OF THINGS (IOT)***

SKRIPSI

DISUSUN OLEH

DARIS FAUZAN ABILA

2109020139



UMSU

Unggul | Cerdas | Terpercaya

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

MEDAN

2025

**PERANCANGAN KANDANG KUCING PINTAR BERBASIS
*INTERNET OF THINGS (IOT)***

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana
Komputer (S.Kom) dalam Program Studi Teknologi Informasi pada
Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas
Muhammadiyah Sumatera Utara**

**DARIS FAUZAN ABILA
NPM. 2109020139**

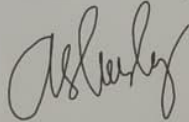
**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN
2025**

LEMBAR PENGESAHAN

LEMBAR PENGESAHAN

Judul Skripsi : PERANCANGAN KANDANG KUCING PINTAR
BERBASIS INTERNET OF THINGS (IOT)
Nama Mahasiswa : DARIS FAUZAN ABILA
NPM : 2109020139
Program Studi : TEKNOLOGI INFORMASI

Menyetujui
Komisi Pembimbing



(Rizaldy Khair, S.Kom., M.Kom)
NIDN. 0116098802

Ketua Program Studi



(Fatma Sari Hutagalung, S.Kom., M.Kom.)
NIDN. 0117019301

Dekan



(Dr. Al-Khowarizmi, S.Kom., M.Kom.)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

PERNYATAAN ORISINALITAS

PERANCANGAN KANDANG KUCING PINTAR BERBASIS *INTERNET OF THINGS (IOT)*

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, 10 September 2025

Yang membuat pernyataan



Daris Fauzan Abila

NPM.2109020139

**PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN
AKADEMIS**

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama	: Daris Fauzan Abila
NPM	: 2109020139
Program Studi	: Teknologi Informasi
Karya Ilmiah	: Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

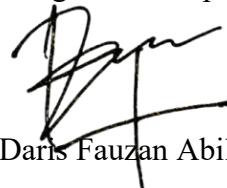
RANCANGAN KANDANG KUCING PINTAR BERBASIS *INTERNET OF THINGS* (IOT)

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, 10 September 2025

Yang membuat pernyataan



Daris Fauzan Abila

NPM. 2109020139

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap : Daris Fauzan Abila
Tempat dan Tanggal Lahir : Lahat, 24 Desember 2003
Alamat Rumah : Jalan Marelan Raya Tanah 600 Komplek
Marelan Sejati no.98M
Telepon/ /HP : 083176003809
E-mail : darisfauzan24@gmail.com
Instansi Tempat Kerja : -
Alamat Kantor : -

DATA PENDIDIKAN

SD	: SD Negeri 067248	TAMAT: 2015
SMP	: SMP Negeri 20 Medan	TAMAT: 2018
SMK	: SMK Negeri 14 Medan	TAMAT: 2021

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji bagi Allah SWT atas karunia-Nya, Penulis akhirnya bisa menyelesaikan skripsi ini. Skripsi ini disusun sebagai salah satu syarat untuk meraih gelar Sarjana Komputer di Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara.

Secara khusus, Penulis ingin berterima kasih kepada kedua orang tua atas dukungan, doa, dan semangat yang tak pernah berhenti. Skripsi ini juga tidak akan terwujud tanpa bimbingan dan bantuan dari banyak pihak, baik selama masa perkuliahan maupun saat penyusunan skripsi. Maka dari itu, dengan tulus Penulis ingin menyampaikan ucapan terima kasih kepada:

1. Bapak Prof. Dr. Agussani, M.AP., selaku Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom., selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Bapak Halim Maulana, S.T., M.Kom., selaku Wakil Dekan I Fakultas Ilmu Komputer dan Teknologi Informasi.
4. Bapak Dr. Lutfi Basit, S.Sos., M.I.Kom., selaku Wakil Dekan III Fakultas Ilmu Komputer dan Teknologi Informasi.
5. Ibu Fatma Sari Hutagalung, S.Kom., M.Kom., selaku Ketua Program Studi Teknologi Informasi yang telah memberikan bimbingan dan bantuan selama Penulis menempuh studi serta yang selalu mengingatkan penulis untuk segera menyelesaikan skripsi ini.

6. Bapak Mhd. Basri, S.Si., M.Kom., selaku Sekretaris Program Studi Teknologi Informasi.
7. Bapak Rizaldy Khair, S.Kom., M.Kom., selaku Dosen Pembimbing yang telah banyak membantu penulis baik dukungan, saran, dan bimbingan dalam penyusunan penelitian ini.
8. Ayahanda Hendri Yuwirsal dan Ibunda Surismawati selaku Kedua Orang Tua penulis yang selalu mendoakan dan memberi dukungan baik mental maupun finansial selama ini.
9. Kepada Nabila Zahra Tusyifa, selaku adik penulis yang senantiasa memberikan dukungan dan semangat dalam penyelesaian skripsi ini.
10. Ucapan terima kasih kepada sahabat baik saya Ahmad Asadel, Ismi Qontas Lubis dan sahabat Kinoy yang telah membantu dan memberi dukungan kepada penulis dalam menyelesaikan skripsi ini.
11. Ucapan terima kasih kepada sahabat baik saya Jefri Pranata dan Bibit Sangadjie Wicak Wiguna yang telah membantu dan memberi dukungan kepada penulis dalam menyelesaikan skripsi ini
12. Terima kasih sebesar-besarnya untuk seluruh teman dan rekan seperjuangan dukungan dan masukan dari kalian semua sangat berarti hingga penulis bisa sampai di titik ini.

Perjalanan untuk sampai pada titik ini tidaklah mudah, dan Penulis sadar bahwa karya yang dihasilkan ini belumlah sempurna. Masih terdapat banyak kekurangan yang perlu diperbaiki. Namun, seperti sebuah langkah pertama, Penulis berharap skripsi ini bisa menjadi pijakan yang bermanfaat, baik bagi pembaca, rekan-rekan mahasiswa, maupun bagi Penulis sendiri untuk terus belajar.

Kritik dan saran yang membangun ibarat lentera yang akan menerangi jalan Penulis ke depan. Dengan segala kerendahan hati, Penulis menantikan masukan dari para pembaca. Terima kasih.

Medan, 10 Agustus 2025

Penulis

A handwritten signature in black ink, appearing to be 'Daris Fauzan Abila', written in a cursive style.

Daris Fauzan Abila

PERANCANGAN KANDANG KUCING PINTAR BERBASIS *INTERNET OF THINGS* (IOT)

ABSTRAK

Seiring dengan meningkatnya minat masyarakat dalam memelihara kucing, tantangan dalam merawatnya juga bertambah, terutama bagi pemilik yang memiliki aktivitas padat. Kesulitan tersebut seringkali meliputi kurangnya jadwal pemberian pakan yang teratur, kurangnya hidrasi, dan kebersihan *litterbox* yang tidak terpantau, yang dapat berdampak negatif pada kesehatan hewan peliharaan. Penelitian ini bertujuan untuk mengatasi masalah tersebut dengan merancang dan mengimplementasikan sebuah sistem kandang kucing pintar berbasis *Internet of Things* (IoT). Sistem ini memiliki tiga fitur utama: pemberian pakan otomatis secara terjadwal, otomasi pemberian minum, serta monitoring dan penanganan bau di *litterbox*. Metode yang digunakan dalam penelitian ini adalah *prototyping*. Sistem dibangun menggunakan mikrokontroler ESP32 sebagai unit kendali utama. Untuk subsistem pakan, digunakan motor servo, sensor *load cell* untuk menakar berat pakan, dan sensor ultrasonik untuk memonitor level pakan. Untuk subsistem minum, digunakan sensor *water level* dan pompa air mini yang dikendalikan oleh relay. Sementara itu, untuk *litterbox*, digunakan sensor ultrasonik untuk mendeteksi keberadaan kucing dan *ultrasonic mist maker* untuk penanganan bau. Seluruh sistem terintegrasi dengan antarmuka pengguna berupa dashboard berbasis web dan notifikasi melalui aplikasi Telegram, yang memungkinkan pemantauan dan kontrol dari jarak jauh secara *real-time*. Hasil dari penelitian ini membuktikan bahwa sistem dapat berjalan dengan baik, berhasil mengotomatiskan proses perawatan, dan memberikan kemudahan bagi pemilik kucing.

Kata Kunci: Kandang Kucing Pintar, *Internet of Things* (IoT), Pakan Otomatis, Website, Telegram Bot, Esp32

DESIGNING A SMART CAT CAGE BASED ON THE INTERNET OF THINGS (IoT)

ABSTRACT

Along with the increasing public interest in keeping cats, the challenges in caring for them also increase, especially for owners with busy schedules. These difficulties often include a lack of regular feeding schedules, insufficient hydration, and unmonitored litterbox cleanliness, which can have a negative impact on the health of the pets. This research aims to address these problems by designing and implementing a smart cat cage system based on the Internet of Things (IoT). The system has three main features: scheduled automatic feeding, automated drinking, and monitoring and odor control in the litterbox. The method used in this research is prototyping. The system is built using an ESP32 microcontroller as the main control unit. For the feeding subsystem, a servo motor, a load cell sensor to measure food weight, and an ultrasonic sensor to monitor food levels are used. For the drinking subsystem, a water level sensor and a mini water pump controlled by a relay are used. Meanwhile, for the litterbox, an ultrasonic sensor is used to detect the presence of cats and an ultrasonic mist maker for odor control. The entire system is integrated with a user interface in the form of a web-based dashboard and notifications via the Telegram application, which allows for remote monitoring and control in real-time. The results of this research prove that the system can run well, successfully automates the care process, and provides convenience for cat owners.

Keywords: Smart Cat Cage, Internet of Things (IoT), Automatic Feeder, Website, Telegram Bot, Esp32

DAFTAR ISI

LEMBAR PENGESAHAN	i
PERNYATAAN ORISINALITAS.....	ii
PERNYATAAN PERSETUJUAN PUBLIKASI.....	iii
RIWAYAT HIDUP	iv
KATA PENGANTAR.....	v
ABSTRAK	viii
ABSTRACT	ix
DAFTAR ISI.....	x
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	4
1.5. Manfaat Penelitian.....	4
BAB II LANDASAN TEORI	6
2.1. Kucing	6
2.2. Internet of Things	6
2.3. Mikrokontroler	7
2.3.1. ESP32	8
2.4. Sensor	10
2.4.1. Ultrasonic HC-SR04.....	10
2.4.2. Sensor Water Level.....	12
2.4.3. Loadcell	13
2.5. Akuator	14
2.5.1. Relay.....	15
2.5.2. Pompa Air Mini	15
2.5.3. Servo.....	16
2.5.4. Ultrasonic Mist Maker.....	17
2.6. Network Time Protocol	19
2.7. Modul HX711	20
2.8. Arduino IDE	21
2.9. Flowchart.....	22

2.10. Penelitian Terdahulu	23
BAB III METODOLOGI PENELITIAN	29
3.1. Metode Penelitian	29
3.2. Alur Penelitian	29
3.3. Analisis Kebutuhan Alat	31
3.4. Perancangan Alur Kerja Sistem	32
3.4.1. Flowchart Sistem Pakan	33
3.4.2. Flowchart Sistem Minum	35
3.4.3. Flowchart Sistem Monitoring dan Penanganan Bau Litterbox	37
3.5. Blok Diagram Sistem	39
3.6. Perancangan Rangkaian Alat	40
3.7. Perancangan Antarmuka Pengguna (User Interface)	43
3.7.1. Perancangan Antarmuka Dashboard Website	43
3.7.2. Perancangan Interaksi Bot Telegram	44
3.8. Teknik Pengumpulan Data	46
3.8.1. Observasi	46
3.8.2. Studi Pustaka	46
3.9. Tempat dan Jadwal Penelitian	46
BAB IV HASIL DAN PEMBAHASAN	48
4.1. Hasil Implementasi Sistem	48
4.1.1. Implementasi Perangkat Keras (Hardware)	48
4.1.2. Implementasi Perangkat Lunak (Software)	52
4.1.3. Tampilan Antarmuka Sistem	53
4.2. Pengujian Sistem	54
4.2.1. Pengujian Sistem Pakan	55
4.2.2. Pengujian Sistem Minum	60
4.2.3. Pengujian Sistem Monitoring dan Penanganan Bau Litterbox	62
4.2.4. Pengujian Kinerja Sistem IoT	65
4.3. Pembahasan Hasil	66
BAB V PENUTUP	68
5.1. Kesimpulan	68
5.2. Saran	69
DAFTAR PUSTAKA	70
LAMPIRAN	73

DAFTAR TABEL

Tabel 2.1 Spesifikasi ESP32	9
Tabel 2.2 Spesifikasi Sensor Ultrasonic	11
Tabel 2.3 Spesifikasi Loadcell.....	14
Tabel 2.4 Simbol Flowchart.....	23
Tabel 2.5 Penelitian Terdahulu	27
Tabel 3.1 Alat dan Bahan.....	31
Tabel 3.2 Skenario dan Pesan Notifikasi Bot Telegram	45
Tabel 3.3 Jadwal Penelitian	47
Tabel 4.1 Data Hasil Uji Penjadwalan Pakan Otomatis.....	55
Tabel 4.2 Hasil Pengujian Akurasi Takaran Pakan.....	57
Tabel 4.3 Hasil Pengujian Deteksi Stok Pakan.....	58
Tabel 4.4 Hasil Pengujian Otomasi Pengisian Air.....	61
Tabel 4.5 Hasil Pengujian Deteksi Keberadaan Kucing.	63
Tabel 4.6 Hasil Pengujian Penanganan Bau Otomatis.....	64
Tabel 4.7 Hasil Pengujian Waktu Respon (Latency)	66

DAFTAR GAMBAR

Gambar 2.1 Internet of Things.....	7
Gambar 2.2 ESP32.....	9
Gambar 2.3 Sensor Ultrasonic HC-SR04	11
Gambar 2.4 Sensor Water Level	12
Gambar 2.5 Sensor Loadcell.....	13
Gambar 2.6 Relay	15
Gambar 2.7 Pompa Air Mini 5V DC	16
Gambar 2.8 Servo	17
Gambar 2.9 Ultrasonic mist maker	19
Gambar 2.10 Modul HX711	20
Gambar 2.11 Tampilan Arduino IDE	22
Gambar 3.1 Alur Penelitian	29
Gambar 3.2 Flowchart Sistem Pakan.....	33
Gambar 3.3 Flowchart Sistem Minum.....	35
Gambar 3.4 Flowchart Sistem Monitoring dan Penanganan Bau Litterbox	37
Gambar 3.5 Blok Diagram Sistem	39
Gambar 3.6 Rangkaian Keseluruhan Sistem.....	41
Gambar 3.7 Tampilan Dashboard Website	44
Gambar 4.1 Implementasi Sistem Pemberian Pakan	49
Gambar 4.2 Implementasi Sistem Otomasi Minum.....	50
Gambar 4.3 Implementasi Sistem Monitoring dan	51
Gambar 4.4 Kandang Kucing Pintar dengan Sistem Terintegrasi	51
Gambar 4.5 Tampilan Arduino IDE dengan Kode Berhasil Diunggah	53
Gambar 4.6 Tampilan Dashboard Website dan Notifikasi Bot Telegram	54
Gambar 4. 7 Notifikasi Pemberian Pakan.....	56
Gambar 4.8 Indikator Berat Pakan Pada Dashboard.....	57

Gambar 4.9 Indikator Status Pada Tabung Pakan.....	59
Gambar 4.10 Notifikasi Peringatan Pada Telegram.....	59
Gambar 4.11 Pengujian Pemberian Pakan Manual.....	60
Gambar 4.12 Indikator Status Pada Minum.....	61
Gambar 4.13 Notifikasi Pengisian Minum Pada Telegram	61
Gambar 4.14 Notifikasi Tandon Air Habis.....	62
Gambar 4.15 Status Pada Litterbox dan Notifikasi di Litterbox Pada Telegram.....	63
Gambar 4.16 Semprotan Menyala dan Notifikasi Pada Telegram.....	65

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Kucing merupakan salah satu hewan peliharaan manusia. Sebagai hewan yang membutuhkan perawatan intensif, kucing tidak hanya memerlukan asupan makanan yang cukup, tetapi juga perhatian dalam hal hidrasi serta kebersihan lingkungan tempat tinggalnya. Pemberian makan kucing yang tidak efektif akan mempengaruhi pola makan hewan peliharaan (Rachmansyah et al., 2022). Perawatan kucing memerlukan perhatian khusus, terutama dalam hal pemberian makan, minum, serta pemantauan kebersihan lingkungan kandang. Pemilik kucing yang memiliki keterbatasan waktu karena kesibukan sehari-hari, sehingga tidak dapat secara rutin memberikan pakan, minuman, ataupun memantau aktivitas kucing di dalam litterbox (Putu et al., 2023).

Pemilik kucing yang sibuk dan beraktivitas di luar rumah dapat berdampak negatif pada kucingnya jika tidak diberi makan dengan cukup, yang dapat menyebabkan pertumbuhan yang tidak optimal dan stres yang cepat (Putri & Darmawan, 2022). Teknologi yang dapat mengendalikan pemberian makan otomatis kandang dari jarak jauh diperlukan untuk mengatasi masalah ini. Pemberian makan hewan peliharaan dari jarak jauh telah digunakan secara efektif dan menghasilkan hasil yang positif.

Selain memberikan pakan kepada kucing. Pemilik perlu memperhatikan juga kebutuhan air minum untuk menjaga kesehatannya tetap baik. Kekurangan asupan air dapat menyebabkan dehidrasi pada kucing. Maka dari itu penting untuk selalu memantau ketersediaan air minum bagi kucing secara teratur. Di sisi lain,

kebersihan tempat buang kotoran juga menjadi fokus utama dalam merawat kucing. Kotak pasir yang tidak sering dibersihkan dapat mengakibatkan ketidaknyamanan bagi kucing karena lingkungan yang kotor.

Teknologi *Internet of Things* (IoT) telah membuka peluang baru dalam otomatisasi berbagai aspek kehidupan, termasuk dalam pemeliharaan hewan peliharaan. IoT memungkinkan objek fisik untuk melihat, mendengar, berpikir, dan melakukan pekerjaan dengan membuat mereka berkomunikasi bersama untuk berbagi informasi dan mengoordinasikan keputusan (Canady et al., 2023). Dengan pemanfaatan IoT, sistem dapat dikendalikan dan dipantau secara *real-time* dari jarak jauh menggunakan perangkat yang terhubung ke internet. *Internet of Things* (IoT) juga telah digunakan dalam berbagai sektor, termasuk dalam sistem manajemen energi, industri, dan otomatisasi lingkungan.

Dalam perkembangan terbaru, penggunaan ESP32 sebagai mikrokontroler utama semakin luas dalam sistem berbasis IoT karena kemampuannya dalam mengintegrasikan Wi-Fi dan Bluetooth secara efisien. Mikrokontroler ESP32 memungkinkan perangkat IoT untuk terhubung ke internet dan berkomunikasi dengan aplikasi berbasis *cloud*, sehingga dapat memantau kondisi secara langsung melalui perangkat seluler atau komputer (Hercog et al., 2023).

Berdasarkan berbagai permasalahan yang telah diidentifikasi dan perkembangan teknologi yang ada, penelitian ini bertujuan untuk merancang dan mengimplementasikan kandang kucing pintar berbasis *Internet of Things* (IoT) dengan ESP32. Sistem ini dirancang untuk mencakup beberapa fitur utama, seperti otomatisasi pemberian makan dengan penjadwalan menggunakan *network time protocol* (NTP), sistem pemberian minum otomatis yang dilengkapi dengan sensor

water level, serta pemantauan aktivitas kucing di litterbox menggunakan sensor ultrasonik. Penelitian ini juga bertujuan untuk mengintegrasikan teknologi *Internet of Things* (IoT) dalam sistem kandang otomatis agar dapat mengirimkan data secara *real-time* kepada pemilik kucing mengenai kondisi makanan, minuman, dan serta informasi terkait aktivitas penggunaan litterbox. Dengan adanya sistem ini, diharapkan dapat meningkatkan efisiensi perawatan kucing dan memberikan kemudahan bagi pemilik kucing yang memiliki keterbatasan waktu atau ketidakhadiran di rumah.

1.2. Rumusan Masalah

Mengenai latar belakang tersebut, maka rumusan masalah yang dapat dijadikan pertimbangan dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang sistem kandang kucing pintar dalam pemberian pakan secara terjadwal, otomasi pemberian minum, *monitoring* dan penanganan bau di *litterbox*?
2. Bagaimana mengintegrasikan IoT dalam sistem kandang kucing pintar dalam mengirimkan data secara *real time*?

1.3. Batasan Masalah

Untuk mencapai tujuan penulisan, penulis membatasi permasalahan yang akan dibahas sebagai berikut:

1. Perancangan kandang kucing pintar berbasis IoT hanya mencakup fitur otomatisasi pemberian pakan terjadwal, pemberian minum otomatis menggunakan sensor *water level*, pemantauan aktivitas kucing di *litterbox* menggunakan sensor ultrasonic dan penanganan bau menggunakan *ultrasonic mist maker*.

2. Tidak mencakup pembersihan otomatis pada litterbox.
3. Kontrol dan pemantauan sistem dilakukan melalui dashboard website yang dapat diakses oleh pemilik kucing untuk memantau kondisi pakan, minum, dan pemantauan aktivitas kucing di litterbox secara real-time.
4. Penelitian ini hanya terbatas pada pemilik kucing yang memiliki keterbatasan waktu atau jarang berada di rumah, dan tidak dirancang untuk pemilik kucing yang melakukan perjalanan jangka panjang.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Merancang dan mengimplementasikan sistem kandang kucing pintar dalam pemberian pakan secara terjadwal, otomasi pemberian minum, *monitoring* dan penanganan bau di *litterbox*.
2. Mengintegrasikan teknologi *Internet of Things* (IoT) dalam sistem kandang kucing pintar agar dapat mengirimkan data secara real-time melalui dashboard website serta notifikasi penting melalui aplikasi telegram.

1.5. Manfaat Penelitian

A. Manfaat bagi peneliti;

1. Dapat meningkatkan pengetahuan teknologi *Internet of Things* (IoT) dan bisa menguntungkan bagi pemilik kucing dengan mudah dan praktis.
2. Menambah wawasan dalam penerapan teknologi berbasis IoT bagi peneliti dalam perancangan sistem kandang kucing pintar.
3. Dapat memberikan inovasi baru terhadap pengetahuan di bidangnya.

B. Manfaat bagi pengguna;

1. Meningkatkan efisiensi waktu dalam perawatan kucing dengan secara otomatis dan penjadwalan.
2. Dapat mengetahui keberadaan kucing saat melakukan defakasi di litterbox.
3. Dapat dikendalikan dan dimonitoring melalui dashboard website dan notifikasi telegram.

BAB II

LANDASAN TEORI

2.1. Kucing

Kucing merupakan salah satu hewan peliharaan yang sangat populer di seluruh dunia. Beberapa jenis kucing ras seperti Persia, Siam, Manx, dan Sphinx sering dibudidayakan di tempat pemeliharaan hewan resmi, namun hanya sekitar 1% dari seluruh populasi kucing yang termasuk dalam ras murni. Sisanya adalah kucing dengan keturunan campuran atau kucing domestik biasa (Ngitung, 2021). Pemilihan makanan yang tepat sangat penting untuk menjaga kesehatan kucing, dengan mempertimbangkan faktor seperti usia, kondisi fisik, tingkat aktivitas, dan riwayat kesehatan. Pemilihan yang tepat dapat membantu mencegah masalah kesehatan seperti obesitas atau gangguan saluran kemih pada kucing.

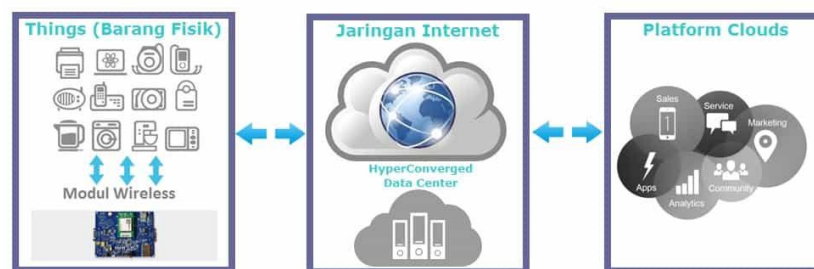
Salah satu jenis makanan yang paling umum diberikan kepada kucing adalah makanan kering (*dryfood*). Makanan ini dirancang dengan formulasi khusus untuk memenuhi kebutuhan gizi kucing, termasuk vitamin, protein, mineral, dan asam amino. Makanan kering ini lebih mudah disimpan dan lebih ramah terhadap pencernaan kucing karena proses pembuatan yang menggunakan suhu tinggi untuk membunuh bakteri (Puspitorini & Sintawati, 2021).

2.2. Internet of Things

Internet of Things (IoT) adalah konsep teknologi yang menghubungkan berbagai objek fisik melalui internet, memungkinkan mereka untuk berkomunikasi, bertukar data, dan melakukan pekerjaan secara otomatis tanpa intervensi manusia secara langsung. IoT merujuk pada jaringan perangkat yang saling terhubung yang memungkinkan objek-objek tersebut untuk bertindak berdasarkan data yang

dikumpulkan dan dikirimkan melalui jaringan internet. Dengan *Internet of Things (IoT)*, berbagai perangkat fisik, seperti sensor, peralatan rumah tangga, dan kendaraan, dapat beroperasi secara efisien dengan memanfaatkan internet untuk berbagi informasi dan mengoordinasikan keputusan (Susanto et al., 2022).

Perkembangan *Internet of Things (IoT)* telah berkembang pesat sejak pertama kali ditemukan oleh Kevin Ashton pada tahun 1999, yang memperkenalkan konsep ini sebagai cara untuk menghubungkan objek fisik ke internet menggunakan teknologi sensor (Ayuningtyas, 2022). Saat ini, IoT telah merambah berbagai sektor kehidupan manusia, seperti industri, kesehatan, rumah tangga, hingga transportasi. Hal ini memungkinkan terciptanya *smart environments* di mana perangkat-perangkat fisik bekerja secara otomatis dan saling terhubung, menjadikan hidup lebih efisien dan praktis (Susanto et al., 2022).



Gambar 2.1 Internet of Things

(Sumber: <https://mobnasesemka.com>)

2.3. Mikrokontroler

Mikrokontroler adalah sebuah chip berbentuk IC (*Integrated Circuit*) yang berfungsi untuk menerima sinyal input, mengolahnya, dan menghasilkan sinyal output sesuai dengan instruksi program yang telah diberikan. Secara umum, mikrokontroler dapat dianggap sebagai otak dari suatu perangkat, yang memiliki kemampuan untuk berinteraksi dengan lingkungan di sekitarnya. Mikrokontroler

terdiri dari beberapa komponen utama, seperti CPU, memori (RAM dan ROM), serta perangkat input dan output yang dapat diprogram (Qureshi, 2024).

Mikrokontroler banyak digunakan dalam berbagai aplikasi yang membutuhkan kontrol otomatis, seperti pada sistem kontrol mesin kendaraan, peralatan medis, remote control, perangkat elektronik, mainan, serta berbagai perangkat lain yang menggunakan sistem tertanam (*embedded system*). Sebagai sirkuit terpadu, mikrokontroler didesain untuk melaksanakan fungsi tertentu sesuai dengan program yang terpasang, memungkinkan perangkat untuk menjalankan tugas secara otonom dan efisien (Makarim et al., 2022).

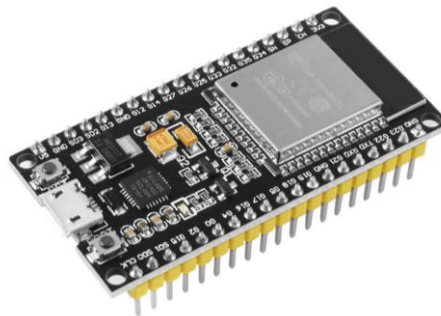
2.3.1. ESP32

ESP32 merupakan mikrokontroler yang dikembangkan oleh *Espressif Systems* dengan berbagai fitur canggih, menjadikannya pilihan unggul dibandingkan dengan mikrokontroler lain seperti Arduino dan NodeMCU ESP8266. Salah satu keunggulan utama ESP32 adalah jumlah pin dan port yang lebih banyak, memungkinkan konektivitas dengan berbagai perangkat eksternal, seperti sensor dan aktuator. ESP32 dikenal sebagai mikrokontroler yang hemat biaya dan memiliki konsumsi daya rendah. ESP32 juga telah dilengkapi dengan konektivitas Wi-Fi berkecepatan tinggi serta fitur *Bluetooth Low Energy* (BLE) yang mendukung dua mode operasi. Dengan adanya fitur ini, tidak diperlukan perangkat tambahan untuk menghubungkan berbagai perangkat eksternal, sehingga dapat mengurangi biaya dan menghemat ruang dalam implementasi sistem berbasis ESP32 (Ruslianto et al., 2023).

Mikrokontroler ini dilengkapi dengan prosesor dual-core dan berbagai kemampuan yang membuatnya ideal untuk aplikasi *Internet of Things* (IoT). ESP32

mendukung multitasking yang efisien, serta mampu menjalankan tugas yang kompleks secara simultan, berkat kinerjanya yang tinggi. Selain itu, mikrokontroler ini memiliki berbagai pin GPIO (General Purpose Input/Output) yang mendukung berbagai protokol komunikasi, seperti SPI, I2C, dan UART (Hercog et al., 2023).

Secara keseluruhan, ESP32 adalah mikrokontroler yang fleksibel dan efisien, dengan kemampuan pemrosesan yang tinggi, konektivitas ganda (Wi-Fi dan Bluetooth), serta manajemen daya yang sangat baik. Semua fitur tersebut membuat ESP32 menjadi pilihan utama dalam pengembangan perangkat IoT yang memerlukan kinerja tinggi, namun tetap hemat energi. Dapat dilihat pada gambar 2.2 mikrokontroler ESP32.



Gambar 2.2 ESP32

(Sumber: <https://grobotronics.com>)

Tabel 2.1 Spesifikasi ESP32

No	Spesifikasi	Keterangan
1	Mikrokontroler	Tensilica 32-bit Single-/Dual-core CPU Xtensa LX6
2	Memory	520 KB SRAM, 448 KB ROM dan 16 KB SRAM RTC
3	Analog Output (DAC) channels	2 channels
4	Digital input output (DIO) pins	25
5	Analog Input (ADC) channels	15 channels
6	UARTs	3

7	SPIs	2
8	Dimensions	51.5x29x5mm
9	Wi-Fi	IEEE 802.11 b/g/n/e/
10	Clock Speed	240 Mhz

2.4. Sensor

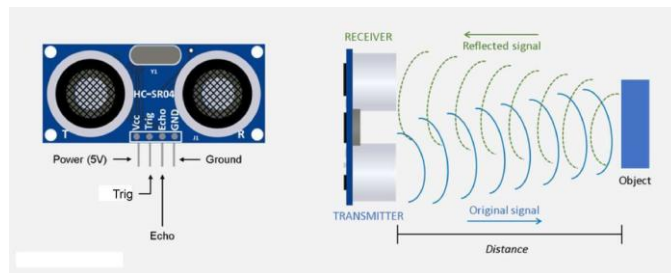
Sensor adalah perangkat yang berfungsi untuk mengubah suatu besaran fisik menjadi sinyal listrik yang dapat diproses lebih lanjut. Dalam berbagai aplikasi, sensor digunakan untuk mendeteksi fenomena fisik seperti suhu, kelembaban udara, tekanan, dan kualitas gas yang ada di lingkungan sekitar. Sistem sensor yang terintegrasi memungkinkan kita untuk memantau dan mengukur perubahan dalam lingkungan secara real-time, yang sangat penting dalam berbagai teknologi, termasuk *Internet of Things* (Gunadi & Rachmawati, 2022).

Sensor bekerja dengan mengubah suatu besaran fisik, seperti suhu, cahaya, tekanan, atau posisi, menjadi sinyal listrik yang dapat diukur dan diproses lebih lanjut oleh sistem. Pada dasarnya, prinsip kerja sensor melibatkan deteksi perubahan dalam suatu parameter fisik yang terjadi di lingkungan sekitar. Sensor berfungsi untuk mendeteksi perubahan ini dengan mengukur respons material atau elemen tertentu terhadap perubahan tersebut. Sebagian sensor menggunakan perubahan resistansi, yang terjadi ketika elemen sensor bereaksi terhadap faktor fisik seperti suhu atau cahaya.. Prinsip dasar ini memungkinkan sensor untuk digunakan dalam berbagai sistem yang mengandalkan pengukuran dan pemantauan otomatis, seperti pada sistem kendali atau pengawasan berbasis IoT (Yeong et al., 2021).

2.4.1. Ultrasonic HC-SR04

Sensor *ultrasonic* HC-SR04 merupakan salah satu jenis sensor yang berfungsi untuk mengukur jarak suatu objek tanpa harus menyentuhnya secara langsung.

Sensor ini bekerja dengan menggunakan gelombang suara frekuensi tinggi yang dipancarkan dari komponen pemancar (*transmitter*) dan diterima kembali oleh komponen penerima (*receiver*) setelah gelombang tersebut dipantulkan oleh objek yang ada di depannya. HC-SR04 banyak digunakan dalam sistem otomatisasi dan robotika karena memiliki kemampuan deteksi yang cukup baik dalam rentang jarak menengah serta akurasi yang relatif tinggi (Yudha Kusuma et al., 2021).



Gambar 2.3 Sensor Ultrasonic HC-SR04

Sumber: (Yudha Kusuma et al., 2021)

Pada gambar 2.3, prinsip kerja sensor ini dimulai saat modul mengirimkan sinyal ultrasonik dengan frekuensi 40 kHz ke arah objek.. Waktu tempuh sinyal dari saat dipancarkan hingga diterima kembali akan diukur, dan dari waktu tersebut, mikrokontroler akan menghitung jarak objek menggunakan rumus kecepatan gelombang suara. HC-SR04 memanfaatkan teknologi *piezoelektrik* untuk menghasilkan dan menerima gelombang ultrasonik, serta mampu mengonversi gelombang suara tersebut menjadi data jarak dalam satuan sentimeter (Yudha Kusuma et al., 2021).

Tabel 2.2 Spesifikasi Sensor Ultrasonic

No	Spesifikasi	Keterangan
1	Frekuensi	40 Khz
2	Arus kerja	15 mA
3	Jarak maksimum	400 cm
4	Koneksi	Vcc, Gnd, Echo dan Trigger
5	Tegangan kerja	5V DC

2.4.2. Sensor Water Level

Sensor *water level* merupakan perangkat elektronik yang berfungsi untuk mendeteksi ketinggian air dalam suatu wadah atau lingkungan tertentu. Prinsip kerja sensor ini didasarkan pada perubahan nilai resistansi atau kapasitansi yang terjadi ketika sensor bersentuhan dengan air. Air sebagai medium penghantar listrik akan menyebabkan perubahan tegangan atau hambatan, yang kemudian dikonversi menjadi informasi mengenai tingkat ketinggian air (Silpa Rani Zain et al., 2024).

Secara umum, *sensor water level* dapat dikategorikan menjadi beberapa jenis berdasarkan metode pendeteksiannya, yaitu sensor resistif, kapasitif, ultrasonik, dan tekanan hidrostatik. Sensor resistif bekerja dengan memanfaatkan jalur konduktif yang akan menutup sirkuit ketika bersentuhan dengan air, sehingga dapat menentukan ketinggian air berdasarkan jumlah jalur yang tertutup. Sensor kapasitif menggunakan perubahan medan listrik yang dipengaruhi oleh kehadiran air untuk mengukur level cairan. Di sisi lain, sensor berbasis tekanan hidrostatik bekerja dengan mengukur tekanan yang dihasilkan oleh ketinggian air dalam suatu wadah. Keunggulan utama sensor water level terletak pada kemudahan instalasi, biaya yang relatif rendah, serta kompatibilitasnya dengan berbagai sistem otomasi dan monitoring. Dapat dilihat pada gambar 2.4 sensor water level.



Gambar 2.4 Sensor Water Level

(Sumber: www.edukasiaelektronika.com)

2.4.3. Loadcell

Loadcell adalah sebuah sensor yang dirancang untuk mendeteksi tekanan atau berat suatu objek. Komponen ini sering digunakan sebagai bagian utama dalam sistem penimbangan digital, termasuk dalam aplikasi seperti jembatan timbang yang mengukur berat berdasarkan tekanan yang diterapkan. *Loadcell* dikenal sebagai sensor pengukur regangan (*strain gauge*), berfungsi untuk mengukur berat statis dengan tingkat akurasi yang tinggi. Ketika suatu gaya diterapkan pada *load cell*, perubahan resistansi terjadi pada pengukur regangan, yang kemudian dikonversi menjadi sinyal tegangan listrik (Majid et al., 2023).

Sebagian besar load cell menggunakan pengukur regangan yang dikonfigurasi dalam bentuk jembatan *Wheatstone* dengan empat lengan. Prinsip kerja load cell didasarkan pada perubahan regangan yang terjadi saat sensor menerima beban. Ketika bagian elastis dari load cell mengalami tekanan, pengukur regangan yang terpasang di bagian lain akan merespons dengan perubahan nilai resistansi akibat gaya yang diterapkan pada sisi berlawanan (Putu et al., 2023). Dapat dilihat sensor *loadcell* pada gambar 2.5.



Gambar 2.5 Sensor Loadcell

Sumber: (Putu et al., 2023)

Tabel 2.3 Spesifikasi Loadcell

No	Spesifikasi	Keterangan
1	Beban maksimum	1 Kg
2	Tegangan	5V – 10V DC
3	Tipe konektor	4 kabel (E+, E-, Sig+, Sig-)

2.5. Aktuator

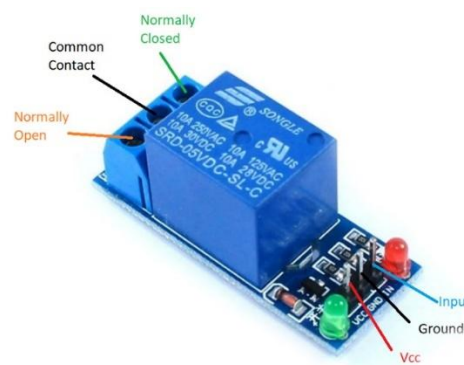
Aktuator adalah perangkat yang mengubah energi atau sinyal kontrol menjadi gerakan atau aksi fisik dalam sebuah sistem. Aktuator berfungsi untuk memanipulasi lingkungan fisik berdasarkan instruksi yang diterimanya, seperti menggerakkan objek atau membuka dan menutup suatu mekanisme. Dalam sistem yang lebih kompleks, aktuator bekerja sebagai penggerak utama dalam banyak aplikasi, termasuk dalam robotika, otomasi, dan kendaraan otonom. Aktuator sering digunakan untuk mengubah sinyal kontrol dari sistem ke dalam bentuk tindakan fisik yang dapat mempengaruhi proses atau lingkungan sekitar (Mukhtar et al., 2023).

Prinsip kerja aktuator didasarkan pada konversi energi dari sumber eksternal menjadi gerakan fisik yang dapat mempengaruhi suatu sistem atau lingkungan. Aktuator elektromagnetik, misalnya, bekerja dengan mengubah energi listrik menjadi gerakan mekanik melalui gaya magnet yang dihasilkan oleh arus listrik. Ketika arus listrik mengalir melalui kumparan yang berada dalam medan magnet, gaya elektromagnetik yang dihasilkan menggerakkan komponen mekanik, seperti rotor dalam motor listrik. Sementara itu, aktuator hidrolik menggunakan tekanan fluida bertekanan untuk menghasilkan gerakan mekanik. Semua jenis aktuator ini mengubah energi yang diterima, baik itu listrik, fluida, udara, atau suhu, menjadi

gerakan fisik yang menggerakkan atau mengendalikan sistem dalam berbagai aplikasi, mulai dari robotika hingga otomasi industri (Mukhtar et al., 2023)..

2.5.1. Relay

Relay adalah suatu komponen elektronik yang berfungsi sebagai saklar otomatis yang dapat menghubungkan atau memutuskan arus listrik berdasarkan perintah dari rangkaian elektronik lainnya. Komponen ini terdiri dari dua jenis kontak listrik, yaitu *normally close* (NC) dan *normally open* (NO), serta dilengkapi dengan kumparan, pegas, dan saklar yang terhubung dengan pegas tersebut. Ketika kumparan relay diberikan tegangan kerja, medan magnet akan terbentuk, menyebabkan saklar berpindah posisi dan memungkinkan arus listrik mengalir ke perangkat yang terhubung. Akibatnya, komponen yang dikendalikan oleh relay akan aktif saat kumparan menerima aliran listrik (Fatahillah Murad et al., 2022). Dapat dilihat pada gambar 2.6 relay.



Gambar 2.6 Relay

Sumber: (Fatahillah Murad et al., 2022)

2.5.2. Pompa Air Mini

Pompa DC Mini merupakan jenis pompa air submersible yang beroperasi dengan menggunakan arus searah (DC) sebagai sumber daya utama untuk menggerakkan motor pompa. Pompa ini umumnya dapat dihubungkan langsung ke

sumber daya listrik DC, seperti baterai atau adaptor dengan spesifikasi yang sesuai. Fungsi utama dari pompa ini adalah untuk mengalirkan atau mentransfer air dari satu tempat ke tempat lain, sehingga sering dimanfaatkan dalam berbagai aplikasi, seperti sistem sirkulasi air pada akuarium serta proyek elektronik yang memerlukan mekanisme pemompaan dalam skala kecil



Gambar 2.7 Pompa Air Mini 5V DC

(Sumber: www.jogjarobotika.com)

Pada gambar 2.7 merupakan gambar pompa air mini yang bertegangan 5v dc. Salah satu keunggulan utama dari pompa air mini adalah desainnya yang ringkas, konsumsi daya yang rendah, serta kemudahan dalam pemasangan. Namun, karena ukurannya yang kecil, pompa ini memiliki kapasitas terbatas, sehingga kurang sesuai untuk aplikasi yang membutuhkan aliran air dalam volume besar (Ariyanti, 2023). Secara keseluruhan, pompa air celup mini merupakan solusi praktis dan efisien untuk kebutuhan pemompaan dalam skala kecil, terutama di lingkungan yang memiliki keterbatasan ruang. Karakteristiknya yang hemat energi serta fleksibilitas dalam pemasangan menjadikannya pilihan ideal untuk berbagai keperluan yang tidak memerlukan kapasitas pompa yang besar.

2.5.3. Servo

Motor servo adalah salah satu komponen elektronik yang bekerja dengan sistem umpan balik tertutup, di mana posisi putaran motor selalu dipantau dan

dikirim kembali ke rangkaian pengendali di dalamnya. Sistem ini memungkinkan motor bergerak dengan tingkat presisi tinggi sesuai dengan sinyal yang diterima. Motor servo terdiri dari beberapa bagian utama, yaitu motor itu sendiri, rangkaian kontrol elektronik, serangkaian roda gigi untuk menyalurkan tenaga, serta potensiometer yang berfungsi sebagai pembatas sudut putaran. Potensiometer ini berperan dalam menentukan posisi akhir dari pergerakan motor dengan membaca perubahan resistansi yang dihasilkan dari sudut putaran porosnya.

Motor servo umumnya menggunakan tegangan DC dan memiliki sistem roda gigi internal yang berfungsi untuk mengontrol pergerakan sudut secara lebih akurat, menjadikannya sangat berguna dalam berbagai aplikasi seperti robotika, otomasi, dan perangkat elektronik lainnya yang membutuhkan kontrol gerakan presisi (Febryanto et al., 2022). Dapat dilihat servo pada gambar 2.8.



Gambar 2.8 Servo

(Sumber: <https://vi.hwlibre.com>)

2.5.4. Ultrasonic Mist Maker

Ultrasonic mist maker merupakan sebuah alat yang berfungsi untuk mengubah air menjadi kabut halus dengan menggunakan gelombang ultrasonik. Alat ini sering digunakan pada sistem pelembap udara (*humidifier*), aromaterapi, dan berbagai aplikasi lain yang membutuhkan penyebaran uap atau cairan dalam

bentuk kabut. Dalam konteks tertentu, alat ini juga disebut sebagai *diffuser*, terutama jika digunakan untuk menyebarkan aroma atau disinfektan ke udara (Nurul Fitri & Yendri, 2023).

Proses pengubahan air menjadi kabut dilakukan dengan menggunakan teknologi piezoelektrik, yaitu sebuah komponen berbahan kristal yang mampu menghasilkan getaran frekuensi tinggi ketika diberi tegangan listrik. Getaran ini kemudian ditransmisikan ke permukaan air melalui elemen diafragma, menyebabkan air terpecah menjadi partikel-partikel kecil dalam bentuk kabut dingin. Kabut yang dihasilkan tidak melalui proses pemanasan, sehingga sangat aman untuk digunakan dalam lingkungan yang sensitif seperti ruang perawatan atau kandang hewan peliharaan (Fernanda & Wildian, 2023).

Ultrasonic mist maker umumnya bekerja pada tegangan 5V hingga 24V tergantung tipe dan desainnya. Perangkat ini dapat diintegrasikan dengan sistem berbasis mikrokontroler seperti Arduino atau ESP32 untuk dikendalikan secara otomatis sesuai dengan kondisi lingkungan atau kebutuhan pengguna. Dalam sistem otomatisasi berbasis IoT, mist maker sering digunakan untuk menyebarkan cairan seperti air, antiseptik, atau pengharum secara otomatis berdasarkan perintah dari sensor atau pemrograman waktu (Nurul Fitri & Yendri, 2023).

Dengan demikian, konsep kerja *piezoelectric* dalam diffuser memanfaatkan getaran frekuensi tinggi yang dihasilkan oleh efek *piezoelektrik* untuk mengubah cairan menjadi uap tanpa pembakaran atau pemanasan berlebih. Dapat dilihat *ultrasonic mist maker* pada gambar 2.9.



Gambar 2.9 Ultrasonic mist maker

Sumber: (Nurul Fitri & Yendri, 2023)

2.6. Network Time Protocol

Network Time Protocol (NTP) merupakan protokol komunikasi yang dirancang untuk menyinkronkan jam komputer dengan sumber waktu yang akurat melalui jaringan. Protokol ini menggunakan hierarki server waktu, di mana sumber waktu referensi seperti jam atom atau GPS, yang meneruskan informasi waktu ke server tingkat berikutnya. Melalui proses pertukaran paket data, komputer klien dapat menyesuaikan waktu sistemnya agar sesuai dengan standar global (Huseng, 2024). *Network Time Protocol* (NTP) memiliki mekanisme koreksi yang cerdas, sehingga tidak hanya menyesuaikan waktu secara tiba-tiba, tetapi juga melakukan penyesuaian bertahap agar perubahan tidak menyebabkan inkonsistensi dalam sistem.

Perbedaan utama antara *Network Time Protocol* (NTP) dan *Real-Time Clock* (RTC) terletak pada mekanisme dan tujuan penggunaannya. *Network Time Protocol* (NTP) adalah solusi berbasis jaringan yang memungkinkan sinkronisasi waktu antarperangkat dalam skala luas, sedangkan *Real-Time Clock* (RTC) merupakan solusi mandiri yang tetap berjalan meskipun sistem tidak terhubung ke jaringan (Huseng, 2024). *Real-Time Clock* (RTC) sangat berguna dalam perangkat yang membutuhkan pencatatan waktu secara konsisten meskipun dalam keadaan

offline, sedangkan *Network Time Protocol* (NTP) lebih cocok digunakan dalam sistem yang memerlukan sinkronisasi dengan waktu global secara real-time.

2.7. Modul HX711

Modul HX711 merupakan perangkat elektronik yang berfungsi sebagai antarmuka untuk membaca dan mengolah data dari sensor beban (*loadcell*). Modul ini mengonversi sinyal dari load cell menjadi format digital yang dapat diinterpretasikan oleh mikrokontroler atau komputer. HX711 banyak digunakan dalam berbagai aplikasi yang membutuhkan pengukuran berat dengan akurasi tinggi, seperti timbangan digital serta peralatan industri yang berkaitan dengan sistem penimbangan (Putu et al., 2023).



Gambar 2.10 Modul HX711

Sumber: (Putu et al., 2023)

Pada gambar 2.10, prinsip kerja modul ini didasarkan pada konversi perubahan resistansi yang terjadi pada load cell menjadi sinyal tegangan. Modul HX711 terdiri dari berbagai komponen elektronik, termasuk kapasitor, resistor, dan transistor, serta sebuah integrated circuit (IC) HX711 yang berperan sebagai penguat sinyal, pengatur osilasi, dan konverter analog ke digital (ADC). Dengan konfigurasi ini, modul HX711 mampu menghasilkan keluaran data digital yang stabil dan akurat, sehingga dapat dengan mudah diintegrasikan ke dalam sistem berbasis mikrokontroler (Putu et al., 2023).

2.8. Arduino IDE

Arduino Integrated Development Environment (Arduino IDE) merupakan perangkat lunak yang berfungsi sebagai lingkungan pengembangan terintegrasi untuk pemrograman mikrokontroler Arduino. Lingkungan ini memungkinkan pengguna untuk menulis, mengunggah, dan mengeksekusi kode pada perangkat Arduino. Arduino IDE menggunakan bahasa pemrograman yang menyerupai bahasa C, namun telah disederhanakan agar lebih mudah dipahami, terutama bagi pengguna pemula (Kurniawan et al., 2020).

Setiap mikrokontroler Arduino telah ditanamkan program *Bootloader* sebelum dipasarkan. Bootloader ini berperan sebagai perantara antara compiler pada Arduino IDE dengan mikrokontroler, sehingga mempermudah proses pemrograman tanpa memerlukan perangkat tambahan. Arduino IDE dikembangkan menggunakan bahasa pemrograman Java dan didukung oleh pustaka C/C++ yang dirancang untuk menyederhanakan operasi input dan output (Kurniawan et al., 2020). Perangkat lunak ini merupakan hasil pengembangan dari *software Processing*, yang kemudian dimodifikasi untuk memenuhi kebutuhan pemrograman pada platform Arduino. Dapat dilihat tampilan Arduino IDE pada gambar 2.11.



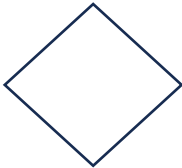
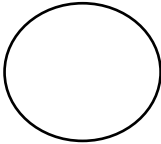
Gambar 2.11 Tampilan Arduino IDE

(Sumber: <https://elibrary.com>)

2.9. Flowchart

Flowchart atau yang sering disebut sebagai bagan alir, merupakan representasi grafis yang menggambarkan alur proses dalam suatu sistem atau program secara logis dan sistematis. Dalam konteks pemrograman, flowchart berfungsi untuk mengilustrasikan algoritma yang digunakan, menunjukkan tahapan-tahapan eksekusi program, serta mengidentifikasi arah aliran data dan proses yang terjadi di dalamnya (Yulianeu & Oktamala, 2022). Dengan menggunakan simbol-simbol standar, flowchart memudahkan pemahaman terhadap struktur logika sebuah prosedur atau sistem, sehingga dapat membantu dalam analisis, perancangan, serta pengoptimalan program sebelum tahap implementasi dilakukan.

Tabel 2.4 Simbol Flowchart

No.	Simbol	Nama Simbol	Keterangan
1		Terminal	Simbol ini digunakan untuk memulai program atau mengakhiri program.
2		Input/Output	Simbol ini digunakan untuk menyatakan input atau output tanpa melihat jenisnya.
3		Decision	Simbol ini digunakan untuk memilih proses untuk penentuan langkah selanjutnya.
4		Process	Simbol ini digunakan untuk menyatakan sebuah proses yang dilakukan komputer.
5		Manual Operation	Simbol ini digunakan untuk menyatakan suatu proses yang tidak dilakukan komputer.
6		Flowline	Simbol ini digunakan untuk menggabungkan antara simbol satu dengan yang lain.
7		Connector Symbol	Simbol ini digunakan untuk keluar-masuk atau penyambungan proses dalam lembar yang sama

2.10. Penelitian Terdahulu

Penelitian terdahulu ini membuat penulis mendapatkan ilmu tambahan dalam melakukan penelitian, Adapun beberapa hasil penelitian yang sudah dilakukan oleh penelitian sebelumnya akan di lampirkan dengan harapan dapat menjelaskan

perbedaan antara penelitian yang akan penulis angkat dengan penelitian yang sudah dilakukan.

Heranof & Yendri, (2023) melakukan penelitian yang berjudul “Alat Kandang Kucing Otomatis Berbasis Mikrokontroler dengan Monitoring Telegram”. Sistem ini menggunakan modul ESP 01 yang bertindak sebagai *client* yang akan mengirim sinyal ke server di kandang. Apabila kucing berada di depan pintu kandang dengan jarak 20 cm hingga 50 cm yang dibaca sensor ultrasonic maka motor servo akan membuka pintu kandang dan saat kucing masuk, kandang akan dibaca oleh sensor PIR, lalu pintu kandang akan tertutup. Di saat kucing berada di dalam kandang, pakan kucing otomatis akan dimulai dari Telegram untuk penginputan profil, data, dan jadwal yang sudah diatur oleh motor servo dan akan ditimbang dibawah loadcell hingga berat sesuai dengan standarnya, sisa pakan kucing akan dihitung dengan sensor infrared.

Putri & Darmawan (2022) melakukan penelitian yang berjudul “Prototipe Sistem Kendali Jarak Jauh Pada Pakan dan Pintu Kandang Kucing”. Penelitian ini menggunakan servo dan ultrasonic untuk alat pemberian pakannya. NodeMCU sebagai mikrokontroller akan memproses nilai atau input dari sensor ultrasonic yang kemudian dikirim ke Blynk sebagai notifikasi jika pakan perlu diisi kembali dan sebagai monitoring banyaknya pakan yang masih tersedia. Motor servo digunakan sebagai komponen yang menggerakkan katup atau penutup pada alat penyimpanan pakan kucing dan dikontrol melalui Blynk. Pada pemberian minum menggunakan beberapa komponen yaitu pompa air mini, relay dan sensor ultrasonic. NodeMCU sebagai mikrokontroler akan memproses nilai atau input dari sensor ultrasonic yang kemudian dikirim ke Blynk sebagai notifikasi jika air minum

perlu diisi kembali dan sebagai monitoring banyaknya air yang masih tersedia. Relay digunakan untuk mengaktifkan pompa air, pada rangkaian ini relay dipasang secara normally open (kondisi dimana kondisi relay terbuka atau tidak terhubung, sehingga arus listrik tidak dapat mengalir). Pompa air digunakan sebagai komponen yang memindahkan air ke tempat minum pada tempat penyimpanan air minum kucing dan dikontrol melalui Blynk. Pada pengontrolan pintu kandang kucing menggunakan dua komponen yaitu motor servo dan sensor infrared yang keduanya saling dihubungkan ke NodeMCU. NodeMCU sebagai mikrokontroler akan memproses nilai atau input dari sensor infrared yang kemudian dikirim ke Blynk sebagai notifikasi keberadaan kucing yang melewati pintu. Motor servo digunakan sebagai komponen yang menggerakkan pintu dan dikontrol melalui Blynk.

Pada penelitian Putu et al., (2023) yang berjudul “Rancang Bangun Alat Pakan Kucing Dengan Mikrokontroller Berbasis *Internet of Things* (IoT)”. Pada penelitian ini menggunakan mikrokontroller ESP32 sebagai pengendalian kinerja sistem. Sistem ini beroperasi dengan mengumpulkan data dari sensor ultrasonik dan sensor berat yang kemudian diproses oleh mikrokontroler. Mikrokontroler menggerakkan motor servo untuk membuka dan menutup tempat pakan kucing. Jika makanan sudah habis, mikrokontroler akan mengirimkan pesan melalui Telegram dan LED akan menyala sebagai indikasi. Pemasangan lampu indikator LED memungkinkan pengguna untuk mengetahui status pakan dalam dispenser. Jika pakan dianggap habis, lampu indikator akan berwarna merah, sedangkan jika pakan dianggap terisi penuh, lampu indikator akan berwarna hijau. Dengan penggunaan RFID pada kalung kucing, setiap kucing yang memakai kalung dengan "tag key" akan memiliki identifikasi unik. Ketika kucing mendekati alat pemberian

pakan dan kalung tersebut menyentuh pembaca RFID, sensor akan otomatis membaca identifikasi kucing, dan servo akan membuka pintu pakan dengan menjatuhkan makanan ke wadah kucing. Hal ini terjadi ketika jarak antara kalung dan pembaca RFID sekitar 1-5 cm.

Yulanda & Kurniawan, (2024) melakukan penelitian yang berjudul “Perancangan Kandang Pintar Menggunakan NodeMCU ESP32 dan Platform Blynk”. Penelitian ini bertujuan untuk mengatasi kesulitan pemilik hewan dalam memberikan pakan secara teratur dengan merancang sebuah kandang pintar berbasis IoT. Sistem ini menggunakan NodeMCU ESP32 sebagai mikrokontroler utama yang terhubung dengan berbagai sensor untuk memantau kondisi kandang. Fitur utamanya meliputi sistem kontrol suhu otomatis menggunakan sensor DHT22 yang dapat mengaktifkan kipas atau lampu pijar, pemantauan ketersediaan pakan dan minum menggunakan sensor ultrasonik dan *water level*, serta pemantauan visual secara *real-time* menggunakan ESP32-CAM. Seluruh data dari sensor dan video dapat diakses dan dikontrol oleh pengguna dari jarak jauh melalui platform aplikasi Blynk.

Alfayed & Purnomo, (2024) melakukan penelitian yang berjudul “Prototipe Alat Pemberi Pakan Kucing Otomatis Berbasis *Internet of Things*”. Penelitian ini bertujuan untuk merancang sebuah alat pemberi pakan kucing yang dapat dikontrol dan dipantau dari jarak jauh. Sistem ini menggunakan mikrokontroler NodeMCU ESP8266 sebagai unit pemrosesan utama. Untuk mekanisme pemberian pakan, motor servo digunakan untuk membuka dan menutup wadah pakan, sedangkan sisa pakan di dalam wadah dipantau menggunakan sensor ultrasonik. Interaksi

pengguna dilakukan melalui aplikasi Android *custom* yang terhubung ke perangkat keras melalui layanan *backend* Google Firebase sebagai jembatan komunikasi.

Tabel 2.5 Penelitian Terdahulu

No.	Nama Penelitian	Judul Penelitian	Perbedaan dan Persamaan Penelitian
1	Putu et al., (2023)	Rancang Bangun Alat Pakan Kucing Dengan Mikrokontroller Berbasis <i>Internet of Things</i> (IoT)	Perbedaannya Pemberian pakan pada sistem saat ini belum terjadwal menggunakan NTP atau RTC, tetapi masih bergantung pada perintah "feed" melalui Telegram serta deteksi RFID yang mengaktifkan dispenser saat kucing mendekat. Penelitian penulis tidak menggunakan sensor RFID. Penelitian ini memiliki kesamaan dalam sistem pemberian pakan kucing yang menggunakan ESP32, sensor load cell, servo, dan sensor ultrasonik (HC-SR04).
2	Putri & Darmawan (2022)	Prototipe Sistem Kendali Jarak Jauh Pada Pakan dan Pintu Kandang Kucing	Perbedaannya adalah tidak digunakannya sensor loadcell untuk pemantauan berat pakan dan sensor water level untuk memantau ketersediaan air minum. Selain itu, sistem ini juga tidak dilengkapi dengan pemantauan di area litterbox. Penelitian penulis tidak menggerakkan pintu kandang. Persamaannya adalah terletak pada penggunaan servo untuk mekanisme pemberian pakan serta pompa air untuk pengisian wadah minum.
3	Heranof & Yendri, (2023)	Alat Kandang Kucing Otomatis Berbasis Mikrokonroller dengan Monitoring Telegram	Perbedaannya adalah terletak pada tidak adanya sistem pemberian minum dan monitoring dalam litterbox, serta penelitian penulis tidak menggunakan sistem penggerak pintu kandang. Persamaannya yaitu sama-sama menggunakan penjadwalan untuk pemberian pakan dan sensor load cell untuk menimbang pakan di wadah.

No.	Nama Penelitian	Judul Penelitian	Perbedaan dan Persamaan Penelitian
4	Yulanda & Kurniawan, (2024)	Perancangan Kandang Pintar Menggunakan NodeMCU ESP32 dan Platform Blynk	Perbedaannya adalah terletak pada antarmuka yang menggunakan platform Blynk dan bukan dashboard website custom, serta fokus utama fitur pada kontrol suhu dan pemantauan video, bukan pada penakaran pakan presisi menggunakan loadcell dan sistem penanganan bau di litterbox. Persamaannya yaitu sama-sama menggunakan mikrokontroler NodeMCU ESP32 untuk sistem kandang kucing berbasis IoT, memiliki fitur pemberian makan dan minum otomatis, serta memanfaatkan sensor ultrasonik dan water level untuk melakukan monitoring.
5	Alfayed & Purnomo, (2024)	Prototipe Alat Pemberi Pakan Kucing Otomatis Berbasis <i>Internet of Things</i>	Perbedaannya adalah penggunaan mikrokontroler NodeMCU ESP8266 dan arsitektur komunikasi yang menggunakan Aplikasi Android dengan Google Firebase sebagai perantaranya, bukan ESP32 dengan <i>dashboard website</i> . Selain itu, penelitian ini cakupannya hanya berfokus pada sistem pakan otomatis dan tidak mencakup sistem minum serta monitoring pada <i>litterbox</i> seperti yang dirancang oleh penulis. Persamaannya yaitu sama-sama merancang alat pakan kucing otomatis berbasis IoT untuk memudahkan pemilik hewan, serta menggunakan motor servo sebagai mekanisme dispenser dan sensor ultrasonik untuk memantau level pakan.

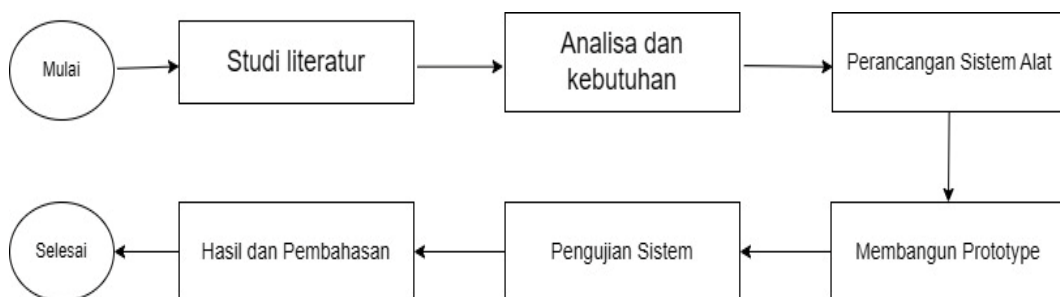
BAB III

METODOLOGI PENELITIAN

3.1. Metode Penelitian

Dalam perancangan kandang kucing pintar berbasis IoT ini, metode prototyping dipilih karena kemampuannya dalam memberikan gambaran awal terhadap sistem yang dikembangkan. Prototipe yang dibuat dapat diuji dan dievaluasi oleh pengguna, yang memungkinkan proses perbaikan berdasarkan umpan balik yang diterima, sehingga sistem yang dihasilkan dapat lebih efektif dan sesuai dengan kebutuhan pengguna.

3.2. Alur Penelitian



Gambar 3.1 Alur Penelitian

Pada gambar 3.1 menjelaskan dimulainya alur penelitian ini dilakukan dengan pendekatan studi literatur, yang bertujuan untuk mengumpulkan dan menganalisis berbagai teori yang terdapat dalam buku, jurnal ilmiah, serta sumber pustaka lainnya yang memiliki relevansi dengan topik penelitian, yaitu mengenai sistem kandang kucing pintar yang memanfaatkan ESP32. Studi ini mencakup konsep *Internet of Things* (IoT), mikrokontroler, berbagai jenis sensor yang akan digunakan.

Setelah itu dilanjutkan dengan menganalisa dan kebutuhan. Pada tahap ini, proses pengumpulan data dilakukan melalui riset awal yang dilanjutkan dengan

observasi terhadap pemilik kucing. Observasi tersebut bertujuan untuk menganalisis permasalahan yang timbul dalam merawat kucing, khususnya ketika pemilik memiliki keterbatasan waktu. Berdasarkan analisis tersebut, didefinisikan kebutuhan fungsional sistem yang akan dirancang, meliputi fitur pemberian pakan terjadwal, pemberian minum otomatis, pemantauan aktivitas *litter box*, dan kemampuan monitoring *real-time* melalui *dashboard website*, sesuai dengan batasan masalah yang ditetapkan.

Proses selanjutnya setelah kebutuhan terpenuhi dilakukan perancangan sistem dan perangkat dalam menerapkan konsep *Internet of Things* pada Kandang Kucing Pintar berbasis mikrokontroler ESP32. Pada tahapan ini dilakukan dalam menyusun rencana dan menggambarkan desain dari sistem kandang kucing pintar yang akan dibangun. Proses perancangan dimulai dengan pembuatan flowchart yang menggambarkan alur proses secara keseluruhan, untuk memberikan gambaran visual mengenai bagaimana sistem tersebut bekerja. Selain itu, skema alur kerja yang berfungsi untuk menggambarkan hubungan antar elemen-elemen yang ada dalam sistem.

Dalam tahap pembuatan prototipe, perancangan sistem disesuaikan dengan kebutuhan pengguna. Prototipe yang dibangun mencakup tiga fitur utama, yaitu pemberian pakan secara terjadwal, otomatisasi penyediaan air minum, serta pemantauan dan penanganan bau pada litterbox. Setiap fitur didukung oleh sejumlah perangkat seperti mikrokontroler ESP32, servo, sensor ultrasonik HC-SR04, *load cell*, sensor *water level*, *diffuser*, pompa air mini, dan relay. Selain itu, sistem ini dilengkapi dengan pengendali jarak jauh yang dapat diakses melalui *dashboard website*, sehingga pengguna dapat mengontrol kandang secara efisien.

Setelah perancangan dan pembuatan prototipe alat telah selesai dilakukan, maka dilakukan lah tahap pengujian. Pengujian sistem dilakukan dengan cara menguji setiap sensor dan komponen lainnya untuk memastikan bahwa hasil yang diperoleh sesuai dengan kinerja maksimal dari sistem yang telah dirancang dan dibangun. Secara keseluruhan, tujuan utama dari pengujian sistem ini adalah untuk meningkatkan kualitas serta keandalan sistem, sekaligus memastikan bahwa sistem yang dikembangkan dapat memenuhi kebutuhan dan harapan pengguna dengan optimal.

Data dan temuan yang diperoleh dari tahap pengujian sistem dikumpulkan, dianalisis, dan disajikan secara sistematis. Pada tahap ini, dilakukan pembahasan mengenai hasil pengujian, mengevaluasi keberhasilan sistem dalam menjawab rumusan masalah, mengidentifikasi kelebihan dan kekurangan sistem yang dibangun.

3.3. Analisis Kebutuhan Alat

Dalam penelitian ini adapun alat dan bahan yang digunakan untuk merancang dan mengimplementasikan sistem kandang kucing pintar berbasis *Internet of Things* (IoT). Berikut ini adalah tabel yang memuat alat dan bahan yang digunakan dalam penelitian ini.

Tabel 3.1 Alat dan Bahan

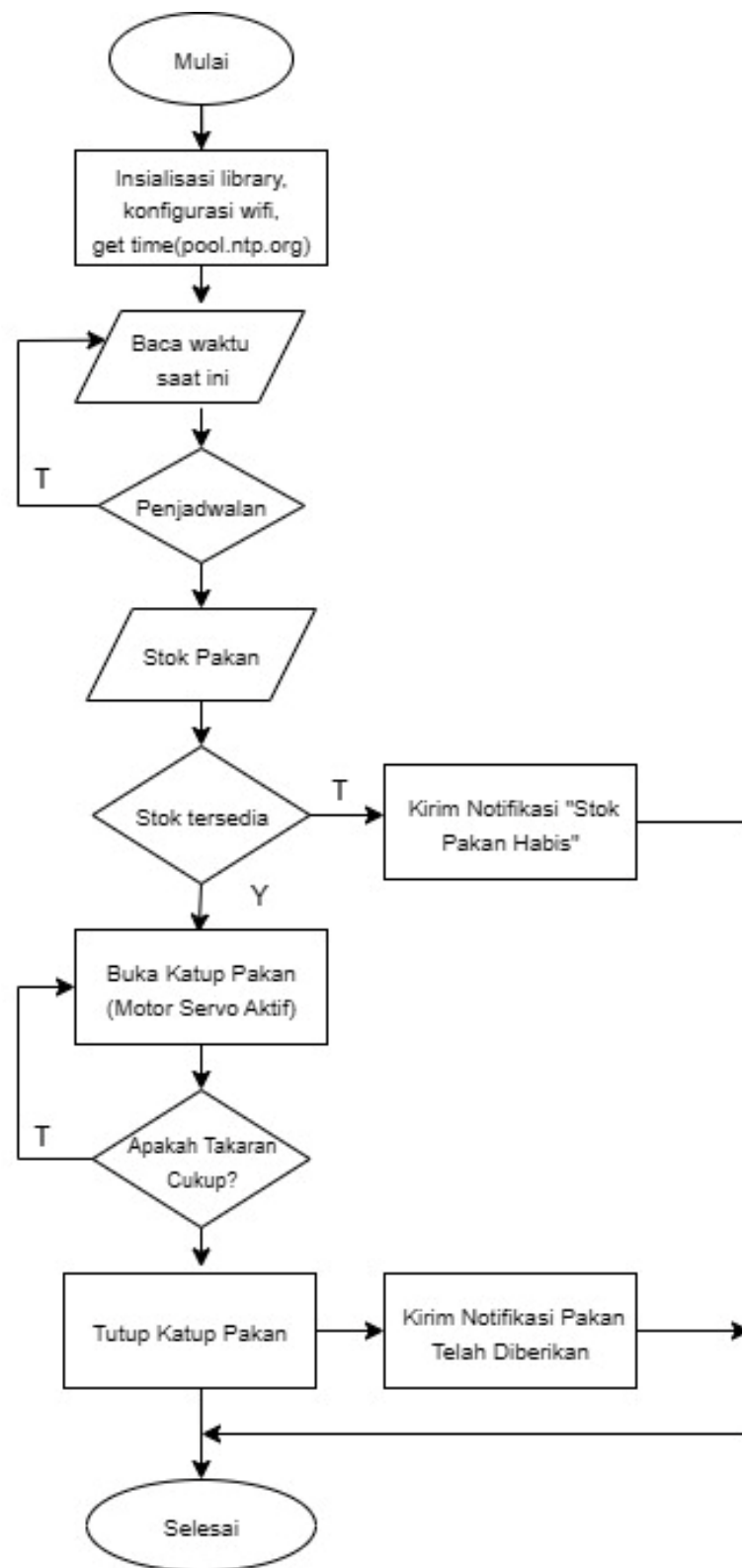
No	Alat dan Bahan	Fungsi
1	ESP32 Devkit	Sebagai mikrokontroler dalam menghubungkan semua komponen dan internet.
2	Ultrasonic HC-SR04	digunakan untuk mengukur ketinggian pakan di dalam tabung pakan serta mendeteksi keberadaan objek berupa kucing pada area litterbox.
3	Sensor water level	berfungsi untuk mengukur ketinggian air pada wadah tempat minum kucing.
4	Loadcell	Digunakan untuk mengukur berat pakan yang terdapat di dalam wadah.

No	Alat dan Bahan	Fungsi
5	Relay	Digunakan untuk menghubungkan dan memutus aliran arus listrik dalam proses pengoperasian pompa air mini serta diffuser.
6	Pompa air mini	berfungsi untuk mengalirkan air dari tangki penampungan menuju ke wadah tempat minum
7	Servo	digunakan untuk menggerakkan mekanisme buka-tutup pada katup tabung pakan.
8	LED Merah dan Hijau	Sebagai indikator visual untuk status sistem, seperti status koneksi WiFi (terhubung/terputus).
10	Ultrasonic Mist Maker	untuk mengeluarkan uap air yang mengandung cairan anti bau dengan tujuan sebagai upaya mengurangi bau tidak sedap di area litterbox
11	Power Adaptor USB	Sebagai sumber catu daya (<i>power supply</i>) untuk menghidupkan seluruh rangkaian sistem melalui mikrokontroler ESP32
12	Kabel Jumper	Untuk menghubungkan antar komponen elektronik pada <i>breadboard</i> atau langsung ke pin mikrokontroler selama tahap pembuatan prototipe.

3.4. Perancangan Alur Kerja Sistem

Pada tahapan ini, penelitian akan mengidentifikasi dan menganalisis tiga alur sistem utama yang terkait dengan fungsi operasional, yang kemudian akan digambarkan dalam bentuk flowchart. Ketiga alur sistem tersebut meliputi sistem pakan, sistem minum, dan sistem monitoring litterbox. Masing-masing sistem ini memiliki peran penting dalam mendukung efisiensi dan efektivitas keseluruhan proses, yang akan dievaluasi lebih lanjut melalui penggambaran alur kerja secara terstruktur dalam flowchart untuk mempermudah pemahaman dan implementasi.

3.4.1. Flowchart Sistem Pakan



Gambar 3.2 Flowchart Sistem Pakan

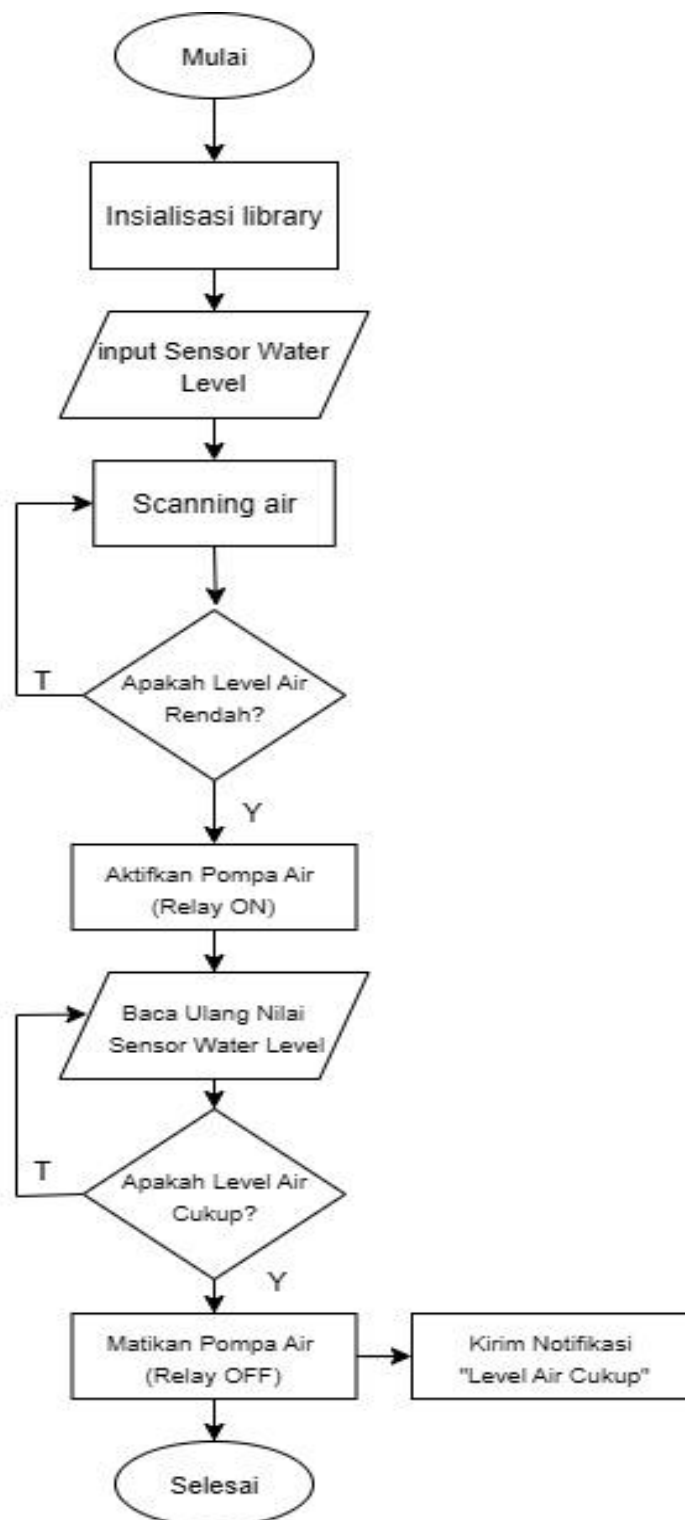
Penjelasan pada gambar 3.2 yaitu Alur kerja pada flowchart sistem pakan diawali dengan tahap inisialisasi. Pada tahap ini, dieksekusi pemanggilan *library* yang relevan, konfigurasi jaringan WiFi, serta sinkronisasi waktu sistem dengan pool.ntp.org untuk memastikan akurasi penjadwalan.

Setelah proses inisialisasi selesai, sistem memasuki alur perulangan utama dengan membaca waktu saat ini secara kontinu. Waktu tersebut kemudian divalidasi terhadap jadwal pakan yang telah diprogram. Apabila waktu aktual belum sesuai dengan jadwal yang ditentukan, sistem akan kembali melakukan pembacaan waktu. Jika kondisi penjadwalan terpenuhi, alur akan dilanjutkan ke tahap berikutnya.

Selanjutnya, sistem akan melakukan pengecekan terhadap stok pakan. Kondisi ketersediaan stok kemudian diverifikasi. Jika stok terdeteksi tidak tersedia (habis), sistem akan mengirimkan notifikasi "Stok Pakan Habis" dan alur proses akan diterminasi. Apabila stok pakan dinyatakan tersedia, mikrokontroler akan memberikan perintah untuk membuka katup pakan dengan mengaktifkan motor servo. Selama katup terbuka, sistem akan melakukan verifikasi secara berulang untuk memastikan takaran pakan telah tercukupi. Setelah takaran yang ditetapkan terpenuhi, perintah selanjutnya adalah menutup katup pakan.

Sebagai tahap akhir, sebuah notifikasi bahwa "Pakan telah Diberikan" akan dikirimkan kepada pengguna, yang menandakan selesainya satu siklus proses pemberian pakan.

3.4.2. Flowchart Sistem Minum



Gambar 3.3 Flowchart Sistem Minum

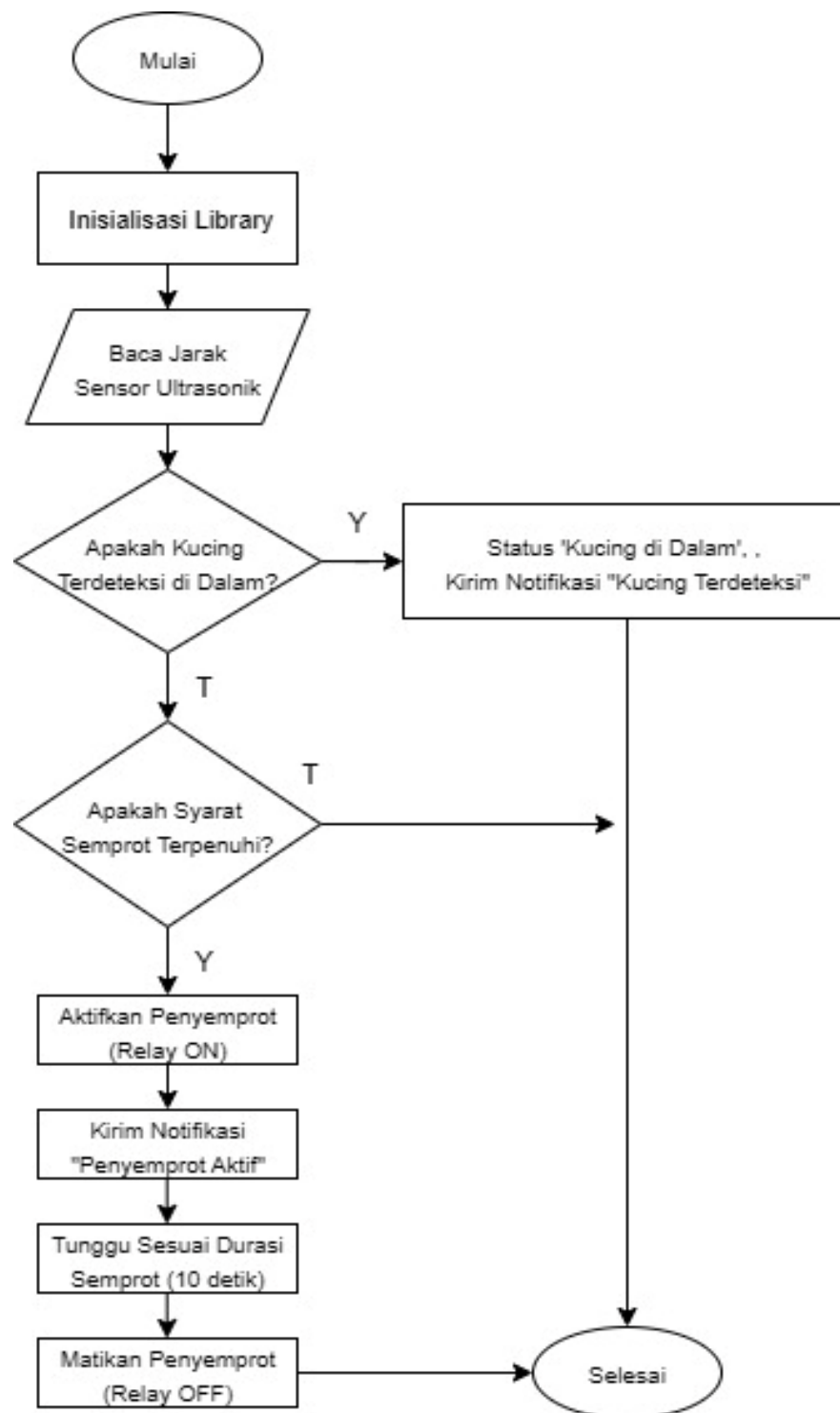
Penjelasan pada gambar 3.3 Alur kerja sistem minum, sebagaimana digambarkan pada flowchart, diawali dengan tahap inisialisasi library yang dibutuhkan oleh mikrokontroler. Setelah inisialisasi, sistem melakukan pembacaan awal melalui input Sensor Water Level dan memasuki siklus pemantauan dengan melakukan proses "Scanning air" secara kontinu.

Pada siklus ini, sistem akan terus melakukan validasi kondisi dengan pertanyaan "Apakah Level Air Rendah?". Jika level air terdeteksi cukup (kondisi 'T' atau Tidak), sistem akan terus melanjutkan pemantauan. Namun, jika level air terdeteksi rendah (kondisi 'Y' atau Ya), sistem akan memulai proses pengisian.

Proses pengisian diawali dengan mengaktifkan pompa air dengan menyalakan relay (Relay ON). Selama pompa aktif, sistem akan secara berulang membaca ulang nilai dari Sensor Water Level untuk memantau kenaikan level air. Pembacaan ini divalidasi dengan pertanyaan "Apakah Level Air Cukup?". Jika belum cukup (kondisi 'T'), proses pembacaan ulang akan terus berlanjut.

Apabila level air terdeteksi sudah cukup (kondisi 'Y'), alur dilanjutkan untuk menghentikan pengisian. Sistem kemudian akan mematikan pompa air dengan menonaktifkan relay (Relay OFF) dan mengirimkan notifikasi "Level Air Cukup" kepada pengguna. Tahap ini menandakan bahwa satu siklus pengisian telah **Selesai** dan sistem kembali ke mode pemantauan.

3.4.3. Flowchart Sistem Monitoring dan Penanganan Bau Litterbox



Gambar 3.4 Flowchart Sistem Monitoring dan Penanganan Bau Litterbox

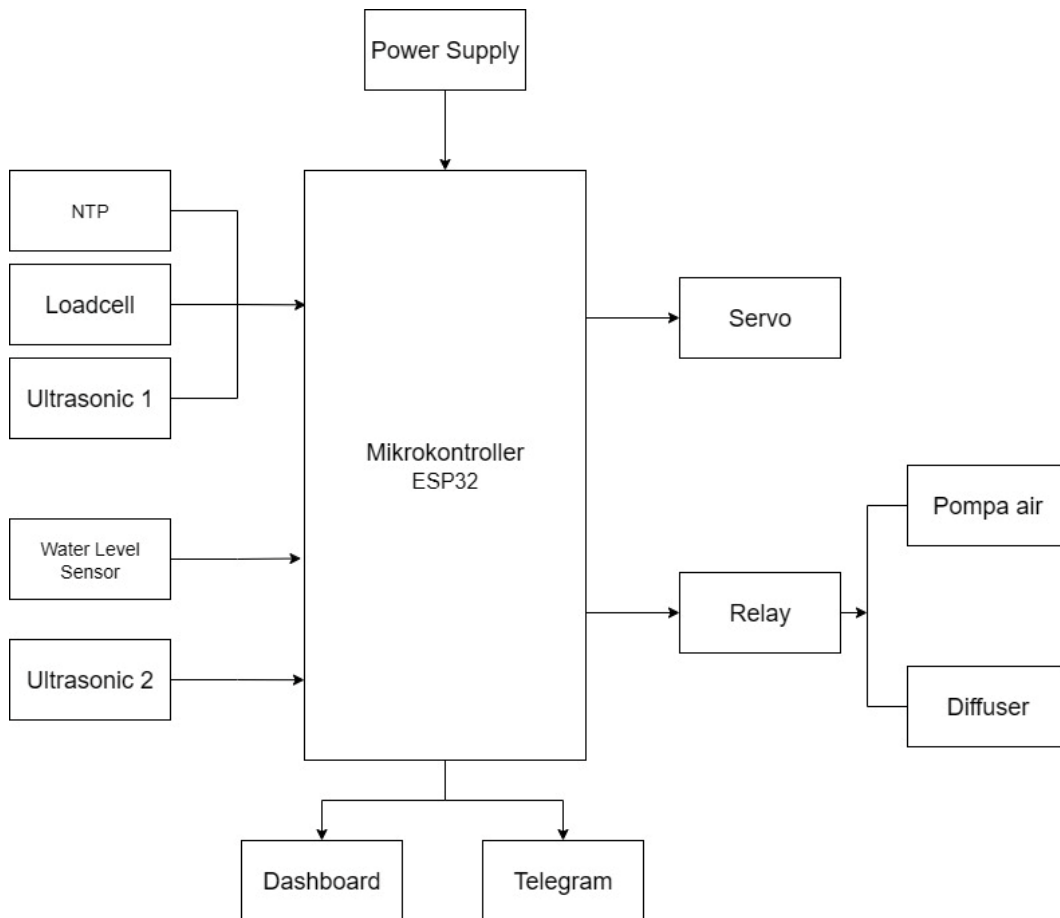
Penjelasan pada gambar 3.4 yaitu alur flowchart diawali dengan tahap inisialisasi library. Setelah itu, sistem akan secara kontinu membaca jarak melalui sensor ultrasonik untuk memantau area *litterbox*.

Data jarak yang diperoleh kemudian digunakan dalam validasi kondisi pertama, yaitu "Apakah Kucing Terdeteksi di Dalam?". Jika kucing terdeteksi (kondisi 'Y' atau Ya), sistem akan memperbarui status internal menjadi 'Kucing di Dalam' dan mengirimkan notifikasi "Kucing Terdeteksi" kepada pengguna, kemudian siklus pemantauan untuk saat itu dianggap selesai.

Apabila pada validasi pertama kucing tidak terdeteksi (kondisi 'T' atau Tidak), alur akan dilanjutkan ke validasi kedua, yaitu "Apakah Syarat Semprot Terpenuhi?". Keputusan ini didasarkan pada logika internal sistem yang mempertimbangkan apakah kucing sebelumnya telah menggunakan *litterbox* dan telah meninggalkan area tersebut selama durasi waktu tunggu yang ditentukan. Jika syarat tidak terpenuhi (kondisi 'T'), maka tidak ada aksi yang diambil dan siklus pemantauan selesai.

Namun, jika syarat terpenuhi (kondisi 'Y'), sistem akan memulai mekanisme penanganan bau. Proses ini diawali dengan mengaktifkan penyemprot (Relay ON) dan mengirimkan notifikasi "Penyemprot Aktif". Sistem kemudian akan menunggu sesuai durasi semprot yang telah ditetapkan, yaitu selama 10 detik. Setelah durasi tercapai, sistem akan mematikan penyemprot (Relay OFF), yang menandakan berakhirnya alur penanganan bau hingga siklus pemantauan berikutnya.

3.5. Blok Diagram Sistem



Gambar 3.5 Blok Diagram Sistem

Pada gambar 3.5 terdapat diagram blok yang memiliki 3 bagian yaitu input, proses dan output yang menggambarkan proses dari perancangan kandang kucing pintar. Berikut keterangan dari penjelasan diagram blok diatas ;

1. Sistem kontrol pakan

Dalam sistem pemberian pakan ini, terdapat tiga input, yaitu *Network Time Protocol* (NTP) yang berfungsi untuk menyinkronkan waktu secara akurat melalui internet. Sensor ultrasonik digunakan untuk mengukur ketinggian pakan di dalam tabung pakan, sementara *loadcell* berfungsi untuk mengukur berat pakan yang terdapat dalam wadah pakan. Pada bagian proses, mikrokontroler ESP32 bertanggung jawab untuk memproses semua

sinyal yang diterima dari sensor-sensor tersebut. Pada bagian output, terdapat servo yang digunakan untuk menggerakkan mekanisme, seperti membuka atau menutup katup pada tabung pakan, yang telah diproses oleh mikrokontroler ESP32.

2. Sistem otomasi minum

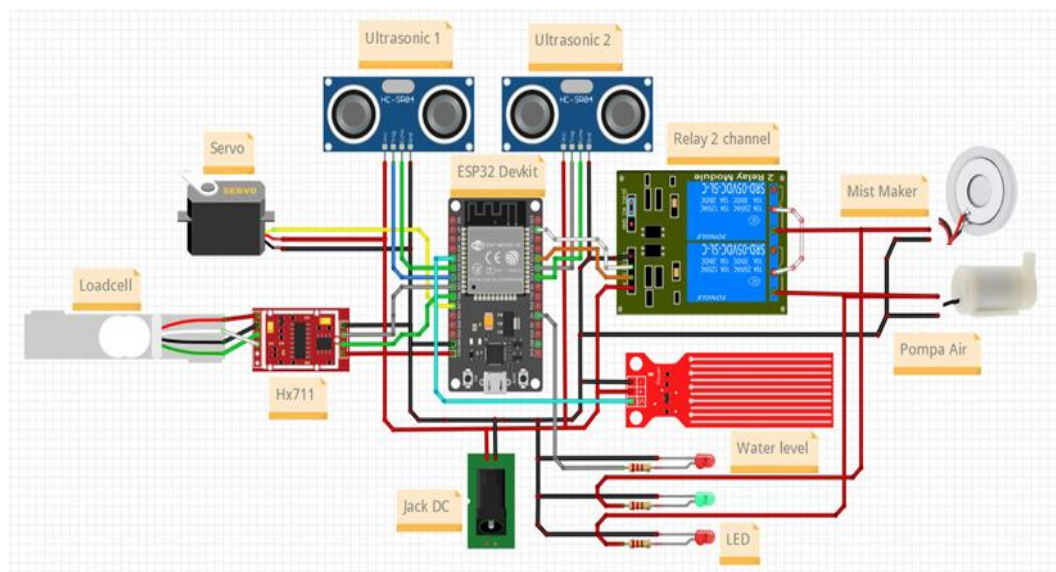
Dalam sistem otomasi minum ini, terdapat satu input, yaitu sensor *water level* yang berfungsi untuk mendeteksi ketinggian air dalam wadah minum. Pada bagian proses, mikrokontroler ESP32 memproses hasil nilai inputan dari sensor *water level*. Pada bagian output, terdapat relay yang akan menghidupkan pompa air.

3. Sistem monitoring dan penanganan bau litterbox

Dalam sistem monitoring litterbox ini, terdapat satu input, yaitu sensor ultrasonic yang berfungsi untuk mendeteksi objek berupa kucing di *litterbox*. Pada bagian proses, mikrokontroler ESP32 memproses hasil inputan dari sensor. Pada bagian output, terdapat relay yang akan menghidupkan *diffuser* untuk mengeluarkan uap air yang mengandung cairan anti bau untuk upaya mengurangi bau di area *litterbox*.

3.6. Perancangan Rangkaian Alat

Tahap perancangan rangkaian alat merupakan visualisasi dari arsitektur sistem yang telah dirancang. Rangkaian ini mengintegrasikan seluruh komponen perangkat keras yang dibutuhkan untuk ketiga fitur utama, sistem pakan, sistem minum, serta sistem monitoring dan penanganan bau litterbox ke dalam satu unit kontrol terpusat menggunakan mikrokontroler ESP32. Diagram rangkaian keseluruhan dari sistem dapat dilihat pada Gambar 3.6.



Gambar 3.6 Rangkaian Keseluruhan Sistem

Pada gambar 3.6 dapat dilihat berdasarkan keterangan berikut:

1. Sensor *ultrasonic* terhubung dengan ESP32 dengan 4 pin yaitu Vcc, Gnd, Pin Trig terhubung pada pin D33, dan Pin ECHO terhubung dengan Pin D32. Sensor *ultrasonic* ini berfungsi untuk mengukur ketinggian pakan pada wadah penampungan pakan.
2. Sensor *loadcell* terhubung dengan ke modul HX711 dengan 4 pin yaitu kabel merah terhubung ke pin E+, kabel hitam terhubung ke pin E-, kabel hijau terhubung ke pin A+ dan kabel putih terhubung ke pin A-. Sensor *load cell* berfungsi untuk mengukur berat pada wadah pakan yang telah ditentukan oleh sistem.
3. Modul HX711 terhubung dengan ESP32 dengan 4 pin yaitu Vcc, Gnd, pin CLK terhubung ke pin D26 dan pin DT terhubung ke pin D25. Modul HX711 ini berfungsi sebagai penguat data yang dihasilkan oleh sensor *loadcell*.

4. Motor servo terhubung dengan ESP32 dengan 3 pin yaitu Vcc, Gnd dan pin data terhubung ke pin D27. Motor servo berfungsi sebagai penggerak buka tutup pada katup penampungan pakan.
5. Sensor *water level* terhubung dengan ESP32 dengan 3 pin yaitu Vcc, Gnd dan pin sinyal terhubung pada pin D4. Sensor ini berfungsi sebagai mendeteksi ketinggian air pada wadah minum.
6. Relay terhubung dengan ESP32 dengan 3 pin yaitu Vcc, Gnd dan pin In terhubung pada pin D13. Relay berfungsi sebagai menyambungkan atau memutuskan arus dan menghidupkan pompa air mini.
7. Pompa air mini terhubung ke relay dengan 1 pin yaitu kabel hitam ke terminal COM dan terminal NO terhubung ke Gnd. Pompa air mini ini berfungsi sebagai mengalirkan air dari penampungan ke wadah minum.
8. Sensor ultrasonic terhubung dengan ESP32 yaitu dengan 4 pin yaitu Vcc, Gnd, Pin Trig terhubung pada pin D19, dan pin Echo terhubung pada pin D18. Sensor ultrasonic berfungsi sebagai mendeteksi keberadaan kucing dalam litterbox.
9. Relay terhubung dengan ESP32 dengan 3 pin yaitu Vcc, Gnd dan pin In terhubung pada pin D21. Relay berfungsi sebagai menyambungkan atau memutuskan arus dan menghidupkan *ultrasonic mist maker* sebagai *diffuser*.
10. *Ultrasonic mist maker* terhubung ke relay dengan 1 pin yaitu kabel hitam ke terminal COM dan terminal NO terhubung ke Gnd. *Ultrasonic mist maker* berfungsi sebagai *diffuser* yang mengeluarkan uap air untuk sebagai upaya mengurangi bau sekitar area *litterbox*.

3.7. Perancangan Antarmuka Pengguna (User Interface)

Selain perancangan pada level perangkat keras dan alur logika sistem, aspek krusial lain dalam penelitian ini adalah perancangan antarmuka pengguna (User Interface). Antarmuka ini memegang peranan vital sebagai jembatan interaksi yang memungkinkan pengguna untuk memonitor dan mengendalikan seluruh fungsionalitas sistem kandang pintar secara intuitif dan efisien. Perancangan antarmuka yang baik menjadi kunci untuk memastikan bahwa teknologi yang diimplementasikan dapat secara efektif menjawab permasalahan yang dihadapi oleh pengguna, yaitu kemudahan dalam melakukan perawatan hewan peliharaan dari jarak jauh. Oleh karena itu, penelitian ini mengembangkan dua platform antarmuka yang saling melengkapi: sebuah dashboard website yang komprehensif sebagai pusat kendali dan monitoring utama, serta Bot Telegram yang berfungsi sebagai media penyampaian notifikasi instan untuk kejadian-kejadian yang memerlukan perhatian segera.

3.7.1. Perancangan Antarmuka Dashboard Website

Antarmuka *dashboard website* dirancang dengan pendekatan minimalis menggunakan tata letak berbasis kartu (*card-based layout*). Tujuannya adalah agar semua informasi vital dan kontrol utama dapat diakses dengan cepat oleh pengguna. Setiap kartu didedikasikan untuk satu fungsi spesifik, seperti monitoring pakan, jadwal, atau kontrol manual.



Gambar 3.7 Tampilan Dashboard Website

3.7.2. Perancangan Interaksi Bot Telegram

Berbeda dengan *dashboard website* yang berfungsi sebagai pusat kontrol interaktif, interaksi dengan Bot Telegram dirancang sebagai kanal komunikasi satu arah yang bersifat proaktif dari sistem kepada pengguna. Platform Telegram dipilih karena beberapa keunggulann kemampuannya mengirimkan notifikasi *push* secara instan ke perangkat seluler pengguna, API (Application Programming Interface) yang matang dan mudah diintegrasikan dengan mikrokontroler seperti ESP32, dan penggunaannya yang luas sehingga tidak memberatkan pengguna untuk menginstal aplikasi baru. Fungsi utama bot ini adalah untuk memberikan peringatan dan laporan status secara *real-time* mengenai kejadian-kejadian penting yang terdeteksi oleh sistem.

Model interaksi yang dirancang bersifat *event-driven*, di mana sistem akan secara otomatis mengirimkan pesan melalui bot ketika sebuah peristiwa (event) spesifik terdeteksi oleh sensor atau saat sebuah aksi otomatis dijalankan. Setiap jenis peristiwa memiliki format pesan notifikasi yang telah ditentukan untuk memberikan informasi yang jelas dan ringkas kepada pengguna. Berdasarkan

model interaksi *event-driven* tersebut, dirancang beberapa skenario notifikasi yang akan dikirimkan oleh sistem. Kondisi pemicu dan contoh format pesan untuk setiap notifikasi dirincikan pada Tabel 3.2

Tabel 3.2 Skenario dan Pesan Notifikasi Bot Telegram

No.	Kondisi Pemicu	Pesan Notifikasi
1	Pemberian pakan (otomatis/manual) berhasil.	Pakan Pagi telah diberikan.
2	Pemberian pakan dilewati karena mangkuk penuh.	Pemberian makan Pagi dilewati, mangkuk masih penuh.
3	Stok pakan di tabung utama terdeteksi habis.	PERINGATAN: Stok pakan di tabung utama habis! Segera isi ulang.
4	Pompa air diaktifkan untuk mengisi tempat minum.	Pompa air aktif untuk mengisi tempat minum.
5	Pompa air dimatikan karena tandon utama kosong.	KRITIS: Air di tandon utama kosong! Pompa dimatikan.
6	Kucing terdeteksi masuk ke dalam <i>litterbox</i> .	Kucing terdeteksi di box pasir.
7	Penyemprot pengharum diaktifkan setelah kucing keluar.	Semprotan penghilang bau telah diaktifkan.
8	Jadwal makan berhasil diubah dari <i>dashboard</i> .	Jadwal makan pagi berhasil di-update dari dashboard.
9	Sistem kembali online setelah koneksi terputus.	Sistem kembali online & terhubung.

3.8. Teknik Pengumpulan Data

3.8.1. Observasi

Salah satu metode pengumpulan data yang digunakan dalam penelitian ini adalah observasi, di mana peneliti melakukan pengamatan langsung terhadap topik penelitian. Observasi ini bertujuan untuk melengkapi dan menyaring fakta-fakta yang relevan dengan permasalahan yang diteliti. Pada penelitian ini, observasi dilakukan di rumah pemilik kucing yang terletak di Jl. Kapten Rahmad Buddin, Kelurahan Terjun, Kecamatan Medan Marelan. Pemilik kucing mengalami kesulitan dalam memonitoring dan memberikan pakan pada kucingnya, terutama karena seringkali kegiatan atau pekerjaan berada di luar rumah.

3.8.2. Studi Pustaka

Proses ini dilakukan dengan cara menganalisis dan mempelajari berbagai referensi yang relevan, seperti jurnal ilmiah, serta penelitian-penelitian terdahulu yang memiliki kaitan erat dengan topik permasalahan yang akan dikaji. Dengan mempelajari referensi-referensi tersebut, diharapkan dapat diperoleh konsep-konsep serta teori-teori yang mendasari permasalahan yang sedang diteliti. Penelaahan terhadap literatur yang ada ini akan memberikan pemahaman yang lebih mendalam dan landasan teori yang kuat untuk mendukung penelitian, serta membantu dalam merancang pendekatan dan metodologi yang tepat untuk menyelesaikan permasalahan yang dihadapi.

3.9. Tempat dan Jadwal Penelitian

Tempat penelitian ini dilakukan di Jl. Kapten Rahmad Buddin, Kelurahan Terjun, Kecamatan Medan Marelan, dan setiap rancangan penelitian perlu

dilengkapi dengan jadwal kegiatan yang akan dilaksanakan, yang dapat dilihat pada tabel 3.3 dalam proposal ini.

Tabel 3.3 Jadwal Penelitian

[illegible]

BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil Implementasi Sistem

Hasil implementasi ini mencakup visualisasi fisik prototipe, deskripsi detail konektivitas antar komponen, serta demonstrasi antarmuka pengguna yang menjadi jembatan interaksi antara pemilik kucing dan sistem yang dibangun. Dengan demikian, bab ini akan memaparkan secara terperinci bagaimana konsep perancangan dapat diwujudkan dalam sebuah sistem yang siap untuk diuji.

4.1.1. Implementasi Perangkat Keras (Hardware)

Perancangan perangkat keras merupakan realisasi fisik dari diagram rangkaian yang telah disusun pada Bab III. Pada tahap ini, semua komponen elektronik dirakit dan diintegrasikan ke dalam satu unit kontrol terpusat menggunakan mikrokontroler ESP32 sebagai otak sistem. Prototipe kandang kucing pintar ini dibagi menjadi tiga subsistem utama, yaitu sistem pakan, sistem minum, dan sistem monitoring *litterbox*. Keseluruhan rangkaian dipasang pada sebuah *breadboard* dan disuplai daya melalui *power adaptor* USB. Berikut adalah deskripsi detail dari setiap subsistem:

1. Sistem Pemberian Pakan

Sistem pakan diimplementasikan dengan menggabungkan sensor ultrasonik, modul *load cell* HX711, dan motor servo. Sebuah sensor ultrasonik (Ultrasonic 1) ditempatkan di bagian atas tabung pakan untuk mengukur ketinggian pakan. Untuk menimbang takaran pakan, digunakan *load cell* yang terpasang di bawah mangkuk pakan. Mekanisme pembuka dan penutup katup pakan digerakkan oleh sebuah motor servo. Gambar 4.1

menunjukkan implementasi fisik dari sistem pemberian pakan otomatis yang terintegrasi dengan wadah dan tabung pakan di dalam kandang.



Gambar 4.1 Implementasi Sistem Pemberian Pakan

2. Sistem Minum

Subsistem otomasi minum berfungsi untuk menjaga ketersediaan air bersih di wadah minum kucing secara otomatis. Sistem ini diimplementasikan dengan memadukan sensor *water level* sebagai input, serta pompa air mini dan relay sebagai aktuator. Sensor *water level* dipasang di dalam wadah minum untuk mendeteksi ketinggian air. Ketika level air terdeteksi rendah, mikrokontroler akan mengaktifkan relay, yang kemudian menyalakan pompa air mini. Pompa ini akan mengalirkan air dari tangki penampungan utama ke wadah minum hingga level air mencapai batas yang ditentukan, lalu secara otomatis dimatikan. Gambar 4.2 menunjukkan implementasi fisik dari sistem ini.



Gambar 4.2 Implementasi Sistem Otomasi Minum

3. Sistem Monitoring dan Penanganan Bau Litterbox

Subsistem ini dirancang untuk memantau aktivitas kucing di area *litterbox* dan secara proaktif menangani masalah bau yang muncul. Implementasinya melibatkan sensor ultrasonik dan *ultrasonic mist maker* yang dikendalikan oleh relay. Sensor ultrasonik diposisikan di atas *litterbox* untuk mendeteksi keberadaan kucing saat menggunakan area tersebut. Setelah kucing meninggalkan area dan jeda waktu yang telah ditentukan terpenuhi, mikrokontroler akan mengaktifkan relay untuk menyalakan *ultrasonic mist maker*. Perangkat ini akan mengeluarkan uap halus yang mengandung cairan anti bau, membantu menjaga kebersihan dan kenyamanan lingkungan kandang. Gambar 4.3 menampilkan implementasi fisik dari subsistem ini, yang berfokus pada kesehatan dan kenyamanan kucing.



Gambar 4.3 Implementasi Sistem Monitoring dan Penanganan Bau Litterbox.

4. Integrasi Keseluruhan Sistem Perangkat Keras

Setelah setiap subsistem dirakit dan diuji secara terpisah, langkah selanjutnya adalah mengintegrasikannya ke dalam satu unit kandang kucing pintar.



Gambar 4.4 Kandang Kucing Pintar dengan Sistem Terintegrasi

Gambar 4.4 memperlihatkan tampilan fisik keseluruhan prototipe kandang kucing pintar yang telah dilengkapi dengan sistem pemberian pakan otomatis, sistem otomasi minum, serta sistem monitoring dan penanganan bau litterbox. Penempatan setiap komponen dipertimbangkan untuk fungsionalitas yang optimal dan kenyamanan kucing.

4.1.2. Implementasi Perangkat Lunak (Software)

Setelah perakitan perangkat keras, tahapan selanjutnya adalah mengimplementasikan perangkat lunak yang berfungsi sebagai sistem kendali utama. Implementasi ini mencakup pemrograman mikrokontroler ESP32 dan pengembangan antarmuka pengguna, yaitu dashboard website dan bot Telegram. Program ini dirancang untuk mengintegrasikan semua komponen perangkat keras, memproses data sensor, menjalankan logika otomatisasi yang kompleks, serta memfasilitasi komunikasi dua arah antara sistem dan pemilik kucing. Berikut adalah detail implementasi fungsionalitas utama dari perangkat lunak yang dibangun.

1. Pengelolaan Data Sensor

Mengambil data *real-time* dari sensor ultrasonik, *load cell*, dan *water level*. Data ini mencakup ketinggian pakan, berat pakan, dan level air minum.

2. Penjadwalan dan Logika Otomatis

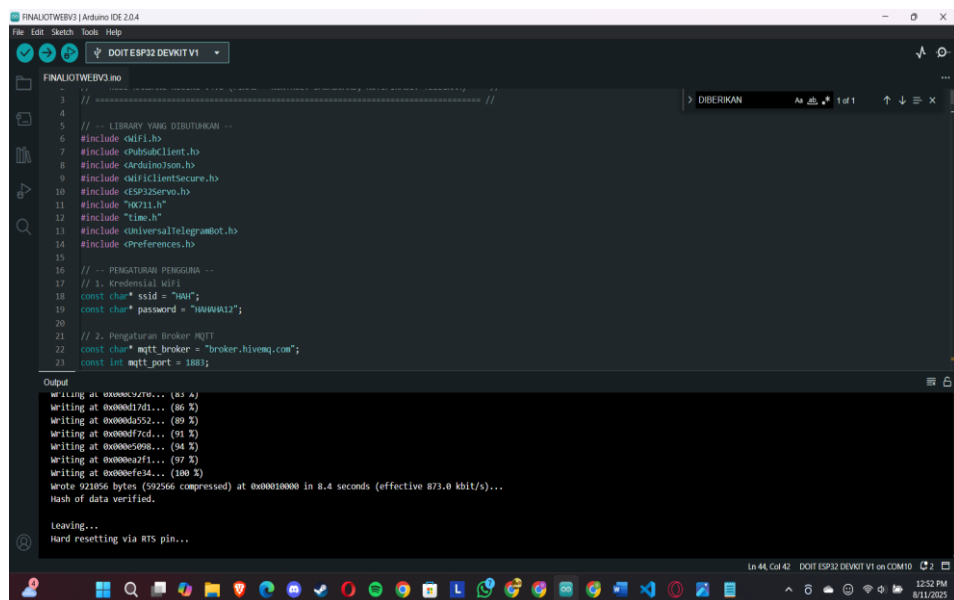
Menggunakan Network Time Protocol (NTP) untuk menjalankan pemberian pakan otomatis pada jadwal yang telah ditentukan (pagi, siang, malam). Serta mengelola logika pengisian air minum dan penyemprotan anti bau.

3. Koneksi Jarak Jauh

Mengelola koneksi Wi-Fi dan protokol MQTT untuk mengirimkan data dan log aktivitas ke dashboard website, sekaligus menerima perintah kontrol dari dashboard.

4. Sistem Notifikasi

Mengirimkan notifikasi penting secara proaktif melalui Telegram Bot ke perangkat pengguna saat terjadi peristiwa tertentu, seperti pakan habis atau kucing terdeteksi di *litterbox*.



Gambar 4.5 Tampilan Arduino IDE dengan Kode Berhasil Diunggah

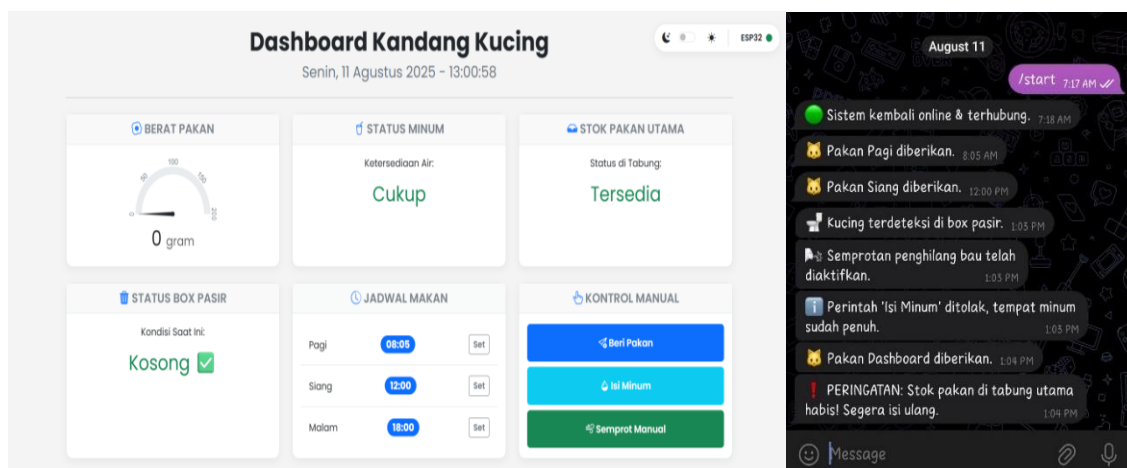
4.1.3. Tampilan Antarmuka Sistem

Dalam penelitian ini, antarmuka sistem dibagi menjadi dua platform utama yang saling melengkapi, yaitu dashboard website yang berfungsi sebagai pusat kendali komprehensif, dan bot Telegram sebagai notifikasi. Tampilan-tampilan ini dirancang untuk menyajikan data sensor secara *real-time* dan memungkinkan pengguna untuk mengintervensi sistem dari jarak jauh, sehingga efektivitas

perawatan kucing dapat tercapai secara maksimal. Terdapat dua media utama yang digunakan dalam tampilan antarmuka sistem ini:

1. Dashboard Website menampilkan data sensor secara visual dan ringkas melalui antarmuka berbasis kartu. Pengguna dapat melihat status pakan (berat dan stok), status air minum, status litterbox, serta melakukan kontrol manual dan mengubah jadwal pakan.
2. Notifikasi Telegram berfungsi sebagai komunikasi yang memberikan laporan status dan peringatan penting secara langsung ke perangkat seluler pengguna.

Pada gambar 4.6 menampilkan kedua media utama yang digunakan, yaitu dashboard website dan bot Telegram.



Gambar 4.6 Tampilan Dashboard Website dan Notifikasi Bot Telegram

4.2. Pengujian Sistem

Pengujian sistem dilakukan untuk mengevaluasi fungsionalitas dan kinerja prototipe kandang kucing pintar yang telah diimplementasikan. Tahap ini bertujuan untuk memastikan bahwa setiap subsistem pakan, minum, dan *litterbox* beroperasi sesuai dengan logika yang dirancang, serta untuk mengukur kinerja sistem secara keseluruhan, termasuk akurasi sensor dan responsivitas antarmuka IoT.

4.2.1. Pengujian Sistem Pakan

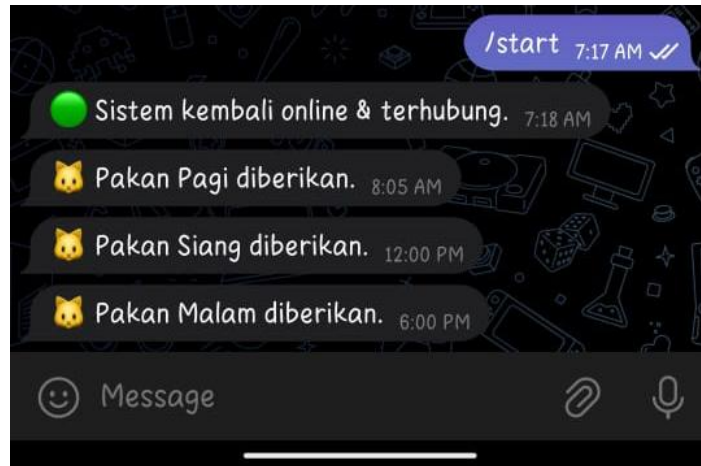
Pengujian fungsionalitas sistem pakan merupakan tahap evaluasi kritis untuk memverifikasi kinerja subsistem pemberian makan otomatis. Bagian ini berfokus pada dua aspek utama: keandalan sistem dalam menjalankan penjadwalan pakan secara otomatis sesuai dengan waktu yang telah ditentukan, serta akurasi takaran pakan yang diberikan ke dalam mangkuk. Data pengujian dikumpulkan untuk menganalisis tingkat kesalahan.

1. Uji Penjadwalan Pakan Otomatis

Pengujian ini dilakukan untuk memastikan sistem dapat memberikan pakan secara otomatis sesuai dengan jadwal yang telah ditetapkan menggunakan Network Time Protocol (NTP). Pengujian difokuskan pada validasi logika penjadwalan, di mana sistem harus mampu mengaktifkan mekanisme pemberian pakan tepat pada waktu yang ditentukan. Adapun hasil pengujian penjadwalan pakan dapat dilihat pada Tabel 4.1.

Tabel 4.1 Data Hasil Uji Penjadwalan Pakan Otomatis

No	Jadwal yang di Tetapkan	Waktu Pemberian Aktual	Keterangan
1	08:05	08:05:03	Berhasil
2	12:00	12:00:05	Berhasil
3	18:00	18:00:02	Berhasil



Gambar 4. 7 Notifikasi Pemberian Pakan

Berdasarkan Tabel 4.1, dapat dianalisis bahwa sistem berhasil menjalankan fungsi penjadwalan pakan otomatis dengan tingkat akurasi yang sangat baik. Setiap waktu yang ditetapkan, yaitu pukul 08:05, 12:00, dan 18:00, direspons oleh sistem dengan selisih waktu aktual yang sangat minim, rata-rata hanya dalam hitungan beberapa detik. Hal ini membuktikan bahwa sinkronisasi waktu menggunakan Network Time Protocol (NTP) berjalan secara efektif dan mikrokontroler ESP32 mampu mengeksekusi perintah sesuai dengan jadwal yang diprogramkan. Dengan demikian, pengujian ini berhasil memvalidasi fungsionalitas penjadwalan pakan otomatis.

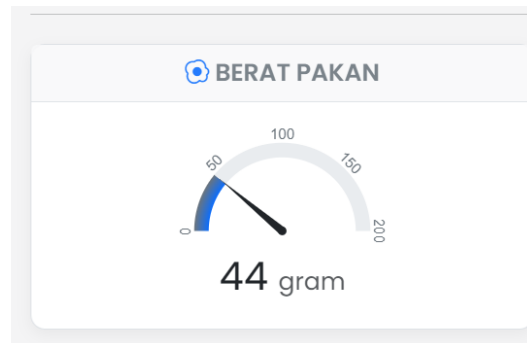
2. Uji Akurasi Takaran Pakan

Pengujian ini bertujuan untuk mengukur tingkat akurasi sistem dalam memberikan pakan sesuai takaran yang ditetapkan. Pengukuran dilakukan dengan menggunakan sensor *load cell* yang terpasang di bawah mangkuk pakan. Sistem diatur untuk mengeluarkan takaran pakan sebesar 40 gram, kemudian berat aktual yang terukur dicatat untuk menghitung

selisih dan persentase kesalahan (*error*). Hasil pengujian ini disajikan pada Tabel 4.2.

Tabel 4.2 Hasil Pengujian Akurasi Takaran Pakan

Percobaan	Takaran yang ditargetkan (gram)	Berat Terukur (gram)	Selisih (gram)	Error (%)
1	40	44	4	10
2	40	43	3	7.5
3	40	45	5	12.5
4	40	42	2	5
5	40	47	7	17.5
Rata rata		44.2	4.2	10.5



Gambar 4.8 Indikator Berat Pakan Pada Dashboard

Berdasarkan Tabel 4.2, rata-rata tingkat kesalahan (*error*) pada sensor load cell adalah 10.5%. Tingkat akurasi ini menunjukkan bahwa sistem mampu memberikan takaran pakan dengan presisi yang cukup baik dan dapat diandalkan untuk memenuhi kebutuhan pakan harian kucing. Tingkat kesalahan ini disebabkan oleh beberapa faktor, salah satunya adalah bentuk dan geometri dari perosotan pakan yang memengaruhi aliran pelet

pakan. Pada beberapa kasus, pelet pakan dapat menumpuk dan tidak jatuh secara merata, sehingga berat yang terukur bisa sedikit lebih besar dari target yang ditetapkan.

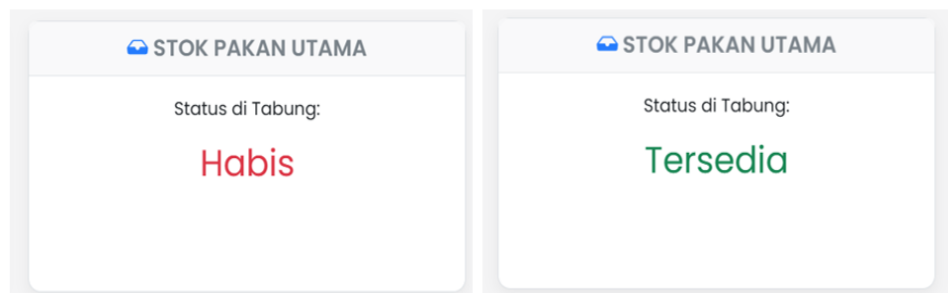
Meskipun demikian, dapat diamati bahwa seluruh percobaan berhasil mengeluarkan pakan dengan berat di atas takaran yang ditargetkan (40 gram). Hal ini memastikan bahwa kebutuhan pakan kucing terpenuhi dan tidak terjadi kekurangan porsi. Dengan demikian, fungsionalitas penakaran pakan, dari sisi keberhasilan pemberian porsi minimum, berhasil divalidasi.

3. Uji Deteksi Stok Pakan

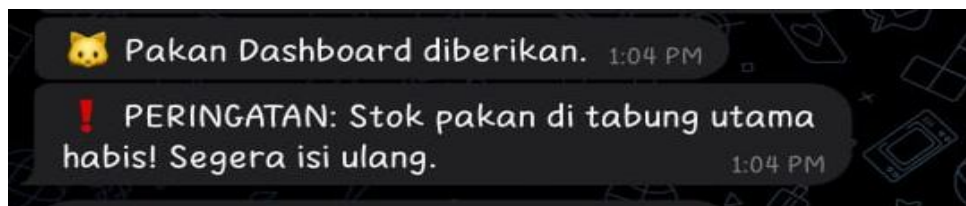
Pengujian ini dilakukan untuk memvalidasi fungsionalitas sensor ultrasonik yang ditempatkan di dalam tabung pakan. Tujuannya adalah memastikan sistem dapat mendeteksi kondisi stok pakan (tersedia atau habis) dan mengirimkan notifikasi peringatan yang sesuai. Data pengujian ini disajikan pada Tabel 4.3.

Tabel 4.3 Hasil Pengujian Deteksi Stok Pakan

No	Kondisi Stok Pakan	Jarak (cm)	Status Notifikasi	Keterangan
1	Terisi Penuh	<15	Tidak ada	Normal
2	Habis/kosong	>15	"PERINGATAN: Stok pakan di tabung utama habis!"	Berhasil



Gambar 4.9 Indikator Status Pada Tabung Pakan



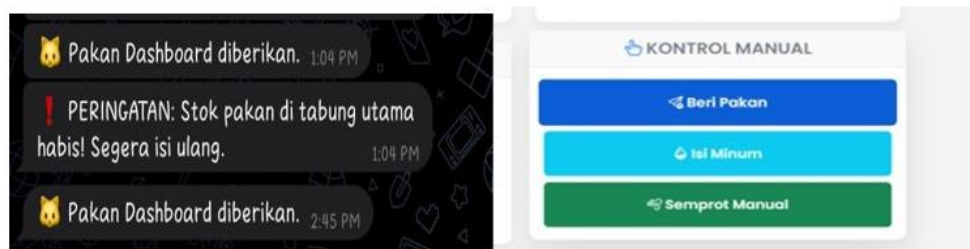
Gambar 4.10 Notifikasi Peringatan Pada Telegram

Berdasarkan Tabel 4.3, sistem berhasil mendeteksi kondisi stok pakan di tabung dengan menggunakan ambang batas jarak 15 cm. Ketika pakan masih terisi, jarak yang diukur berada di bawah 15 cm sehingga tidak ada notifikasi yang dikirimkan. Sebaliknya, saat pakan habis dan jarak terukur melebihi 15 cm, sistem berhasil mengirimkan notifikasi peringatan ke Telegram, membuktikan fungsionalitas deteksi stok pakan berjalan dengan baik.

4. Uji Pemberian Pakan Manual

Pengujian ini memastikan sistem merespons perintah manual yang dikirimkan melalui dashboard website dengan benar. Prosedur pengujian dimulai dengan mengklik tombol "Beri Pakan" pada antarmuka dashboard. Hasilnya, motor servo pada dispenser pakan segera berputar, membuka katup, dan mengeluarkan pakan. Setelah proses selesai, notifikasi konfirmasi 'Pakan telah Diberikan' juga terkirim ke bot Telegram. Pengujian

ini membuktikan bahwa fungsionalitas kontrol manual dari antarmuka pengguna ke perangkat keras berjalan dengan baik.



Gambar 4.11 Pengujian Pemberian Pakan Manual

4.2.2. Pengujian Sistem Minum

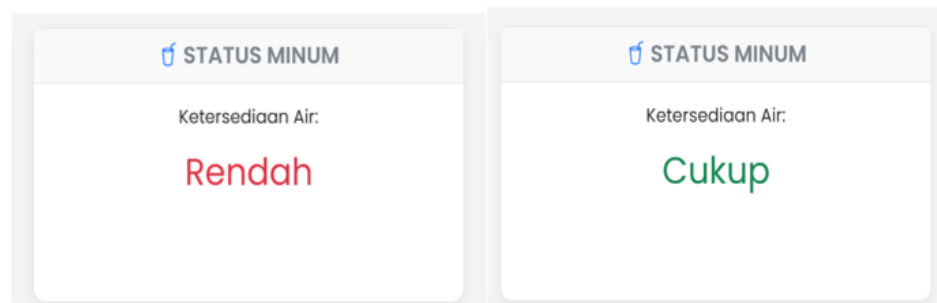
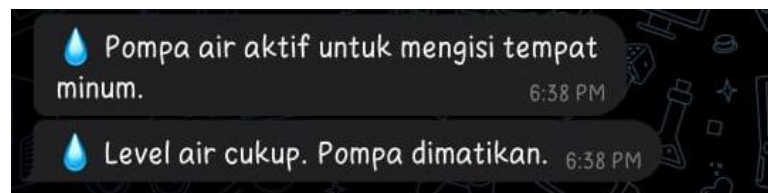
Pengujian fungsionalitas sistem minum dilakukan untuk memverifikasi kinerja subsistem otomasi pengisian air. Fokus utama pengujian ini adalah memastikan sistem dapat mendeteksi level air yang rendah dan mengaktifkan pompa air secara otomatis, serta menghentikan proses pengisian ketika level air sudah mencukupi. Selain itu, pengujian juga mencakup validasi notifikasi yang dikirimkan ke Telegram saat terjadi peristiwa penting, seperti tandon air utama yang kosong.

1. Uji Otomasi Pengisian Air

Pengujian ini dilakukan untuk memvalidasi logika program yang mengontrol pompa air mini berdasarkan pembacaan dari sensor *water level*. Tujuannya adalah memastikan sistem dapat secara otomatis mengisi wadah minum saat level air rendah dan mematikan pompa saat level air sudah cukup. Hasil pengujian ini disajikan pada Tabel 4.4.

Tabel 4.4 Hasil Pengujian Otomasi Pengisian Air

No	Kondisi Sensor Water Level	Nilai Analog Terukur	Aksi Sistem	Notifikasi Telegram	Hasil
1	Rendah	<1500	Relay On	" Pompa air aktif..."	Berhasil
2	Cukup	>1600	Relay Off	"Level air cukup..."	Berhasil

**Gambar 4.12 Indikator Status Pada Minum****Gambar 4.13 Notifikasi Pengisian Minum Pada Telegram**

Berdasarkan Tabel 4.4, sistem berhasil memvalidasi kondisi level air dan meresponsnya dengan tepat. Ketika sensor mendeteksi level air rendah, yang ditandai dengan nilai analog di bawah 1500, pompa diaktifkan secara otomatis. Sebaliknya, saat nilai analog terukur di atas 1600 yang menandakan level air sudah cukup, pompa dimatikan. Notifikasi yang sesuai juga berhasil terkirim ke Telegram pada setiap perubahan status, menunjukkan bahwa sistem otomasi pengisian air berfungsi dengan baik.

2. Uji Notifikasi Tandon Air Habis

Pengujian ini bertujuan untuk memvalidasi fungsionalitas pengaman sistem saat tandon air utama terdeteksi kosong. Prosedur pengujian dilakukan dengan mensimulasikan kondisi di mana pompa air aktif namun tidak ada air yang mengalir dari tandon utama. Dalam skenario ini, sistem dirancang untuk mengaktifkan timer. Jika kondisi level air rendah terus terdeteksi selama lebih dari 20 detik, sistem akan menyimpulkan bahwa tandon air utama telah habis. Pengujian dapat dilihat pada gambar 4.14.



Gambar 4.14 Notifikasi Tandon Air Habis

Hasil pengujian pada gambar 4.14 menunjukkan bahwa sistem berhasil mengirimkan notifikasi kritis " KRITIS: Air di tandon utama kosong! Pompa dimatikan." ke Telegram dan secara otomatis mematikan pompa. Hal ini membuktikan bahwa sistem memiliki mekanisme pengaman yang efektif untuk mencegah pompa berjalan dalam kondisi kering, yang dapat menyebabkan kerusakan.

4.2.3. Pengujian Sistem Monitoring dan Penanganan Bau Litterbox

Pengujian subsistem ini bertujuan untuk memvalidasi deteksi keberadaan kucing di area *litterbox* dan respons sistem dalam penanganan bau. Pengujian dilakukan untuk memastikan bahwa sensor ultrasonik dapat mendeteksi keberadaan

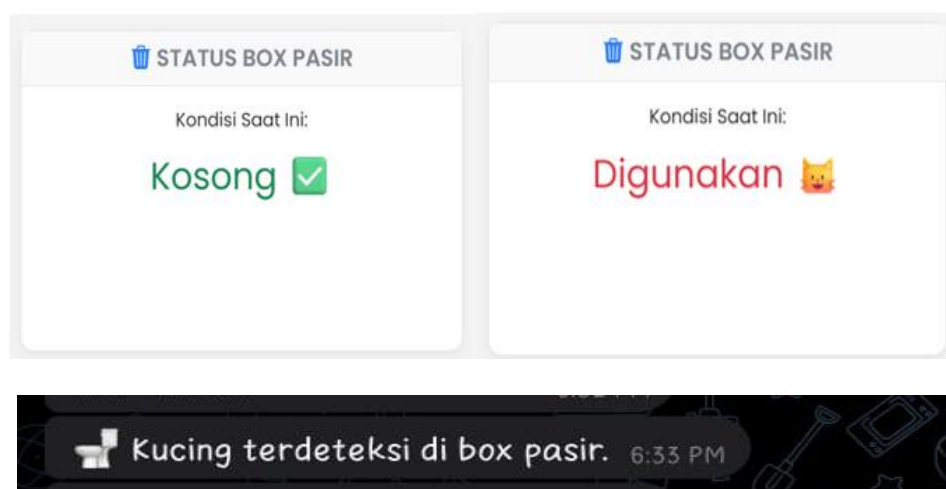
kucing dengan akurat, dan mekanisme penyemprotan pengharum dapat aktif secara otomatis setelah kucing meninggalkan area.

1. Uji Deteksi Keberadaan Kucing.

Pengujian ini dilakukan dengan menempatkan objek yang mensimulasikan kucing di area litterbox untuk menguji sensor ultrasonik. Tujuannya adalah memvalidasi ambang batas jarak yang telah ditentukan untuk mendeteksi kucing dan memastikan notifikasi terkirim dengan benar. Ambang batas jarak yang digunakan dalam sistem adalah 21 cm. Hasil pengujian ini disajikan pada Tabel 4.5.

Tabel 4.5 Hasil Pengujian Deteksi Keberadaan Kucing.

No	Jarak Objek (cm)	Status Terdeteksi	Notifikasi ke Telegram	Hasil
1	>21	Tidak Terdeteksi	Tidak ada	Berhasil
2	<21	Terdeteksi	"Kucing terdeteksi di box pasir."	Berhasil



Gambar 4.15 Status Pada Litterbox dan Notifikasi di Litterbox Pada Telegram

Berdasarkan Tabel 4.5, sistem berhasil mendeteksi keberadaan objek dengan akurat berdasarkan ambang batas jarak 21 cm. Saat objek berada di dalam jarak deteksi, sistem mengirimkan notifikasi. Ketika objek dijauhkan, status deteksi berubah menjadi normal dan sistem tidak lagi mengirimkan notifikasi, membuktikan fungsionalitas deteksi kucing berjalan dengan baik.

2. Uji Penanganan Bau Otomatis

Pengujian ini menguji logika penyemprotan otomatis menggunakan ultrasonic mist maker setelah kucing meninggalkan *litterbox*. Tujuannya adalah memastikan bahwa penyemprot hanya akan aktif setelah kucing pergi dan periode waktu tunggu yang ditentukan telah terpenuhi, serta berfungsi saat diaktifkan secara manual. Hasil pengujian ini disajikan pada Tabel 4.6.

Tabel 4.6 Hasil Pengujian Penanganan Bau Otomatis

No	Skenario	Aksi Sistem	Notifikasi Telegram	Hasil
1	Kucing terdeteksi masuk ke <i>litterbox</i> , lalu pergi.	Menunggu 10 detik, lalu penyemprot aktif selama 10 detik	"Semprotan penghilang bau telah diaktifkan"	Berhasil
2	Perintah 'Semprot Manual' dari dashboard	Penyemprot aktif selama 10 detik	"Semprotan manual diaktifkan dari dashboard."	Berhasil



Gambar 4.16 Semprotan Menyala dan Notifikasi Pada Telegram

Berdasarkan Tabel 4.6, sistem berhasil menjalankan mekanisme penanganan bau dengan benar. Penyemprotan otomatis hanya terjadi setelah kucing meninggalkan area dan memenuhi periode waktu tunggu yang ditetapkan, serta berfungsi dengan baik saat diaktifkan secara manual melalui dashboard. Hal ini membuktikan bahwa sistem dapat membantu menjaga kebersihan dan kenyamanan lingkungan kandang.

4.2.4. Pengujian Kinerja Sistem IoT

Pengujian ini dilakukan untuk menganalisis performa sistem secara keseluruhan, berfokus pada aspek konektivitas dan respons waktu (*latency*). Tujuannya adalah untuk memastikan bahwa sistem dapat beroperasi dengan stabil, merespons perintah dengan cepat, dan mengirimkan notifikasi secara andal melalui jaringan internet.

1. Uji Waktu Respon (Latency)

Pengujian ini mengukur waktu tunda (*delay*) yang dibutuhkan sistem untuk menanggapi sebuah perintah dari dashboard atau mengirimkan notifikasi. Waktu diukur dari saat perintah dikirimkan hingga aksi fisik

terjadi atau notifikasi diterima di Telegram. Hasil pengujian ini disajikan pada Tabel 4.7.

Tabel 4.7 Hasil Pengujian Waktu Respon (Latency)

No	Perintah/Aktivitas	Waktu Respon (detik)
1	Beri Pakan Manual (Dashboard ke Servo)	1.7
2	Isi Minum Manual (Dashboard ke Pompa)	2.1
3	Semprot Manual (Dashboard ke Mist Maker)	1.9
4	Notifikasi Pemberian Pakan (Sistem ke Telegram)	2.9
5	Notifikasi Kucing Terdeteksi (Sistem ke Telegram)	2.8
6	Notifikasi Tandon Air Habis (Sistem ke Telegram)	3.5

Berdasarkan Tabel 4.7, dapat dianalisis bahwa rata-rata waktu respons sistem berada di kisaran 2 hingga 4 detik. Waktu tunda ini masih dalam batas wajar untuk aplikasi IoT dan tidak mengganggu fungsionalitas sistem. Keterlambatan respons ini dipengaruhi oleh faktor jaringan internet dan proses komunikasi data melalui protokol MQTT.

4.3. Pembahasan Hasil

Berdasarkan seluruh rangkaian pengujian yang telah dilakukan, prototipe kandang kucing pintar ini terbukti berfungsi dengan baik dan andal sesuai dengan tujuan perancangan. Dalam pengujian sistem pakan, mekanisme penjadwalan otomatis menggunakan Network Time Protocol (NTP) berhasil mengeksekusi pemberian pakan pada waktu yang telah ditetapkan dengan akurasi tinggi, yaitu

dengan selisih waktu rata-rata hanya dalam hitungan detik. Di samping itu, fungsionalitas penakaran pakan dengan sensor *load cell* juga berhasil, di mana meskipun terdapat tingkat kesalahan (*error*) rata-rata sebesar 10.5%, sistem secara konsisten berhasil memberikan porsi pakan di atas takaran minimum yang ditargetkan, memastikan kebutuhan nutrisi kucing selalu terpenuhi.

Pada subsistem minum, sistem otomasi pengisian air bekerja secara efektif. Sensor *water level* berhasil mendeteksi kondisi air rendah dan mengaktifkan pompa secara otomatis, dan mematikannya kembali saat level air sudah cukup. Mekanisme pengaman juga berfungsi dengan baik, di mana sistem berhasil mengirimkan notifikasi kritis dan mematikan pompa saat tandon air utama terdeteksi kosong. Sementara itu, pada subsistem *litterbox*, sensor ultrasonik terbukti akurat dalam mendeteksi keberadaan kucing berdasarkan ambang batas jarak 21 cm, dan berhasil memicu notifikasi ke Telegram. Fitur penanganan bau otomatis juga berfungsi sesuai logika, di mana *ultrasonic mist maker* menyembprotkan pengharum setelah kucing meninggalkan area dan periode waktu tunggu terpenuhi.

Secara keseluruhan, integrasi *Internet of Things* (IoT) pada sistem ini berjalan dengan efektif. Sistem menunjukkan waktu respons (*latency*) yang cepat, rata-rata sekitar 2-4 detik, untuk setiap perintah dari dashboard website dan pengiriman notifikasi ke Telegram. Hal ini memvalidasi bahwa sistem dapat dikontrol dan dimonitoring secara *real-time* dari jarak jauh, sehingga berhasil menjawab semua rumusan masalah dan mencapai tujuan penelitian yang telah ditetapkan.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem kandang kucing pintar, dapat ditarik beberapa kesimpulan utama sebagai berikut:

1. Sistem berhasil diimplementasikan sesuai dengan rancangan. Fungsionalitas pemberian pakan otomatis, otomasi minum, serta *monitoring* dan penanganan bau di *litterbox* telah terintegrasi dengan baik dan berfungsi sesuai dengan logika yang diprogram.
2. Pengujian fungsionalitas sistem pakan menunjukkan akurasi yang tinggi. Sistem berhasil menjalankan penjadwalan pakan secara otomatis menggunakan Network Time Protocol (NTP) dengan selisih waktu yang sangat minim. Akurasi takaran pakan menggunakan sensor *load cell* juga berada dalam batas toleransi yang dapat diterima, memastikan porsi pakan terpenuhi.
3. Sistem otomasi minum berfungsi dengan efektif. Sensor *water level* berhasil mendeteksi kondisi air rendah dan memicu pengisian air secara otomatis. Mekanisme pengaman untuk notifikasi tandon air habis juga bekerja dengan andal, mengirimkan peringatan kritis saat terdeteksi kondisi kosong.
4. Fungsionalitas *monitoring* dan penanganan bau pada *litterbox* berjalan dengan baik. Sensor ultrasonik terbukti mampu mendeteksi keberadaan kucing dan memicu notifikasi. Selain itu, *ultrasonic mist maker* berhasil diaktifkan secara otomatis setelah kucing meninggalkan area, membantu menjaga kebersihan lingkungan kandang.

5. Sistem IoT yang dibangun menunjukkan kinerja yang stabil. Interaksi melalui dashboard website dan notifikasi Telegram memiliki waktu respons yang cepat dan andal, membuktikan bahwa sistem dapat dikendalikan dan dimonitoring secara *real-time* dari jarak jauh.

5.2. Saran

Meskipun sistem telah berhasil diimplementasikan dan berfungsi dengan baik, terdapat beberapa saran untuk pengembangan dan perbaikan di masa mendatang guna meningkatkan fungsionalitas dan efektivitas sistem:

1. Mengintegrasikan modul kamera seperti ESP32-CAM untuk memungkinkan pemilik memantau kondisi kucing dan kandang secara visual melalui *video streaming*.
2. Mengintegrasikan mekanisme mekanik untuk membersihkan kotoran kucing secara otomatis.
3. Merancang mekanisme perosotan atau katup pakan yang lebih baik untuk mengurangi tingkat kesalahan penakaran yang disebabkan oleh pelet pakan yang tidak seragam, sehingga akurasi dapat ditingkatkan.

DAFTAR PUSTAKA

- Ariyanti, Y. R. (2023). Sistem Monitoring Kelembaban Dan Penyiraman Otomatis Pada Tanaman Hias Lidah Mertua (Sansivera) Berbasis Iot Yusrina.
- Ayuningtyas, A. A. (2022). Penerapan Internet Of Things (Iot) Dalam Upaya Mewujudkan Perpustakaan Digital Di Era Society 5.0.
- Canady, R., Danendra, D. R., Indrawan, V. M., & Rochadiani, T. H. (2023). Rancang Bangun Alat Pemberi Makan Hewan Peliharaan Pintar Menggunakan Mikrokontroler ESP32 Berbasis *Internet of Things* (IoT) dengan Platform Blynk. In *Jurnal Teknik Informatika Unis* (Vol. 11, Issue 2).
- Fatahillah Murad, R., Almasir, G., Ronald Harahap, C., Komputer, T., Ratu, L., & Lampung, B. (2022). Pendeteksi Gas Amonia Untuk Pembesaran Anak Ayam Pada Box Kandang Menggunakan Mq-135. *Jurnal Ilmiah Mahasiswa Kendali Dan Listrik*, 3(1). <https://doi.org/10.33365/jimel.v1i1>
- Febryanto, Y., Radillah, T., & Ameliza, K. (2022). Perancangan Alat Pemberi Pakan Ikan Otomatis Dengan Rtc Ds3231 Berbasis Microcontroller Arduino Uno. *Indonesia Journal of Computer Science*, 11.
- Fernanda, M. I., & Wildian. (2023). Rancang Bangun Mesin Desinfeksi Kabut Otomatis untuk Menghemat Air. 12(2), 304–310. <https://doi.org/10.25077/jfu.12.2.304-310.2023>
- Gunadi, I. G. A., & Rachmawati, D. O. (2022). Review Penggunaan Sensor Pada Aplikasi IOT. Wahana Matematika Dan Sains: Jurnal Matematika, Sains, Dan Pembelajarannya, 16(3), 1858–0629. <https://ejournal.undiksha.ac.id/index.php/JPM/article/view/51037>
- Hercog, D., Lerher, T., Truntič, M., & Težak, O. (2023). Design and Implementation of ESP32-Based IoT Devices. *Sensors*, 23(15). <https://doi.org/10.3390/s23156739>
- Huseng, R. (2024). Skripsi Rancang Bangun Sistem Pemberian Pakan Dan Pemantauan Kualitas Air Tambak Udang Berbasis Iot Menggunakan Algoritma Fuzzy Logic Design And Construction Of Iot-Based Shrimp Pond Feeding And Water Quality Monitoring System Using Fuzzy Logic Algorithm.
- Kurniawan, S., Halim, D. K., Dicky, H., & Tang, C. M. (2020). Multicore Development Environment For Embedded Processor In Arduino IDE. *Telkomnika (Telecommunication Computing Electronics and Control)*, 18(2), 870–878. <https://doi.org/10.12928/TELKOMNIKA.V18I2.14873>
- Majid, A., Androva, A., & Mukhtar, A. (2023). Rancang Bangun Instrumentasi Load Cell Strain Gauge Half Bridge pada Dynamometer Prony Brake dengan Sistem Monitoring LCD 16x4 Display Berbasis Mikrokontroler Arduino.

Cross-Border, 6(2), 834–842.

- Makarim, M. F., Nurmuslimah, S., Danang, D., Sulaksono, H., Adhi, I. T., & Surabaya, T. (2022). Sistem Kontrol Otomatis Penyemprotan Pestisida Pada Lahan Pertanian Padi Menggunakan Mikrokontroler Arduino Berbasis Internet of Things.
- Mukhtar, A., Hermana, R., Burhanudin, A., & Setyoadi, Y. (2023). Sensor Dan Aktuator: Konsep Dasar Dan Aplikasi. *Cv Widina Media Utama*, 1.
- Ngitung, R. (2021). Karakteristik Perilaku Kucing Domestik. *Sainsmat : Jurnal Ilmiah Ilmu Pengetahuan Alam*, 10(1), 78. <https://doi.org/10.35580/sainsmat101362152021>
- Nurul Fitri, A., & Yendri, D. (2023). Rancang Bangun Pelembab Udara Ruangan (Humidifier) berbasis Mikrokontroler. *CHIPSET*, 4(01), 61–70. <https://doi.org/10.25077/chipset.4.01.61-70.2023>
- Puspitorini, I., & Sintawati, I. D. (2021). Penerapan Data Mining Untuk Klasifikasi Prediksi Produk Jenis Makanan Kucing Yang Sesuai Kebutuhan Dengan Algoritma Decision Tree (ID3). *Akrab Juara*, 75(17), 399–405.
- Putu, L., Dewi, A. C., Ketut, I., Arthana, R., & Setemen, K. (2023). Rancang Bangun Alat Pakan Kucing Dengan Menggunakan Mikrokontroler Berbasis *Internet Of Things* (Iot). *Kumpulan Artikel Mahasiswa Pendidikan Teknik Informatika (KARMAPATI)*, 12(3).
- Qureshi, M. H. R. A. (2024). *Microcontrollers*.
- Rachmansyah, A., Satra, R., & Mude, M. A. (2022). Buletin Sistem Informasi dan Teknologi Islam Perancangan Alat Pemberi Makan dan Monitoring Sisa Pakan Hewan Pemeliharaan Berbasis Microcontroller *INFORMASI ARTIKEL ABSTRAK*. 3(1), 26.
- Ruslianto, I., Ristian, U., Hasfani, H., & Sari, K. (2023). Rekayasa Sistem Fotosintesis dan Ekosistem pada Media Aquascape Berbasis *Internet of Things*. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, 9(1), 136. <https://doi.org/10.26418/jp.v9i1.61746>
- Silpa Rani Zain, Zaenudin, Ardiyallah Akbar, & Lalu Delsi Samsumar. (2024). Rancang Bangun Smart Pet Feeder Pada Kandang Kucing Berbasis *Internet of Things*. *Journal Of Computer Science And Technology (JOCSTEC)*, 2(3), 150–158. <https://doi.org/10.59435/jocstec.v2i3.415>
- Susanto, F., Komang Prasiani, N., & Darmawan, P. (2022). Implementasi *Internet Of Things* Dalam Kehidupan Sehari-Hari. In *Jurnal Imagine* (Vol. 2, Issue 1). Online. <https://jurnal.std-bali.ac.id/index.php/imagine>
- Yeong, D. J., Velasco-hernandez, G., Barry, J., & Walsh, J. (2021). Sensor And Sensor Fusion Technology In Autonomous Vehicles: A review. *Sensors*,

21(6), 1–37. <https://doi.org/10.3390/s21062140>

Yudha Kusuma, D., Bayu Permatasari, N., Rostira Pebriani, R., & Hudati, I. (2021). Sensor Ultrasonik Waterproof A02YYUW Berbasis Arduino Uno Pada Sistem Pengukuran Jarak. 2(2).

Yulianeu, A., & Oktamala, R. (2022). Sistem Informasi Geografis Trayek Angkutan Umum Di Kota Tasikmalaya Berbasis Web. *Jutekin (Jurnal Teknik Informatika)*, 10(2). <https://doi.org/10.51530/jutekin.v10i2.669>

LAMPIRAN

A. Lampiran Code Program

1. Library dan Insialisasi

```
// -- LIBRARY YANG DIBUTUHKAN --
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
#include <WiFiClientSecure.h>
#include <ESP32Servo.h>
#include "HX711.h"
#include "time.h"
#include <UniversalTelegramBot.h>
#include <Preferences.h>
```

2. Koneksi Wifi, Mqtt dan Bot Telegram

```
// -- PENGATURAN PENGGUNA --
// 1. Kredensial WiFi
const char* ssid = "HAH";
const char* password = "HAHAHA12";

// 2. Pengaturan Broker MQTT
const char* mqtt_broker = "broker.hivemq.com";
const int mqtt_port = 1883;

// 3. Topik MQTT
const char* dataTopic = "kandang-kucing/data";
const char* commandTopic = "kandang-kucing/perintah";
const char* logTopic = "kandang-kucing/log";

// 4. Pengaturan Telegram (HANYA UNTUK NOTIFIKASI)
#define BOT_TOKEN
"7814718304:AAEphP5KyapjwqcDSft2XvIEb8Yf5ZB03WQ" // Token Telegram
#define CHAT_ID "1401212890" // chat ID Telegram
```

3. Deklarasi Pin Hardware dan Variable

```
// 5. Pengaturan Pin Hardware
const int PIN_LED_WIFI = 13;
const int PIN_SERVO_PAKAN = 27;
const int PIN_ULTRASONIC_PAKAN_TRIG = 33;
const int PIN_ULTRASONIC_PAKAN_ECHO = 32;
const int PIN_LOADCELL_DT = 25;
const int PIN_LOADCELL_SCK = 26;
const int PIN_SENSOR_AIR = 35;
const int PIN_RELAY_POMPA = 21;
```

```

const int PIN_ULTRASONIC_PASIR_TRIG = 19;
const int PIN_ULTRASONIC_PASIR_ECHO = 18;
const int PIN_RELAY_SEMPROT = 22;
    // 6. Pengaturan Fungsional
const float FAKTOR_KALIBRASI_LOADCELL = -944.96;
const float TAKARAN_PAKAN_GRAM = 40.0;
const int JARAK_STOK_HABIS_CM = 15;
const float BATAS_PAKAN_SISA_GRAM = 40.0;
int waktuMakanPagi = 6, menitMakanPagi = 27;
int waktuMakanSiang = 12, menitMakanSiang = 17;
int waktuMakanMalam = 18, menitMakanMalam = 19;
const int JARAK_DETEKSI_KUCING_CM = 21;
const unsigned long WAKTU_TUNGGU_SEMPROT_MS = 10000;
const unsigned long DURASI_SEMPROT_MS = 10000;
const int NILAI_SENSOR_AIR_KERING = 1500;
const char* ntpServer = "pool.ntp.org";
const long  gmtoffset_sec = 25200;
const int  daylightOffset_sec = 0;

    // === Variabel & Objek Global ===
WiFiClient espClient;
PubSubClient client(espClient);
WiFiClientSecure clientSecure;
UniversalTelegramBot bot(BOT_TOKEN, clientSecure);
Preferences preferences;
Servo pakanServo;
HX711 scale;

unsigned long waktuCobaReconnectTerakhir = 0; // Untuk reconnect MQTT non-
blocking

unsigned long waktuBooting = 0;
bool sistemSiap = false;
unsigned long intervalCekWaktu = 1000;
unsigned long waktuCekWaktuTerakhir = 0;
bool sudahMakanPagi = false, sudahMakanSiang = false, sudahMakanMalam = false;
unsigned long waktuAirMulaiRendah = 0;
bool pompaAktif = false;
bool manualFillActive = false;
unsigned long intervalCekMinum = 2000;
unsigned long waktuCekMinumTerakhir = 0;
unsigned long intervalCekPasir = 200;
unsigned long waktuCekPasirTerakhir = 0;
bool kucingDianggapDiDalam = false;
bool sedangMenyemprot = false;
unsigned long waktuMulaiSemprot = 0;
unsigned long waktuCekJarakPakanTerakhir = 0;
const unsigned long intervalCekJarakPakan = 1000;

```

4. Fungsi dan Logika Fitur

```

void kirimNotifikasi(String pesan) {
    // Kirim ke Telegram
    bot.sendMessage(CHAT_ID, pesan, "");

    // Kirim juga ke Dashboard via MQTT dan cek statusnya
    bool berhasil_kirim = client.publish(logTopic, pesan.c_str());

    // Tampilkan status pengiriman di Serial Monitor
    if (berhasil_kirim) {
        Serial.println("[MQTT LOG] Berhasil dikirim: " + pesan);
    } else {
        Serial.println("[MQTT LOG] GAGAL dikirim: " + pesan);
    }
}

// Deklarasi fungsi
void jalankanPemberianPakan(String jadwal, bool dariPerintahManual = false);
bool cekStokPakan();
void kirimNotifikasi(String pesan);

//
=====
// ==    FUNGSI-FUNGSI MQTT (Terima Perintah)    ==
//
=====

void callback(char* topic, byte* payload, unsigned int length) {
    String message = "";
    for (int i = 0; i < length; i++) { message += (char)payload[i]; }
    Serial.printf("Pesan diterima [%s]: %s\n", topic, message.c_str());

    if (String(topic) == commandTopic) {
        // Coba parsing pesan sebagai JSON
        StaticJsonDocument<200> doc;
        DeserializationError error = deserializeJson(doc, message);

        // Jika berhasil di-parse dan ada event 'set-jadwal-single'
        if (!error && doc["event"] == "set-jadwal-single") {
            Serial.println("[MQTT] Menerima perintah set jadwal baru!");

            String waktu = doc["waktu"]; // "pagi", "siang", atau "malam"
            int jam = doc["jam"].as<int>();
            int menit = doc["menit"].as<int>();

            preferences.begin("jadwal-kucing", false); // Buka memori untuk menulis

            if (waktu == "pagi") {
                waktuMakanPagi = jam;
                menitMakanPagi = menit;
                preferences.putInt("pagi_j", jam);
                preferences.putInt("pagi_m", menit);
            }
        }
    }
}

```

```

        } else if (waktu == "siang") {
            waktuMakanSiang = jam;
            menitMakanSiang = menit;
            preferences.putInt("siang_j", jam);
            preferences.putInt("siang_m", menit);
        } else if (waktu == "malam") {
            waktuMakanMalam = jam;
            menitMakanMalam = menit;
            preferences.putInt("malam_j", jam);
            preferences.putInt("malam_m", menit);
        }

        preferences.end(); // Simpan perubahan

        sudahMakanPagi = sudahMakanSiang = sudahMakanMalam = false; // Reset
        status makan harian

        // Kirim notifikasi konfirmasi ke Telegram
        kirimNotifikasi("📅 Jadwal makan " + waktu + " berhasil di-update dari
        dashboard.");
    }
    // Jika bukan JSON atau event lain, proses sebagai perintah biasa
    else {
        if (message == "feed") {
            jalankanPemberianPakan("Dashboard", true);
        } else if (message == "drink") {
            int nilaiSensor = analogRead(PIN_SENSOR_AIR);
            if (nilaiSensor > 1500) {
                kirimNotifikasi("⚠️ Perintah 'Isi Minum' ditolak, tempat minum sudah
                penuh.");
            } else {
                manualFillActive = true;
            }
        } else if (message == "spray") {
            Serial.println("[MQTT] Menerima perintah: semprot!");

            digitalWrite(PIN_RELAY_SEMPROT, LOW);

            kirimNotifikasi("☂️ Semprotan manual diaktifkan dari dashboard.");

            delay(DURASI_SEMPROT_MS);
            digitalWrite(PIN_RELAY_SEMPROT, HIGH);
        }
    }
}

//
=====
// ==  FUNGSI MANAJEMEN KONEKSI  ==
//
=====

```

```

// FUNGSI BARU: Untuk menjaga koneksi WiFi tetap hidup
void checkWiFi() {
  if (WiFi.status() != WL_CONNECTED) {
    digitalWrite(PIN_LED_WIFI, LOW);
    Serial.print("Koneksi WiFi terputus. Mencoba menyambungkan kembali...");
    WiFi.reconnect();
    unsigned long startAttemptTime = millis();
    // Tunggu maksimal 10 detik untuk konek
    while (WiFi.status() != WL_CONNECTED && millis() - startAttemptTime <
10000) {
      // --> INI LOGIKA KEDIP YANG DITAMBAHKAN <--
      digitalWrite(PIN_LED_WIFI, HIGH);
      delay(300);
      digitalWrite(PIN_LED_WIFI, LOW);
      delay(300);
      Serial.print(".");
      Serial.print(".");

    }

    if(WiFi.status() == WL_CONNECTED) {
      Serial.println("\nWiFi berhasil terhubung kembali!");
      digitalWrite(PIN_LED_WIFI, HIGH);
    } else {
      Serial.println("\nGagal menyambungkan kembali WiFi.");
      digitalWrite(PIN_LED_WIFI, LOW); // Tetap mati jika gagal
    }
  }
}

// FUNGSI DIPERBARUI: Reconnect MQTT menjadi non-blocking
void reconnect() {
  if (millis() - waktuCobaReconnectTerakhir > 5000) {
    waktuCobaReconnectTerakhir = millis();
    if (!client.connected()) {
      Serial.print("Mencoba koneksi ke MQTT Broker...");
      String clientId = "esp32-kandang-kucing-" + String(random(0xffff), HEX);
      if (client.connect(clientId.c_str())) {
        Serial.println("terhubung!");
        client.subscribe(commandTopic);
        kirimNotifikasi("● Sistem kembali online & terhubung.");
      } else {
        Serial.printf("gagal, rc=%d. Coba lagi nanti...\n", client.state());
      }
    }
  }
}
}

```

```

//
=====

// ==          SETUP          ==
//
=====

void setup() {
  Serial.begin(115200);
  waktuBooting = millis();
  Serial.println("\n\nInisialisasi Sistem Kandang Kucing Pintar V4.8...");
  pakanServo.attach(PIN_SERVO_PAKAN);
  pakanServo.write(0);
  pinMode(PIN_ULTRASONIC_PAKAN_TRIG, OUTPUT);
  pinMode(PIN_ULTRASONIC_PAKAN_ECHO, INPUT);
  scale.begin(PIN_LOADCELL_DT, PIN_LOADCELL_SCK);
  scale.set_scale(FAKTOR_KALIBRASI_LOADCELL);
  scale.tare();
  pinMode(PIN_SENSOR_AIR, INPUT);
  pinMode(PIN_RELAY_POMPA, OUTPUT);
  digitalWrite(PIN_RELAY_POMPA, HIGH);
  pinMode(PIN_ULTRASONIC_PASIR_TRIG, OUTPUT);
  pinMode(PIN_ULTRASONIC_PASIR_ECHO, INPUT);
  pinMode(PIN_RELAY_SEMPROT, OUTPUT);
  digitalWrite(PIN_RELAY_SEMPROT, HIGH);
  pinMode(PIN_LED_WIFI, OUTPUT);

  Serial.print("Menghubungkan ke WiFi: ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);

  digitalWrite(PIN_LED_WIFI, HIGH);
  Serial.println("\nWiFi terhubung!");
  Serial.print("Alamat IP: ");
  Serial.println(WiFi.localIP());

  loadJadwalFromMemory();
  configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
  clientSecure.setInsecure();

  client.setServer(mqtt_broker, mqtt_port);
  client.setCallback(callback);

  Serial.println("Sistem Siap!");
}

//
=====

// ==          LOOP UTAMA          ==
//
=====

unsigned long waktuKirimDashboardTerakhir = 0;
const long intervalKirimDashboard = 1000;

void loop() {

```

```

    // 1. Selalu cek koneksi WiFi terlebih dahulu
    checkWiFi();
    if (WiFi.status() == WL_CONNECTED) {
        if (!client.connected()) {
            reconnect();
        }
        client.loop();
    }

    unsigned long currentTime = millis();
    if (!sistemSiap && currentTime - waktuBooting > 5000) {
        sistemSiap = true;
    }

    if (sistemSiap) {
        handleLitterBox();
        handlePakan();
        handleMinum();
        handleJarakPakanMonitor();
    }

    if (currentTime - waktuKirimDashboardTerakhir >= intervalKirimDashboard) {
        waktuKirimDashboardTerakhir = currentTime;

        StaticJsonDocument<512> doc;
        float beratPakan = scale.get_units(5);
        if (beratPakan < 0) beratPakan = 0;

        // Baca jarak ultrasonik
        digitalWrite(PIN_ULTRASONIC_PAKAN_TRIG, LOW); delayMicroseconds(2);
        digitalWrite(PIN_ULTRASONIC_PAKAN_TRIG, HIGH);
        delayMicroseconds(10);
        digitalWrite(PIN_ULTRASONIC_PAKAN_TRIG, LOW);
        float jarakPakan = pulseIn(PIN_ULTRASONIC_PAKAN_ECHO, HIGH, 25000)
            * 0.034 / 2;

        digitalWrite(PIN_ULTRASONIC_PASIR_TRIG, LOW); delayMicroseconds(2);
        digitalWrite(PIN_ULTRASONIC_PASIR_TRIG, HIGH); delayMicroseconds(10);
        digitalWrite(PIN_ULTRASONIC_PASIR_TRIG, LOW);
        float jarakPasir = pulseIn(PIN_ULTRASONIC_PASIR_ECHO, HIGH, 25000) *
            0.034 / 2;

        doc["berat"] = String(beratPakan, 0);
        doc["nilaiAir"] = analogRead(PIN_SENSOR_AIR);
        doc["stok"] = cekStokPakan() ? "Tersedia" : "Habis";
        doc["pasir"] = kucingDianggapDiDalam ? "Digunakan 🐱" : "Kosong ✅";
        doc["jarakPakan"] = jarakPakan; // <-- DATA BARU
        doc["jarakPasir"] = jarakPasir; // <-- DATA BARU

        JsonObject jadwal = doc.createNestedObject("jadwal");

```

```

    JsonObject pagi = jadwal.createNestedObject("pagi");
    pagi["jam"] = String(waktuMakanPagi);
    pagi["menit"] = String(menitMakanPagi);
    JsonObject siang = jadwal.createNestedObject("siang");
    siang["jam"] = String(waktuMakanSiang);
    siang["menit"] = String(menitMakanSiang);
    JsonObject malam = jadwal.createNestedObject("malam");
    malam["jam"] = String(waktuMakanMalam);
    malam["menit"] = String(menitMakanMalam);

    char buffer[512];
    serializeJson(doc, buffer);
    client.publish(dataTopic, buffer);
}
}

//
=====

// ==          SEMUA FUNGSI ==
//
=====

void loadJadwalFromMemory() {
    preferences.begin("jadwal-kucing", true);
    waktuMakanPagi = preferences.getInt("pagi_j", waktuMakanPagi);
    menitMakanPagi = preferences.getInt("pagi_m", menitMakanPagi);
    waktuMakanSiang = preferences.getInt("siang_j", waktuMakanSiang);
    menitMakanSiang = preferences.getInt("siang_m", menitMakanSiang);
    waktuMakanMalam = preferences.getInt("malam_j", waktuMakanMalam);
    menitMakanMalam = preferences.getInt("malam_m", menitMakanMalam);
    preferences.end();
    Serial.println("Jadwal makan berhasil dimuat dari memori.");
}

void handlePakan() {
    if (millis() - waktuCekWaktuTerakhir >= intervalCekWaktu) {
        waktuCekWaktuTerakhir = millis();
        struct tm timeinfo;
        if (!getLocalTime(&timeinfo)) {
            Serial.println("Gagal mendapatkan waktu lokal!");
            return;
        }
        int jamSekarang = timeinfo.tm_hour;
        int menitSekarang = timeinfo.tm_min;

        if (jamSekarang == waktuMakanPagi && menitSekarang == menitMakanPagi
            && !sudahMakanPagi) {
            jalankanPemberianPakan("Pagi", false);
            sudahMakanPagi = true;
        }
        else if (jamSekarang == waktuMakanSiang && menitSekarang ==
            menitMakanSiang && !sudahMakanSiang) {
            jalankanPemberianPakan("Siang", false);
            sudahMakanSiang = true;
        }
    }
}

```

```

    }
    else if (jamSekarang == waktuMakanMalam && menitSekarang ==
    menitMakanMalam && !sudahMakanMalam) {
        jalankanPemberianPakan("Malam", false);
        sudahMakanMalam = true;
    }
    if (jamSekarang == 0 && menitSekarang == 1) { // Reset jam 00:01
        sudahMakanPagi = false;
        sudahMakanSiang = false;
        sudahMakanMalam = false;
    }
}
}

void handleMinum() {
    if (millis() - waktuCekMinumTerakhir >= intervalCekMinum) {
        waktuCekMinumTerakhir = millis();
        int nilaiSensor = analogRead(PIN_SENSOR_AIR);
        static bool airHabisDiTandon = false;
        static bool notifTandonHabisTer kirim = false;
        if (airHabisDiTandon && nilaiSensor > NILAI_SENSOR_AIR_KERING) {
            airHabisDiTandon = false;
            notifTandonHabisTer kirim = false;
            waktuAirMulaiRendah = 0;
        }
        if (pompaAktif && nilaiSensor < NILAI_SENSOR_AIR_KERING) {
            if (waktuAirMulaiRendah == 0) {
                waktuAirMulaiRendah = millis();
            } else if (millis() - waktuAirMulaiRendah > 20000) {
                airHabisDiTandon = true;
                if (!notifTandonHabisTer kirim) {
                    kirimNotifikasi(" ! KRITIS: Air di tandon utama kosong! Pompa
                    dimatikan.");
                    notifTandonHabisTer kirim = true;
                }
            }
        } else {
            waktuAirMulaiRendah = 0;
        }
        if (airHabisDiTandon && !manualFillActive) {
            digitalWrite(PIN_RELAY_POMPA, HIGH);
            pompaAktif = false;
        } else {
            if (nilaiSensor > NILAI_SENSOR_AIR_KERING + 100) {
                digitalWrite(PIN_RELAY_POMPA, HIGH);
                if (pompaAktif) {
                    if (!manualFillActive) { // Hanya kirim notif jika bukan manual
                        kirimNotifikasi(" 💧 Level air cukup. Pompa dimatikan.");
                    }
                }
                pompaAktif = false;
                manualFillActive = false;
            }
        }
    }
}

```

```

    }
    else if (nilaiSensor < NILAI_SENSOR_AIR_KERING || manualFillActive) {
        digitalWrite(PIN_RELAY_POMPA, LOW);
        if (!pompaAktif) {
            kirimNotifikasi("💧 Pompa air aktif untuk mengisi tempat minum.");
            pompaAktif = true;
        }
    }
}
}
}

void handleLitterBox() {
    if (pompaAktif) return;
    static unsigned long waktuKucingMasuk = 0;
    static bool sudahSemprotSetelahPup = true;
    static bool notifKucingMasukTer kirim = false;
    if (millis() - waktuCekPasirTerakhir >= intervalCekPasir) {
        waktuCekPasirTerakhir = millis();
        digitalWrite(PIN_ULTRASONIC_PASIR_TRIG, LOW); delayMicroseconds(2);
        digitalWrite(PIN_ULTRASONIC_PASIR_TRIG, HIGH); delayMicroseconds(10);
        digitalWrite(PIN_ULTRASONIC_PASIR_TRIG, LOW);
        long durasi = pulseIn(PIN_ULTRASONIC_PASIR_ECHO, HIGH, 25000);
        float jarak = durasi * 0.034 / 2;
        if (jarak <= 0) return;
        if (jarak < JARAK_DETEKSI_KUCING_CM && !kucingDianggapDiDalam
            && !sedangMenyemprot) {
            if (!notifKucingMasukTer kirim) {
                kirimNotifikasi("🐾 Kucing terdeteksi di box pasir.");
                notifKucingMasukTer kirim = true;
            }
            kucingDianggapDiDalam = true;
            waktuKucingMasuk = millis();
            sudahSemprotSetelahPup = false;
        }
        if (jarak >= JARAK_DETEKSI_KUCING_CM && kucingDianggapDiDalam) {
            kucingDianggapDiDalam = false;
            notifKucingMasukTer kirim = false;
        }
        if (!kucingDianggapDiDalam && !sudahSemprotSetelahPup && (millis() -
            waktuKucingMasuk > WAKTU_TUNGGU_SEMPROT_MS)) {
            kirimNotifikasi("💧 Semprotan penghilang bau telah diaktifkan.");
            digitalWrite(PIN_RELAY_SEMPROT, LOW);
            sedangMenyemprot = true;
            waktuMulaiSemprot = millis();
            sudahSemprotSetelahPup = true;
        }
    }
}
if (sedangMenyemprot && (millis() - waktuMulaiSemprot >=
    DURASI_SEMPROT_MS)) {
    digitalWrite(PIN_RELAY_SEMPROT, HIGH);
    sedangMenyemprot = false;
}

```

```

    }
}

void handleJarakPakanMonitor() {
    if (millis() - waktuCekJarakPakanTerakhir >= intervalCekJarakPakan) {
        waktuCekJarakPakanTerakhir = millis();
        // Fungsi ini bisa dikosongkan jika tidak ingin ada log serial terus menerus
    }
}

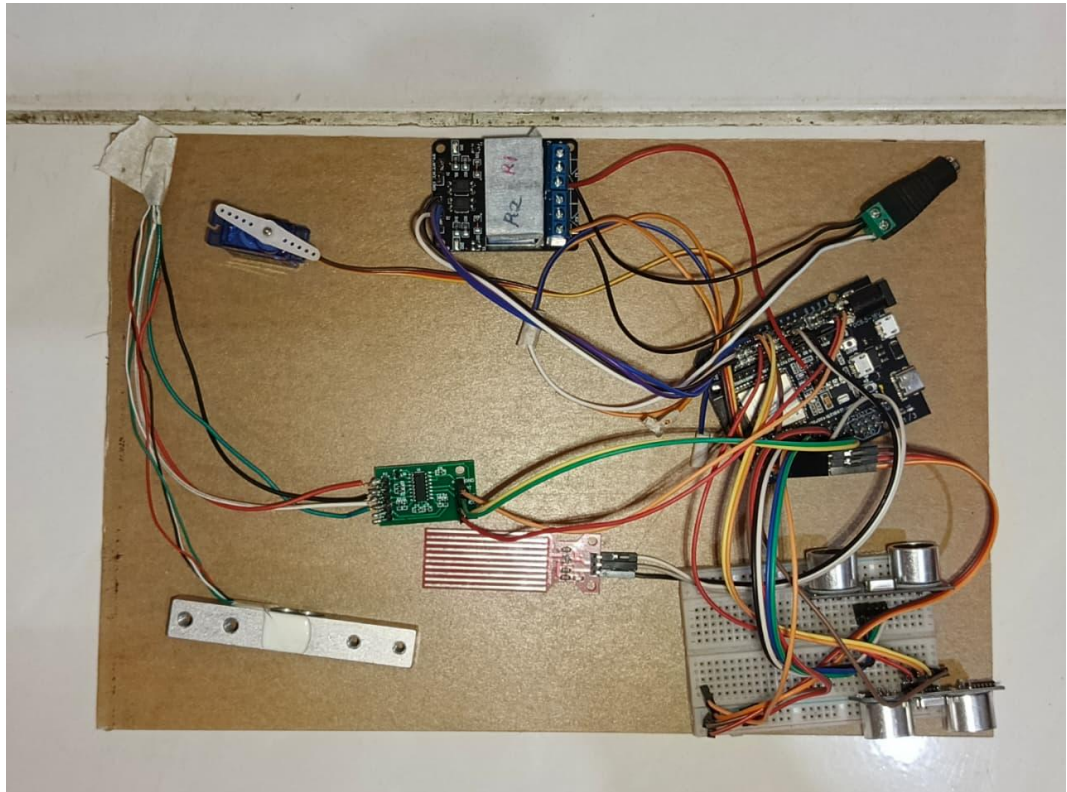
void jalankanPemberianPakan(String jadwal, bool dariPerintahManual) {
    if (cekStokPakan()) {
        float beratAwal = scale.get_units(10);
        if (beratAwal < 0) beratAwal = 0;
        if (beratAwal < BATAS_PAKAN_SISA_GRAM || dariPerintahManual) {
            pakanServo.write(45);
            long startAttempt = millis();
            while (true) {
                float beratSekarang = scale.get_units();
                if (beratSekarang < 0) beratSekarang = 0;
                if (beratSekarang - beratAwal >= TAKARAN_PAKAN_GRAM) {
                    break;
                }
                if (millis() - startAttempt > 15000) { // Timeout 15 detik
                    Serial.println("Pemberian pakan timeout, pakan mungkin macet.");
                    break;
                }
                delay(100);
            }
            pakanServo.write(0);
            delay(1000);
            float beratAkhir = scale.get_units(10);
            if (beratAkhir < 0) beratAkhir = 0;
            String msg = "🐱 Pakan " + jadwal + " diberikan.";
            kirimNotifikasi(msg);
        } else {
            String msg = "⚠ Pemberian makan " + jadwal + " dilewati, mangkuk masih penuh (" + String(beratAwal, 0) + "g).";
            kirimNotifikasi(msg);
        }
    } else {
        kirimNotifikasi("❗ PERINGATAN: Stok pakan di tabung utama habis! Segera isi ulang.");
    }
}

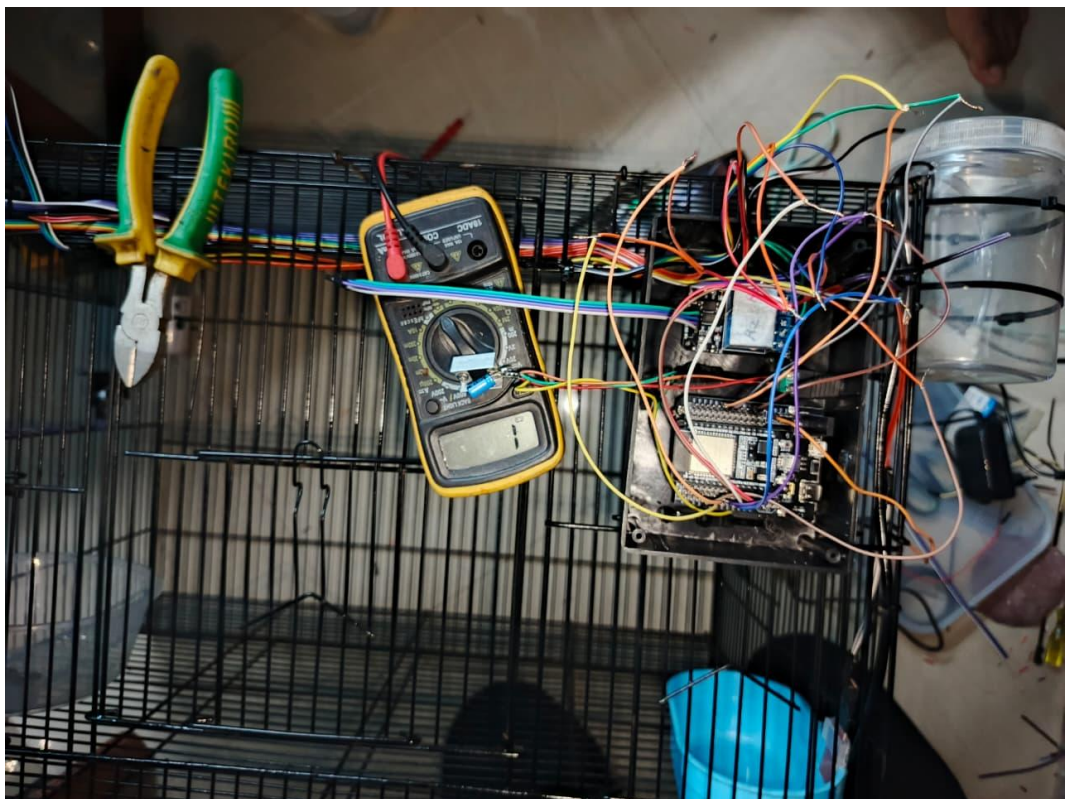
bool cekStokPakan() {
    digitalWrite(PIN_ULTRASONIC_PAKAN_TRIG, LOW);
    delayMicroseconds(2);
    digitalWrite(PIN_ULTRASONIC_PAKAN_TRIG, HIGH);
    delayMicroseconds(10);
    digitalWrite(PIN_ULTRASONIC_PAKAN_TRIG, LOW);
}

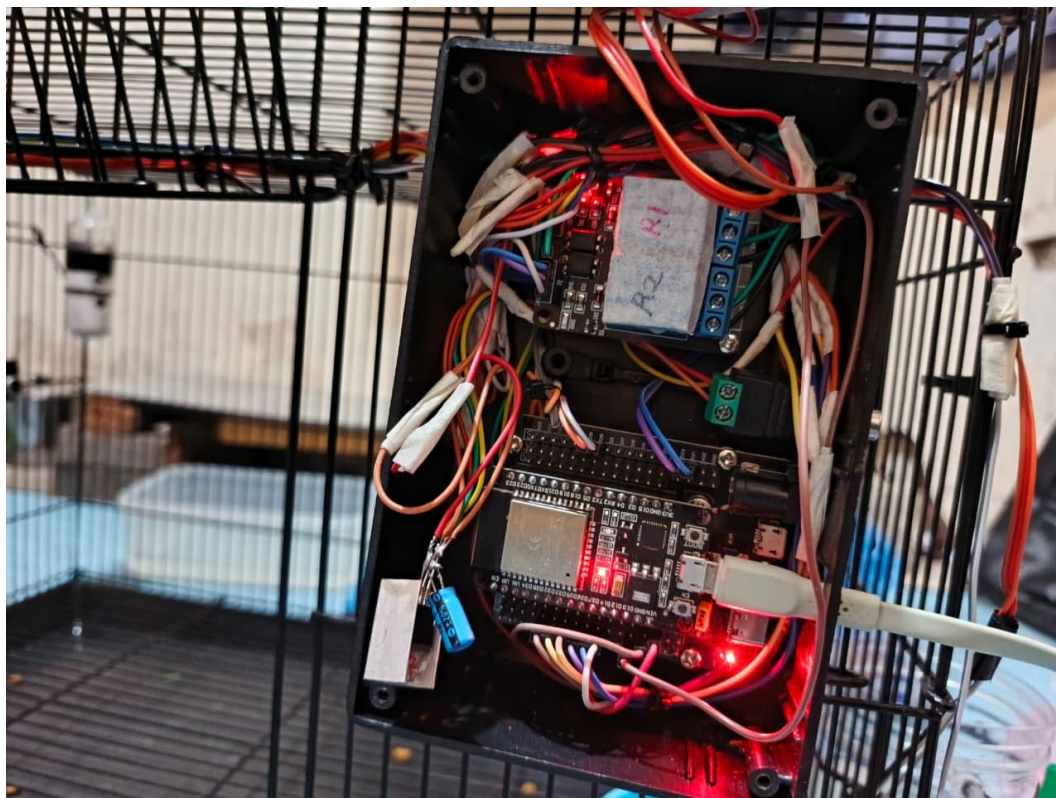
```

```
long durasi = pulseIn(PIN_ULTRASONIC_PAKAN_ECHO, HIGH, 25000);  
float jarakCm = durasi * 0.034 / 2;  
  
return (jarakCm > 0 && jarakCm < JARAK_STOK_HABIS_CM);  
}
```

B. Proses Pembuatan







ORIGINALITY REPORT

21 %

SIMILARITY INDEX

19 %

INTERNET SOURCES

7 %

PUBLICATIONS

8 %

STUDENT PAPERS

PRIMARY SOURCES

1	ejournal.undiksha.ac.id Internet Source	2 %
2	stt-pln.e-journal.id Internet Source	2 %
3	Submitted to Universitas Brawijaya Student Paper	1 %
4	eprints.utdi.ac.id Internet Source	<1 %
5	eprints.undip.ac.id Internet Source	<1 %
6	docplayer.info Internet Source	<1 %
7	Submitted to Universitas Diponegoro Student Paper	<1 %
8	portaldata.org Internet Source	<1 %
9	Submitted to itera Student Paper	<1 %
10	jurnal.untan.ac.id Internet Source	<1 %
11	repository.umsu.ac.id Internet Source	<1 %
12	openlibrarypublications.telkomuniversity.ac.id Internet Source	<1 %

13	text-id.123dok.com Internet Source	<1 %
14	ejurnal.methodist.ac.id Internet Source	<1 %
15	jurnal.wicida.ac.id Internet Source	<1 %
16	repository.uhamka.ac.id Internet Source	<1 %
17	repository.unhas.ac.id Internet Source	<1 %
18	Submitted to University of Sydney Student Paper	<1 %
19	eprints.perbanas.ac.id Internet Source	<1 %
20	oa.upm.es Internet Source	<1 %
21	repo.itera.ac.id Internet Source	<1 %
22	repo.palcomtech.ac.id Internet Source	<1 %
23	ojs.trigunadharma.ac.id Internet Source	<1 %
24	jtiik.ub.ac.id Internet Source	<1 %
25	Muhammad Arkan Fauzi, Setyawan Ajie Sukarno. "PENGEMBANGAN SISTEM PENDETEKSI KEBOCORAN GAS LPG BERBASIS IOT: INTEGRASI SENSOR MQ-02 DAN DHT11 UNTUK PEMANTAUAN REAL-TIME", Jurnal Informatika dan Teknik Elektro Terapan, 2025	<1 %

26	journal.artei.or.id Internet Source	<1 %
27	lib.unnes.ac.id Internet Source	<1 %
28	Submitted to Universitas Muria Kudus Student Paper	<1 %
29	Submitted to Universitas Bengkulu Student Paper	<1 %
30	Submitted to Politeknik Kesehatan Kemenkes Surabaya Student Paper	<1 %
31	proceedings2.upi.edu Internet Source	<1 %
32	repo.undiksha.ac.id Internet Source	<1 %
33	sisformik.atim.ac.id Internet Source	<1 %
34	ojs.stttexmaco.ac.id Internet Source	<1 %
35	www.scribd.com Internet Source	<1 %
36	Muhammad Aqsha Rizki Sugiarto, Moh Muhtarom, Anindhiasti Ayu Kusuma Asri. "Implementasi Sistem Pemberian Pakan Ikan Hias Otomatis Menggunakan Esp32 Berbasis IoT (Internet of Things)", Jurnal Indonesia : Manajemen Informatika dan Komunikasi, 2024 Publication	<1 %

37	repo.darmajaya.ac.id Internet Source	<1 %
38	repository.dinamika.ac.id Internet Source	<1 %
39	repository.poliupg.ac.id Internet Source	<1 %
40	Submitted to Fakultas Ekonomi Universitas Indonesia Student Paper	<1 %
41	Submitted to Universitas Putera Batam Student Paper	<1 %
42	www.beritasore.co.id Internet Source	<1 %
43	Submitted to Southville International School and Colleges Student Paper	<1 %
44	Submitted to Universitas Budi Luhur Student Paper	<1 %
45	Submitted to Universitas Muhammadiyah Sumatera Utara Student Paper	<1 %
46	jurnal.fikom.umi.ac.id Internet Source	<1 %
47	Azran Budi Arief. "RANCANG BANGUN SISTEM KENDALI OTOMATIS DAN MONITORING SUHU DAN KELEMBAPAN PADA RUMAH BURUNG WALET BERBASIS BLYNK", Jurnal Informatika dan Teknik Elektro Terapan, 2024 Publication	<1 %
48	Submitted to Institut Teknologi Kalimantan Student Paper	<1 %

49	repository.unugiri.ac.id Internet Source	<1 %
50	Submitted to Institut Teknologi Nasional Malang Student Paper	<1 %
51	ejournal.uin-suka.ac.id Internet Source	<1 %
52	Submitted to Daegu Gyeongbuk Institute of Science and Technology Student Paper	<1 %
53	Epin Aji, Syamsul Bahri, Suhardi Suhardi. "SISTEM MONITORING DAN KONTROL PADA KANDANG KELINCI REX BERBASIS INTERNET OF THINGS", Coding Jurnal Komputer dan Aplikasi, 2023 Publication	<1 %
54	Submitted to Institute of Research & Postgraduate Studies, Universiti Kuala Lumpur Student Paper	<1 %
55	Submitted to Universitas Islam Majapahit Student Paper	<1 %
56	Submitted to Universitas Muslim Indonesia Student Paper	<1 %
57	geograf.id Internet Source	<1 %
58	jurnal.itpln.ac.id Internet Source	<1 %
59	jurusan.tik.pnj.ac.id Internet Source	<1 %

60

Internet Source

<1 %

61

Submitted to California State University,
Fresno

Student Paper

<1 %

62

Ilham Santoso, Miftah Farid Adiwisastra,
Bambang Kelana Simpony, Deddy Supriadi,
Dini Silvi Purnia. "IMPLEMENTASI NodeMCU
DALAM HOME AUTOMATION DENGAN
SISTEM KONTROL APLIKASI BLYNK", Swabumi,
2021

Publication

<1 %

63

Saharman Saharman, Fatma Agus
Setyaningsih, Suhardi Suhardi. "MONITORING
DAN KONTROL PEMBERIAN PAKAN DAN
MINUM PADA PETERNAKAN BURUNG PUYUH
BERBASIS INTERNET OF THINGS", Coding
Jurnal Komputer dan Aplikasi, 2022

Publication

<1 %

64

Submitted to Universitas Negeri Jakarta

Student Paper

<1 %

65

acikerisim.uludag.edu.tr

Internet Source

<1 %

66

alissonsuwfoley.blogspot.com

Internet Source

<1 %

67

ejournal.unis.ac.id

Internet Source

<1 %

68

kikipermatasari04.wordpress.com

Internet Source

<1 %

69

Submitted to Udayana University

Student Paper

<1 %

70	batrisyaedu.com Internet Source	<1 %
71	ecampus.iainbatusangkar.ac.id Internet Source	<1 %
72	eclipse.itbu.ac.id Internet Source	<1 %
73	eprints.unugha.ac.id Internet Source	<1 %
74	journal.mediapublikasi.id Internet Source	<1 %
75	Submitted to Politeknik Negeri Jember Student Paper	<1 %
76	join.if.uinsgd.ac.id Internet Source	<1 %
77	pdfcoffee.com Internet Source	<1 %
78	repository.unja.ac.id Internet Source	<1 %
79	Submitted to Universitas Mercu Buana Student Paper	<1 %
80	Submitted to Universitas Muhammadiyah Palembang Student Paper	<1 %
81	aijpkm.iaiq.ac.id Internet Source	<1 %
82	digilib.unila.ac.id Internet Source	<1 %
83	doaj.org Internet Source	<1 %

84	ejournal.itn.ac.id Internet Source	<1 %
85	jurnal.padangtekno.com Internet Source	<1 %
86	mulok.library.um.ac.id Internet Source	<1 %
87	ojs.stmik-banjarbaru.ac.id Internet Source	<1 %
88	repository.its.ac.id Internet Source	<1 %
89	repository.unib.ac.id Internet Source	<1 %
90	scholar.unand.ac.id Internet Source	<1 %
91	sfe-electronics.com Internet Source	<1 %
92	123dok.com Internet Source	<1 %
93	Dahira Unsulbar. "PROTOTYPE OF AUTOMATIC WATERGATE MONITORING SYSTEM IN DAMS BASED ON THE INTERNET OF THINGS", Jurnal Informatika dan Teknik Elektro Terapan, 2024 Publication	<1 %
94	Lisa Choirunnanda, Triana Rahmawati, Lamidi Lamidi. "Alat Ukur Tinggi untuk Pengujian Status Gizi Balita dengan Metode Anthropometry", Jurnal Teknokes, 2020 Publication	<1 %

95	Moch Firman hidayat, Basuki Rahmat, Andreas Nugroho Sihananto. "MONITORING DAN KONTROL RUANGAN OBAT DENGAN SISTEM IOT DAN FUZZY", Jurnal Informatika dan Rekayasa Elektronik, 2024	<1 %
----	--	------

Publication

96	adoc.pub Internet Source	<1 %
----	---	------

97	afiasi.unwir.ac.id Internet Source	<1 %
----	---	------

98	artikelpendidikan.id Internet Source	<1 %
----	---	------

99	digilib.uinkhas.ac.id Internet Source	<1 %
----	--	------

100	digilib.uinsby.ac.id Internet Source	<1 %
-----	---	------

101	dspace.uii.ac.id Internet Source	<1 %
-----	---	------

102	eprints.akakom.ac.id Internet Source	<1 %
-----	---	------

103	eprints.itn.ac.id Internet Source	<1 %
-----	--	------

104	eprints.polsri.ac.id Internet Source	<1 %
-----	---	------

105	eprints.ums.ac.id Internet Source	<1 %
-----	--	------

106	eprints.walisongo.ac.id Internet Source	<1 %
-----	--	------

107	es.scribd.com Internet Source	<1 %
-----	--	------

108 greenpub.org <1 %
Internet Source

109 id.wikihow.com <1 %
Internet Source

110 id.wondershare.com <1 %
Internet Source

111 jim.unindra.ac.id <1 %
Internet Source

112 journal.ipm2kpe.or.id <1 %
Internet Source

113 jurnalunibi.unibi.ac.id <1 %
Internet Source

114 moam.info <1 %
Internet Source

115 pt.scribd.com <1 %
Internet Source

116 repository.polman-babel.ac.id <1 %
Internet Source

117 repository.unj.ac.id <1 %
Internet Source

118 www.cantika.com <1 %
Internet Source

119 www.masa.biz.id <1 %
Internet Source

120 www.microstrain.com <1 %
Internet Source

121 Regita Aulia Safitri. "Smart Aquaponic:
Pemberi Pakan Ikan Serta Pengontrol ph
Otomatis Berbasis Energi Matahari dan Iot", <1 %

-
- 122 Sendi Agus Setiawan, Manurul Hidayat, Sutarti. "PROTOTYPE LAMPU PENERANGAN JALAN OTOMATIS MENGGUNAKAN SENSOR LDR BERBASIS ARDUINO UNO", PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer, 2024

Publication

<1 %

-
- 123 Zulkifli Zulkifli, Muhlis Muhallim, Hasnahwati Hasnahwati. "PENGEMBANGAN SISTEM ALARM DAN PEMADAM KEBAKARAN OTOMATIS MENGGUNAKAN INTERNET OF THINGS", Jurnal Informatika dan Teknik Elektro Terapan, 2024

Publication

<1 %

-
- 124 www.slideshare.net

Internet Source

<1 %

Exclude quotes On

Exclude matches Off

Exclude bibliography Off