

Automatic Cat Feeding System Based on The Internet of Things (IOT) With Time and Feed Weight Control

Sefi Ayuk Wijayanti¹, Joni Maulindar², Nurohman³

^{1,2,3} Teknik Informatika, Fakultas Ilmu Komputer, Universitas Duta Bangsa, Surakarta, 57155, Indonesia

Informasi Artikel

Diterima : 1 Agustus 2025
Revisi : 28 Agustus 2025
Publikasi : 30 September 2025

Kata Kunci:

Internet of Things
Pakan Kucing Otomatis
ESP32
Website
Kucing

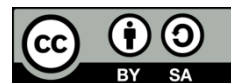
ABSTRAK

Permasalahan umum dalam perawatan kucing peliharaan adalah ketidakteraturan pemberian pakan yang dapat memengaruhi keseimbangan nutrisi dan kesehatan hewan. Untuk mengatasi hal ini, dikembangkan sebuah sistem pemberian pakan otomatis berbasis *Internet of Things* (IoT) yang memanfaatkan mikrokontroler ESP32. Sistem ini terintegrasi dengan sensor Passive Infrared (PIR) untuk deteksi gerakan kucing, *Load Cell* untuk pengukuran berat pakan, dan sensor ultrasonik untuk memantau stok pakan. Data disimpan dan ditampilkan melalui platform web yang mendukung pengaturan jadwal makan serta monitoring *real-time*. Pengujian sistem menggunakan pendekatan *black box* menunjukkan bahwa semua komponen bekerja dengan baik, termasuk sinkronisasi waktu dengan NTP, komunikasi Wi-Fi, dan fungsi sensor secara keseluruhan. Sistem ini diharapkan mampu meningkatkan konsistensi pemberian pakan serta mendukung kesehatan dan kesejahteraan kucing secara optimal.

ABSTRACT

A common problem in pet cat care is irregular feeding, which can affect the animal's nutritional balance and health. To address this issue, an Internet of Things (IoT)-based automatic feeding system utilizing an ESP32 microcontroller has been developed. The system integrates a Passive Infrared (PIR) sensor for cat motion detection, a Load Cell for measuring food weight, and an ultrasonic sensor to monitor food stock levels. Data is stored and displayed through a web-based platform that supports meal scheduling and real-time monitoring. System testing using a blackbox approach shows that all components function properly, including time synchronization via NTP, Wi-Fi communication, and overall sensor functionality. This system is expected to improve feeding consistency and optimally support the health and well-being of cats.

This is an open-access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license



*Penulis Koresponden

Email: 210103036@mhs.udb.ac.id

Cara sitasi IEEE:

S.A. Wijayanti, J. Maulindar & Nuurohman, "Automatic Cat Feeding System Based on The Internet Of Things (IOT) With Time And Feed Weight Control," *Journal of Artificial Intelligence and Software Engineering (J-AISE)*, vol. 5, no. 3, pp. 1151-1165, September 2025, doi: 10.30811/jaise.v5i3.7358

1. PENDAHULUAN

Salah satu permasalahan nyata yang masih sering ditemui dalam perawatan kucing adalah pemberian makan secara manual yang kurang konsisten. Banyak pemilik yang lupa memberi makan tepat waktu, kesulitan menakar jumlah pakan yang sesuai, atau terlalu sibuk sehingga tidak sempat memberi makan sama sekali.[1] Ketidakteraturan ini, apabila terus berlanjut, dapat mengganggu keseimbangan nutrisi dan berdampak negatif pada kesehatan kucing.

Indera pengecap kucing memainkan peran penting dalam memilih pakan yang memenuhi kebutuhan gizi mereka[2]. Sebagai karnivora obligat, kucing secara alami cenderung memilih makanan dengan kandungan protein dan lemak tinggi serta karbohidrat rendah. Penelitian menunjukkan bahwa kucing membutuhkan sekitar 26 gram protein, 9 gram lemak, dan 8 gram karbohidrat per hari, yang sesuai dengan kebutuhan kalori 52% dari protein, 36% dari lemak, dan 12% dari karbohidrat.

Menurut (Kartika et al., 2020) Untuk kucing peliharaan dengan berat sekitar 3,5 kg, mereka hanya membutuhkan sekitar 50 gram pakan kering per hari. Ketidaksesuaian dalam pemberian jumlah pakan tidak hanya menyebabkan kekurangan nutrisi, tetapi juga dapat berujung pada obesitas [3]. Obesitas pada kucing menjadi masalah yang serius karena dapat meningkatkan risiko berbagai penyakit metabolik. Salah satu solusi yang umum digunakan untuk mencegah obesitas adalah dengan mengatur asupan kalori secara ketat disertai peningkatan aktivitas fisik. Namun, tantangan utama tetap terletak pada konsistensi pemilik dalam menjalankan program diet untuk hewan peliharaannya.

Dengan memfasilitasi pola makan alami kucing di lingkungan rumah, strategi ini dapat menekan risiko masalah terkait stres maupun makan berlebih (*overeating*). Salah satu pendekatan efektif adalah dengan memberikan porsi kecil tetapi lebih sering.

Sejalan dengan perkembangan teknologi, berbagai inovasi telah dikembangkan untuk membantu pemilik kucing mengelola pemberian pakan. Salah satunya adalah penelitian oleh Habillah Abbas, Kusnadi, dan Wanda Ilham (2021) tentang sistem pemberi pakan otomatis berbasis IoT menggunakan modul NodeMCU ESP8266. Sistem ini dilengkapi modul RTC untuk pengaturan jadwal, sensor ultrasonik untuk memantau stok pakan, dan motor servo untuk membuka penutup sesuai jadwal. Selain beroperasi secara otomatis, sistem dapat dikendalikan jarak jauh melalui bot Telegram yang memberi notifikasi saat pakan diberikan atau dikonsumsi kucing. Hasil pengujian menunjukkan sistem berfungsi baik dalam kedua mode, meskipun kecepatan notifikasi tergantung pada stabilitas koneksi internet[4].

Penelitian serupa juga dilakukan oleh (Kasmawaru et al., 2024) dalam karya Sistem Kendali Cerdas Pemberian Pakan dengan Penerapan *Internet of Things*, di mana mereka mengintegrasikan NodeMCU ESP8266 sebagai pengendali utama yang bekerja sama dengan sensor ultrasonik, *load cell*, motor servo, dan mini pump untuk mendistribusikan pakan serta air. Sistem ini dapat dioperasikan dan dipantau melalui aplikasi Android, sehingga memungkinkan pengaturan pakan secara *real-time* dan jarak jauh [5].

Meskipun beberapa sistem pemberi pakan otomatis berbasis IoT telah dikembangkan, sejumlah keterbatasan masih teridentifikasi. Sistem oleh Habillah Abbas et al. (2021) mengandalkan RTC lokal yang rentan terhadap drift waktu dan tidak menggunakan pengukuran berat pakan secara langsung, sementara notifikasi melalui Telegram terbatas dalam fleksibilitas akses dan pemantauan. Di sisi lain, sistem Kasmawaru et al. (2024) telah menggunakan *load cell* dan antarmuka Android, namun pengukuran berat dilakukan sebelum dispensing, sehingga tidak memastikan takaran akhir yang diberikan

Berdasarkan keterbatasan tersebut, penelitian ini mengidentifikasi beberapa kesenjangan Utama Belum adanya sinkronisasi waktu akurat melalui NTP, yang penting untuk ketepatan jadwal meskipun terjadi restart perangkat. Tidak ada verifikasi takaran pakan secara *real-time* setelah distribusi, sehingga takaran aktual yang diterima kucing belum terjamin. Minimnya deteksi keberadaan kucing berbasis PIR, yang dapat mencegah pemborosan pakan. Keterbatasan antarmuka monitoring yang masih bergantung pada aplikasi mobile atau notifikasi telegram.

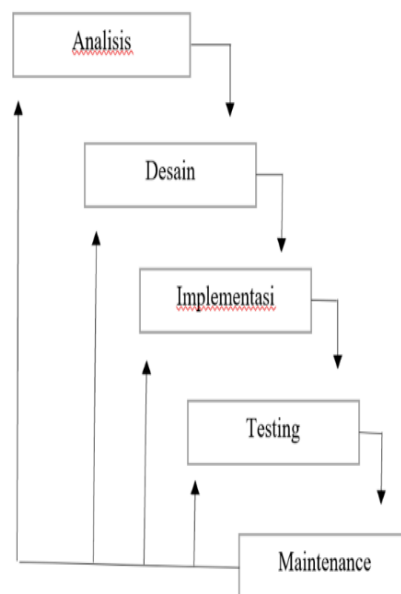
Oleh karena itu, penelitian ini menawarkan kontribusi kebaruan melalui rancangan sistem yang Menggunakan sinkronisasi waktu via NTP untuk ketepatan jadwal yang konsisten. Mengukur berat pakan secara *real-time* setelah servo membuka penutup wadah, memastikan takaran sesuai target. Mengintegrasikan sensor PIR untuk mendeteksi keberadaan kucing berdasarkan panas tubuh. Menghadirkan antarmuka berbasis web yang mendukung pengaturan jadwal dan pemantauan stok secara *real-time*.

Berdasarkan permasalahan tersebut, sekaligus melihat peluang pemanfaatan IoT dalam mendukung perawatan hewan peliharaan, penulis merancang sistem pemberian pakan kucing otomatis yang tidak hanya mampu mengatur waktu pemberian makan, tetapi juga memastikan takaran pakan. sistem ini dikendalikan dengan melalui platform berbasis web. Sistem ini memanfaatkan tiga jenis sensor, yaitu *Passive Infrared Sensor* (PIR) untuk mendeteksi gerakan dan suhu tubuh kucing, *load cell* untuk mengukur berat pakan secara presisi setelah servo membuka penutup wadah, serta sensor ultrasonik untuk memantau stok pakan. Seluruh

sistem disinkronkan dengan waktu server melalui NTP, sehingga setiap proses berjalan teratur layaknya jam jaringan yang selaras. Dengan rancangan ini, pemberian makan diharapkan dapat berlangsung konsisten, terkontrol, dan mampu menjaga kesehatan serta kesejahteraan kucing peliharaan secara optimal.

2. METODE

Rosa dan Shalahuddin (2014:28) menjelaskan bahwa metode waterfall, atau sering disebut sebagai classic life cycle (siklus hidup klasik), adalah pendekatan yang mengembangkan perangkat lunak secara berurutan, melalui tahapan-tahapan yang saling terhubung. Proses dimulai dengan tahap analisis kebutuhan (*analytics*), dimana kebutuhan sistem dirinci dan dipahami. Tahapan ini dilanjutkan dengan desain sistem (*design*), yang mencakup perencanaan struktur dan tampilan perangkat lunak. Setelah itu, dilakukan pengkodean (Implementasi), yaitu proses implementasi desain ke dalam bentuk kode program. Selanjutnya, perangkat lunak diuji dalam tahap pengujian (*testing*) untuk memastikan fungsionalitasnya sesuai spesifikasi. Terakhir, sistem memasuki tahap dukungan atau pemeliharaan (*Maintenance*), yang berfokus pada perawatan dan pembaruan perangkat lunak agar tetap berfungsi optimal. Pendekatan ini memastikan bahwa setiap tahap pengembangan dilakukan dengan urutan yang sistematis dan terstruktur.



Gambar 1. Metode *Waterfall*

3. HASIL DAN PEMBAHASAN

3.1. Analisis Kebutuhan

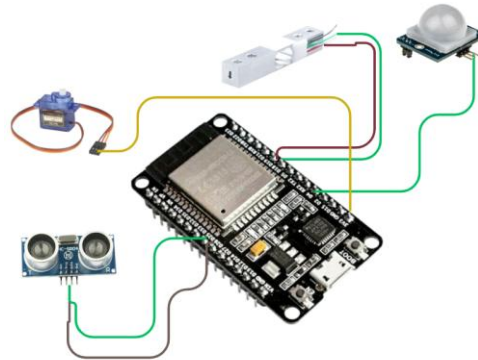
Analisis sistem membantu mengidentifikasi kebutuhan perangkat keras dan perangkat lunak agar sistem berjalan optimal. Perangkat keras utama meliputi mikrokontroler ESP32 sebagai pusat kendali, sensor berat untuk mengukur pakan, sensor PIR untuk mendeteksi keberadaan kucing, motor servo untuk membuka wadah makanan, serta sensor ultrasonik untuk memantau sisa pakan. Semua komponen dihubungkan secara efisien menggunakan kabel jumper dan modul ekspansi ESP32 DevKit.

Kemudian ada perangkat lunak, menggunakan Bahasa C++ pada Arduino IDE untuk mengatur logika kerja, dengan data disimpan dalam database MySQL seperti jadwal dan riwayat pemberian pakan. Antarmuka web dibangun dengan HTML dan CSS agar tampilan menarik dan responsif di berbagai perangkat, sementara JavaScript digunakan untuk menampilkan data *real-time* dari sistem IoT. Integrasi antara perangkat keras dan perangkat lunak ini memungkinkan pengelolaan pemberian pakan secara otomatis dan terpantau dengan baik.

3.2. Desain Sistem

Pada tahapan desain ini, penulis melakukan perancangan rangkaian sistem secara menyeluruh, termasuk komponen elektronik dan alur kerjanya, serta mengembangkan desain yang mencakup aspek fungsional dan kesesuaian dengan kebutuhan. Proses tersebut dilakukan dengan mempertimbangkan faktor teknis dan kemudahan

3.2.1. Desain Rangkaian Sistem



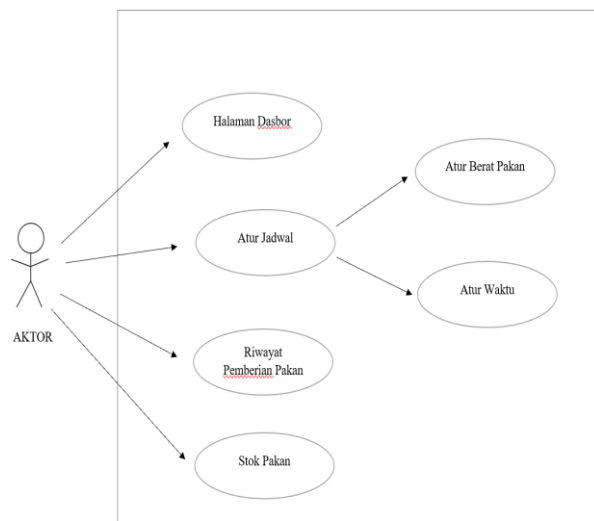
Gambar 2. Skema Rangkaian Alat

Mikrokontroler ESP32 menjadi otak sistem yang terhubung via Wi-Fi untuk komunikasi dengan server, sinkronisasi waktu lewat NTP, serta mengirim data pemberian pakan dan stok ke server. Sensor berat (*load cell*) memastikan pakan dikeluarkan sesuai berat yang ditentukan, sementara motor servo mengatur buka-tutup dispenser. Saat kucing mendekat, sensor PIR mendeteksi keberadaannya dan memicu pemberian pakan otomatis. Stok pakan dipantau secara berkala oleh sensor ultrasonik yang mengukur ketinggian pakan, memberi notifikasi jika isi wadah hampir habis. Semua komponen bekerja terintegrasi untuk sistem pemberian pakan yang akurat dan otomatis.

3.2.2. Perancangan Sistem UML (Unified Modeling Language)

Perancangan sistem pemberian pakan otomatis berbasis IoT dengan kontrol waktu dan berat pakan menggunakan pemodelan berbasis *Unified Modeling Language* (UML) sebagai standar desain. UML membantu memvisualisasikan sistem melalui diagram seperti use case untuk menunjukkan interaksi antara pengguna dan sistem, serta *activity diagram* yang menggambarkan alur proses secara jelas. Dengan pendekatan ini, aplikasi dapat dirancang secara efektif, mudah dipahami, dan fleksibel untuk dikembangkan di masa depan.

4.2.2.1 Use Case Diagram



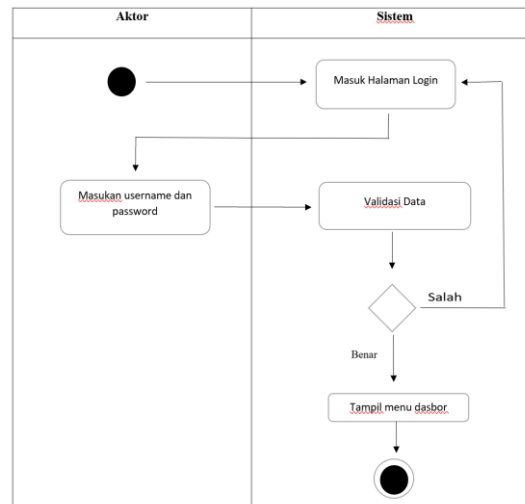
Gambar 3. Use Case Diagram

Gambar 3 menunjukkan satu aktor utama, yaitu pengguna, yang memiliki akses penuh ke seluruh fitur sistem. Pengguna dapat mengakses halaman dasbor untuk memantau informasi terkini, mengatur jadwal dan waktu pemberian pakan, serta menyesuaikan pengaturan sesuai kebutuhan. Selain itu, pengguna juga bisa melihat riwayat pemberian pakan dan memeriksa stok pakan untuk memastikan ketersediaan makanan. Dengan akses yang lengkap ini, pengguna memiliki kendali penuh atas pengelolaan sistem pemberian pakan otomatis.

4.2.2.2 Activity Diagram

Activity Diagram adalah salah satu alat dalam pemodelan sistem yang digunakan untuk menggambarkan alur proses atau logika kerja secara fungsional. Diagram ini memperlihatkan urutan kegiatan yang terjadi dalam suatu use case, sekaligus memvisualisasikan alur dinamis sistem sehingga memudahkan pemahaman tentang cara kerja setiap fungsi di dalamnya.

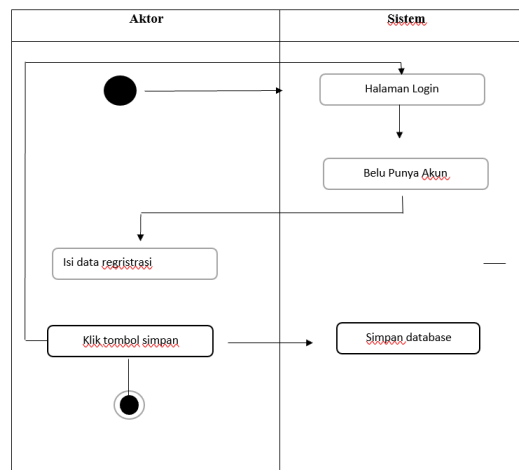
1. Login



Gambar 4. Activity Diagram Login

Pada gambar di atas menggambarkan proses login pengguna ke dalam sistem. Proses dimulai saat pengguna masuk ke halaman login, lalu memasukkan username dan password. Sistem kemudian melakukan validasi terhadap data yang dimasukkan. Jika username dan password benar, pengguna diarahkan ke menu dashboard sebagai tanda berhasil masuk. Namun jika data salah, pengguna diminta mencoba kembali hingga informasi yang dimasukkan valid. Proses berakhir setelah pengguna berhasil masuk dan menu dashboard ditampilkan.

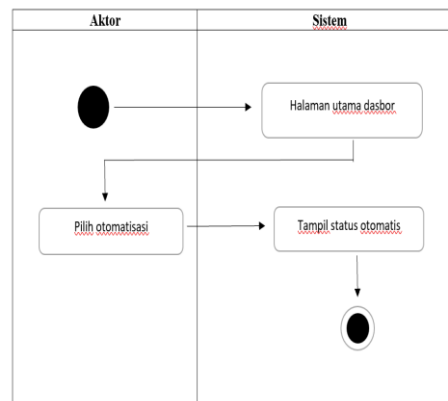
2. Daftar



Gambar 5. Activity Diagram Daftar

Pada gambar 5, menunjukkan proses registrasi pengguna, mulai dari pengguna memilih opsi "Belum Punya Akun" hingga data berhasil disimpan ke database. Pengguna diarahkan untuk mengisi formulir registrasi, lalu menekan tombol "Simpan" agar data disimpan ke sistem. Proses berakhir saat data pengguna tersimpan dengan sukses.

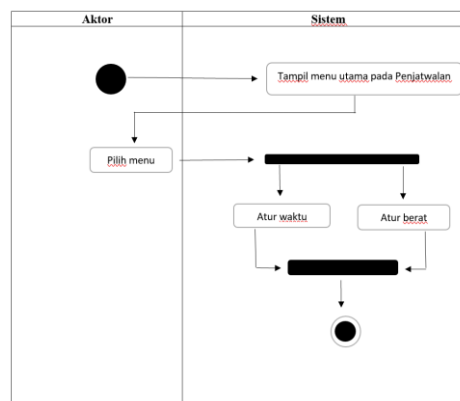
3. Dasbor



Gambar 6. Activity Diagram Dashboard

Dapat dilihat pada gambar 6, Diagram ini menunjukkan proses pengelolaan fitur otomatisasi, mulai dari pengguna masuk ke halaman dashboard, lalu memilih opsi "Pilih otomatisasi". Setelah itu, sistem menampilkan status otomatisasi kepada pengguna. Proses berakhir saat informasi status otomatis berhasil ditampilkan.

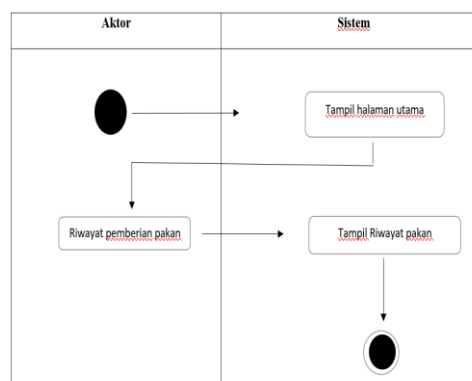
4. Atur Jadwal



Gambar 7. Activity Diagram Atur Jadwal

Pada gambar 7, menunjukkan proses penjadwalan, mulai dari pengguna melihat menu utama penjadwalan hingga mereka mengatur waktu dan berat sesuai kebutuhan. Setelah pengaturan selesai, proses berakhir dengan semua konfigurasi tersimpan.

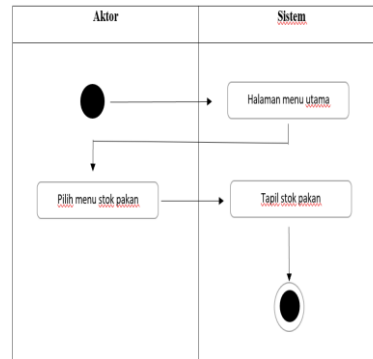
5. Riwayat Pemberian Pakan



Gambar 8. Activity Diagram Riwayat Pemberian Pakan

Diagram ini menunjukkan proses pemanggilan riwayat pemberian pakan, mulai dari pengguna melihat halaman utama hingga mereka memilih menu "Riwayat pemberian pakan". Setelah itu, sistem menampilkan riwayat pakan sesuai permintaan. Proses berakhir saat riwayat pakan ditampilkan dengan sukses.

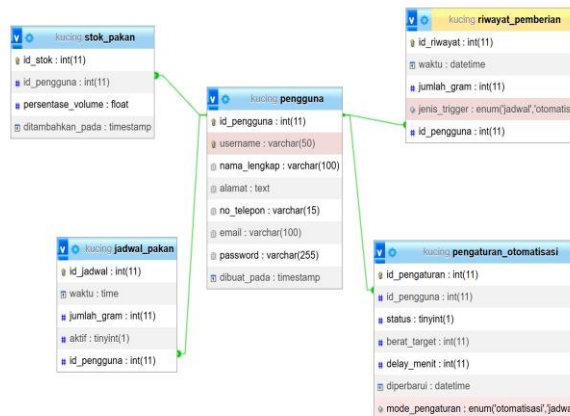
6. Stok Pakan



Gambar 8. Activity Diagram Stok Pakan

Diagram ini menunjukkan proses pemanggilan informasi stok pakan, mulai dari pengguna melihat halaman menu utama hingga mereka memilih menu "Pilih menu stok pakan". Setelah itu, sistem menampilkan data stok pakan sesuai permintaan. Proses berakhir saat informasi stok pakan ditampilkan dengan sukses

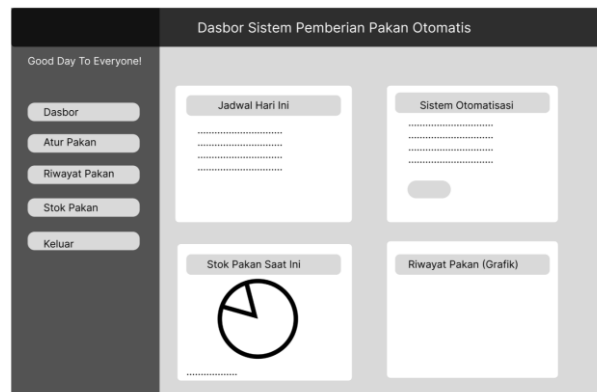
4.2.2.3 Class diagram



Gambar 9. Class Diagram

Terlihat bahwa kelas Pengguna menjadi inti sistem yang terhubung dengan seluruh fitur utama. Pengguna dapat mengatur jadwal pakan melalui kelas JadwalPakan, memantau aktivitas pemberian pakan di Riwayat Pemberian, serta mengelola konfigurasi otomatis di Pengaturan Otomatisasi. Stok pakan dipantau lewat kelas Stok Pakan yang mencatat volume tersisa. Semua kelas memiliki relasi langsung dengan Pengguna, menunjukkan struktur yang terpusat, terorganisir, dan memudahkan pengelolaan sistem secara personal.

3.2.3. Perancangan Antar Muka



Gambar 10. Perancangan Antar Muka

Pada gambar 10, mencerminkan desain antarmuka yang fokus pada efisiensi dan kemudahan penggunaan. Setiap elemen memiliki tujuan spesifik, mulai dari navigasi yang intuitif hingga presentasi data yang visual dan informatif. Dengan demikian, pengguna dapat dengan cepat memantau dan mengelola sistem pemberian pakan otomatis sesuai kebutuhan mereka.

3.3. Implementasi

Implementasi sistem dilakukan secara bertahap, dimulai dari perakitan perangkat keras seperti mikrokontroler ESP32, sensor berat, sensor PIR, motor servo, dan sensor ultrasonik yang disusun menjadi satu sistem terintegrasi. Setelah perangkat keras siap, dilanjutkan dengan penulisan program menggunakan C++ di Arduino IDE untuk mengatur logika sistem, termasuk pembacaan sensor, kendali aktuator, dan komunikasi via Wifi. Tahap terakhir adalah pengembangan antarmuka pengguna berupa aplikasi web yang memudahkan pengguna dalam memantau dan mengelola sistem secara *real-time*.

3.3.1. Perakitan Perangkat Keras

Dibawah ini merupakan gambar Tampilan Perangkat Keras



Gambar 11. Rangkaian Perangkat Keras

Pada gambar di atas, hanya expansion board dengan ESP32 yang tampak terpasang. Hal ini karena semua sensor dan modul pendukung sudah terintegrasi pada posisi masing-masing. Motor servo dipasang antara tabung penyimpanan dan pipa pengeluaran pakan, berfungsi sebagai katup otomatis untuk mengontrol aliran pakan. Setelah pakan dikeluarkan, ia mengalir melalui pipa ke wadah di bawahnya. Di bawah wadah, terdapat sensor load cell bersama modul HX711 untuk mengukur berat pakan secara akurat. Di depan wadah, sensor PIR mendeteksi keberadaan hewan peliharaan; ketika hewan mendekat, sensor ini memicu sistem otomatisasi. Di atas tabung penyimpanan, sensor ultrasonik memantau volume pakan tersisa, sehingga pengguna tahu kapan harus mengisi ulang. Semua komponen dirancang rapi dan efisien, menjadikan sistem fungsional serta mudah dioperasikan.

3.3.2. Penulisan Kode Program

Program untuk sistem pemberian pakan otomatis berbasis IoT ini dibuat menggunakan Arduino IDE, merupakan seperti gambar 12, dibawah ini.

```
#include <HTTPClient.h>
#include <WiFiUdp.h>
#include <NTPCClient.h>
#include <ESP32Servo.h>
#include <HX711_ADC.h>
#include <ArduinoJson.h>
#include <EEPROM.h>

// ===== KONFIGURASI WIFI =====
const char* ssid = "espwifi"; // Nama WiFi
const char* password = "esp12345"; // Password WiFi
// ===== ALAMAT SERVER API =====
const char* uriRiwayat = "https://maui-counsel-is-chronic.trycloudflare.com/kucing/backend/api/riwayat_pakan.php";
const char* serverUrl = "https://maui-counsel-is-chronic.trycloudflare.com/kucing/backend/api/jadwal_esp.php";
const char* urlStokPakan = "https://maui-counsel-is-chronic.trycloudflare.com/kucing/backend/api/update_stok.php";
const char* esp_secret = "esp_lucky_ayuk"; // Kunci rahasia untuk validasi
// ===== KONFIGURASI SERVO =====
Servo myServo; // Objek servo
const int SERVO_PIN = 27; // Pin servo terhubung ke GPIO 27
// ===== KONFIGURASI LOAD CELL =====
#define DT_PIN 32 // Pin DT Load Cell ke GPIO 32
#define SCK_PIN 26 // Pin SCK Load Cell ke GPIO 26
HX711_ADC LoadCell(DT_PIN, SCK_PIN); // Inisialisasi sensor Load Cell
float calibrationValue = 650.75; // Nilai kalibrasi dari hasil uji (sudah akurat)
// ===== KONFIGURASI PIR (SENSOR GERAK) =====
const int PIR_PIN = 25; // Pin PIR terhubung ke GPIO 25
int status_pir_sebelumnya = LOW; // Untuk deteksi perubahan (rising edge)
bool servo_sudah_jalan = false; // Cegah servo aktif berulang
unsigned long waktuTerakhirPakan = 0; // Catat waktu terakhir makanan dikeluarkan
// ===== KONFIGURASI WAKTU (NTP) =====
WiFiUDP ntpUDP; // UDP untuk sinkronisasi waktu
NTPCClient timeClient(ntpUDP, "pool.ntp.org", 25280); // Server NTP + offset +7 (WIB)
// ===== VARIABEL SISTEM =====
String jenis_trigger = "jadwal"; // Mode default: jadwal
int berat_target = 0; // Target berat pakan (dari server)
```

Gambar 12. Program Arduino

```
int delay_menit = 0; // Delay antar pakan (dari server)
// ===== KONFIGURASI ULTRASONIK (UKUR STOK) =====
#define ULTRA_TRIG_PIN 4 // Pin Trig ultrasonik ke GPIO 4
#define ULTRA_ECHO_PIN 16 // Pin Echo ultrasonik ke GPIO 16
const float STOK_PENUH_JARAK_CM = 6.88; // Jarak saat wadah penuh (cm)
const float STOK_KOSONG_JARAK_CM = 19.20; // Jarak saat wadah kosong (cm)
#define PIN_TOMBOL_REKALKULASI 13 // Tombol manual untuk rekalkulasi stok
// ===== Fungsi bantu: hitung median untuk stabilisasi jarak =====
float medianOf(float *arr, int n) {
  for (int i = 0; i < n - 1; i++) {
    for (int j = 0; j < n - i - 1; j++) {
      if (arr[j] > arr[j + 1]) {
        float t = arr[j];
        arr[j] = arr[j + 1];
        arr[j + 1] = t;
      }
    }
  }
  if (n % 2 == 1) return arr[n / 2];
  return (arr[n / 2 - 1] + arr[n / 2]) / 2.0;
}
// ===== SETUP (DIJALANKAN SAAT START) =====
void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) delay(500);
  pinMode(PIR_PIN, INPUT);
  timeClient.begin();
  timeClient.update();
  myServo.setPeriodHertz(60);
  myServo.attach(SERVO_PIN, 500, 2400);
  myServo.write(0);
  // Inisialisasi Load Cell
  LoadCell.begin();
  LoadCell.start(2000);
```

Gambar 13. Program Arduino

```

while (!LoadCell.update());
LoadCell.setCalFactor(calibrationValue);
LoadCell.tare(); // Nolkan sensor
// Set pin ultrasonik dan tombol
pinMode(ULTRA_TRIG_PIN, OUTPUT);
pinMode(ULTRA_ECHO_PIN, INPUT);
pinMode(PIN_TOMBOL_REKALKULASI, INPUT_PULLUP);
Serial.println("✅ Setup selesai. Load Cell siap dengan kalibrasi: 650.75");
}
// ===== FUNGSI UTAMA: KELUARKAN PAKAN SESUAI TARGET =====
void dispensasiSampaiTarget(int targetGram) {
const unsigned long TIMEOUT_MS = 30000; // Timeout 30 detik
unsigned long waktuMulai = millis();
const int POS_BUKA = 35; // Sudut servo saat BUKA
const int POS_TUTUP = 0; // Sudut servo saat TUTUP
// 🐛 PENGATURAN DELAY - UBAH DI SINI
const int DELAY_BUKA_MS = 180; // 🕒 Waktu servo TERBUKA (ms) - pakan jatuh
// 🕒 Coba 100 (lebih cepat) atau 250 (lebih lambat)
const int DELAY_TUTUP_MS = 4000; // 🕒 Waktu TERTUTUP + delay (ms) - 4000 = 4 detik
// 🕒 Kurangi untuk lebih cepat, tambah untuk lebih lama
Serial.println("🔥 Dispensasi BUKA-TUTUP + DELAY dimulai sampai target: " + String(targetGram) + "g");
float beratSaatIni = 0.0;
while (millis() - waktuMulai < TIMEOUT_MS) {
// --- 1. BUKA Servo (cepat) ---
myServo.write(POS_BUKA);
delay(DELAY_BUKA_MS);
// 📖 BACA LOAD CELL TERUS-MENERUS SELAMA BUKA (lebih sensitif)
Serial.println("📖 BUKA - Membaca berat secara terus-menerus...");
unsigned long waktuBaca = millis();
float totalBaca = 0;
int jumlahBaca = 0;
while (millis() - waktuBaca < 200) { // Baca 200ms saat terbuka
if (LoadCell.update()) {
float b = LoadCell.getData();

```

Gambar 14. Program Arduino

```

servo_sudah_jalan = raise;
Serial.println("🕒 Delay selesai. Servo bisa aktif lagi.");
}
}
if (!servo_sudah_jalan && status_pir_sebelumnya == LOW && status == HIGH) {
Serial.println("🟢 Gerakan PIR terdeteksi (rising edge)!");
dispensasiSampaiTarget(berat_target);
rekalkulasiStokPakan();
kirimRiwayat(ambilBeratRataRata(), "otomatis");
servo_sudah_jalan = true;
waktuTerakhirPakan = millis();
Serial.println("🔥 Pakan selesai didistribusikan via PIR.");
}
status_pir_sebelumnya = status;
}
// === MODE JADWAL ===
else if (jenis_trigger == "jadwal") {
JSONArray jadwal = doc["jadwal"];
for (JsonObject slot : jadwal) {
int jam_jadwal = slot["jam"];
int menit_jadwal = slot["menit"];
int targetGram = slot["gram"];
if (jam_jadwal == jam && menit_jadwal == menit) {
dispensasiSampaiTarget(targetGram);
rekalkulasiStokPakan();
kirimRiwayat(ambilBeratRataRata(), "jadwal");
delay(60000);
break;
}
}
}
}
}
http.end();
}
delay(50);
}

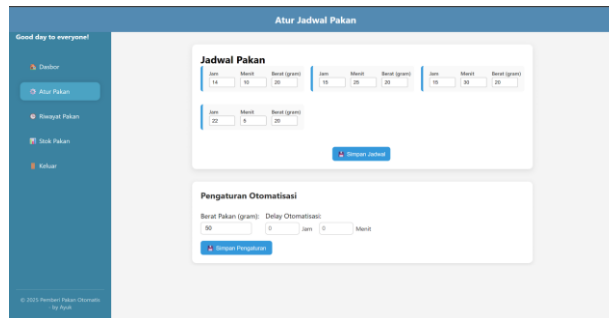
```

Gambar 15. Program Arduino

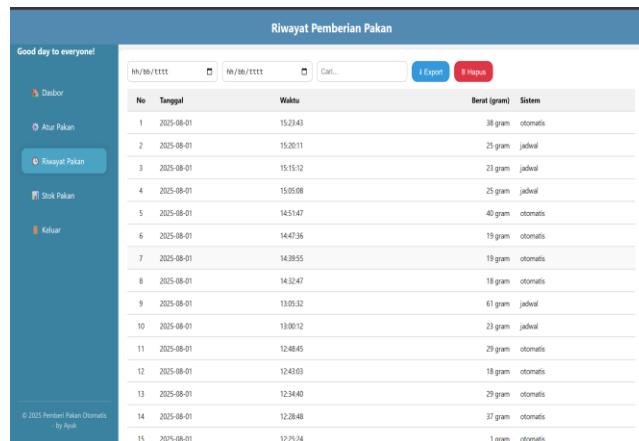
3.3.3. Pengembangan Antar Muka



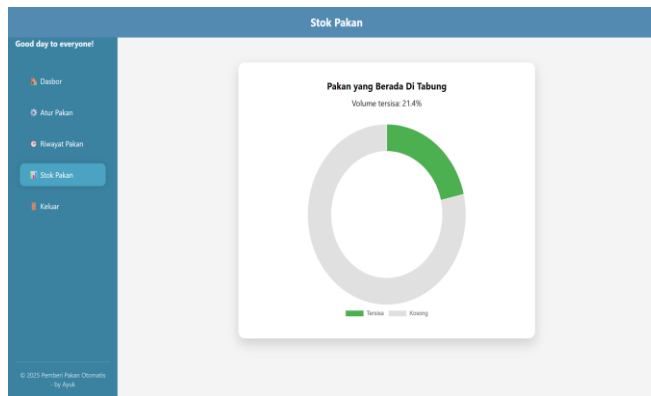
Gambar 16. Halaman Dasbor.



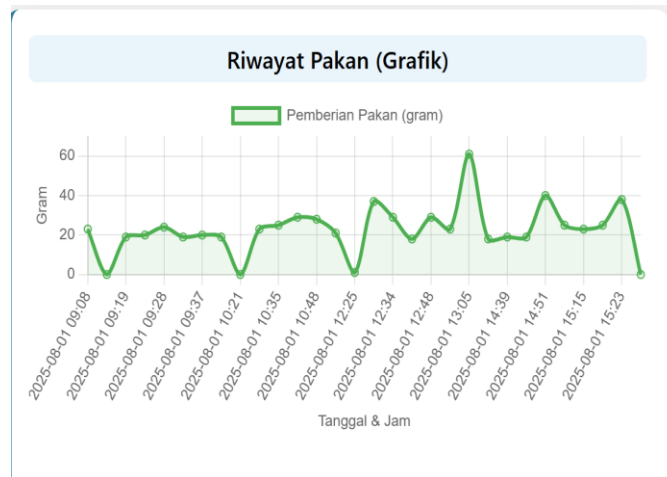
Gambar 17. Halaman Atur Pakan



Gambar 18. Halaman Riwayat Pakan



Gambar 19. Halaman Stok Pakan



Gambar 20. Grafik Riwayat Pakan

Antarmuka sistem dirancang intuitif dan mudah digunakan, dengan halaman login, dasbor, atur jadwal, riwayat, dan stok pakan yang saling terintegrasi. Berdasarkan pengujian menggunakan *Chrome DevTools*, seluruh halaman memiliki waktu muat di bawah 100ms seperti halaman dasbor (71 ms) dan login (72 ms). Interaksi seperti menyimpan jadwal atau ekspor riwayat merespons dalam <100 ms, sementara pembaruan data stok pakan dari sensor ultrasonik hanya membutuhkan rata-rata 7 ms. Sistem menunjukkan kinerja responsif dan stabil, dengan komunikasi real-time antara perangkat keras (ESP32) dan antarmuka web berjalan lancar.

3.4. Pengujian

Dalam tahap ini, digunakan pendekatan black box testing sebagai metode utama, yaitu pengujian fungsional yang difokuskan pada input dan output sistem tanpa melibatkan analisis terhadap kode program di dalamnya. Pengujian dilakukan secara langsung pada perangkat yang telah terintegrasi, dengan mempertimbangkan skenario penggunaan nyata. Fokus pengujian meliputi kinerja sensor seperti sensor PIR, load cell, dan ultrasonik, ketepatan sistem dalam menjalankan jadwal pemberian pakan, berat pakan yang dikeluarkan, serta penerapan delay antar pemberian dalam mode otomatisasi.

Tabel 1. Pengujian Dengan Mode Jadwal

Waktu	Jadwal	Target Berat (G)	Status	Keterangan
08.15	08.15	20g	Berhasil	Motor servo aktif, pakan sesuai target yaitu melebihi target
08.20	08.20	20g	Berhasil	Motor servo aktif, pakan sesuai target yang diharapkan
08.25	08.25	15g	Berhasil	Motor servo aktif, pakan sesuai target yang diharapkan
08.30	08.30	20g	Gagal	Motor servo aktif, pakan sesuai target yang diharapkan

Tabel 2. Pengujian Mode Otomatisasi

Delay	Target Berat (G)	Status	Keterangan
4 menit	15 g	Berhasil	Motor servo aktif, delay sesuai target dan pakan sesuai target yang diharapkan
5 menit	20g	Berhasil	Motor servo aktif, delay sesuai target dan pakan sesuai target yang diharapkan

Tabel 3. Pengujian Fungsi Sensor dan Modul

Sensor	kondisi	Langkah Pengujian	Output yang Diharapkan	Status
--------	---------	-------------------	------------------------	--------

Sensor PIR	Perangkat dinyalakan dalam jangkauan Wi-Fi, delay dalam masa non aktif	Mendekatkan Tangan ke depan	Gerakan Terdeteksi Mengeluarkan Pakan	Berhasil
Sensor PIR	Delay Dalam masa aktif	Mendekatkan Tangan ke depan	Sensor pir tidak akan mendeteksi meskipun aja Gerakan.	Berhasil
Motor servo	Waktu telah masuk jam di server	Memberikan jadwal	Servo akan membuka Ketika jadwal sudah memasuki waktu yang ditentukan	Berhasil
Motor servo	Delay aktif	Tunggu sampai delay selesai	Servo Kembali aktif setelah delay	Berhasil
Sensor Loadcell	Servo membuka dan loadcell membaca terus menerus berat	Pemberian pakan dimulai	<i>Load Cell</i> membaca berat	Berhasil
Sensor Ultrasonik	Servo telah selesai menutup	Ultrasonik membaca jarak	Data dikirim ke server	Berhasil

Tabel 5. Respon Sensor Pir

Percobaan	Waktu Respon (Detik)
1	0,86
2	0,55
3	0,77
4	0,30

Tabel 6. Pengujian Akurasi *Load Cell*

Target Berat	Berat yang Dikeluarkan oleh Sistem	Nilai Selisih	Presentase Akurasi
20 (G)	22 (G)	+2 (G)	95.45 %
20 (G)	20 (G)	0	100 %
20 (G)	25 (G)	+ 5 (G)	80 %
15 (G)	19 (G)	+ 4 (G)	78.95%
15 (G)	16 (G)	+ 1 (G)	93.75 %

Pengujian sistem menunjukkan bahwa seluruh komponen dan fitur utama telah berfungsi sesuai rancangan. Koneksi Wi-Fi stabil dan sinkronisasi waktu melalui NTP berjalan dengan akurat, memungkinkan sistem menjalankan jadwal pemberian pakan secara tepat waktu. Sensor PIR mampu mendeteksi gerakan hewan dengan baik dan memicu pemberian pakan dalam mode otomatisasi, termasuk penerapan delay untuk mencegah overfeeding. Motor servo bekerja responsif dalam membuka dan menutup dispenser sesuai perintah, baik dari jadwal maupun deteksi PIR

Pada tabel 6 dapat disimpulkan bahwa Sistem dirancang dengan pakan dikeluarkan secara bertahap hingga berat mencapai atau melebihi target yang telah ditentukan oleh pengguna. Proses pemberian pakan dimulai saat servo membuka katup, dan sensor load cell terus membaca berat secara *real-time*. Ketika berat pakan yang keluar sudah mencapai atau sedikit melebihi target, sistem langsung memerintahkan servo untuk menutup katup. Ini berarti, servo hanya akan menutup setelah target tercapai, meskipun kadang berat akhir sedikit melebihi karena waktu respons sistem. Logika ini memastikan bahwa kucing tidak kekurangan pakan, meskipun terjadi toleransi kecil di atas target

4. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian, sistem pemberian pakan kucing berbasis Internet of Things dengan pengendalian waktu dan berat pakan telah berhasil dibangun dan berfungsi sesuai dengan tujuan. Integrasi antara perangkat keras dan perangkat lunak berjalan dengan baik, didukung oleh koneksi Wi-Fi yang stabil dan sinkronisasi waktu melalui NTP. Sistem mampu menjalankan pemberian pakan secara otomatis berdasarkan jadwal yang ditentukan atau dipicu oleh deteksi keberadaan kucing menggunakan sensor PIR. Kontrol berat pakan dilakukan secara real-time dengan sensor load cell, cukup akurat. Fitur delay mencegah pemberian pakan berulang, sementara sensor ultrasonik memantau stok pakan secara berkala. Data

riwayat dan stok pakan berhasil dikirim ke server, memungkinkan pemantauan jarak jauh melalui antarmuka web. Secara keseluruhan, sistem ini telah terbukti fungsional, akurat

REFERENSI

- [1] M. P. Utami, M. H. H. Ichsan, and D. Syaury, "Implementasi Regresi Linear terhadap Optimalisasi Food Dosage Estimation pada Pemberi Pakan Kucing," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 4, pp. 2029–2037, 2023, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [2] M. H. Sabila, R. Afriadi, S. W. Tampubolon, and N. Nuha, "Pemanfaatan Ikan dan Tempe untuk Kebutuhan Nutrisi Pakan Kucing Ramah Lingkungan," *JRIKUF J. Ris. Ilmu Kesehat. Umum*, vol. 3, pp. 23–32, 2025.
- [3] G. R. Saudale, *Saudale, G. R. (2018). Aplikasi Liposuction Pada Kucing (Felis Catus) Overweight Terhadap Kadar Blood Urea Nitrogen (Bun) Dan Kreatinin.* 2018.
- [4] H. Abbas, K. Kusnadi, W. Ilham, and S. Parman, "Sistem Kendali Alat Pemberi Pakan Kucing Otomatis Menggunakan Modul Nodemcu," *J. Digit*, vol. 11, no. 2, p. 166, 2021, doi: 10.51920/jd.v11i2.202.
- [5] K. Kasmawaru, H. Husain, H. Herlinda, N. Nurdiansah, A. Ahmad, and A. Asran, "Sistem Kendali Cerdas Pemberian Pakan Dengan Penerapan Internet of Things," *JIKA (Jurnal Inform.*, vol. 8, no. 3, p. 272, 2024, doi: 10.31000/jika.v8i3.10828.
- [6] M. Syaifudin, I. Permatasari, and A. Prakasa, "Rancang Bangun Pakan Kucing Pintar Untuk Pemantauan Jarak Jauh Menggunakan Aplikasi Mqtt Dashboard," *Teodolita Media Komunkasi Ilm. di Bid. Tek.*, vol. 24, no. 1, pp. 1–11, 2023, doi: 10.53810/jt.v24i1.471.
- [7] H. Suhendi and R. Saputro, "Sistem Monitoring Dan Automatic Feeding Hewan Peliharaan Menggunakan Android Berbasis Internet of Things," *Naratif J. Nas. Ris. Apl. dan Tek. Inform.*, vol. 3, no. 01, pp. 1–8, 2021, doi: 10.53580/naratif.v3i01.112.
- [8] Tresna Umar Syamsuri, Rahma Nur Amalia, Mudjiono, and Aly Imron, "Rancang Bangun Alat Monitoring Daya Listrik di Asrama Berbasis Web Menggunakan ESP32," *Elposys J. Sist. Kelistrikan*, vol. 9, no. 3, pp. 139–145, 2023, doi: 10.33795/elposys.v9i3.648.
- [9] I. Gunawan, H. Ahmadi, and M. R. Said, "Rancang Bangun Sistem Monitoring Dan Pemberi Pakan Otomatis Ayam Anakan Berbasis Internet Of Things (IoT)," *Infotek J. Inform. dan Teknol.*, vol. 4, no. 2, pp. 151–162, 2021, doi: 10.29408/jit.v4i2.3562.
- [10] I. A. Nasyrullah, A. Soetedjo, and K. A. Widodo, "Rancang Bangun Alat Penghitung Bibit Lele Dan Jumlah Berat Menggunakan Arduino Berbasis Iot," vol. 8, pp. 5–6, 2022, [Online]. Available: www.ricelake.com
- [11] Mochamad Andri Prasetyo, Henna Nurdiansari, and Kuntoro Bayu Ajie, "Rancang Bangun Alat Pemilah Ikan Berbasis Internet Of Things (IoT)," *J. Tek. Mesin, Ind. Elektro dan Inform.*, vol. 3, no. 2, pp. 35–52, 2024, doi: 10.55606/jtmei.v3i2.3768.
- [12] Putri Mayang Sari, Novriyenni Novriyenni, and Milli Alfhi Syari, "Rancang Bangun Alat Pendeteksi dan Pengusir Hama Tanaman Kangkung menggunakan Sensor Pir dan Cairan Peptisida Berbasis Internet Of Things (IOT)," *Repeater Publ. Tek. Inform. dan Jar.*, vol. 2, no. 4, pp. 221–230, 2024, doi: 10.62951/repeater.v2i4.248.
- [13] R. Safarudin, D. Setiawan, and D. Saripurna, "Rancang Bangun Sistem Kendali Dan Monitoring Smartclassroom Menggunakan Radio Frequency Identification Berbasis Internet Of Things," *J. Sist. Komput. Triguna Dharma (JURSIK TGD)*, vol. 2, no. 6, pp. 298–306, 2023, doi: 10.53513/jursik.v2i6.9152.
- [14] H. M. Syaban, T. Mufizar, and R. Ruuhwan, "Rancang Bangun Alat Keamanan Rumah Menggunakan Sensor Pir Dengan Notifikasi Telegram Berbasis Iot Dan Catu Daya Plts," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, 2024, doi: 10.23960/jitet.v12i2.4126.
- [15] B. J. VALENDIO, "PERANCANGAN PROTOTYPE PEMBERI PAKAN KUCING GATEWAY SKRIPSI OLEH : BAYU JAKA VALENDIO FAKULTAS TEKNIK UNIVERSITAS MEDAN AREA MEDAN Diajukan sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana di Fakultas Teknik Universitas Medan Area," 2022.
- [16] R. Sudrajat and F. Rofifah, "Rancang Bangun Sistem Kendali Kipas Angin dengan Sensor Suhu dan Sensor Ultrasonik Berbasis Arduino Uno," *Remik*, vol. 7, no. 1, pp. 555–564, 2023, doi: 10.33395/remik.v7i1.12082.
- [17] A. P. Saifullah, "Design and Build Automatic Fish Feeding Device Based on Internet of Things," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, 2024, doi: 10.23960/jitet.v12i2.3991.
- [18] I. Hasan and B. Hariadi, "Rancang Bangun Peningkat Ganti Oli Sepeda Motor Menggunakan Speed Sensor dan ESP 32," *J. Tek. Elektro dan Komputasi*, vol. 6, no. 1, pp. 28–35, 2024, doi: 10.32528/elkom.v6i1.20294.