

Perancangan Sistem Otomasi Pakan Kucing Berbasis Web dan Platform IoT Blynk

Ahmad Farrel Mahardika¹, Muhammad Hadya Alleredha¹, Muhammad Alvin Dzakyanta¹, Jaka Fitra¹, Oki Arifin¹

¹Program Studi Teknologi Rekayasa Perangkat Lunak, Politeknik Negeri Lampung, Indonesia.

Artikel Info

Kata Kunci:

Blynk;
ESP32;
Internet of things;
Sistem Otomasi Pakan;
Web Monitoring.

Keywords:

Blynk;
ESP32;
Internet of things;
Feed Automation System;
Web Monitoring.

Riwayat Artikel:

Submitted: 2 Juni 2025
Accepted: 22 Juni 2025
Published: 24 Juni 2025

Abstrak: Teknologi *Internet of Things* (IoT) semakin banyak diterapkan dalam kehidupan sehari-hari, termasuk dalam bidang perawatan hewan peliharaan. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem pemberian pakan kucing otomatis berbasis mikrokontroler ESP32 yang terintegrasi dengan platform Blynk dan monitoring berbasis web. Pengembangan dilakukan menggunakan metode *Software Development Life Cycle* (SDLC) model *Waterfall* agar setiap tahap dapat berjalan secara sistematis. Sistem ini memungkinkan pemilik hewan untuk menjadwalkan pemberian pakan secara otomatis dan tetap memiliki opsi kontrol manual melalui aplikasi *smartphone*. Komponen utama sistem meliputi ESP32, motor servo, dan LCD 16x2. Blynk digunakan untuk kontrol jarak jauh, sedangkan PHP dan MySQL mencatat aktivitas melalui web. Hasil pengujian menunjukkan bahwa sistem bekerja stabil dalam mode otomatis dan manual, serta mampu mencatat data secara *real-time*. Dalam uji coba selama tujuh hari, sistem berhasil mencatat 21 eksekusi pemberian pakan tanpa kehilangan data. Kendala seperti gangguan Wi-Fi dapat diatasi melalui penyesuaian interval pengiriman data. Sistem ini efektif bagi pengguna yang memiliki mobilitas tinggi dan dapat dikembangkan lebih lanjut dengan fitur notifikasi, sensor berat, serta integrasi *cloud* untuk pemantauan lanjutan.

Abstract: The *Internet of Things* (IoT) technology is applied increasingly in daily life, including in the field of pet care. This study aims to design and implement an automatic cat feeding system based on the ESP32 microcontroller, integrated with the Blynk platform and web-based monitoring. The development process follows the *Software Development Life Cycle* (SDLC) using the *Waterfall* model to ensure that each stage is carried out systematically. The system enables pet owners to schedule automatic feeding and still provides a manual control option via a *smartphone* application. The core components include ESP32, a servo motor, and a 16x2 LCD. Blynk is used for remote control, while PHP and MySQL are utilized to log feeding activity via web. Test results show that the system operates reliably in both automatic and manual modes and capable of recording data in *real-time*. During a seven-day trial, the system successfully logged 21 feeding executions without any data loss. Issues such as Wi-Fi connectivity interruptions were resolved by adjusting the data transmission interval. This system is considered effective for users with high mobility and can be further enhanced by adding a notification feature, weight sensors, and cloud integration for extended monitoring and data analysis.



Corresponding Author:

Jaka Fitra

Email: jakafitra@polinela.ac.id

PENDAHULUAN

Popularitas kucing (*Felis catus*) sebagai hewan peliharaan terus meningkat berkat sifatnya yang adaptif dan menghibur. Seiring dengan popularitas tersebut, pemilik kucing memiliki tanggung jawab penting untuk memastikan kesejahteraannya, terutama dalam hal manajemen pakan. Menjaga jadwal makan yang teratur merupakan hal krusial, namun seringkali sulit dilakukan oleh pemilik dengan mobilitas tinggi. Kegagalan dalam menjaga konsistensi pakan ini dapat memicu berbagai masalah kesehatan, mulai dari stres, gangguan pencernaan, hingga kondisi kronis seperti obesitas dan malnutrisi (Pico-Valencia & Holgado-Terriza, 2025).

Seiring dengan kemajuan teknologi, konsep *Internet of things* (IoT) menawarkan solusi inovatif untuk mengatasi tantangan dalam perawatan hewan peliharaan. Penerapan IoT memungkinkan otomasi dan pemantauan perangkat dari jarak jauh. Penelitian awal di bidang ini telah berhasil mengembangkan perangkat fungsional menggunakan mikrokontroler seperti ESP32 yang dipadukan dengan modul *Real Time Clock* (RTC) untuk penjadwalan pakan yang presisi (Muhammad Rayhan Arisma, Lifwarda, 2024). Beberapa pengembangan juga telah mengintegrasikan berbagai sensor untuk menakar volume pakan secara lebih akurat, membuktikan efektivitas teknologi ini dalam menjaga keteraturan nutrisi hewan (Jacomini, 2024).

Perkembangan selanjutnya membawa pemanfaatan mikrokontroler yang lebih canggih seperti ESP32, yang memiliki keunggulan konektivitas Wi-Fi terintegrasi (Hannani et al., 2021). Hal ini membuka jalan bagi sistem yang tidak hanya bekerja secara otonom, tetapi juga dapat dikontrol dan dipantau secara *real-time*. Berbagai penelitian telah memanfaatkan platform mobile seperti Blynk untuk memungkinkan pengguna memberi pakan dari jarak jauh dan menerima notifikasi status perangkat langsung ke ponsel pintar (Maulana, Muhammad Rizki; Prasetyo, 2022). Di sisi lain, beberapa peneliti berfokus pada aspek pemantauan jangka panjang dengan mengembangkan sistem *monitoring* dan pencatatan data (*data logging*) untuk melacak aktivitas secara historis. Namun, masih terdapat peluang untuk mengintegrasikan kontrol *real-time* via aplikasi dan *monitoring* data historis via web secara lebih sinergi (Valendio, 2022).

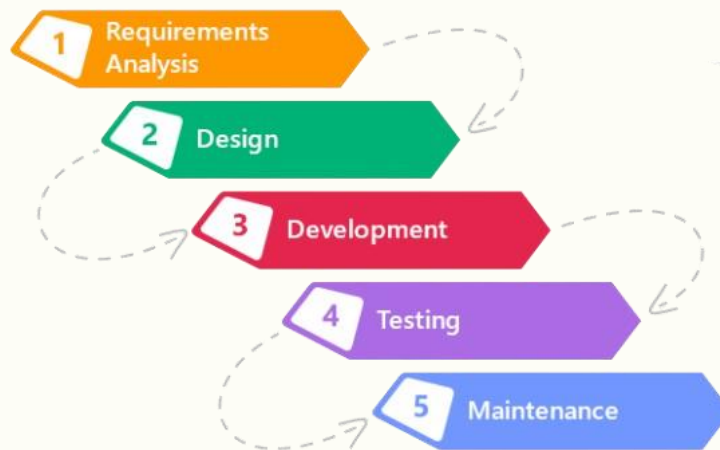
Menjawab peluang tersebut, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem pakan kucing otomatis yang komprehensif dengan mengintegrasikan platform IoT Blynk dan *monitoring* berbasis web. Sistem ini menggunakan mikrokontroler ESP32 sebagai unit pemrosesan utama. Keunggulan dan nilai inovasi dari penelitian ini terletak pada sinergi dua platform antarmuka: Aplikasi Blynk yang berfungsi sebagai alat kontrol langsung (*direct control*) untuk intervensi pakan di luar jadwal dan penerima notifikasi instan serta dashboard web berbasis PHP yang berfungsi sebagai pusat data untuk analisis historis, memungkinkan pemilik memantau tren pola makan kucing (Priatna, 2018). Integrasi ini memberikan solusi yang tidak hanya fleksibel untuk kebutuhan harian, tetapi juga informatif untuk pengambilan keputusan perawatan jangka panjang. Beberapa sistem sebelumnya belum menggabungkan antarmuka kontrol langsung dan pemantauan historis secara terpadu, yang menyebabkan kurangnya fleksibilitas dalam evaluasi pola makan hewan secara menyeluruh.

Dengan demikian, tujuan utama dari penelitian ini adalah untuk menghasilkan sebuah prototipe sistem pakan kucing otomatis yang andal, efisien, dan mudah diakses. Secara spesifik, penelitian ini akan merancang arsitektur perangkat keras dan lunak, mengimplementasikan sistem kontrol

penjadwalan dan manual, serta mengembangkan antarmuka *monitoring* berbasis web yang interaktif dan juga informatif. Diharapkan sistem ini dapat memberikan kontribusi praktis bagi para pemilik hewan peliharaan dalam meningkatkan kualitas perawatan, serta menjadi referensi model pengembangan perangkat dalam domain *smart pet care* yang mengintegrasikan kontrol real-time dengan analisis data historis. Pada akhirnya, Sistem terintegrasi ini memberikan solusi praktis yang memungkinkan pemilik hewan tidak hanya menjadi pengguna, tetapi juga pengelola aktif dalam pemantauan dan perawatan hewan peliharaan mereka (Priatna, 2018).

METODE

Pengembangan sistem dilakukan dengan mengikuti tahapan dalam *Software Development Life Cycle* (SDLC) menggunakan *Waterfall Model* (Sugiarto et al., 2024). Model ini dipilih karena pendekatannya yang sistematis dan berurutan, di mana setiap tahapan harus diselesaikan sebelum melanjutkan ke tahapan berikutnya. Hal ini memastikan bahwa setiap fase pengembangan, mulai dari analisis hingga pemeliharaan, memiliki landasan yang kuat dan terdefinisi dengan baik. Untuk tahapan metode *waterfall* dapat dilihat pada Gambar 1.



Gambar 1. Metode *Waterfall*

Gambar 1 menampilkan *Model Waterfall* yang digunakan dalam pengembangan sistem. Penjelasan dari setiap tahapan dalam model tersebut adalah sebagai berikut:

1. Analisis Kebutuhan

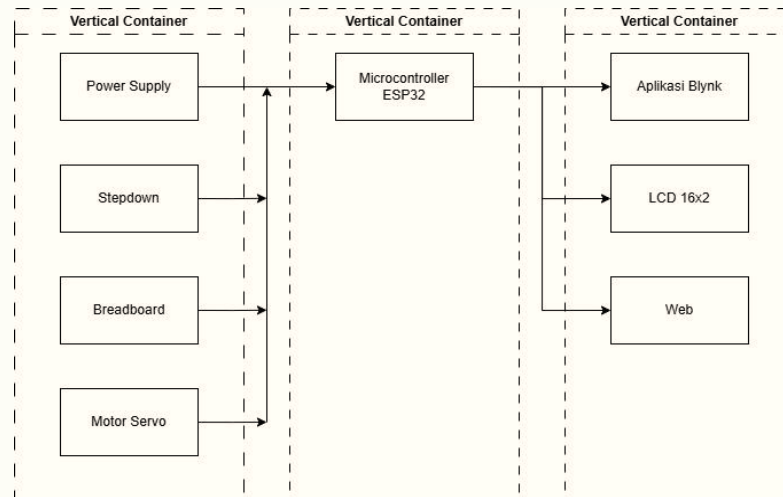
Tahap ini mencakup identifikasi dan pemahaman mendalam mengenai kebutuhan sistem pakan kucing otomatis berbasis web dan IoT. Analisis dilakukan untuk menentukan fitur dan fungsi utama yang harus dimiliki sistem, baik dari sisi perangkat keras maupun perangkat lunak. Proses ini melibatkan pendefinisian kebutuhan fungsional, seperti kemampuan penjadwalan pakan, pemberian pakan manual jarak jauh, dan *monitoring* volume pakan. Selain itu, kebutuhan non-fungsional seperti keandalan sistem, kemudahan penggunaan antarmuka, dan keamanan data juga diidentifikasi untuk memastikan produk akhir dapat memenuhi ekspektasi pengguna (Priatna, 2018).

2. Design

Tahap perancangan bertujuan untuk merancang konsep dan struktur sistem secara terperinci. Proses ini mencakup perancangan arsitektur sistem, skema sensor, perancangan antarmuka berbasis web dan aplikasi Blynk sebagai platform pemantauan dan pengendalian perangkat IoT (Fitra et al., 2024). Pada perancangan perangkat keras, dilakukan pemilihan komponen seperti mikrokontroler ESP32 sebagai unit pemrosesan utama, motor servo sebagai aktuator, dan sensor ultrasonik untuk mendeteksi level pakan (Dewi et al., 2023). Untuk perangkat lunak, dirancang alur kerja program pada

mikrokontroler, struktur *database* untuk *data logging*, serta desain *User Interface* (UI) dan *User Experience* (UX) pada aplikasi Blynk dan *dashboard* web. Tahapan ini dilakukan untuk memastikan bahwa sistem memiliki kerangka kerja yang terstruktur dan efisien untuk mempermudah proses implementasi.

2.1 Blok Diagram Proses Sistem



Gambar 2. Blok Diagram Proses Sistem

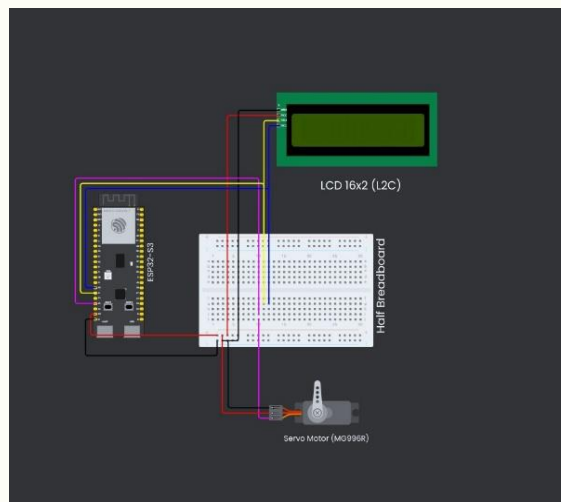
Dari blok diagram proses sistem yang ditunjukkan pada Gambar 2, dapat dijelaskan sebagai berikut:

- 1) *Power Supply* dan *Step-down Converter* berfungsi untuk memberikan tegangan masukan sebesar 5 Volt DC terhadap mikrokontroler ESP32, motor servo, dan komponen lainnya. *Step-down* digunakan untuk menurunkan tegangan dari sumber daya utama agar sesuai dengan kebutuhan perangkat elektronik yang digunakan.
- 2) Breadboard berfungsi sebagai media perakitan sementara yang menghubungkan berbagai komponen seperti ESP32, motor servo, LCD, dan kabel jumper tanpa perlu menyolder. Breadboard sangat membantu dalam proses pengujian dan *prototyping* karena memungkinkan perubahan sambungan secara fleksibel dan cepat.
- 3) Mikrokontroler ESP32 berfungsi sebagai pusat pemrosesan sistem. Komponen ini mengatur proses pemberian pakan baik secara otomatis berdasarkan jadwal yang telah ditentukan, maupun secara manual melalui aplikasi Blynk. Selain itu, ESP32 juga mengelola data yang dikirimkan ke web *server* dan mengatur tampilan status ke LCD.
- 4) Motor Servo digunakan sebagai aktuator yang akan membuka dan menutup wadah makanan secara otomatis berdasarkan perintah dari ESP32, baik melalui jadwal maupun perintah manual dari pengguna.
- 5) LCD 16x2 digunakan untuk menampilkan status sistem secara langsung, seperti informasi "feeding berhasil/gagal", mode aktif (otomatis/manual), serta waktu eksekusi pemberian pakan.
- 6) Aplikasi Blynk berfungsi sebagai kontrol jarak jauh yang memungkinkan pengguna mengaktifkan sistem secara manual melalui *smartphone*.
- 7) Web Monitoring (PHP + Database) digunakan untuk mencatat dan menampilkan log aktivitas pemberian pakan. Data seperti waktu, mode pemberian (otomatis/manual), dan status keberhasilan akan disimpan dalam *database* dan dapat diakses melalui antarmuka web kapan pun dibutuhkan oleh pengguna.

2.2 Perancangan Perangkat Keras

Ada beberapa tahapan dalam proses perancangan perangkat keras pada sistem pakan kucing otomatis ini. Perancangan dilakukan dengan memperhatikan integrasi antara komponen *hardware*,

software, dan mekanik agar sistem dapat bekerja dengan stabil dan responsif. Tahapan perancangan dimulai dari pembuatan *schematic* rangkaian menggunakan software Wokwi, kemudian dilanjutkan dengan perancangan fisik alat, penentuan tata letak pemasangan komponen, pengkabelan dan penyolderan kabel penghubung, hingga pemasangan rangkaian ke dalam sebuah *box panel* yang berisi *power supply*, mikrokontroler ESP32, LCD, dan motor servo (Nurul Huda, M., Syarifuddin Zuhrie, M., Kholis, N., & Wanarti Rusimanto, 2023).



Gambar 3. Schematic Rangkaian Alat

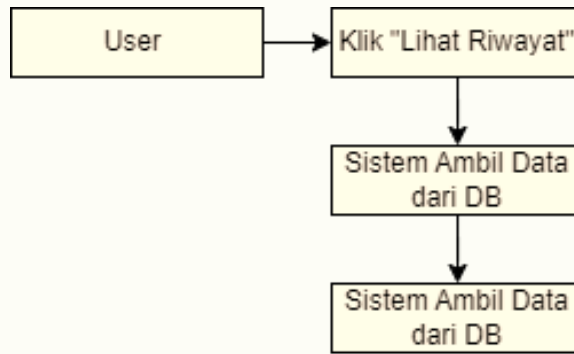
Berdasarkan Gambar 3, skematik rangkaian alat ini bekerja melalui interaksi tiga bagian penting, yaitu perangkat keras (*hardware*), perangkat lunak (*software*), dan mekanik. Pusat kendali dari keseluruhan sistem ini adalah mikrokontroler ESP32 yang berfungsi sebagai otak dari sistem. ESP32 bertugas membaca dan memproses semua perintah sesuai logika pemrograman, kemudian mengontrol komponen keluaran (*output*). Komponen keluaran utama meliputi:

- 1) LCD 16x2, yang digunakan sebagai antarmuka tampilan untuk menyajikan informasi status secara *real-time* kepada pengguna, seperti "Ready", "Feeding", dan "Last Feed".
- 2) Motor Servo, yang berfungsi sebagai aktuator untuk menggerakkan mekanisme pembuka wadah pakan.

Saat sistem diaktifkan, ESP32 akan mengeksekusi program yang tersimpan di memorinya. Sesuai dengan perintah, motor servo akan bergerak ke posisi tertentu untuk membuka wadah pakan selama durasi yang telah ditentukan. Secara bersamaan, LCD 16x2 akan menampilkan status dari proses yang sedang berlangsung. Untuk mempermudah perakitan dan pengujian, seluruh komponen dihubungkan menggunakan *breadboard* yang berfungsi untuk mendistribusikan tegangan dan sebagai penghubung antar komponen. Dalam konfigurasinya, pin VCC dan GND dari ESP32 dihubungkan ke jalur positif dan negatif pada *breadboard* untuk menyuplai daya ke semua komponen.

2.3 Perancangan Aplikasi Pakan Kucing Otomatis Berbasis Web

Aplikasi Pakan Kucing Otomatis berbasis web hanya memiliki satu fungsi utama sistem yang digambarkan melalui use case diagram pada Gambar 4, yang menunjukkan bahwa 'User' (pemilik kucing) memiliki akses tunggal untuk melihat riwayat (*view history*) pemberian pakan. Fungsionalitas ini dirancang secara spesifik untuk memberikan referensi yang jelas bagi pemilik dalam mengevaluasi jadwal dan efektivitas pola makan hewan peliharaannya.

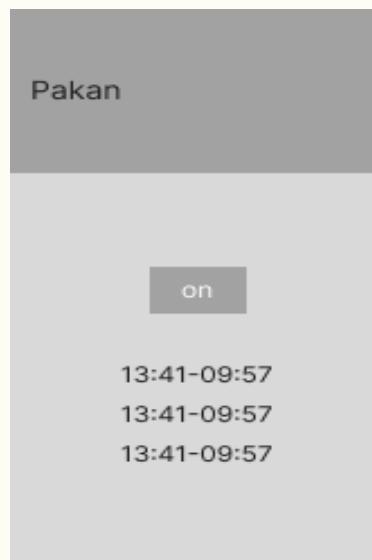


Gambar 4. Diagram Aktivitas Tampilan Riwayat Pakan

Diagram pada Gambar 4 menunjukkan proses alur aktivitas pengguna ketika ingin melihat riwayat pemberian pakan melalui antarmuka web. Pengguna akan menekan tombol "Lihat Riwayat", sistem kemudian mengambil data log dari database, dan menampilkannya dalam bentuk tabel yang informatif. Hal ini dirancang untuk memudahkan pemilik dalam memantau serta mengevaluasi efektivitas jadwal pemberian pakan secara berkala.

2.4 Perancangan Halaman *Monitoring*

Halaman *monitoring* pakan kucing otomatis ini didesain untuk dapat mengatur secara langsung jadwal otomatis pemberian pakannya (Priatna, 2018). Halaman ini juga memiliki satu tombol on off untuk kita mengatur makan kucing secara manual. Rancangan halaman *monitoring* pakan kucing otomatis ditunjukkan pada Gambar 5 di bawah ini.



Gambar 4. Rancangan Halaman *Monitoring*

2.5 Perancangan Halaman *History*

Halaman ini dirancang untuk menampilkan Riwayat aktivitas pemberian pakan otomatis dalam bentuk tabel, sehingga dapat memudahkan pengguna untuk melihat data pemberian pakan. Tabel Utama pada halaman ini akan memvisualisasikan jumlah eksekusi pemberian pakan perhari mau itu secara otomatis ataupun manual serta menampilkan status keberhasilan (Berhasil/Gagal) sesuai dengan data yang tersimpan pada *database* (Austin et al., 2024). Di tunjukan pada Gambar 6 di bawah ini.

Riwayat Pakan Kucing			
ID	Tanggal	Waktu	Status
1	2025-05-18	07:00	Berhasil
2	2025-05-18	12:00	Tidak Berhasil
3	2025-05-18	20:00	Berhasil

Gambar 5. Rancangan Pengelolaan *History*

3. Implementasi

Pada tahap implementasi, semua hasil dari tahap perancangan diwujudkan menjadi produk nyata. Proses ini meliputi dua kegiatan utama: perakitan perangkat keras dan pengkodean perangkat lunak. Komponen-komponen elektronik seperti ESP32, motor servo, dan sensor dirangkai sesuai dengan skema yang telah dirancang. Secara paralel, dilakukan penulisan kode program menggunakan bahasa C++ di Arduino IDE untuk fungsionalitas mikrokontroler. Selain itu, pengembangan sisi *server* dan antarmuka web dilakukan menggunakan bahasa pemrograman PHP dan MySQL untuk *database*. Pengaturan *widget* pada aplikasi Blynk juga dilakukan pada tahap ini untuk menghubungkannya dengan perangkat keras (Syukhron, 2021).

3.1 Konfigurasi Perangkat Keras

Pada tahap ini, dilakukan perakitan fisik seluruh komponen elektronik. Mikrokontroler ESP32, motor servo, LCD 16x2, *step-down converter*, dan *power supply* dirangkai pada *breadboard* dan ditempatkan pada sebuah wadah atau *casing* purwarupa sesuai dengan skema rangkaian yang telah dirancang sebelumnya.

3.2 Pengembangan Perangkat Lunak

Pengembangan perangkat lunak mencakup dua hal:

1. **Firmware ESP32:** Penulisan kode program menggunakan bahasa C++ pada Arduino IDE. Kode ini berisi logika untuk mengontrol motor servo, menampilkan status ke LCD, membaca input dari Blynk, serta mengirim data log ke *server* web.
2. **Backend Web:** Pengembangan skrip PHP yang berfungsi untuk menerima data dari ESP32 dan menyimpannya ke dalam *database* MySQL, serta untuk menampilkan data riwayat pakan pada halaman *history*.

3.3 Pengembangan Aplikasi dan Antarmuka

Tahap ini berfokus pada konfigurasi antarmuka yang akan digunakan oleh pengguna. Proyek baru dibuat pada aplikasi Blynk, kemudian *widget* seperti *Timer* untuk penjadwalan dan *Button* untuk pemberian pakan manual dikonfigurasi dan dihubungkan dengan *virtual pin* pada kode ESP32.

4. Integrasi dan Testing

Setelah tahap implementasi selesai, proses dilanjutkan dengan mengintegrasikan seluruh komponen yang telah dikembangkan mencakup perangkat keras, *firmware* mikrokontroler, *server* web,

dan aplikasi Blynk untuk memastikan semua bagian dapat berkomunikasi dan bekerja sama sebagai satu kesatuan sistem yang utuh.

Setelah integrasi, dilakukan pengujian sistem secara komprehensif. Pendekatan pengujian fungsionalitas secara keseluruhan menggunakan metode *Black Box Testing*, di mana pengujian difokuskan pada kesesuaian input dan output sistem tanpa melihat struktur kode internalnya. Proses pengujian ini dilakukan secara bertingkat, dimulai dari pengujian unit (*unit testing*) untuk memverifikasi setiap fungsi pada kode, hingga pengujian sistem (*system testing*) secara menyeluruh setelah semua komponen terintegrasi.

Pengujian fungsionalitas mencakup verifikasi jadwal pakan, akurasi takaran, responsivitas kontrol manual melalui aplikasi Blynk, konektivitas jaringan WiFi, dan keakuratan data yang ditampilkan pada *dashboard monitoring* serta LCD 16x2. Tujuan dari tahap ini adalah untuk menemukan dan memperbaiki *bug* atau kesalahan sebelum sistem diserahkan kepada pengguna (Rizal & Pramudita, 2023).

Tabel 1. Pengujian

Objek Pengujian	Skenario Pengujian	Hasil yang Diharapkan
Tambah Jadwal Pakan	Input waktu (jam/hari)	Jadwal tersimpan dan muncul pada halaman <i>monitoring</i>
Mode Otomatis	Sistem aktif dan mengeksekusi pakan sesuai waktu	Motor servo aktif, LCD tampilkan status, <i>log</i> tercatat di web
Mode Manual	Menekan tombol " <i>Feed Now</i> " dari Blynk	Servo langsung aktif & status muncul di LCD & web
<i>Monitoring Status</i>	Menampilkan status pakan terakhir di web & LCD	Informasi status sesuai eksekusi terakhir (berhasil/gagal)
Koneksi Blynk	Sistem terhubung ke jaringan & Blynk	Status alat tampil, kontrol manual bisa dilakukan
<i>Web Log History</i>	Menampilkan daftar eksekusi pakan harian	<i>Log</i> tanggal, waktu, status ditampilkan

5. Verifikasi

Tahap verifikasi bertujuan untuk memastikan bahwa sistem yang dibangun telah sesuai dengan semua kebutuhan yang didefinisikan pada tahap analisis. Proses ini sering kali melibatkan *User Acceptance Test* (UAT), di mana calon pengguna mencoba sistem dalam skenario penggunaan yang disimulasikan. Masukan dan umpan balik dari pengguna dikumpulkan untuk menilai apakah fungsionalitas, kemudahan penggunaan, dan kinerja sistem secara keseluruhan telah memenuhi harapan. Verifikasi ini menjadi penentu apakah sistem siap untuk dioperasikan atau memerlukan revisi lebih lanjut.

6. Operasi dan Pemeliharaan

Ini adalah tahap akhir dari *Waterfall Model*, di mana sistem yang telah lolos verifikasi diserahkan kepada pengguna untuk dioperasikan dalam lingkungan yang sebenarnya. Tahap pemeliharaan dimulai setelah sistem berjalan, yang mencakup berbagai kegiatan seperti perbaikan *bug* yang baru ditemukan, pembaruan sistem keamanan, serta penambahan fitur minor berdasarkan permintaan pengguna di masa depan. Pemeliharaan berkelanjutan sangat penting untuk memastikan sistem tetap andal dan relevan seiring berjalannya waktu.

HASIL DAN PEMBAHASAN

Pada bagian ini, diuraikan hasil dari implementasi sistem yang telah dirancang. Pembahasan mencakup hasil implementasi antarmuka pengguna pada aplikasi Blynk dan web, struktur *database*, wujud fisik perangkat keras, serta penjelasan potongan kode kunci yang menjalankan fungsionalitas sistem

4.1 Hasil Implementasi antarmuka Aplikasi

Hasil implementasi pada sistem pakan kucing otomatis berbasis IoT menunjukkan bahwa halaman *monitoring* berhasil menampilkan status sistem secara *real-time* melalui aplikasi Blynk (Fadhli Adam et al., 2024). Tampilan halaman *monitoring* dirancang secara sederhana namun informatif, agar pengguna dapat dengan mudah memantau dan mengontrol aktivitas pemberian pakan kapan saja dan di mana saja. Antarmuka aplikasi didesain ringkas dan intuitif, sehingga pengguna dapat langsung memantau serta mengendalikan aktivitas pemberian pakan tanpa perlu proses adaptasi yang kompleks. Hasil akhir dari antarmuka pada aplikasi Blynk ditunjukkan pada Gambar 7.



Gambar 6. Halaman *Monitoring* pada Aplikasi *Blynk*

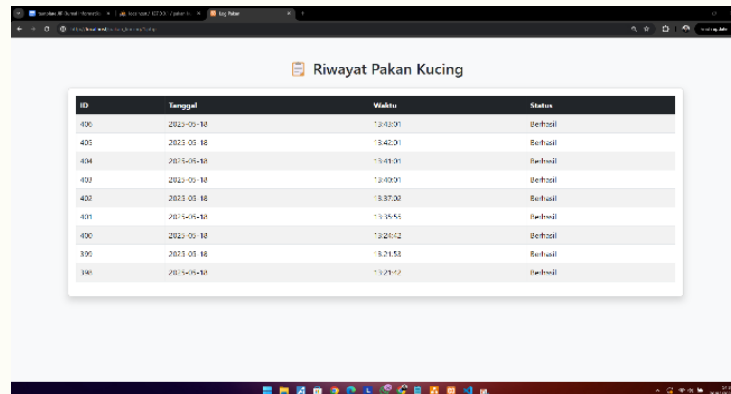
Fungsionalitas pada antarmuka ini diatur melalui *firmware* ESP32. Potongan Kode 1 menampilkan logika utama untuk menangani input dari pengguna, yaitu tombol pakan manual (V0) dan pengaturan jadwal (V1).

```
BLYNK_WRITE(V0) {  
  int tombol = param.asInt();  
  if (tombol == 1 && !feedingInProgress) {  
    beriPakan(); // Panggil fungsi utama untuk memberi pakan  
    Blynk.virtualWrite(V0, 0); // Reset tombol di aplikasi  
  }  
}  
  
BLYNK_WRITE(V1) {  
  TimeInputParam t(param);  
  if (t.hasStartTime()) {  
    // Simpan waktu jadwal ke variabel  
    startHour1 = t.getStartHour();  
    startMinute1 = t.getStartMinute();  
    done1 = false;  
  }  
}
```

Gambar 8. Potongan Kode 1 Logika Kontrol dari Aplikasi Blynk

Selain antarmuka kontrol, sistem ini juga menyediakan halaman *history* untuk menampilkan riwayat aktivitas pemberian pakan otomatis dalam bentuk tabel. Tabel ini memvisualisasikan data

eksekusi pakan harian, baik secara otomatis maupun manual, beserta status keberhasilannya sesuai dengan data yang tersimpan pada Gambar 8.

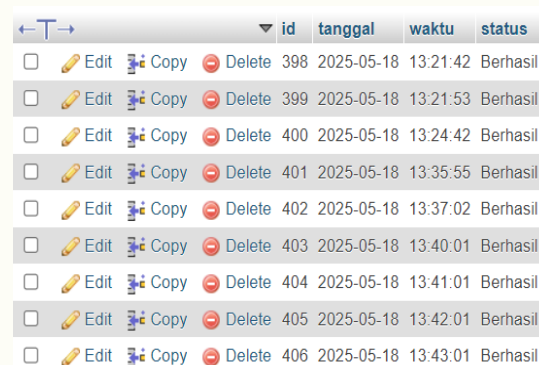


ID	Tanggal	Waktu	Status
406	2025-05-18	13:42:01	Berhasil
405	2025-05-18	13:42:01	Berhasil
404	2025-05-18	13:41:01	Berhasil
403	2025-05-18	13:40:01	Berhasil
402	2025-05-18	13:37:02	Berhasil
401	2025-05-18	13:35:55	Berhasil
400	2025-05-18	13:24:42	Berhasil
399	2025-05-18	13:21:53	Berhasil
398	2025-05-18	13:21:42	Berhasil

Gambar 7. Halaman *History*

4.2 Hasil Implementasi Backend dan Database

Sistem *backend* menyimpan log aktivitas yang dilakukan setelah pemberian makan secara otomatis ke dalam sebuah *database*. Struktur dari tabel pakan_kucing dapat dilihat pada Gambar 9.



	id	tanggal	waktu	status
☐ Edit ☐ Copy ☐ Delete	398	2025-05-18	13:21:42	Berhasil
☐ Edit ☐ Copy ☐ Delete	399	2025-05-18	13:21:53	Berhasil
☐ Edit ☐ Copy ☐ Delete	400	2025-05-18	13:24:42	Berhasil
☐ Edit ☐ Copy ☐ Delete	401	2025-05-18	13:35:55	Berhasil
☐ Edit ☐ Copy ☐ Delete	402	2025-05-18	13:37:02	Berhasil
☐ Edit ☐ Copy ☐ Delete	403	2025-05-18	13:40:01	Berhasil
☐ Edit ☐ Copy ☐ Delete	404	2025-05-18	13:41:01	Berhasil
☐ Edit ☐ Copy ☐ Delete	405	2025-05-18	13:42:01	Berhasil
☐ Edit ☐ Copy ☐ Delete	406	2025-05-18	13:43:01	Berhasil

Gambar 8. Struktur Tabel *Database* pakan_kucing

Struktur tabel log_pakan dibuat menggunakan perintah SQL, dan data dari ESP32 diterima oleh skrip PHP untuk disimpan, seperti ditunjukkan pada Gambar 10 dan 11.

```
-- Struktur Tabel SQL untuk log_pakan
CREATE TABLE `log_pakan` (
  `id` int(11) NOT NULL,
  `tanggal` date NOT NULL,
  `waktu` time NOT NULL,
  `status` varchar(50) DEFAULT NULL
);
```

Gambar 10. Struktur Database

```
<?php
// Logika utama pada file PHP untuk menerima data dari ESP32
if (isset($_GET['tanggal']) && isset($_GET['jam'])) {
    $tanggal = $_GET['tanggal'];
    $jam = $_GET['jam'];
    $status = $_GET['status'] ?? 'Berhasil';

    // Siapkan dan eksekusi perintah SQL untuk menyimpan data
    $stmt = $conn->prepare("INSERT INTO log_pakan (tanggal, waktu, status) VALUES
    (?, ?, ?)");
    $stmt->bind_param("sss", $tanggal, $jam, $status);
    $stmt->execute();
    $stmt->close();
}
?>
```

Gambar 11. Potongan Kode 2 Logika *Backend* PHP

4.3 Hasil Implementasi Backend dan Database

Gambar dibawah ini menunjukkan wujud fisik dari perangkat sistem pakan kucing otomatis berbasis IoT yang telah dikembangkan. Perangkat ini menggunakan mikrokontroler ESP32 sebagai pusat kendali utama, yang terhubung dengan motor servo untuk membuka dan menutup wadah pakan, serta LCD 16x2 untuk menampilkan informasi status alat. Seluruh komponen dirangkai pada sebuah *breadboard* untuk mempermudah proses perakitan dan pengujian. Tampilan dari belakang memperlihatkan penempatan ESP32, motor servo, dan LCD yang diatur untuk memudahkan pemantauan oleh pengguna secara langsung.



Gambar 12. Wujud Fisik Sistem Pakan Kucing Otomatis dari Tiga Sudut Pandang (Belakang, Depan, dan Samping)

KESIMPULAN

Berdasarkan hasil perancangan dan implementasi yang telah dilakukan, dapat disimpulkan bahwa sistem otomatisasi pakan kucing berbasis mikrokontroler ESP32, aplikasi Blynk, dan monitoring berbasis web PHP telah berhasil direalisasikan dan mampu berfungsi secara optimal. Sistem ini mampu menjalankan pemberian pakan secara otomatis sesuai dengan jadwal yang telah ditentukan, serta memungkinkan pengguna melakukan pemberian pakan secara manual melalui aplikasi *smartphone*.

Fitur-fitur utama yang diimplementasikan, seperti penjadwalan pakan, pengendalian motor servo, tampilan status melalui LCD 16x2, serta pencatatan aktivitas ke dalam sistem monitoring berbasis web, telah diuji dan menunjukkan performa yang stabil dan responsif. Kemampuan sistem dalam memberikan informasi secara real-time serta akses kendali jarak jauh sangat membantu pengguna, khususnya mereka yang memiliki mobilitas tinggi, untuk tetap memantau dan mengatur pola makan hewan peliharaannya dari mana saja. Tampilan halaman monitoring dan halaman riwayat (*history*) yang berhasil ditampilkan melalui platform web dan aplikasi Blynk memberikan nilai tambah dalam bentuk informasi historis yang akurat dan mudah dipahami. Hal ini memungkinkan pengguna untuk melakukan evaluasi terhadap efektivitas pemberian pakan secara berkala dan lebih terstruktur.

Sistem ini masih memiliki ruang untuk pengembangan lebih lanjut. Beberapa saran pengembangan mencakup penambahan sensor berat untuk mendeteksi sisa pakan secara otomatis, integrasi notifikasi ke *smartphone* apabila terjadi kegagalan distribusi, serta peningkatan desain antarmuka web agar lebih dinamis dan interaktif. Selain itu, untuk mendukung skalabilitas dan ketersediaan sistem dalam jangka panjang, integrasi dengan *platform cloud* direkomendasikan sebagai solusi penyimpanan dan pemantauan yang lebih luas. Meskipun telah berfungsi dengan baik, sistem ini masih menghadapi beberapa keterbatasan, seperti kestabilan koneksi Wi-Fi dalam kondisi tertentu serta kapasitas penyimpanan log yang masih bergantung pada memori lokal. Oleh karena itu, pengembangan selanjutnya dapat difokuskan pada optimasi efisiensi transmisi data dan perluasan penyimpanan berbasis *cloud* guna meningkatkan keandalan dan fleksibilitas sistem secara keseluruhan.

DAFTAR PUSTAKA

- Austin, C., Mulyadi, M., & Octaviani, S. (2024). Implementasi IoT dengan ESP 32 Untuk Pemantauan Kondisi Suhu Secara Jarak Jauh Menggunakan MQTT Pada AWS. *Jurnal Elektro*, 15(2), 46–55. <https://doi.org/10.25170/jurnalelektro.v15i2.5141>
- Dewi, C., 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), 177–199. <https://doi.org/10.23887/karmapati.v12i3.70010>
- Fadhli Adam, M., Purwantoro, P., & Yusup, D. (2024). Simulasi Sistem Data Logging Berbasis Internet of things Untuk Melacak Aktivitas Kendaraan Dengan Aplikasi Android. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(5), 10646–10652. <https://doi.org/10.36040/jati.v8i5.11136>
- Fitra, J., Rofianto, D., & Amaliah, K. (2024). Implementasi Sistem Telemetry Monitoring Gas serta Suhu dan Kelembaban Pada Kandang Ayam Closed House Berbasis IoT. *JoMMiT : Jurnal Multi Media Dan IT*, 8(1), 001–006. <https://doi.org/10.46961/jommit.v8i1.1266>
- Hannani, N., Rasyidi, M., & Zin, R. M. (2021). Cat's Monitoring and Feeding Systems via Internet of things (IoT). *Evolution in Electrical and Electronic Engineering*, 2(2), 93–101. <http://publisher.uthm.edu.my/periodicals/index.php/eeee>
- Jacomini, R. de S. (2024). Sistema embarcado aplicado à automação na alimentação de pets: ML e IoT. *UNESP Repository.*, 15(01), 133–158. <https://repositorio.unesp.br/items/49960535-35c7-48e3-844e-4730a6a2bd8b>
- Maulana, Muhammad Rizki; Prasetyo, E. (2022). Perencanaan Sistem Pemberi Pakan Otomatis Berbasis Internet of things. 1–14. https://eprints.ums.ac.id/116484/1/Naskah_Publikasi_D400180164.pdf
- Muhammad Rayhan Arisma, Lifwarda, A. A. D. (2024). Design of a Microcontroller-Based Automatic Cat Feeding and Monitoring Device. 1(1), 1–7. <https://jieta.org/index.php/jieta/article/view/1>
- Nurul Huda, M., Syarifuddin Zuhrie, M., Kholis, N., & Wanarti Rusimamto, P. (2023). Rancang Bangun Kit Mikrokontroler Esp32 Berbasis Iot Sebagai Media Pembelajaran Pada Elemen Pemrograman Dan Aplikasi Mikrokontroler Di Smkn 7 Surabaya.

<https://ejournal.unesa.ac.id/index.php/jurnal-pendidikan-teknik-elektro/article/view/67359>

- Pico-Valencia, P., & Holgado-Terriza, J. A. (2025). The Internet of things Empowering the Internet of Pets—An Outlook from the Academic and Scientific Experience. *Applied Sciences (Switzerland)*, 15(4), 1–39. <https://doi.org/10.3390/app15041722>
- Priatna, P. (2018). Analisis Tingkat Keberhasilan Sistem Monitoring Pakan Kucing Otomatis Menggunakan Metode Prototyping. [https://repository.uinjkt.ac.id/dspace/bitstream/123456789/54950/1/PERDANA PRIATNA-FST.pdf](https://repository.uinjkt.ac.id/dspace/bitstream/123456789/54950/1/PERDANA_PRIATNA-FST.pdf)
- Rizal, M., & Pramudita, R. (2023). Perancangan Alat Pemberi Pakan Kucing Otomatis Berbasis IoT Pada Sadewa Pet Care Bekasi. *Jurnal Mahasiswa Bina Insani*, 7(2), 117–128. <https://ejournal-binainsani.ac.id/index.php/JMBI/article/view/2604/1672>
- Sugiarto, M. A. R., Muhtarom, M., & Asri, A. A. K. (2024). Implementasi Sistem Pemberian Pakan Ikan Hias Otomatis Menggunakan Esp32 Berbasis IoT (Internet of things). *Jurnal Indonesia : Manajemen Informatika Dan Komunikasi*, 5(3), 2781–2791. <https://doi.org/10.35870/jimik.v5i3.1008>
- Syukhron, I. (2021). Penggunaan Aplikasi Blynk untuk Sistem Monitoring dan Kontrol Jarak Jauh pada Sistem Kompos Pintar berbasis IoT. *Electrician*, 15(1), 1–11. <https://doi.org/10.23960/elc.v15n1.2158>
- Valendio, B. J. (2022). Perancangan Prototype Pemberi Pakan Kucing Otomatis Berdasarkan Umur Berbasis Sms Gateway [Universitas Medan Area]. [https://repositori.uma.ac.id/bitstream/123456789/18985/1/178160107 - Bayu Jaka Valendio - Fulltext.pdf](https://repositori.uma.ac.id/bitstream/123456789/18985/1/178160107-Bayu%20Jaka%20Valendio-Fulltext.pdf)