

OTOMATISASI SISTEM PAKAN KUCING BERBASIS IOT UNTUK PENGATURAN PORSI PAKAN DAN MONITORING SECARA *REAL-TIME*

1st Yuwan Cornelius

Direktorat Kampus Purwokerto

Universitas Telkom Purwokerto

Purwokerto, Indonesia

yuwancornelius@student.telkomuniversity.ac.id

2nd Caca Surya

Direktorat Kampus Purwokerto

Universitas Telkom Purwokerto

Purwokerto, Indonesia

cacasurya@telkomuniversity.ac.id

3th Refia Nada Fitri

Direktorat Kampus Purwokerto

Universitas Telkom Purwokerto

Purwokerto, Indonesia

refianada@student.telkomuniversity.ac.id

4th Irsad Farhan Nulhakim

Direktorat Kampus Purwokerto

Universitas Telkom Purwokerto

Purwokerto, Indonesia

irsadfarhannulhakim@telkomuniversity.ac.id

5th Yulian Zetta Maulana, S.T., M.T

Direktorat Kampus Purwokerto

Universitas Telkom Purwokerto

Purwokerto, Indonesia

yulianm@telkomuniversity.ac.id

6th Zein Hanni Pradana, S.T., M.T

Direktorat Kampus Purwokerto

Universitas Telkom Purwokerto

Purwokerto, Indonesia

zeindana@ittelkom-pwt.ac.id

Abstrak — Perkembangan teknologi *Internet of Things* (IoT) dan *Artificial Intelligence* (AI) mendorong terciptanya sistem otomatisasi cerdas dan responsif. Penelitian ini mengusulkan sistem pakan kucing otomatis berbasis IoT dan AI yang dirancang untuk mengatasi permasalahan keterbatasan waktu pemilik hewan dalam memberi pakan secara teratur. Sistem menggunakan YOLOv11 untuk mendeteksi kucing secara *real-time* melalui *webcam*, sehingga hanya kucing teridentifikasi yang dapat menerima pakan. Model AI dilatih menggunakan platform *Roboflow* dan dikonversi ke format *TensorFlow* agar kompatibel dengan perangkat *edge*. *Raspberry Pi 4* digunakan sebagai pengendali utama untuk sensor dan aktuator, serta terhubung ke server lokal melalui integrasi *Cloudflare Tunnel* guna menyediakan akses *website* secara aman. Sensor *loadcell* dan ultrasonik digunakan untuk mendeteksi berat pakan dan ketersediaan pakan, sedangkan motor *stepper* mengatur proses distribusi pakan baik secara otomatis maupun manual. Sistem dapat dikendalikan dan dipantau dari jarak jauh melalui antarmuka *web* dan notifikasi Telegram. Hasil pengujian menunjukkan akurasi *loadcell* sebesar 97,16%, sensor ultrasonik 98,28%, akurasi pemberian pakan otomatis 91,92%, manual 89,13%, dan akurasi deteksi kucing 95%. *Website* memiliki waktu *connect time* rata-rata 109,2 ms, notifikasi Telegram 3,4 ms, dan *latency* akses ke server lokal sebesar 373,6 ms. Sistem ini menunjukkan performa yang andal dalam pengelolaan pakan kucing secara selektif dan efisien melalui integrasi teknologi IoT dan AI.

Kata kunci — AI, *Cloudflare Tunnel*, IoT, *Raspberry Pi 4*, YOLOv11

I. PENDAHULUAN

Pemberian makan yang tepat dan teratur merupakan hal penting dalam menjaga kesehatan hewan peliharaan, termasuk kucing [1]. Kucing yang tidak diberi makan secara teratur dapat mengalami masalah kesehatan seperti obesitas atau gangguan pencernaan yang serius [2]. Namun,

pemberian makan yang konsisten menjadi tantangan bagi pemilik kucing yang memiliki jadwal yang padat dan kesibukan tinggi, sehingga mereka sulit memberi perhatian penuh terhadap pola makan hewan peliharaannya [3]. Permasalahan ini semakin relevan mengingat Indonesia merupakan negara dengan populasi pemilik kucing yang cukup besar, sekitar 47% masyarakat Indonesia memelihara kucing, dilansir dari artikel kompas.id tahun 2023. Hal tersebut menjadikannya hewan peliharaan yang paling banyak dipelihara di Indonesia. Oleh karena itu, diperlukan sebuah solusi yang dapat membantu pemilik kucing untuk memberikan pakan secara otomatis, sehingga kucing tetap mendapatkan asupan gizi yang teratur tanpa kehadiran fisik pemilik [4].

Seiring dengan perkembangan teknologi, konsep otomatisasi pemberian pakan melalui *Internet of Things* (IoT) menjadi solusi yang potensial untuk mengatasi masalah ini. Sistem pakan kucing otomatis berbasis IoT memungkinkan pemilik kucing untuk memonitor dan mengendalikan pemberian pakan secara *real-time* melalui internet [5]. Selain itu, sistem ini dapat dilengkapi dengan fitur-fitur canggih seperti notifikasi kepada pemilik kucing, sensor pemantauan ketersediaan pakan, pemrosesan data menggunakan *cloud computing* dan penggunaan teknologi *Artificial Intelligence* (AI) untuk mendeteksi kucing yang sudah teridentifikasi, sehingga hanya kucing yang terdaftar yang dapat mengakses pakan yang diberikan [6] [7].

Penelitian terdahulu menunjukkan bahwa berbagai teknologi telah digunakan untuk mengembangkan sistem pemberian pakan otomatis berbasis IoT. Misalnya, penelitian oleh A. A. Mulky, dkk [3], yang menggunakan teknologi *You Only Look Once* (YOLO) diintegrasikan dengan *webcam* agar mampu mendeteksi kucing. Sistem tersebut berhasil mendeteksi wajah kucing yang mendekat pada perangkat, kemudian mengaktifkan motor *servo* untuk membuka *port* makanan, memungkinkan pengaturan makan kucing secara otomatis. Namun, meskipun berbagai penelitian telah diusulkan, sistem tersebut masih memiliki keterbatasan,

seperti kurangnya sensor yang diintegrasikan untuk memantau sisa pakan, platform untuk melakukan monitoring maupun *controlling* sistem yang belum ada, serta tidak ada kombinasi notifikasi sebagai *alert* pengguna.

II. KAJIAN TEORI

A. Sistem Pakan Kucing Otomatis

Sistem pemberian pakan kucing otomatis merupakan suatu perangkat yang dirancang untuk mengatur proses pemberian pakan kepada kucing secara terjadwal dan teratur tanpa memerlukan intervensi langsung dari pemilik. Sistem ini bekerja dengan membagi porsi makanan yang telah ditentukan sebelumnya dan mendistribusikannya pada waktu tertentu sesuai dengan jadwal yang telah diprogram. Tujuan utama sistem ini adalah memastikan kucing memperoleh asupan makanan secara konsisten dan sesuai kebutuhan, sekaligus mengurangi ketergantungan pada kehadiran fisik pemilik dalam aktivitas pemberian pakan harian [8].

B. Internet of Things (IoT)

Internet of Things (IoT) merupakan sebuah sistem tertanam (*embedded system*) yang dikembangkan untuk memperluas cakupan konektivitas internet secara berkelanjutan. Teknologi ini memungkinkan berbagai objek fisik, seperti bahan pangan, perangkat elektronik, dan peralatan lainnya yang dilengkapi dengan sensor, untuk saling terhubung melalui jaringan. Melalui konektivitas ini, objek-objek tersebut mampu melakukan pertukaran data, pemantauan jarak jauh (*remote monitoring*), serta pengendalian secara *real-time* [9].

C. Artificial Intelligence (AI)

Artificial Intelligence (AI) atau kecerdasan buatan merupakan suatu sistem komputasi yang dirancang untuk menjalankan tugas-tugas yang umumnya memerlukan kemampuan kognitif manusia. Teknologi ini mampu mengambil keputusan melalui analisis data yang tersedia di dalam sistem. Mekanisme kerja AI melibatkan proses pembelajaran (*learning*), penalaran (*reasoning*), serta koreksi diri (*self-correction*), yang secara fungsional menyerupai cara manusia dalam mengevaluasi informasi sebelum menghasilkan suatu keputusan [10].

D. You Only Look Once (YOLO)

You Only Look Once (YOLO) adalah suatu teknologi untuk mendeteksi objek yang berlandaskan *deep learning*. YOLO sendiri merupakan suatu teori yang diturunkan dari *computer vision* (CV) dengan fokus sistem mampu memahami, mengenali, dan melihat gambar maupun video. Selain itu, diturunkan pula berdasarkan *convolutional neural networks* (CNN) yang merupakan algoritma untuk dapat mengklasifikasikan suatu gambar [11].

E. Cloud Computing

Cloud computing merupakan teknologi untuk penggunaannya mampu melakukan komputasi, seperti akses server, aplikasi, ataupun penyimpanan melalui internet secara fleksibel. Model ini memungkinkan penyediaan layanan secara cepat dan sesuai permintaan tanpa mengharuskan pengguna memiliki atau mengelola infrastruktur fisik secara langsung. Layanan *cloud* mencakup berbagai fungsi seperti penyimpanan data, pemrosesan informasi, dan kecerdasan komputasional [12].

F. Cloudflare Tunnel

Cloudflare merupakan perusahaan penyedia layanan di bidang keamanan, kinerja, dan privasi digital yang bertujuan

untuk melindungi serta mengoptimalkan situs *web* dan aplikasi berbasis *web*. Layanan yang ditawarkan mencakup perlindungan terhadap ancaman *siber*, peningkatan kecepatan akses, dan solusi atas berbagai permasalahan keamanan di internet. Salah satu layanan unggulannya adalah *Cloudflare Tunnel*, yaitu sebuah metode yang memungkinkan koneksi aman dari perangkat atau server lokal ke internet tanpa perlu membuka *port* jaringan secara langsung, sehingga meningkatkan keamanan sekaligus memudahkan akses dari jarak jauh [13].

G. Website

Website merupakan kumpulan halaman yang menyajikan informasi diakses melalui jaringan internet, memungkinkan pengguna mengakses dari berbagai lokasi selama terhubung ke internet. Pengembangan proses sistem maupun sistem informasi, merupakan serangkaian kegiatan yang bertujuan merancang dan membangun suatu sistem informasi yang berlandaskan komputer untuk membantu penyelesaian permasalahan organisasi atau memanfaatkan peluang yang ada [14].

H. Telegram

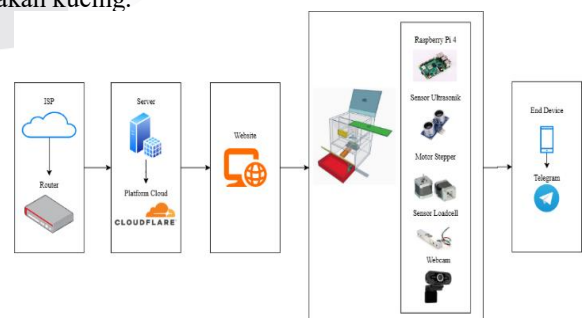
Telegram merupakan aplikasi pesan instan yang berbasis *cloud* dilengkapi fitur enkripsi. Beberapa fitur unggulannya meliputi enkripsi *end-to-end*, kemampuan menghapus pesan secara otomatis, serta penggunaan infrastruktur pusat data yang tersebar di berbagai lokasi. Telegram menjadi pilihan populer karena mendukung berbagai platform, performan ringan, serta menyediakan *Application Programming Interface* (API) yang terdokumentasi dengan baik sehingga memudahkan proses pengembangan aplikasi [15].

III. METODE

Metode yang digunakan dalam implementasi penelitian analisis komunikasi data pada sistem pakan kucing otomatis tersebut adalah melakukan pencarian referensi, pembuatan konsep sistem, perancangan sistem, dan pengujian untuk mendapatkan hasil data agar mampu dianalisis. Rangkaian tersebut menggunakan metode kuantitatif.

A. Desain Sistem Pakan Kucing Otomatis

Desain sistem yang akan dirancang dibuat berdasarkan aspek kebutuhan yang perlu dipenuhi, hal tersebut mencakup aspek ekonomi untuk mengurangi *cost* agar tidak perlu *hire* orang atau menitipkan ke *pet shop*, aspek teknologi untuk memudahkan pengelolaan dan penggunaan sistem, aspek fungsionalitas untuk menawarkan kenyamanan bagi pengguna, dan aspek kesehatan hewan untuk mengatur pola makan kucing.



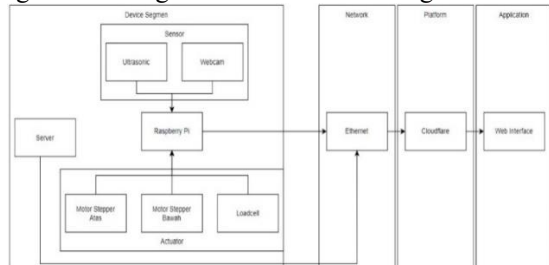
Gambar 1. Desain Sistem

Pada Gambar 1 menunjukkan desain sistem yang terdiri dari jaringan internet melalui *Internet Service Provider* (ISP) dan *router* Mikrotik, server *cloud* menggunakan *Cloudflare* sebagai penghubung antara server dan *client*, serta *website*

yang terintegrasi dengan sistem pakan kucing otomatis. Sistem ini dikendalikan oleh *Raspberry Pi 4* yang terhubung dengan sensor ultrasonik, sensor *loadcell*, dan kamera *webcam*. Notifikasi sistem dikirimkan ke pengguna melalui aplikasi Telegram di *handphone*.

B. Blok Diagram Sistem Pakan Kucing Otomatis

Untuk memberikan gambaran menyeluruh mengenai alur kerja dan komponen utama yang terlibat dalam sistem, pakan kucing otomatis digunakan sebuah blok diagram.

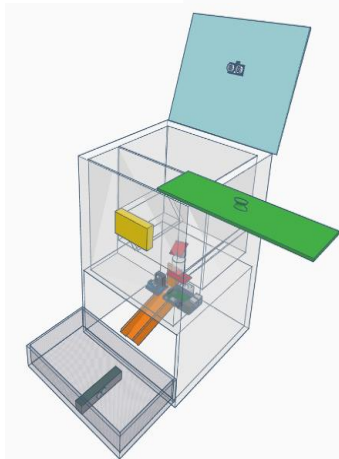


Gambar 2. Blok Diagram Sistem Pakan Kucing Otomatis

Pada Gambar 2. menunjukkan blok diagram sistem, terlihat bahwasannya sistem terintegrasi berdasarkan beberapa blok, yaitu *device system* yang terdiri dari komponen IoT sistem pakan kucing otomatis, *network* yang terdiri dari *ethernet* untuk komunikasi ke internet, platform menggunakan *cloudflare tunnel* untuk komunikasi data antar komponen sistem, dan *application* untuk mengakses *website* sistem.

C. Desain Hardware Sistem

Bagian ini menjelaskan secara ringkas desain perangkat keras pada sistem pemberi pakan kucing otomatis, yang meliputi komponen utama untuk proses deteksi, pengukuran, dan distribusi pakan guna mendukung kinerja sistem secara menyeluruh.



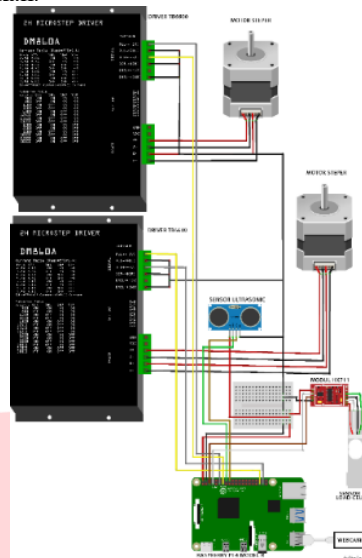
Gambar 3. Desain Hardware Sistem Pakan Kucing Otomatis

Gambar 3. menunjukkan desain prototipe sistem pakan kucing otomatis yang terdiri dari *Raspberry Pi 4* sebagai pengendali utama yang terhubung ke server, *webcam* untuk mendeteksi dan memantau kucing, sensor ultrasonik untuk mendeteksi sisa pakan dalam tabung utama, *loadcell* memantau sisa pada wadah akhir, serta dua motor *stepper* untuk menyisir pakan dalam tabung agar tidak menyumbat lubang keluaran, dan untuk mengatur distribusi pakan secara presisi melalui mekanisme *slider* atau katup.

D. Desain Sistem Perangkat IoT

Bagian ini membahas perancangan sistem perangkat IoT yang digunakan dalam sistem pemberi pakan kucing

otomatis, untuk mengetahui integrasi setiap komponen IoT yang digunakan.

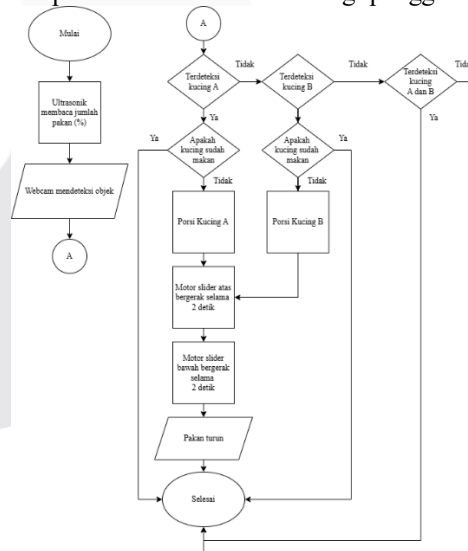


Gambar 4. Desain Sistem Perangkat IoT

Pada Gambar 4. Merupakan desain komponen IoT yang digunakan oleh sistem, integrasi komponen perlu diperhatikan agar sistem mampu berjalan sesuai dengan spesifikasi yang direncanakan. Hal tersebut, antara lain mikrokontroler *Raspberry Pi 4* untuk mengontrol sensor dan aktuator, ultrasonik dan *loadcell* untuk cek gram pakan, motor *stepper* untuk menggerakkan distribusi pakan.

E. Cara Kerja Sistem Pakan Kucing Otomatis

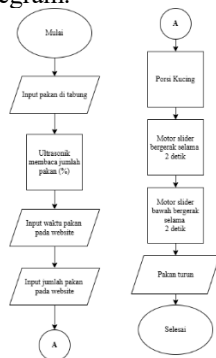
Cara kerja sistem pakan kucing otomatis ini terbagi menjadi dua mode, yaitu otomatis dan manual. Kedua metode ini dirancang untuk memastikan pemberian pakan yang tepat waktu dan sesuai kebutuhan, dengan tetap memberikan kemudahan pemantauan dan kontrol bagi pengguna.



Gambar 5. Cara Kerja Sistem Mode Otomatis

Gambar 5. memperlihatkan alur kerja sistem dengan mode otomatis, pengguna hanya perlu mengaktifkan pengontrol AI melalui *website*. Sistem kemudian membaca jadwal dan porsi pakan *default* (50 gram per makan, dua kali sehari). Kamera *webcam* mendeteksi keberadaan kucing menggunakan AI, lalu *Raspberry Pi 4* mengaktifkan motor *stepper* untuk mengeluarkan pakan. Sensor ultrasonik memantau sisa pakan dalam tabung utama, sementara *loadcell* memantau berat pakan di wadah akhir. Setelah dua

kali makan dalam sehari, sistem otomatis berhenti dan aktif kembali keesokan harinya. Notifikasi pemberian pakan dikirim melalui Telegram.

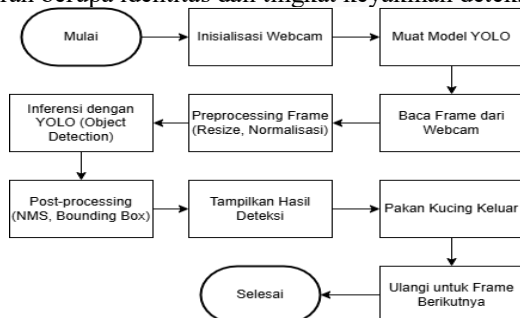


Gambar 6. Cara Kerja Sistem Mode Manual

Gambar 6. menunjukkan alur kerja dengan mode manual, pengguna menginput jadwal dan porsi pakan secara langsung via *website*. Sistem memproses input tersebut dan mengaktifkan sensor serta motor untuk mendistribusikan pakan sesuai waktu dan jumlah yang ditentukan. Proses selanjutnya berjalan seperti pada metode otomatis, memastikan pemberian pakan tetap konsisten dan sesuai kebutuhan.

F. Cara Kerja Deteksi Objek

Bagian ini menjelaskan cara kerja sistem deteksi objek yang digunakan untuk mengenali identitas kucing secara otomatis. Penjelasan difokuskan pada alur proses mulai dari pengambilan citra, pemrosesan oleh model AI, hingga hasil keluaran berupa identitas dan tingkat keyakinan deteksi.

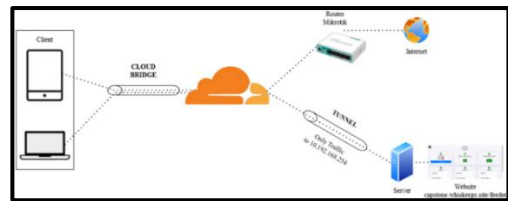


Gambar 7. Cara Kerja Deteksi Objek dengan YOLOv11

Gambar 7. menggambarkan cara kerja sistem deteksi kucing yang menggunakan kamera *webcam* dan AI berbasis YOLOv11. Kamera menangkap pergerakan objek secara *real-time*, lalu gambar diproses untuk mengenali kucing berdasarkan model yang telah dilatih melalui *Roboflow* dan *Ultralytics*. Jika *confident score* $\geq 0,88$, objek dianggap kucing dan sistem melanjutkan proses pemberian pakan. Sebaliknya, jika nilai $< 0,88$, objek dianggap bukan kucing dan sistem akan terus mengulang proses deteksi hingga akurasi tercapai. Mekanisme ini memastikan pemberian pakan hanya dilakukan saat kucing teridentifikasi dengan benar.

G. Cara Kerja Cloud Computing

Bagian ini membahas cara kerja *cloud computing* dalam sistem, khususnya pada proses integrasi antara server lokal dan akses internet melalui *Cloudflare Tunnel*. Penjelasan difokuskan pada alur komunikasi data, pengelolaan akses jarak jauh, serta peran *cloud* dalam mendukung pemantauan dan kontrol sistem secara *real-time*.

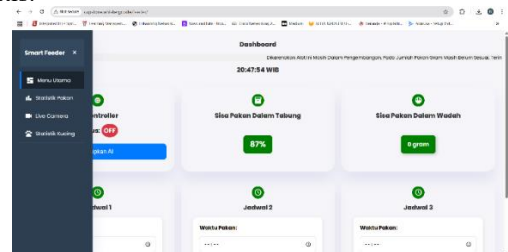


Gambar 8. Cara Kerja Cloud Computing

Pada Gambar 8. Menunjukkan cara kerja *cloud computing* dalam sistem ini berfungsi sebagai pusat pemrosesan dan penyimpanan data secara *real-time*. Server *cloud* publik terhubung melalui internet dengan dukungan HTTPS, FastAPI, dan *database SQL*. *Cloudflare Tunnel* digunakan untuk menghubungkan server lokal ke internet secara aman tanpa IP publik. *Website* sistem dilakukan *hosting* dan dapat diakses kapan pun untuk monitoring dan *controlling*. Router Mikrotik mengatur konektivitas jaringan melalui konfigurasi IP, DHCP, NAT. Integrasi tersebut memungkinkan pengguna mengakses sistem dari mana saja dengan performa yang andal dan responsif.

H. Platform Monitoring dan Controlling Sistem

Bagian ini menjelaskan bahwa *website* berfungsi sebagai platform monitoring dan *controlling* sistem pakan kucing otomatis.



Gambar 9. Tampilan Website

Pada Gambar 9. tampilan *website* tersebut memudahkan pengguna dalam mengelola dan memantau sistem secara jarak jauh, kapan pun dan di mana pun, selama terhubung dengan internet.



Gambar 10. Tampilan Notifikasi Telegram

Pada Gambar 10. konsep kerja notifikasi *real-time* melalui Telegram dalam sistem pakan kucing otomatis berbasis IoT dan AI. Sistem akan mengirim pesan notifikasi ketika terdeteksi kejadian tertentu seperti stok pakan, perubahan jadwal, status AI, atau keberhasilan pemberian pakan. *Backend* memproses data dan mengirimkan informasi kontekstual secara tepat waktu agar pengguna bisa segera merespon.

IV. HASIL DAN PEMBAHASAN

Pada bagian ini akan dijelaskan mengenai hasil pengujian dari sistem pakan kucing otomatis berbasis IoT dan AI yang telah dirancang dan diimplementasikan. Pengujian dilakukan untuk mengevaluasi performa dan fungsionalitas sistem berdasarkan parameter-parameter yang telah ditentukan. Hasil pengujian ini akan menjadi dasar dalam menilai efektivitas, efisiensi, serta keandalan sistem dalam kondisi nyata.



Gambar 11. Prototipe Sistem Pakan Kucing Otomatis

Gambar 11. menunjukkan hasil prototipe sistem pemberi pakan kucing otomatis yang telah berhasil dirancang dan diimplementasikan. Prototipe ini mencakup integrasi perangkat keras seperti *Raspberry Pi 4*, sensor *loadcell*, sensor ultrasonik, motor *stepper*, serta kamera *webcam* yang dikemas dalam satu sistem fungsional. Seluruh komponen terhubung ke sistem berbasis *website* dan AI yang mendukung proses pendeteksian kucing, pemberian pakan otomatis, serta pemantauan jarak jauh secara *real-time*.

A. Pengujian Sistem Pakan Kucing Otomatis

Pengujian pada sistem pakan kucing otomatis ini terbagi menjadi dua mode utama, yaitu mode otomatis dan mode manual. Masing-masing mode memiliki mekanisme kerja yang berbeda dalam menentukan dan menyalurkan porsi pakan. Selain itu, juga terdapat pengujian akurasi sensor untuk memantau sisa pakan.

Tabel 1. Pengujian Sistem Pakan Kucing Mode Otomatis

Jumlah Pengujian	Rata-rata Keakuratan
20 Kali	91,92%

Pada Tabel 1. pengujian mode otomatis, sistem menunjukkan performa yang baik dengan rata-rata akurasi sebesar 91,92% untuk target berat 50 gram. Distribusi pakan terbukti stabil dan presisi, dengan selisih hasil terhadap timbangan digital hanya 3–5 gram. Akurasi berkisar antara 90,91% hingga 94,34% dalam 20 kali pengujian. Kendala yang ditemukan adalah tersangkutnya pakan akibat desain tabung distribusi yang kurang besar, namun mekanisme lima kali pergerakan *slider* membantu mendorong sisa pakan keluar secara bertahap sehingga tetap menghasilkan distribusi yang konsisten. Hasil ini menunjukkan sistem bekerja optimal pada mode otomatis untuk porsi pakan besar.

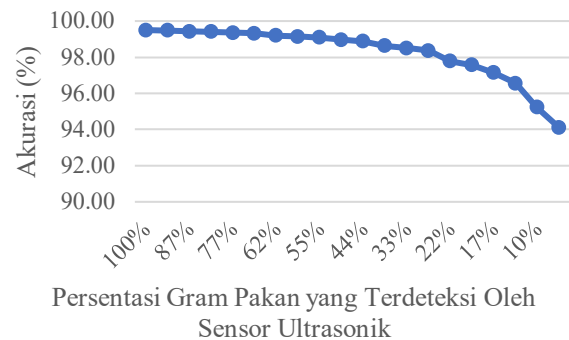
Tabel 2. Pengujian Sistem Pakan Kucing Mode Manual

Jumlah Pengujian	Rata-rata Keakuratan
20 Kali	89,13%

Berdasarkan Tabel 2. mode manual, pengguna mengatur langsung waktu dan takaran pakan melalui *website* tanpa bergantung pada deteksi AI. Hasil pengujian menunjukkan akurasi distribusi sebesar 89,13% dari 20 kali pengujian, sedikit lebih rendah dibanding mode otomatis. Penurunan

akurasi disebabkan oleh keterlambatan input pengguna dan hambatan mekanis. Meski demikian, mode ini tetap berfungsi baik sebagai alternatif fleksibel untuk pemberian pakan secara langsung.

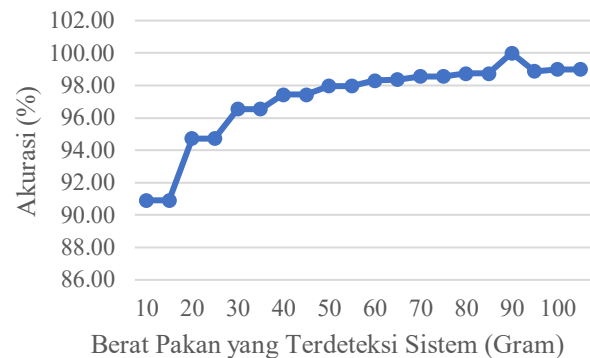
Sensor Ultrasonik



Gambar 12. Hasil Pengujian Keakuratan Sensor Ultrasonik

Pada Gambar 12. menunjukkan grafik pengujian sensor ultrasonik menunjukkan akurasi rata-rata sebesar 98,57% dalam 20 kali pengujian. Sensor mengukur ketinggian pakan dan mengkonversinya ke satuan gram dengan selisih hanya ± 1 gram. Akurasi tertinggi tercapai pada kapasitas penuh (99,50%) dan terendah pada kapasitas rendah 10% (95,12%), menunjukkan penurunan presisi pada volume kecil akibat batas deteksi pantulan gelombang. Secara keseluruhan, sensor terbukti sangat andal untuk memantau sisa pakan dalam tabung penyimpanan hingga kapasitas 2 kg untuk pakan *dry food* besar dan 4 kg untuk pakan *dry food* kecil.

Sensor Loadcell



Gambar 13. Hasil Pengujian Keakuratan Sensor *Loadcell*

Pada Gambar 13. sensor *loadcell* menunjukkan performa sangat baik dengan rata-rata akurasi 97,48% dalam 20 kali pengujian. Akurasi meningkat seiring bertambahnya berat pakan yang ditimbang, dari 90,91% pada 10 gram hingga 98,99% pada 100 gram. Penurunan akurasi pada berat kecil disebabkan oleh sensitivitas tinggi terhadap gangguan seperti getaran dan gesekan, yang memengaruhi hasil pembacaan. Meskipun demikian, sensor terbukti andal untuk pengukuran berat pakan, terutama pada massa menengah hingga besar.

B. Pengujian Akurasi Deteksi

Pada pengujian ini model AI YOLOv11 akan diuji seberapa akurat dalam mendeteksi kucing. Hasil keakuratan akan dilihat berdasarkan nilai *confident score* yang diperoleh oleh objek yang terdeteksi.

Tabel 3. Pengujian Akurasi Deteksi Berdasarkan Jarak

Nomor Pengujian	Status Hasil Pengujian	Nama Kucing	Jarak	Confident Score
Pengujian 1	Terdeteksi	Kucing A	20 cm	0.92
Pengujian 2	Terdeteksi	Kucing B	20 cm	0.90
Pengujian 3	Terdeteksi	Kucing A	50 cm	0.91
Pengujian 4	Terdeteksi	Kucing B	50 cm	0.89
Pengujian 5	Terdeteksi	Kucing A	100 cm	0.90
Pengujian 6	Terdeteksi	Kucing B	100 cm	0.89

Pada Tabel 3. hasil pengujian menunjukkan bahwa sistem deteksi kucing memiliki akurasi 100% pada jarak 20 cm dan 50 cm, menandakan performa yang memadai pada jarak dekat dan menengah. Namun, seiring jauhnya jarak seperti 100 cm terdapat kegagalan deteksi sebanyak 2 kali dari total 6 pengujian akibat keterbatasan visual dari objek gambar yang digunakan. Secara keseluruhan, sistem mencapai akurasi rata-rata sebesar 95%. Akurasi tersebut diperoleh dari total pengujian berhasil dibagi dengan total pengujian yang dilakukan lalu dikali 100%.



Gambar 14. Hasil Pengujian Deteksi Kucing Berdasarkan Jarak

Pada Gambar 14. Menampilkan hasil pengujian deteksi kucing berdasarkan jarak yang telah ditentukan sebelumnya, yaitu 20 cm, 50 cm, dan 100 cm. Terlihat pada gambar bahwa kucing yang telah teridentifikasi sebelumnya dapat dideteksi dan memiliki *confident score* $\geq 0,88$.

Tabel 4. Pengujian Akurasi Deteksi Berdasarkan Gangguan

Nomor Pengujian	Kondisi Pengujian	Status Hasil Pengujian	Confident Score
Pengujian 1	Menghadirkan Objek Lain	Tidak Terdeteksi	Tidak Terdeteksi
Pengujian 2	Cahaya Redup	Terdeteksi	Kucing A 0.93 dan Kucing B 0.89
Pengujian 3	Dua Kucing Teridentifikasi	Terdeteksi dan Sistem Non Aktif	Kucing A 0.94 dan Kucing B 0.88

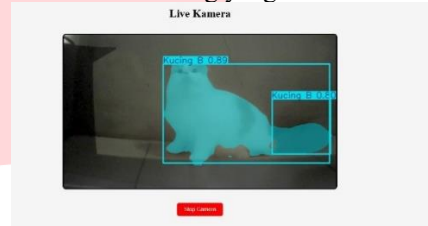
Pada Tabel 4. Pengujian dengan objek selain kucing menunjukkan bahwa sistem tidak salah mendeteksi, membuktikan kemampuan klasifikasi yang baik. Sistem juga tetap akurat dalam pencahayaan redup pada jarak dekat hingga menengah. Selain itu, saat dua kucing teridentifikasi bersamaan, sistem berhasil menonaktifkan distribusi pakan

otomatis ketika terdapat dua kucing yang telah teridentifikasi sebelumnya terdeteksi oleh kamera, menunjukkan logika pengendalian pakan ganda bekerja dengan baik.



Gambar 15. Hasil Deteksi Kucing Belum Teridentifikasi

Gambar 15. menampilkan hasil deteksi kucing yang belum pernah diidentifikasi oleh sistem sebelumnya. Pada hasil tersebut kucing tidak dapat dikenali oleh sistem dengan bukti tidak adanya *bounding box* dan *confident score* yang menunjukkan keaslian kucing yang telah teridentifikasi.



Gambar 16. Hasil Deteksi Kucing Pada Cahaya Redup

Gambar 16. menampilkan hasil deteksi kucing yang telah teridentifikasi, namun dalam kondisi cahaya yang redup. Hasil menunjukkan bahwa sistem tetap berjalan optimal untuk dapat mendeteksi kucing, meskipun cahaya di area sistem redup.



Gambar 17. Hasil Deteksi Dua Kucing yang Telah Teridentifikasi

Gambar 17. Menampilkan hasil deteksi dua kucing yang telah teridentifikasi sebelumnya oleh sistem. Dari hasil kedua kucing tersebut yang tertangkap kamera sistem berhasil menonaktifkan pakan sehingga dapat dikatakan *logic* sistem bekerja sesuai dengan desain spesifikasi untuk mencegah pemberian pakan ganda.

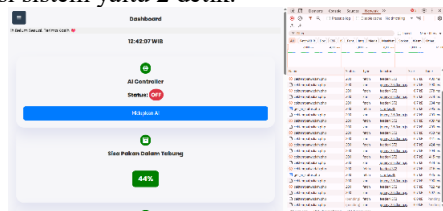
C. Pengujian Platform Monitoring dan Controlling

Pengujian integrasi platform mencakup *website* dan notifikasi Telegram menunjukkan hasil sangat baik dengan tingkat keberhasilan 100% pada seluruh fitur yang diuji. *Website* berfungsi optimal dalam hal fungsionalitas dan kompatibilitas, mencakup fitur *AI controller*, *live clock*, *live feed camera*, pengaturan jadwal, tampilan sisa pakan, serta statistik *history*, semuanya berjalan tanpa kendala. Pengaturan jadwal berhasil diuji sebanyak tiga kali dengan hasil sukses seluruhnya. Antarmuka *website* juga responsif dan *user-friendly*, dapat diakses dengan baik melalui berbagai perangkat seperti laptop, *handphone*, *tablet*, dan komputer.

Tabel 5. Hasil Pengujian Connect Time Website

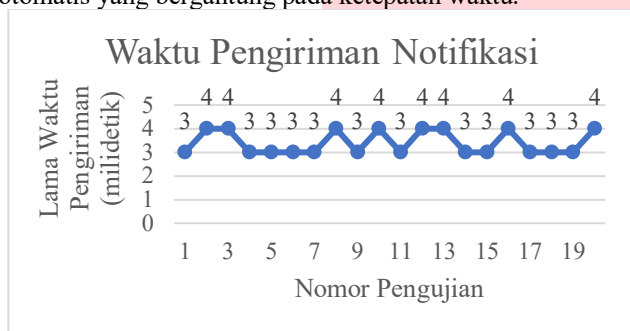
Komponen Pengujian	Time (milidetik) Pada Connect Time Website
Rata-rata	109.2

Pada Tabel 5. menunjukkan pengujian *connect time* dengan rata-rata waktu respon *website* sebesar 109,2 milidetik dari 20 permintaan, jauh di bawah ambang batas spesifikasi sistem yaitu 2 detik.



Gambar 18. Pengujian *Connect Time Website*

Gambar 18. menunjukkan pengujian dilakukan menggunakan *Inspect* pada *tab Network* ketika akses *website*. Hasil ini membuktikan bahwa platform *website* beroperasi cepat dan andal sebagai sistem *real-time*, memungkinkan proses monitoring dan *controlling* jarak jauh secara efisien. Kecepatan ini penting untuk memastikan perintah dan data ditangani dengan akurat, khususnya dalam pemberian pakan otomatis yang bergantung pada ketepatan waktu.



Gambar 19. Hasil Pengujian Waktu Pengiriman Notifikasi

Pada Gambar 19. menunjukkan sistem notifikasi memiliki performa sangat baik dengan rata-rata waktu pengiriman hanya 3,4 milidetik dari 20 pengujian, jauh di bawah standar 5–7 detik. Hasil ini membuktikan bahwa pengiriman notifikasi berlangsung secara *real-time* dan efisien, mendukung pengelolaan pakan jarak jauh secara instan dan responsif.

D. Pengujian *Latency* Komunikasi Data dan Akses *Website*

Bagian ini menjelaskan pengujian terhadap waktu akses dan *latency website* saat *website* tersebut diakses diluar jaringan lokal. Tujuannya adalah untuk mengetahui responsivitas sistem serta seberapa cepat data ditampilkan ke pengguna dalam kondisi aktual.

Tabel 6. Pengujian *Latency* dan Akses *Website*

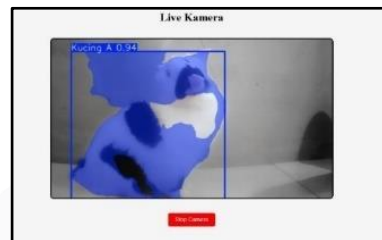
Sample	Status	Bytes	Sent Bytes	Latency (milidetik)
1	Success	14301	269	403
2	Success	14299	269	361
3	Success	14285	269	389
4	Success	14293	269	351
5	Success	14296	269	372
6	Success	14281	269	380
7	Success	14308	269	373
8	Success	14290	269	349
9	Success	14294	269	376
10	Success	14298	269	372
11	Success	14301	269	361
12	Success	14298	269	369
13	Success	14282	269	347
14	Success	14296	269	376

15	Success	14293	269	380
16	Success	14295	269	388
17	Success	14296	269	413
18	Success	14298	269	374
19	Success	14297	269	363
20	Success	14283	269	375
Rata-rata	Success	14294,2	269	373,6 milidetik

Pada Tabel 6. menunjukkan hasil pengujian *latency* pada komunikasi data berupa *cloud computing*. Pengujian tersebut menggunakan *Apache JMeter* menunjukkan rata-rata *latency* sebesar 373,6 milidetik, dengan seluruh pengujian berhasil akses *website*. Meski melebihi standar ideal <150 milidetik menurut TIPHON dan ITU-T G.114, nilai ini masih tergolong responsif dan tidak mengganggu fungsi sistem. *Latency* dipengaruhi oleh kualitas koneksi internet, dan peningkatan jaringan atau infrastruktur server.

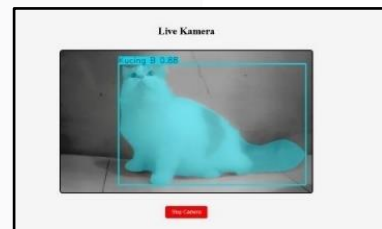
E. Pengujian Optimalisasi Penggunaan

Pengujian pada optimalisasi penggunaan dilakukan agar mengetahui apakah sistem benar-benar valid untuk mendeteksi antara kucing yang telah teridentifikasi sebelumnya atau belum.



Gambar 20. Hasil Deteksi Kucing A

Gambar 20. menunjukkan Kucing A berhasil terdeteksi dengan *confident score* $\geq 0,88$ setelah melalui proses pelabelan dan pelatihan *dataset*. Sistem mengenali identitasnya secara otomatis dan memicu pemberian pakan sesuai logika sistem.



Gambar 21. Hasil Deteksi Kucing B

Gambar 21. menampilkan deteksi Kucing B yang juga telah dilatih sebelumnya. Dengan *confident score* $\geq 0,88$, sistem menampilkan label “Kucing B” dan menjalankan distribusi pakan sesuai jatah yang ditentukan.



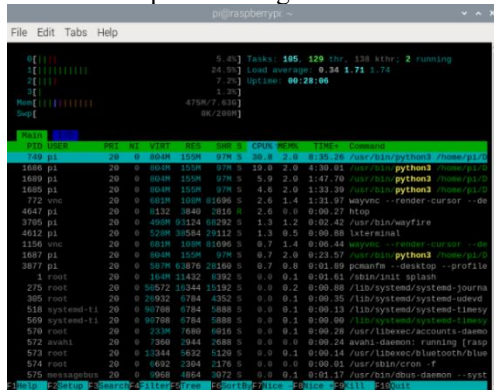
Gambar 22. Hasil Deteksi Kucing Tidak Teridentifikasi

Gambar 22. memperlihatkan seekor kucing yang tidak dikenali karena belum terdaftar di *database*. Sistem tidak menampilkan label atau *confident score*, dan pakan tidak diberikan, sebagai bentuk pengamanan distribusi.

F. Pengujian Penyimpanan Data

Pengujian penyimpanan data dilakukan untuk mengetahui apakah mikrokontroler *Raspberry Pi 4* mampu menyimpan *history* data sensor, deteksi objek tanpa adanya hambatan serta proses penyimpanan dapat berjalan dengan baik.

Sistem mampu mendeteksi dan membedakan individu kucing yang telah teridentifikasi dengan akurat, serta menolak pemberian pakan pada kucing yang tidak dikenal. Fitur ini penting dalam mengelola pakan secara selektif dan adil, terutama untuk pemilik dengan lebih dari satu kucing.



PID	PPID	UID	GID	NAME	STATE	VSZ	RSS	TTY	TIME	COMMAND
1	0	root	0	init	S	0	0		0:00	/sbin/init
2	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
3	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
4	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
5	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
6	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
7	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
8	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
9	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
10	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
11	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
12	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
13	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
14	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
15	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
16	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
17	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
18	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
19	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
20	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
21	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
22	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
23	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
24	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
25	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
26	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
27	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
28	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
29	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
30	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
31	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
32	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
33	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
34	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
35	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
36	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
37	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
38	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
39	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
40	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
41	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
42	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
43	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
44	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
45	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
46	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
47	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
48	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
49	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
50	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
51	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
52	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
53	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
54	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
55	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
56	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
57	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
58	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
59	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
60	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
61	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
62	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
63	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
64	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
65	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
66	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
67	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
68	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
69	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
70	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
71	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
72	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
73	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
74	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
75	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
76	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
77	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
78	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
79	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
80	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
81	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
82	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
83	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
84	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
85	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
86	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
87	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
88	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
89	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
90	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
91	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
92	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
93	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
94	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
95	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
96	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
97	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
98	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
99	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd
100	0	root	0	systemd	S	0	0		0:00	/usr/lib/systemd/systemd

Gambar 23. Hasil Pengujian Penyimpanan Data

Pada Gambar 23. sistem pakan kucing otomatis menunjukkan performa yang efisien dalam penyimpanan data. Dengan kapasitas 32 GB, sistem mampu menyimpan data sensor dan deteksi objek secara berkelanjutan tanpa kendala. Seluruh pengujian berhasil 100%, tanpa terjadi *crash*, *error*, atau *overload*, serta tanpa penggunaan memori cadangan berlebih. Hal ini menandakan optimasi memori dan pemrosesan yang baik, mendukung kestabilan dan kelancaran fungsi sistem.

V. KESIMPULAN

Berdasarkan implementasi sistem pakan kucing otomatis dengan integrasi IoT dan AI berhasil beroperasi melalui platform *website* yang terhubung dengan *Cloudflare Tunnel* secara *real-time*, dengan waktu respon rata-rata 109,2 ms dan notifikasi Telegram 3,4 ms. Sistem deteksi AI dengan YOLOv11 mencapai akurasi 95% dalam mengenali kucing terdaftar dan mencegah pemberian pakan ganda. Distribusi pakan tercatat akurat, masing-masing 91,92% untuk mode otomatis dan 89,13% untuk mode manual, serta mendukung jadwal makan harian dan penyimpanan histori. Sistem juga dapat dipantau dari jarak jauh melalui antarmuka *website* interaktif. Selain itu, sensor ultrasonik dan *loadcell* mencatat akurasi tinggi, yakni 98,57% dan 97,48%, serta aktuator motor *stepper* bekerja stabil dalam mendistribusikan pakan sesuai logika sistem.

REFERENSI

- [1] 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.
- [2] W. Aditya and S. Subektiningsih, "Scheduled Cat Feeder Berbasis Internet of Things Menggunakan Wemos D1 Mini dan Telegram," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 11, no. 1, pp. 183–190, 2024, doi: 10.25126/jtiik.20241117847.
- [3] A. A. Mulky, R. Tulloh, and A. Alfaruq, "Smart

Feeder Kucing Menggunakan Metode You Only Look Once (YOLO)," *Appl. Sci.*, vol. 5, no. 5, pp. 2535–2538, 2023.

- [4] S. I. Baehaki, Mochamad Hilman Lestaringati, "Pemberi Pakan Hewan Peliharaan Berbasis Web," *Komputika*, vol. 6, no. 1, pp. 13–16, 2017.
- [5] M. A. Sadad, L. Narpulaela, and R. Rahmadewi, "Analisis Metode Fuzzy Logic Pada Sistem Pemberi Makan Kucing Otomatis Studi Kasus Makanan Kering," *J. Tek. Elektro dan Komputasi*, vol. 5, no. 1, pp. 16–27, 2023.
- [6] D. Nafis Alfari, R. Agung Pangestu, D. Aditya, M. Adi Setiawan, and P. Rosyani, "Penggunaan Metode YOLO Pada Deteksi Objek: Sebuah Tinjauan Literatur Sistematis," *J. Artif. Intel. dan Sist. Penunjang Keputusan*, vol. 1, no. 1, pp. 54–63, 2023.
- [7] Bayu Prastyo, Faiz Syaikhoni Aziz, Wahyu Pribadi, and A.N. Afandi, "Desain Banyumas Smart City Berbasis Internet of Things (IoT) Menggunakan Fog Computing Architecture," *J. JEETech*, vol. 1, no. 2, pp. 6–13, 2020, doi: 10.48056/jeeitech.v1i2.7.
- [8] A. S. P. Muhammed Hafid Alfayed, "Prototipe Alat Pemberi Pakan Kucing Otomatis Berbasis Internet of Things," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 8, no. 1, pp. 937–944, 2024.
- [9] F. Susanto, N. K. Prasiani, and P. Darmawan, "Implementasi Internet of Things Dalam Kehidupan Sehari-Hari," *J. Imagine*, vol. 2, no. 1, pp. 35–40, 2022, doi: 10.35886/imagine.v2i1.329.
- [10] M. Sobron and Lubis, "Implementasi Artificial Intelligence Pada System Manufaktur Terpadu," *Semin. Nas. Tek. UISU*, vol. 4, no. 1, pp. 1–7, 2021, [Online]. Available: <https://jurnal.uisu.ac.id/index.php/semnastek/article/view/4134>
- [11] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv*, 2020, [Online]. Available: <http://arxiv.org/abs/2004.10934>
- [12] E. Barus, K. M. Pardede, and J. A. Putri Br. Manjorang, "Transformasi Digital: Teknologi Cloud Computing dalam Efisiensi Akuntansi," *J. Sains dan Teknol.*, vol. 5, no. 3, pp. 904–911, 2024, doi: 10.55338/saintek.v5i3.2862.
- [13] Tri Yulianto B, Quraisy M, Daulay A, and Puspita Sari A, "Private Server Design Using Proxmox Platform and Implementation of Zero Trust Model with Cloudflare," no. November 2023, pp. 195–207, 2023, [Online]. Available: <https://www.researchgate.net/publication/379957385>
- [14] E. R. Rahmi, E. Yumami, and N. Hidayasari, "Analisis Metode Pengembangan Sistem Informasi Berbasis Website: Systematic Literature Review," *Remik*, vol. 7, no. 1, pp. 821–834, 2023, doi: 10.33395/remik.v7i1.12177.
- [15] M. Furqan, S. Sriani, and M. N. Shidqi, "Chatbot Telegram Menggunakan Natural Language Processing," *Walisongo J. Inf. Technol.*, vol. 5, no. 1, pp. 15–26, 2023, doi: 10.21580/wjit.2023.5.1.14793.