

LAPORAN IOT



PERANCANGAN DAN IMPLEMENTASI SISTEM PEMBERI PAKAN AYAM KAMPUNG OTOMATIS MENGGUNAKAN MEKANISME AUGER SPIRAL

Oleh :

**Eggy Alfan Ananta /1103223194
Jaka Kelana Wijaya / 1103223048**

**PRODI S1 TEKNIK KOMPUTER
FAKULTAS TEKNIK ELEKTRO
UNIVERSITAS TELKOM
2025**

DAFTAR ISI

DAFTAR ISI.....	2
DAFTAR GAMBAR.....	3
DAFTAR TABEL	4
1.1. Gambaran Umum Pemberian Ayam Kampung dengan Sistem Auger.....	5
1.2. Deskripsi Umum Sistem	6
1.3. Arsitektur Sistem.....	7
1.4. Desain Perangkat Keras.....	8
1.5. Rencana Anggaran Biaya (RAB)	14
1.6. Desain Workflow Perangkat Lunak.....	15
1.6.1. Workflow Utama Sistem	15
1.6.2. Flowchart Pemberian Pakan Manual	17
1.6.3. Flowchart subproses feeding	18
1.6.4. Flowchart Pemantauan Menggunakan Kamera	19
1.6.5. Sequence Diagram Manajemen Jadwal.....	20
1.6.6. Sequence Diagram Permintaan Status Sistem	21
1.7. Gambaran Source Code Modul Hardware	22
1.8. Petunjuk Instalasi Perangkat Keras	23
DAFTAR PUSTAKA.....	24
LAMPIRAN.....	26

DAFTAR GAMBAR

Gambar 1. 1 Diagram blok arsitektur sistem	8
Gambar 1. 2 Rangkaian Sistem.....	9
Gambar 1. 3 Skematik Rangkaian Sistem.....	10
Gambar 1. 4 Desain PCB Sistem	11
Gambar 1. 5 State Transition Diagram Pakan Ayam.....	12
Gambar 1. 6 Flowchart Utama.....	16
Gambar 1. 7 Flowchart pemberian pakan manual	17
Gambar 1. 8 Flowchart Subproses Feeding	18
Gambar 1. 9 Flowchart pemantauan menggunakan kamera.....	19
Gambar 1. 10 Sequence Diagram Tambah Jadwal	20
Gambar 1. 11 Sequence Diagram Hapus Jadwal	20
Gambar 1. 12 Sequence Diagram Lihat Jadwal	21
Gambar 1. 13 Sequence Diagram Permintaan Status Sistem.....	22

DAFTAR TABEL

Tabel 1. 1 Parameter Auger Spiral + Telegram Bot	5
Tabel 1. 2 Transisi State.....	13
Tabel 1. 3 RAB Komponen dan Perangkat.....	14

1.1. Gambaran Umum Pemberian Ayam Kampung dengan Sistem Auger

Sistem ini menggunakan auger spiral sebagai mekanisme utama untuk mengangkut pakan dari hooper ke wadah makan ayam secara otomatis. Pengguna dapat mengatur jadwal pemberian pakan melalui Telegram Bot, serta memiliki opsiuntuk mengaktifkan sistem secara manual jika dibutuhkan. Auger berputar sesuai durasi yang telah diatur, dan sensor berat di wadah pakan memonitor jumlah pakan yang telah disalurkan [2], [4]. Ketika jumlah pakan atau durasi waktu pemberian tercapai, servo secara otomatis menutup jalur pakan untuk mencegah pemborosan.

Setiap kali pemberian pakan selesai, kamera akan mengambil foto kondisi wadah pakan dan mengirimkannya ke Telegram Bot. Foto ini memungkinkan peternak memantau kondisi pakan secara visual dari jarak jauh. Selain itu, Telegram Bot mengirimkan notifikasi status operasi, termasuk status mulai, selesai, gagal, atau stok menipis. Peternak dapat menambah, menghapus, atau memodifikasi jadwal secara fleksibel langsung melalui Telegram, sehingga seluruh kontrol berada di ujung jari tanpa perlu aplikasi tambahan atau server rumit [3].

Tabel 1. 1 Parameter Auger Spiral + Telegram Bot

Parameter	Deskripsi
Mekanisme Pemberian Pakan	Auger spiral otomatis
Kontrol Jadwal	Telegram Bot, otomatis dan manual
Monitoring Visual	Kamera real time melalui Telegram
Sensor	Sensor berat di wadah pakan
Notifikasi	Status dikirim otomatis ke Telegram
Skala Kandang	3–6 kandang, panjang 3–6 m, 30–40 ayam/kandang
Biaya Implementasi	Relatif rendah, komponen sederhana, Telegram gratis
Kompleksitas	Mudah diimplementasikan untuk usaha kecil

Analisis Keunggulan

- Kontrol jarak jauh: Penuh, karena semua pengaturan bisa dilakukan melalui Telegram.
- Presisi pemberian pakan: Tinggi, sensor berat dan servo menjamin jumlah pakan sesuai jadwal.
- Monitoring visual: Kamera memberi umpan balik langsung melalui Telegram.
- Skalabilitas: Cocok untuk 3–6 kandang, panjang 3–6 m, 30–40 ayam/kandang.
- Biaya: Relatif rendah, auger spiral dan komponen elektronik sederhana, Telegram gratis.
- Implementasi usaha kecil: Mudah, karena tidak memerlukan server atau aplikasi rumit, hanya mikrokontroler dan koneksi internet.

Kelemahan

- Membutuhkan perawatan auger agar tidak macet.
- Telegram terbatas pada pengiriman foto dan teks, tidak ada dashboard data histori yang kompleks.

1.2. Deskripsi Umum Sistem

Sistem smart feeder ayam ini dirancang sebagai sistem otomatis yang mampu menyalurkan pakan sesuai dengan jadwal atau perintah manual yang dikirimkan oleh pengguna melalui Telegram Bot. Mekanisme utama penyaluran pakan menggunakan auger spiral, yang berfungsi untuk memindahkan pakan dari wadah penyimpanan (hopper) menuju tempat pakan ayam secara terkontrol [7].

Pengendalian sistem dilakukan oleh sebuah mikrokontroler yang berperan sebagai pusat kendali. Mikrokontroler ini menerima input berupa perintah dari Telegram Bot dan data dari sensor, kemudian memproses input tersebut untuk mengaktifkan atau menonaktifkan motor penggerak auger. Untuk meningkatkan akurasi, sistem dilengkapi dengan sensor berat, sehingga jumlah pakan yang dikeluarkan dapat disesuaikan dengan target yang telah ditentukan [1], [4].

Selain fungsi pemberian pakan, sistem juga mendukung monitoring visual kondisi kandang melalui kamera. Hasil pengambilan gambar dikirimkan kepada

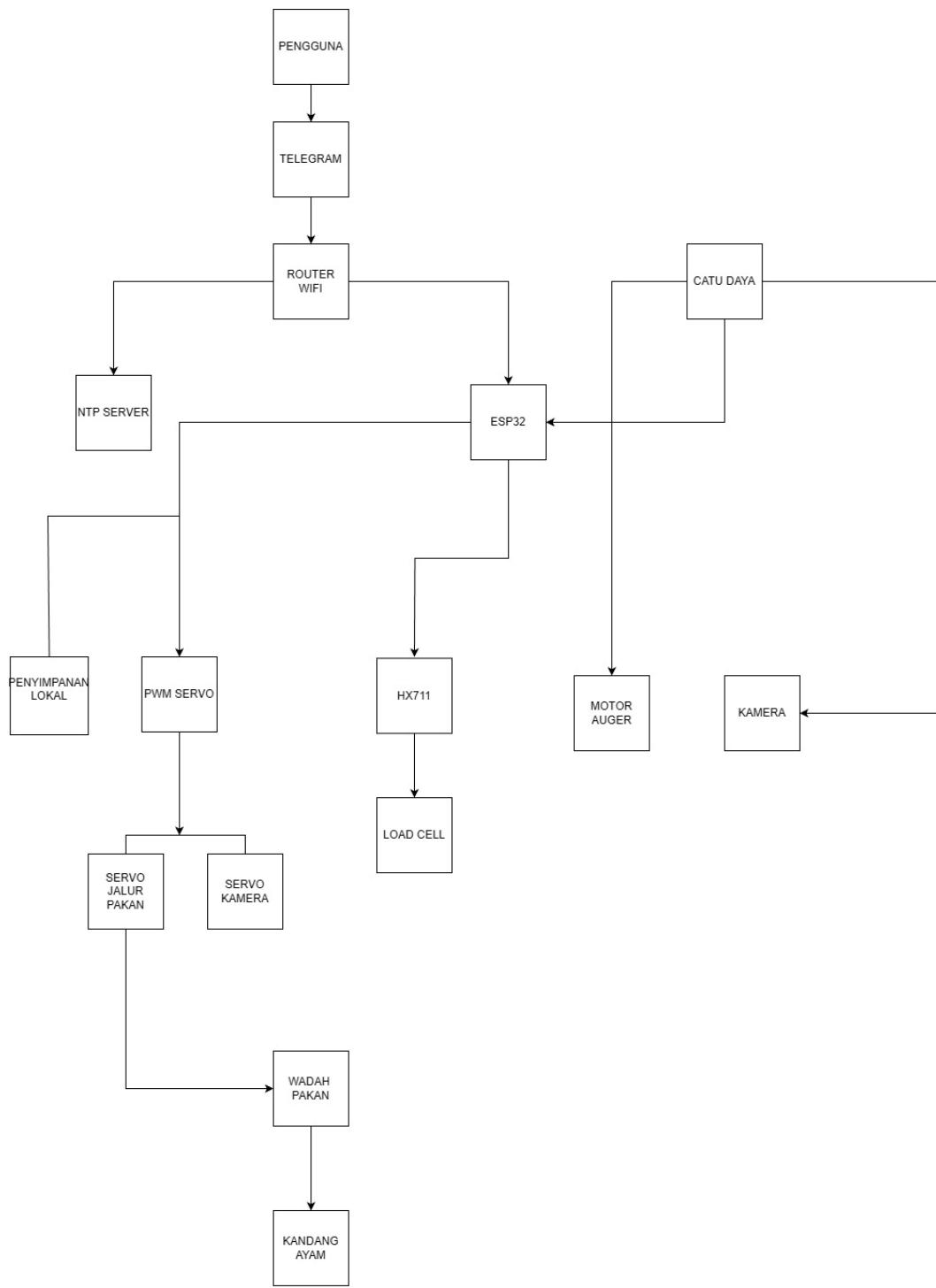
pengguna melalui Telegram sebagai bentuk umpan balik visual, sehingga pengguna dapat memantau kondisi kandang dan aktivitas sistem tanpa harus berada langsung di lokasi peternakan [10].

1.3. Arsitektur Sistem

Pada arsitektur sistem ini menjelaskan gambaran umum struktur dan keterkaitan antar komponen dalam sistem pemberian pakan ayam otomatis berbasis Internet of Things (IoT). Arsitektur sistem disajikan dalam bentuk diagram blok yang menunjukkan hubungan fungsional antara unit pengendali, sensor, aktuator, serta sistem pendukung lainnya. Penyajian arsitektur ini bertujuan untuk memberikan pemahaman menyeluruh mengenai bagaimana sistem dibangun dan bagaimana setiap komponen saling terhubung dalam satu kesatuan sistem, sebagaimana pendekatan arsitektur sistem IoT pada bidang peternakan [3].

Diagram blok arsitektur sistem menggambarkan alur interaksi antar komponen utama tanpa menampilkan detail kelistrikan. Hubungan yang ditampilkan bersifat konseptual dan fungsional, sehingga diagram ini berperan sebagai representasi tingkat sistem (system-level view). Pendekatan ini umum digunakan dalam perancangan sistem IoT untuk memudahkan pemahaman alur kerja sistem sebelum masuk ke tahap perancangan perangkat keras secara detail [2], [4].

Selain sebagai visualisasi hubungan antar komponen, arsitektur sistem juga menjadi dasar dalam perancangan rangkaian dan implementasi perangkat keras. Setiap komponen yang ditampilkan pada diagram blok akan direalisasikan dalam bentuk rangkaian sistem yang mencakup koneksi fisik, skematik, dan desain PCB. Tahapan ini penting untuk memastikan bahwa desain konseptual dapat diimplementasikan secara teknis dan berfungsi sesuai dengan kebutuhan operasional sistem pemberian pakan ayam otomatis [5], [7].



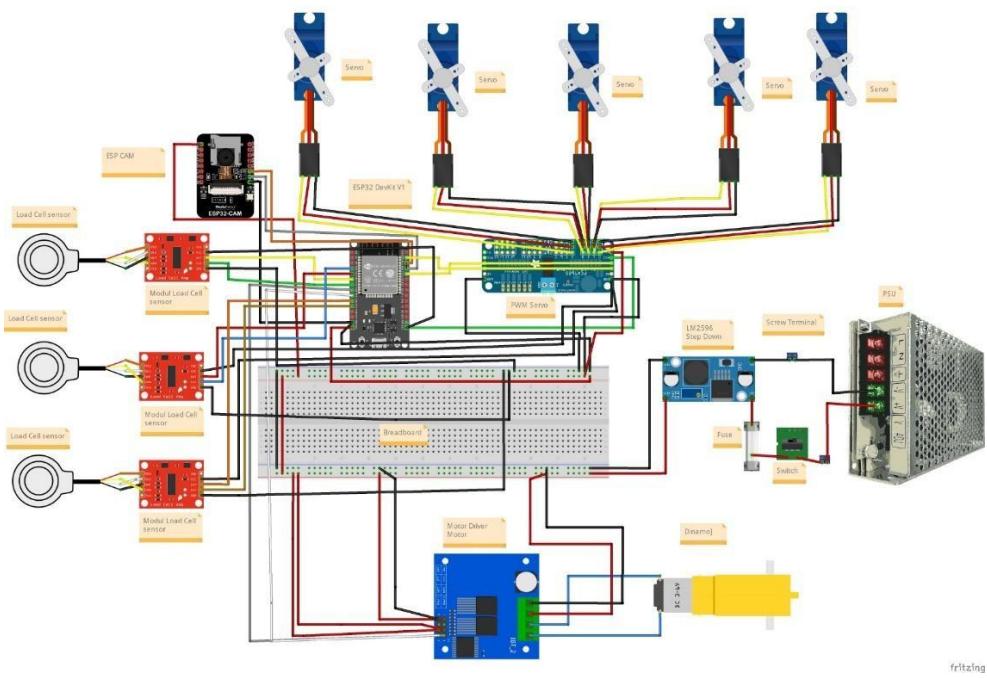
Gambar 1. 1 Diagram blok arsitektur sistem

1.4. Desain Perangkat Keras

Rangkaian sistem membahas perancangan rangkaian sistem secara menyeluruh yang mencakup rangkaian fisik, skematik, dan desain Printed Circuit Board (PCB). Rangkaian sistem menunjukkan bagaimana seluruh komponen perangkat keras

dihubungkan secara langsung untuk membentuk sistem pemberian pakan ayam otomatis yang terintegrasi, mulai dari unit pengendali hingga perangkat input dan output sistem [4], [11].

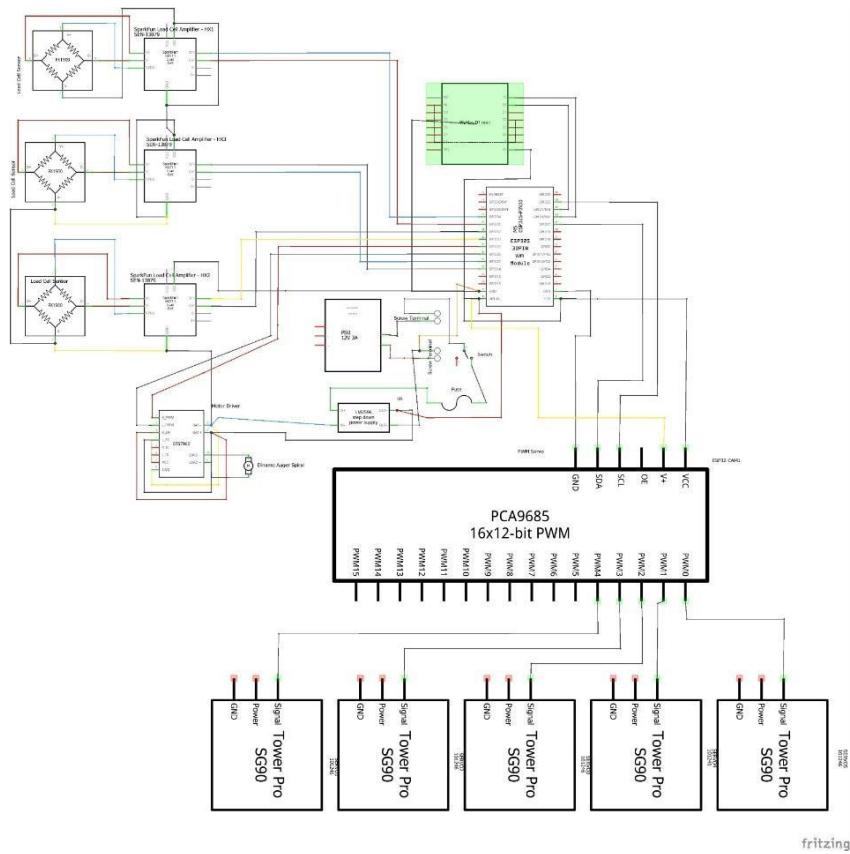
Rangkaian sistem disajikan dalam bentuk visual koneksi antar komponen untuk memperlihatkan hubungan fisik antara mikrokontroler, sensor, aktuator, catu daya, serta perangkat pendukung lainnya. Penyajian visual ini bertujuan memberikan gambaran awal mengenai jalur daya dan jalur sinyal antar komponen sebelum dituangkan ke dalam bentuk skematik teknis. Pendekatan ini mempermudah proses analisis dan identifikasi hubungan antar perangkat dalam sistem IoT berbasis perangkat keras [6], [8]. Visualisasi rangkaian fisik tersebut direalisasikan menggunakan aplikasi Fritzing, yang memungkinkan representasi koneksi antar komponen secara intuitif dan mudah dipahami, khususnya pada tahap awal perancangan sistem perangkat keras [13].



Gambar 1. 2 Rangkaian Sistem

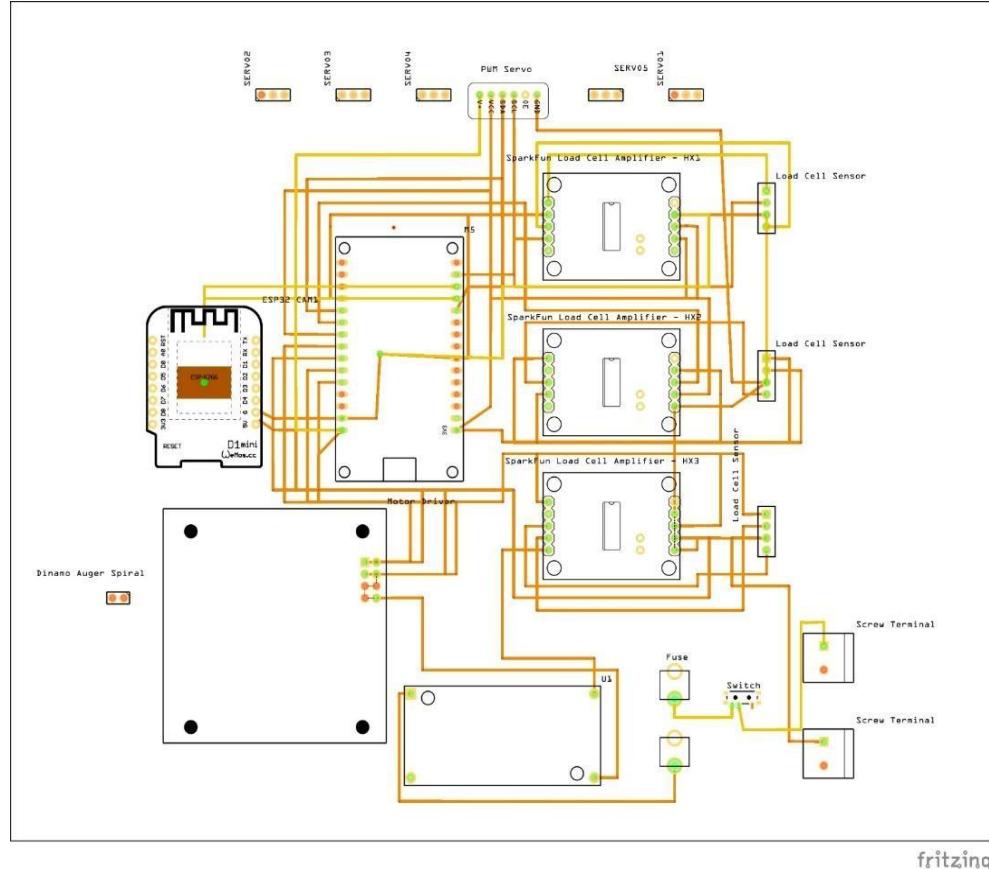
Skematik rangkaian menampilkan hubungan antar komponen secara lebih rinci dengan menggunakan simbol elektronik standar. Skematik ini menunjukkan konfigurasi pin, distribusi catu daya, serta jalur komunikasi data antar perangkat, sehingga dapat digunakan sebagai acuan teknis utama dalam proses perakitan dan pengujian sistem. Penyusunan skematik yang tepat sangat penting untuk memastikan kompatibilitas antar komponen dan mencegah kesalahan koneksi pada sistem otomatis [1], [9]. Penyusunan

skematik ini juga dilakukan menggunakan aplikasi Fritzing, sehingga konsistensi antara rangkaian fisik dan skematik dapat terjaga dengan baik [13].



Gambar 1. 3 Skematik Rangkaian Sistem

Berdasarkan skematik rangkaian, dilakukan perancangan desain PCB sebagai implementasi fisik dari rangkaian sistem. Desain PCB disusun dengan mempertimbangkan tata letak komponen, kerapihan jalur penghantar, serta kestabilan sistem secara keseluruhan. Penggunaan PCB bertujuan untuk meningkatkan keandalan sistem, mempermudah proses perakitan, serta mendukung implementasi sistem dalam jangka panjang, khususnya pada lingkungan peternakan yang menuntut kestabilan dan ketahanan perangkat [5], [12]. Desain PCB pada sistem ini dirancang menggunakan aplikasi Fritzing sebagai tahap awal perancangan, yang selanjutnya dapat dikembangkan atau diproduksi sesuai kebutuhan implementasi perangkat keras [13].



Gambar 1. 4 Desain PCB Sistem

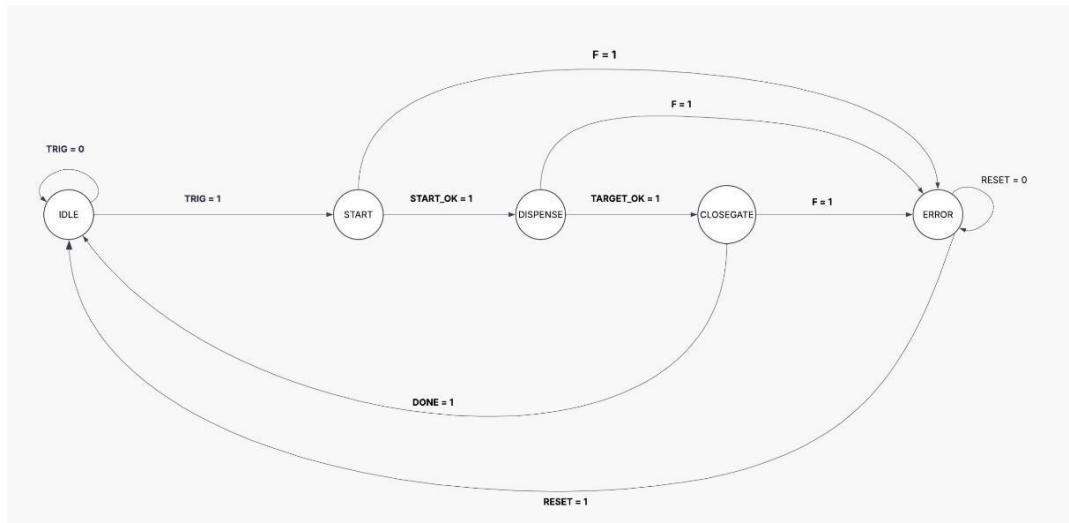
Selain perancangan perangkat keras, sistem ini juga dilengkapi dengan perancangan logika kendali menggunakan *Finite State Machine (FSM)* untuk memastikan proses pemberian pakan berjalan secara terstruktur, aman, dan terkendali. *FSM* digunakan untuk merepresentasikan perilaku sistem dalam bentuk state dan transisi antar state berdasarkan kondisi input tertentu, sehingga memudahkan analisis alur kerja sistem secara formal [4], [9].

Model *FSM* yang digunakan pada sistem ini adalah model Moore, di mana keluaran sistem ditentukan sepenuhnya oleh state aktif dan tidak bergantung langsung pada kombinasi input. Pendekatan ini dipilih karena memberikan kestabilan keluaran dan mengurangi kemungkinan perubahan output secara tiba-tiba akibat fluktuasi input sensor, yang penting dalam sistem otomasi pemberian pakan [1], [5].

State utama yang digunakan dalam *FSM* sistem pemberi pakan ayam ini meliputi IDLE, START, DISPENSE, CLOSEGATE, dan ERROR. State IDLE merepresentasikan kondisi sistem dalam keadaan siaga, menunggu pemicu pemberian pakan baik dari jadwal maupun perintah manual. Ketika pemicu (trigger)

terpenuhi, sistem berpindah ke state START untuk melakukan proses inisialisasi dan pengecekan kesiapan sistem. Setelah sistem dinyatakan siap, FSM berpindah ke state DISPENSE, di mana motor auger dan servo jalur pakan diaktifkan untuk menyalurkan pakan secara bertahap. Apabila target pemberian pakan telah tercapai, sistem masuk ke state CLOSEGATE untuk menutup jalur pakan dan menyelesaikan proses feeding. Setelah seluruh proses selesai dengan normal, sistem kembali ke state IDLE untuk menunggu siklus berikutnya. State ERROR digunakan sebagai mekanisme pengamanan apabila terdeteksi gangguan sistem, seperti kesalahan sensor, aktuator tidak merespons, atau kondisi operasional yang tidak normal. Sistem akan tetap berada pada state ini hingga dilakukan reset atau intervensi untuk mengembalikan sistem ke kondisi aman [6], [8].

Perilaku FSM tersebut direpresentasikan secara visual dalam bentuk *State Transition Diagram (STD)*, yang menggambarkan hubungan antar state serta kondisi transisi yang mungkin terjadi selama sistem beroperasi. STD memberikan gambaran konseptual mengenai alur kerja sistem dan keterkaitan antar state, sehingga memudahkan pemahaman logika kontrol secara keseluruhan [4], [9].



Gambar 1. 5 State Transition Diagram Pakan Ayam

Untuk melengkapi representasi visual tersebut, disusun pula tabel transisi state sebagai deskripsi formal dari FSM. Tabel transisi state menjelaskan hubungan antara current state, kombinasi input, next state, serta keluaran sistem secara sistematis. Setiap baris pada tabel merepresentasikan satu kondisi evaluasi FSM.

Penggunaan simbol don't care (X) pada kolom input menunjukkan bahwa nilai input tersebut tidak memengaruhi keputusan transisi pada kondisi tertentu, yang

merupakan praktik umum dalam perancangan FSM untuk menyederhanakan analisis logika [1], [9]. Output sistem pada FSM ini meliputi sinyal kendali motor auger dan servo jalur pakan. Karena FSM menggunakan model Moore, nilai output ditentukan oleh state aktif, sehingga perilaku aktuator menjadi lebih stabil dan mudah diprediksi selama proses pemberian pakan berlangsung. Dengan integrasi antara perancangan rangkaian sistem dan logika kendali berbasis FSM, sistem pemberi pakan ayam otomatis ini dirancang memiliki tingkat keandalan, keteraturan, dan keselamatan operasi yang lebih baik [5], [12].

Tabel 1. 2 Transisi State

Current State	TR IG	START _OK	TARGE T_OK	DO NE	FAU LT	RES ET	Next State	A	G
IDLE	0	X	X	X	X	X	IDLE	0	0
IDLE	1	X	X	X	X	X	START	0	0
START	X	X	X	X	1	X	ERROR	0	0
START	X	1	X	X	X	X	DISPENSE	0	0
DISPENSE	X	X	X	X	1	X	ERROR	1	1
DISPENSE	X	X	1	X	X	X	CLOSE GATE	1	1
CLOSE GATE	X	X	X	X	1	X	ERROR	0	0
CLOSE GATE	X	X	X	1	X	X	IDLE	0	0
ERROR	X	X	X	X	X	0	ERROR	0	0
ERROR	X	X	X	X	X	1	IDLE	0	0

1.5. Rencana Anggaran Biaya (RAB)

Rencana Anggaran Biaya (RAB) disusun untuk memberikan gambaran total biaya yang dibutuhkan dalam perancangan dan implementasi sistem pemberi pakan ayam otomatis. RAB ini mencakup seluruh kebutuhan komponen dan perangkat yang digunakan selama proses pembangunan sistem, mulai dari tahap perancangan hingga implementasi.

Perincian RAB disajikan dalam bentuk tabel yang memuat daftar komponen, jumlah, serta estimasi biaya yang diperlukan. Penyusunan RAB bertujuan sebagai acuan perencanaan pengadaan komponen sekaligus sebagai alat pengendalian biaya agar proses pengembangan sistem dapat berjalan secara efektif dan efisien.

Tabel 1. 3 RAB Komponen dan Perangkat

No	Komponen dan Perangkat	Jumlah	Estimasi Biaya
1	pca9685 (pwm servo)	1	Rp50.000
2	esp32 cam	1	Rp96.000
3	motor servo MG996	2	Rp70.800
4	jumper female to female	2	Rp32.000
5	jumper male to female	2	Rp32.000
6	jumper male to male	2	Rp32.000
7	ESP32	1	Rp80.000
8	power supply	1	Rp85.000
9	Motor Driver BTS7960	1	Rp55.000
10	Load cell Sensor Berat 10KG	2	Rp110.000
11	PSU 24V 10A SMPS (240W)	1	Rp220.000
12	Buck Converter 24V → 5V (logika)	1	Rp40.000
13	Buck Converter 24V → 6V 10A (servo)	1	Rp150.000
14	Kapasitor 2200µF 10V	3	Rp36.000
15	Kapasitor 1000µF 10V	3	Rp24.000
16	Fuse holder + Fuse 15A	1	Rp30.000
17	Fuse 10A (servo)	1	Rp12.000
18	Power switch	1	Rp15.000
19	Dinamo	1	Rp300.000
20	Auger	6	Rp258.000
21	Pipa besi ada bolongan	2	Rp220.000
22	Bearing	2	Rp50.000
23	As Besi	1	Rp26.000
24	drum besi 60L	1	Rp60.000
25	baut kleman auger	1	Rp35.000

No	Komponen dan Perangkat	Jumlah	Estimasi Biaya
26	klem pipa feeder	1	Rp17.000
27	tali tambang 15 meter	1	Rp30.000
Total			Rp2.165.800,00

1.6. Desain Workflow Perangkat Lunak

Desain perangkat lunak ini menjelaskan perancangan alur kerja perangkat lunak pada sistem pemberi pakan ayam otomatis berbasis Internet of Things (IoT). Desain workflow digunakan untuk menggambarkan bagaimana sistem merespons perintah pengguna, menjalankan proses pemberian pakan secara otomatis, serta menyampaikan informasi dan notifikasi melalui Telegram Bot. Penyusunan workflow bertujuan memastikan sistem dapat beroperasi secara terstruktur, responsif, dan mudah dipahami oleh pengguna [2], [3].

Perancangan perangkat lunak mengadopsi pendekatan event-driven system, di mana setiap proses dijalankan sebagai respons terhadap suatu kejadian tertentu, baik berupa perintah pengguna maupun pemicu waktu berdasarkan jadwal. Pendekatan ini umum digunakan pada sistem IoT karena mampu meningkatkan efisiensi pemrosesan, fleksibilitas kontrol, serta keandalan sistem dalam lingkungan operasional yang dinamis [6], [9].

Secara umum, alur kerja sistem direpresentasikan menggunakan workflow utama sebagai gambaran global sistem. Selanjutnya, setiap fungsi utama pada workflow tersebut dijabarkan lebih rinci dalam bentuk flowchart dan sequence diagram sebagai pecahan dari workflow utama agar logika sistem dapat dipahami secara lebih detail.

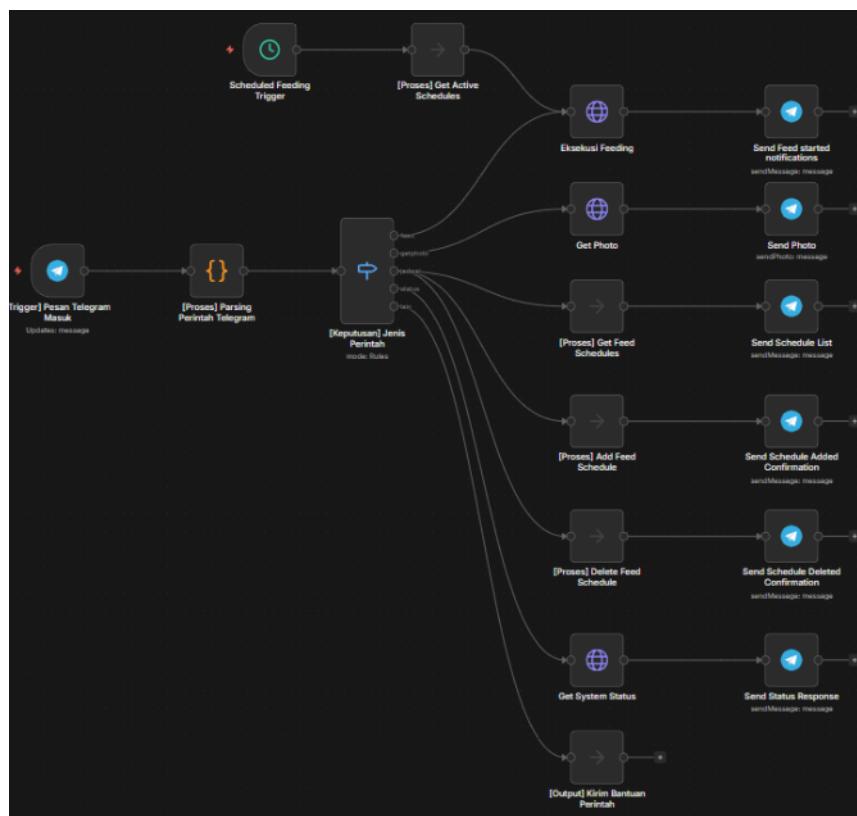
1.6.1. Workflow Utama Sistem

Workflow utama sistem menggambarkan alur kerja perangkat lunak secara keseluruhan, mulai dari kondisi sistem dalam keadaan siaga hingga eksekusi proses pemberian pakan dan pengiriman informasi kepada pengguna. Pada kondisi awal, sistem berada dalam mode siaga sambil memantau dua pemicu utama, yaitu pesan masuk dari Telegram Bot dan waktu sistem yang dibandingkan dengan jadwal pemberian pakan [2].

Apabila sistem menerima perintah dari pengguna melalui Telegram Bot, alur kerja diarahkan ke proses pengolahan perintah sesuai jenis instruksi yang diberikan. Sebaliknya, jika tidak terdapat perintah manual, sistem akan melakukan pengecekan

jadwal pemberian pakan yang tersimpan dan mengeksekusi proses feeding secara otomatis ketika waktu telah sesuai. Mekanisme ini mencerminkan konsep otomasi pada sistem smart farming berbasis IoT [3], [10].

Alur kerja utama tersebut direpresentasikan dalam bentuk diagram workflow yang dirancang menggunakan platform n8n sebagai alat bantu otomasi. Penggunaan n8n memungkinkan pemodelan alur proses berbasis peristiwa (event-driven) secara visual, sehingga setiap pemicu, pengambilan keputusan, dan eksekusi proses dapat digambarkan secara terstruktur dan mudah dipahami. Selain itu, n8n berperan dalam mengelola integrasi antara Telegram Bot, sistem penjadwalan, dan proses pengendalian perangkat, sehingga keseluruhan alur kerja perangkat lunak dapat berjalan secara otomatis dan terkoordinasi [14].

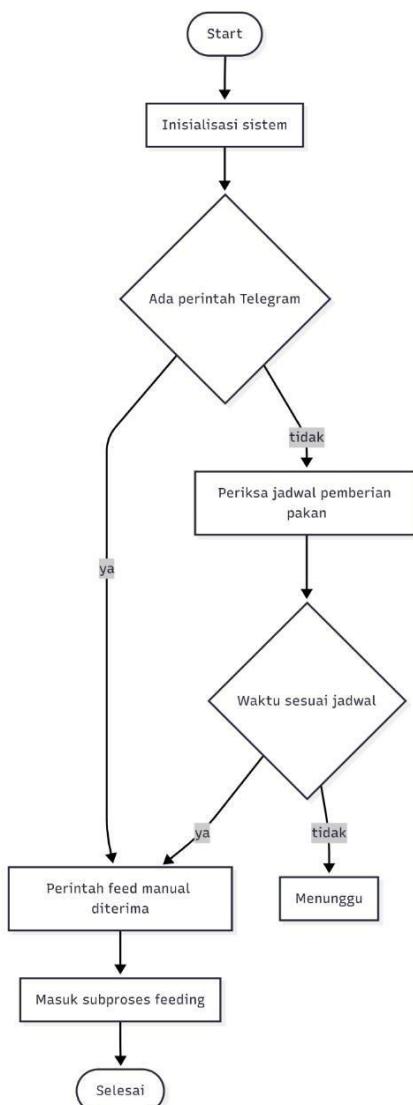


Gambar 1. 6 Flowchart Utama

1.6.2. Flowchart Pemberian Pakan Manual

Flowchart pemberian pakan manual menjelaskan mekanisme sistem ketika pengguna secara langsung memberikan perintah feeding melalui Telegram Bot. Proses dimulai saat sistem menerima pesan dari pengguna dan mengidentifikasi perintah tersebut sebagai perintah pemberian pakan manual.

Setelah perintah dikenali, sistem langsung memicu proses feeding tanpa menunggu jadwal otomatis. Pendekatan ini memberikan fleksibilitas kepada pengguna untuk menyesuaikan pemberian pakan sesuai kondisi lapangan, yang merupakan salah satu keunggulan sistem IoT berbasis kontrol jarak jauh [3], [6].

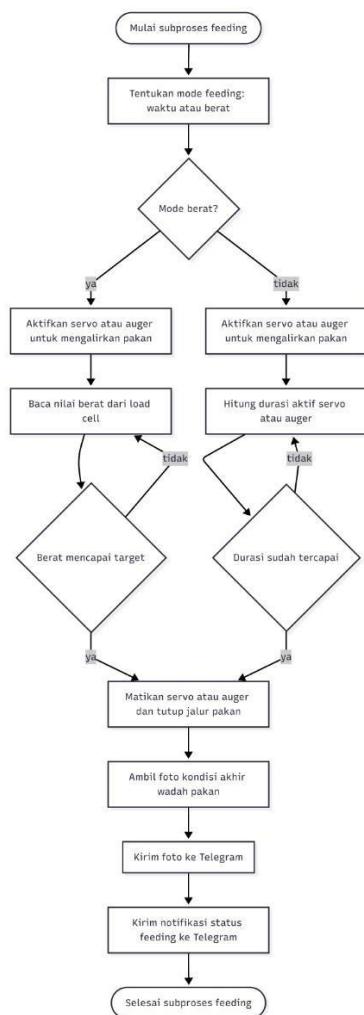


Gambar 1. 7 Flowchart pemberian pakan manual

1.6.3. Flowchart subproses feeding

Flowchart subproses feeding menggambarkan proses inti sistem dalam mengeksekusi pemberian pakan ayam setelah dipicu oleh workflow utama. Subproses ini bertanggung jawab terhadap pengaturan jumlah pakan dan pengendalian aktuator auger spiral.

Sistem dapat menjalankan pemberian pakan berbasis waktu atau berbasis berat sesuai konfigurasi yang ditetapkan. Pendekatan ini umum digunakan pada sistem pemberi pakan otomatis karena mampu meningkatkan konsistensi pemberian pakan serta mengurangi ketergantungan pada tenaga kerja manual [4], [5].

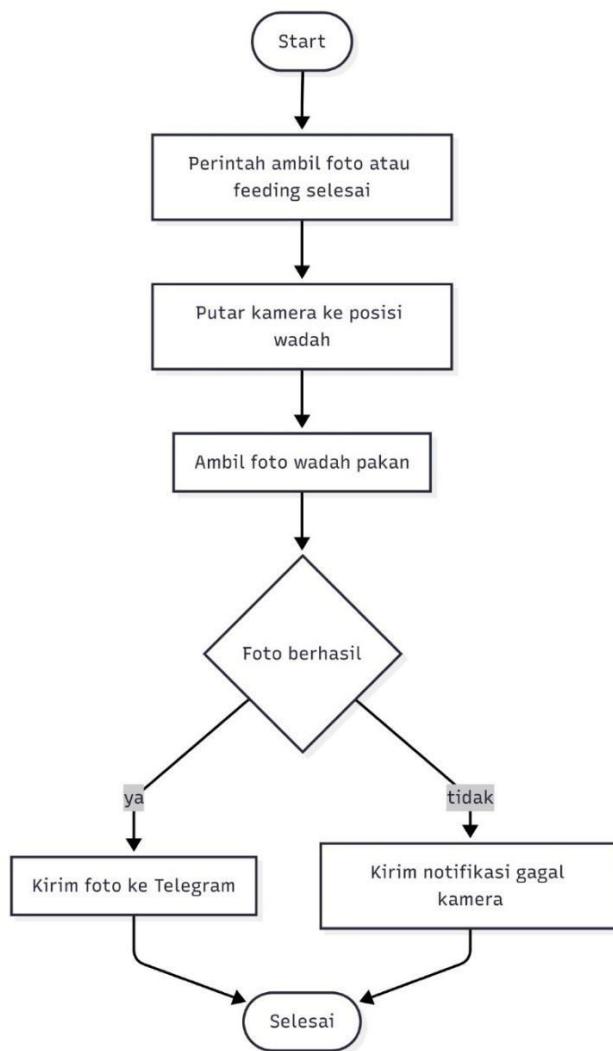


Gambar 1. 8 Flowchart Subproses Feeding

1.6.4. Flowchart Pemantauan Menggunakan Kamera

Flowchart pemantauan menggunakan kamera menjelaskan mekanisme sistem dalam mengambil dan mengirimkan gambar kondisi wadah pakan kepada pengguna melalui Telegram Bot. Proses ini dapat dipicu setelah proses feeding selesai maupun melalui perintah manual pengguna.

Pemantauan visual berfungsi sebagai umpan balik untuk memastikan bahwa proses pemberian pakan telah berjalan dengan baik. Pendekatan monitoring visual jarak jauh ini banyak diterapkan pada sistem peternakan berbasis IoT untuk meningkatkan kepercayaan pengguna terhadap sistem otomatis [2], [10].

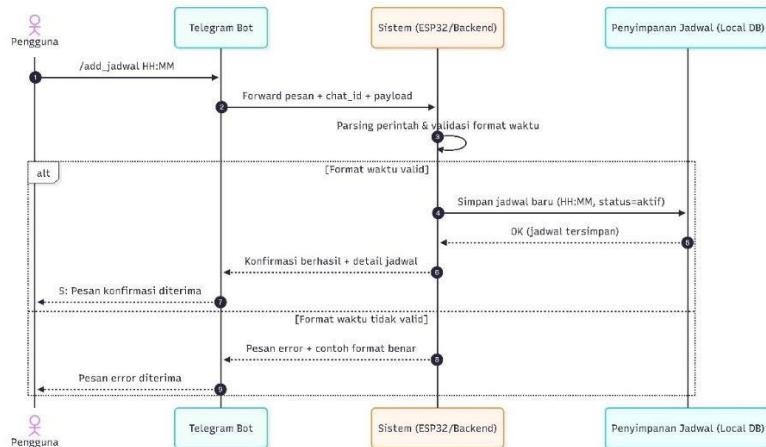


Gambar 1. 9 Flowchart pemantauan menggunakan kamera

1.6.5. Sequence Diagram Manajemen Jadwal

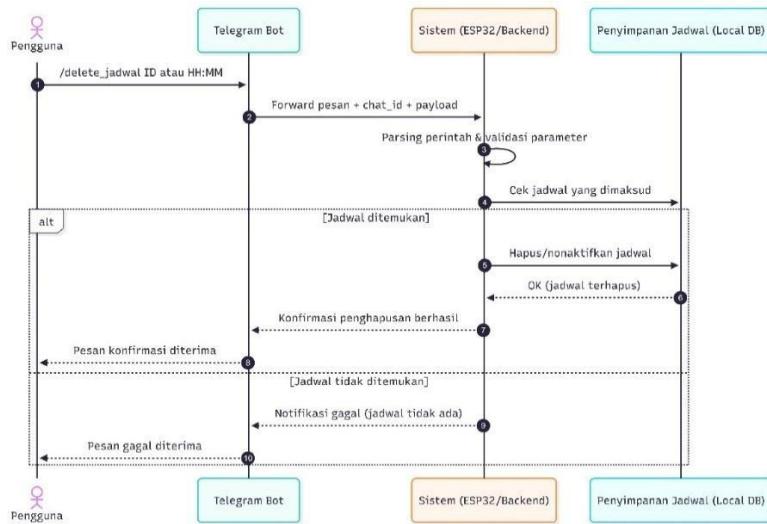
Sequence diagram digunakan untuk menggambarkan interaksi antara pengguna, Telegram Bot, dan sistem dalam proses manajemen jadwal pemberian pakan. Diagram ini menunjukkan alur pertukaran pesan secara kronologis.

1. Sequence diagram tambah jadwal menggambarkan proses penambahan jadwal baru dan penyimpanan data ke memori lokal sistem.



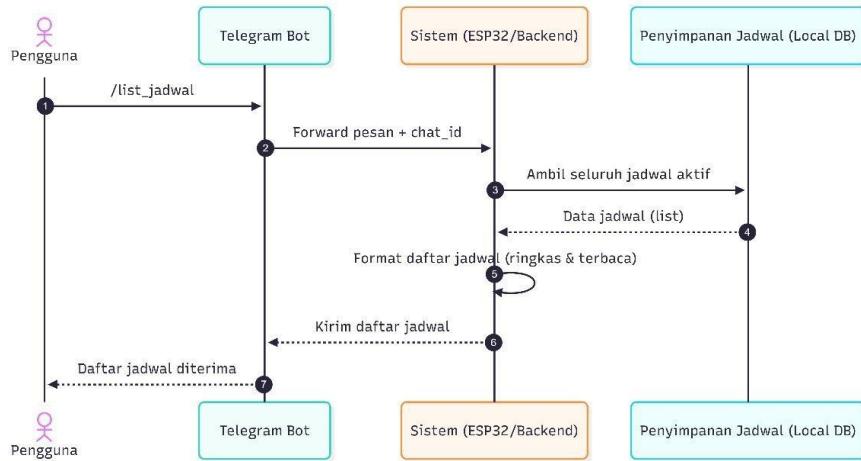
Gambar 1. 10 Sequence Diagram Tambah Jadwal

2. Sequence diagram hapus jadwal menunjukkan mekanisme penghapusan jadwal yang telah tersimpan untuk menjaga konsistensi data sistem.



Gambar 1. 11 Sequence Diagram Hapus Jadwal

3. Sequence diagram lihat jadwal menjelaskan proses permintaan dan penyajian daftar jadwal kepada pengguna melalui Telegram Bot.

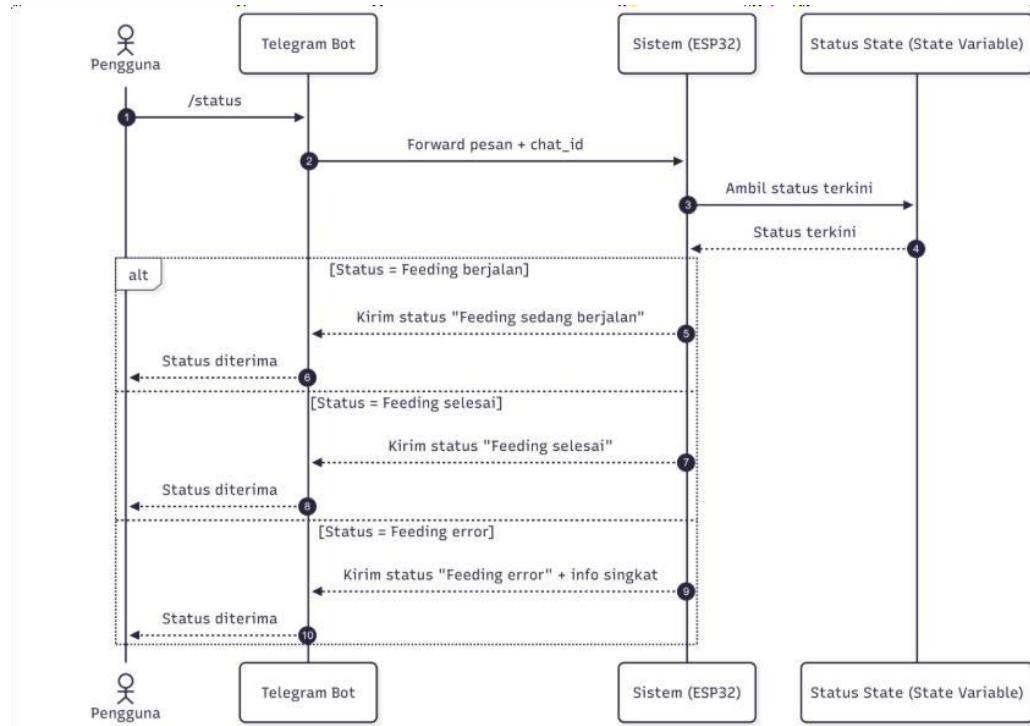


Gambar 1. 12 Sequence Diagram Lihat Jadwal

1.6.6. Sequence Diagram Permintaan Status Sistem

Sequence diagram permintaan status sistem menggambarkan mekanisme ketika pengguna meminta informasi kondisi terkini sistem melalui Telegram Bot. Sistem mengambil informasi status dari variabel internal (*state variable*) dan mengirimkan respon yang sesuai tanpa memicu proses feeding baru.

Pendekatan ini mendukung mekanisme monitoring sesuai permintaan yang efisien dan aman, serta umum diterapkan pada sistem IoT berbasis komunikasi pesan instan [6], [9].



Gambar 1. 13 Sequence Diagram Permintaan Status Sistem

1.7. Gambaran Source Code Modul Hardware

Source code modul hardware merupakan kode program yang digunakan untuk mengendalikan sistem pemberi pakan ayam kampung otomatis berbasis IoT dengan mekanisme auger spiral. Kode program ini ditanamkan pada mikrokontroler ESP32/NodeMCU yang berfungsi sebagai pusat kendali sistem, pembacaan sensor, pengendalian aktuator, serta komunikasi dengan Telegram Bot sebagai antarmuka pengguna.

Secara umum, source code modul hardware terdiri dari beberapa proses utama sebagai berikut:

1. Deklarasi library yang dibutuhkan untuk mendukung koneksi WiFi, komunikasi Telegram Bot, pembacaan sensor berat (load cell), pengendalian motor auger spiral, servo penutup jalur pakan, serta pengoperasian kamera.
2. Pendefinisian kredensial WiFi dan Telegram Bot agar sistem dapat terhubung ke jaringan internet dan berkomunikasi dengan pengguna.
3. Pendefinisian pin sensor dan aktuator, meliputi load cell, motor DC auger spiral, servo, dan kamera.
4. Deklarasi objek sensor dan aktuator untuk mempermudah pengolahan data dan pengendalian perangkat keras.

5. Fungsi `setup()`, yang digunakan untuk inisialisasi komunikasi serial, koneksi WiFi, inisialisasi Telegram Bot, sensor, aktuator, dan kamera.
6. Fungsi `loop()`, yang berisi proses utama sistem berupa pengecekan jadwal dan perintah manual, pembacaan berat pakan, pengendalian auger dan servo, pengambilan gambar kondisi wadah pakan, serta pengiriman notifikasi ke pengguna melalui Telegram Bot.

1.8. Petunjuk Instalasi Perangkat Keras

Berikut merupakan tahapan instalasi modul perangkat keras pada sistem pemberi pakan ayam kampung otomatis:

1. Letakkan modul perangkat keras (mikrokontroler, driver motor, dan rangkaian kontrol) pada lokasi yang aman dan terlindung dari air serta debu berlebih di area kandang.
2. Pasang mekanisme auger spiral pada wadah penyimpanan pakan dan pastikan dapat berputar dengan lancar.
3. Pasang sensor berat (load cell) pada dudukan wadah pakan dengan posisi stabil agar hasil pengukuran akurat.
4. Pasang servo motor pada jalur keluaran pakan sebagai mekanisme buka-tutup aliran pakan.
5. Pasang kamera 180 derajat pada posisi yang dapat menangkap kondisi wadah pakan secara menyeluruh.
6. Hubungkan seluruh sensor dan aktuator ke mikrokontroler sesuai dengan diagram skematik sistem.
7. Sambungkan modul perangkat keras ke catu daya untuk menyalakan sistem dan pastikan sistem berhasil terhubung ke jaringan WiFi.

DAFTAR PUSTAKA

- [1] T. Setiadi and T. I. Arifiandi, “Development of an IoT-Enabled Automatic Poultry Feeder and Cage Temperature Monitoring System Using Microcontroller Technology,” *TIERS Journal*, vol. 5, no. 2, 2024, doi:10.38043/tiers.v5i2.5631.
- [2] A. K. Nalendra and H. P. Waspada, “Smart Poultry Farming: A Mobile-Based IoT System for Real-Time Broiler Monitoring and Management,” *International Journal of Electronics and Communications System*, vol. 5, no. 1, 2025, doi:10.24042/ijecs.v5i1.27622.
- [3] E. E. Prasetya, N. Fadillah, S. Satriyo, R. Rusman, dan M. Y. Tharam, “Sistem Monitoring dan Smart Farming untuk Peternakan Anak Ayam Berbasis Internet of Things (IoT),” *ENTRIES Journal*, vol. 3, no. 2, 2023, doi: 10.58466/entries.v3i2.1632.
- [4] I. Gunawan, H. Ahmadi, and M. R. Said, “Rancang Bangun Sistem Monitoring dan Pemberi Pakan Otomatis Ayam Anakan Berbasis Internet of Things (IoT),” *Infotek: Jurnal Informatika dan Teknologi*, vol. 4, no. 2, 2021. doi: 10.29408/jit.v4i2.3562.
- [5] E. B. Agustina, D. A. Rachman, R. Nofillah, dan L. I. Fitri, “Desain Sistem Smart Feeder Ayam Berbasis Internet of Things (IoT) Guna Mencegah Keterlambatan Pemberian Pakan,” *Jurnal Fisika Unand*, vol. 13, no. 2, pp. 297–302, Mar. 2024, doi: 10.25077/jfu.13.2.297-302.2024.
- [6] S. Asyari, J. W. Leksono, N. Kholis, dan E. Indahwati, “Rancang Bangun Sistem Pakan dan Pengontrol Suhu Otomatis Berbasis Internet of Things (IoT),” *Elconika: Jurnal Teknik Elektronika dan Informatika*, vol. 2, no. 1, 2023, doi: 10.33752/elconika.v2i1.5630.
- [7] L. Perdanasaki, B. Etikasari, T. D. Puspitasari, dan R. Ayuninghemi, “Implementasi Internet of Things pada Otomasi Pemberian Pakan Ayam Ras Petelur,” *JTIM: Jurnal Teknologi Informasi dan Multimedia*, vol. 6, no. 3, Nov. 2024, doi: 10.35746/jtim.v6i3.452.
- [8] J. D. Susatyono and Y. Fitrianto, “Sistem Monitoring Kualitas Udara dan Otomatisasi Pemberian Pakan Ayam Berbasis IoT,” *KREA-TIF: Jurnal Teknik*

- Informatika*, vol. 9, no. 2, pp. 1–10, Nov. 2021, doi: 10.32832/kreatif.v9i2.5650.
- [9] N. A. Kesvika, Y. Dimas, R. Fadli, B. Setiyono, dan I. Santoso, “Perancangan Pengendalian Sistem Otomatisasi dan Monitoring yang Efektif pada Autofeeds Smart Farming Kandang Ayam untuk Pakan dan Minum secara Real Time,” *Transient: Jurnal Ilmiah Teknik Elektro*, vol. 14, no. 3, pp. 112–119, Sep. 2025, doi: 10.14710/transient.v14i3.112-119.
- [10] A. Bootwong, S. Suttibak, W. Kaewka, S. Somwong, N. Bootthanu, N. Bhumiphan, dan K. Tantichukiad, “Evaluation and Implementation of an Automatic Feeding System for Jariya Broiler Farm, Nakhon-Ratchasima, Thailand,” *Brazilian Journal of Poultry Science*, vol. 27, no. 3, 2025, doi: 10.1590/1806-9061-2025-2103.
- [11] F. Ariani, A. Y. Vandika, and H. Widjaya, “Implementasi alat pemberi pakan ternak menggunakan IoT untuk otomatisasi pemberian pakan ternak,” *Explore: Jurnal Sistem Informasi dan Telematika*, vol. 10, no. 2, 2019, doi: 10.36448/josit.v10i2.1315.
- [12] S. H. G. Vebi *et al.*, “Alat pemberi pakan ayam kampung otomatis dengan sistem monitoring dan kontrol berbasis Internet of Thing,” *Jurnal Muara Sains, Teknologi, Kedokteran dan Ilmu Kesehatan*, vol. 8, no. 1, 2024, doi: 10.24912/jmstkk.v8i1.20452.
- [13] Fritzing GmbH, “Fritzing – Open Source Hardware Initiative,” 2025. [Online]. Available: <https://fritzing.org>
- [14] n8n GmbH, “n8n – Workflow Automation Tool,” 2025. [Online]. Available: <https://n8n.io>

LAMPIRAN

Source code :

```
/*
=====
```

SIMULASI FSM FEEDING SYSTEM

Setiap gate memiliki sensor berat sendiri

Target: ±600 gram per gate

```
*/
=====
```

```
enum State {
```

```
    IDLE,
```

```
    INIT_SYSTEM,
```

```
    START_FEEDING,
```

```
    DISPENSE_GATE1,
```

```
    DISPENSE_GATE2,
```

```
    DISPENSE_GATE3,
```

```
    CLOSEGATE,
```

```
    ERROR_STATE
```

```
};
```

```
State currentState = INIT_SYSTEM; // Mulai dari INIT_SYSTEM
```

```
// ----- Parameter -----
```

```
const int GATE_COUNT = 3;
```

```
const int TARGET_WEIGHT = 600; // gram
```

```
const int WEIGHT_STEP_MIN = 40;
```

```
const int WEIGHT_STEP_MAX = 80;
```

```

// ----- Variabel -----
int gateWeight[GATE_COUNT] = {0, 0, 0};
bool firstRun = true; // Flag untuk run pertama kali

// ----- Helper -----
void logMsg(const String& tag, const String& msg) {
    Serial.println("[ " + tag + " ] " + msg);
}

void printSeparator() {
    Serial.println("=====");
}

// ----- Aktuator (Simulasi) -----
void augerOn() { logMsg("ACTUATOR", "Auger motor ON"); }
void augerOff() { logMsg("ACTUATOR", "Auger motor OFF"); }

void gateOpen(int i) {
    logMsg("GATE", "Gate " + String(i + 1) + " OPEN");
}

void gateClose(int i) {
    logMsg("GATE", "Gate " + String(i + 1) + " CLOSED");
}

// ----- Setup -----

```

```

void setup() {
    Serial.begin(115200);
    delay(2000); // Beri waktu lebih lama untuk serial monitor terbuka

    printSeparator();
    logMsg("SYSTEM", "SIMULASI FSM FEEDING SYSTEM");
    logMsg("SYSTEM", "Target: 600 g per gate");
    printSeparator();

    delay(1000); // Delay tambahan sebelum mulai
}

// ----- FSM -----
void fsmTick() {

    switch (currentState) {

        case IDLE:
            logMsg("STATE", "IDLE");
            logMsg("SYSTEM", "Sistem siap, menunggu trigger..."); 
            delay(2000);
            printSeparator();
            currentState = INIT_SYSTEM;
            break;

        case INIT_SYSTEM:
            logMsg("STATE", "INIT_SYSTEM");
            logMsg("SYSTEM", "Melakukan pengecekan sistem..."); 
            delay(1000);
    }
}

```

```

logMsg("CHECK", "Sensor berat: OK");
delay(500);
logMsg("CHECK", "Motor auger: OK");
delay(500);
logMsg("CHECK", "Servo gate: OK");
delay(500);

// Reset berat
for (int i = 0; i < GATE_COUNT; i++) {
    gateWeight[i] = 0;
}

logMsg("SYSTEM", "Inisialisasi selesai");
printSeparator();
currentState = START_FEEDING;
break;

case START_FEEDING:
logMsg("STATE", "START_FEEDING");
logMsg("SYSTEM", "Memulai proses feeding...");
delay(1000);

// Buka semua gate
for (int i = 0; i < GATE_COUNT; i++) {
    gateOpen(i);
    delay(500);
}

augerOn();
delay(1000);

```

```

printSeparator();

currentState = DISPENSE_GATE1;
break;

case DISPENSE_GATE1: {
    logMsg("STATE", "DISPENSE");

    // Update berat Gate 1
    gateWeight[0] += random(WEIGHT_STEP_MIN, WEIGHT_STEP_MAX);
    logMsg("SENSOR", "Berat Gate 1: " + String(gateWeight[0]) + " g");

    delay(800);

    // Cek apakah Gate 1 mencapai target
    if (gateWeight[0] >= TARGET_WEIGHT) {
        gateClose(0);
        logMsg("EVENT", "Gate 1 ditutup (target 600 g tercapai)");
        printSeparator();
        delay(1000);
        currentState = DISPENSE_GATE2;
    }
}

break;
}

case DISPENSE_GATE2: {
    logMsg("STATE", "DISPENSE");
}

```

```
// Update berat Gate 2
gateWeight[1] += random(WEIGHT_STEP_MIN, WEIGHT_STEP_MAX);
logMsg("SENSOR", "Berat Gate 2: " + String(gateWeight[1]) + " g");

delay(800);

// Cek apakah Gate 2 mencapai target
if (gateWeight[1] >= TARGET_WEIGHT) {
    gateClose(1);
    logMsg("EVENT", "Gate 2 ditutup (target 600 g tercapai)");
    printSeparator();
    delay(1000);
    currentState = DISPENSE_GATE3;
}

break;
}

case DISPENSE_GATE3: {
    logMsg("STATE", "DISPENSE");

    // Update berat Gate 3
    gateWeight[2] += random(WEIGHT_STEP_MIN, WEIGHT_STEP_MAX);
    logMsg("SENSOR", "Berat Gate 3: " + String(gateWeight[2]) + " g");
}
```

```
delay(800);

// Cek apakah Gate 3 mencapai target
if (gateWeight[2] >= TARGET_WEIGHT) {
    gateClose(2);
    logMsg("EVENT", "Gate 3 ditutup (target 600 g tercapai)");
    printSeparator();
    delay(1000);
    currentState = CLOSEGATE;
}

break;
}

case CLOSEGATE:
    logMsg("STATE", "CLOSEGATE");
    augerOff();
    delay(1000);
    logMsg("SYSTEM", "Feeding selesai");
    printSeparator();

    currentState = IDLE;
    delay(3000);
    break;

case ERROR_STATE:
    augerOff();
    logMsg("ERROR", "Kesalahan sistem");
    currentState = IDLE;
```

```

        break;

    }

}

// ----- Loop -----
void loop() {
    fsmTick();
}

```

Output :

```

[STATE] INIT_SYSTEM
[SYSTEM] Melakukan pengecekan sistem...
[CHECK] Sensor berat: OK
[CHECK] Motor auger: OK
[CHECK] Servo gate: OK
[SYSTEM] Inisialisasi selesai
=====
[STATE] START_FEEDING
[SYSTEM] Memulai proses feeding...
[GATE] Gate 1 OPEN
[GATE] Gate 2 OPEN
[GATE] Gate 3 OPEN
[ACTUATOR] Auger motor ON
=====
[STATE] DISPENSE
[SENSOR] Berat Gate 1: 69 g
[STATE] DISPENSE
[SENSOR] Berat Gate 1: 127 g
[STATE] DISPENSE
[SENSOR] Berat Gate 1: 196 g
[STATE] DISPENSE
[SENSOR] Berat Gate 1: 241 g
[STATE] DISPENSE
[SENSOR] Berat Gate 1: 294 g
[STATE] DISPENSE
[SENSOR] Berat Gate 1: 367 g
[STATE] DISPENSE
[SENSOR] Berat Gate 1: 439 g
[STATE] DISPENSE
[SENSOR] Berat Gate 1: 508 g
[STATE] DISPENSE
[SENSOR] Berat Gate 1: 556 g

```

```
[STATE] DISPENSE
[SENSOR] Berat Gate 1: 609 g
[GATE] Gate 1 CLOSED
[EVENT] Gate 1 ditutup (target 600 g tercapai)
=====
[STATE] DISPENSE
[SENSOR] Berat Gate 2: 43 g
[STATE] DISPENSE
[SENSOR] Berat Gate 2: 86 g
[STATE] DISPENSE
[SENSOR] Berat Gate 2: 129 g
[STATE] DISPENSE
[SENSOR] Berat Gate 2: 185 g
[STATE] DISPENSE
[SENSOR] Berat Gate 2: 245 g
[STATE] DISPENSE
[SENSOR] Berat Gate 2: 290 g
[STATE] DISPENSE
[SENSOR] Berat Gate 2: 355 g
[STATE] DISPENSE
[SENSOR] Berat Gate 2: 425 g
[STATE] DISPENSE
[SENSOR] Berat Gate 2: 482 g
[STATE] DISPENSE
[SENSOR] Berat Gate 2: 525 g
[STATE] DISPENSE
[SENSOR] Berat Gate 2: 603 g
[GATE] Gate 2 CLOSED
```

```
[SENSOR] Berat Gate 2: 603 g
[GATE] Gate 2 CLOSED
[EVENT] Gate 2 ditutup (target 600 g tercapai)
=====
[STATE] DISPENSE
[SENSOR] Berat Gate 3: 66 g
[STATE] DISPENSE
[SENSOR] Berat Gate 3: 127 g
[STATE] DISPENSE
[SENSOR] Berat Gate 3: 200 g
[STATE] DISPENSE
[SENSOR] Berat Gate 3: 257 g
[STATE] DISPENSE
[SENSOR] Berat Gate 3: 313 g
[STATE] DISPENSE
[SENSOR] Berat Gate 3: 380 g
[STATE] DISPENSE
[SENSOR] Berat Gate 3: 458 g
[STATE] DISPENSE
[SENSOR] Berat Gate 3: 534 g
[STATE] DISPENSE
[SENSOR] Berat Gate 3: 585 g
[STATE] DISPENSE
[SENSOR] Berat Gate 3: 649 g
[GATE] Gate 3 CLOSED
[EVENT] Gate 3 ditutup (target 600 g tercapai)
=====
[STATE] CLOSEGATE
[ACTUATOR] Auger motor OFF
```

```
=====
[STATE] IDLE
[SYSTEM] Sistem siap, menunggu trigger...
=====
```