

HALAMAN SAMPUL LAPORAN PROYEK AKHIR LAPORAN PROYEK AKHIR II



LAPORAN PENGEMBANGAN PRODUK PROYEK AKHIR II

**eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan
Kontrol via Telegram**

Disusun Oleh:

13323010	:	Calista Yosephin Panjaitan
13323043	:	Chesia P.M.Silaen
13323035	:	Glen Rafael Situmorang
13323045	:	Dealova Zevanya Manurung

**PROGRAM STUDI DIII TEKNOLOGI KOMPUTER
FAKULTAS VOKASI
INSTITUT TEKNOLOGI DEL
2025**

HALAMAN JUDUL LAPORAN PROYEK AKHIR LAPORAN PROYEK AKHIR II



LAPORAN PENGEMBANGAN PRODUK PROYEK AKHIR II

**eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT
dengan Kontrol via Telegram**

Disusun Oleh:

13323010	:	Calista Yosephin Panjaitan
13323043	:	Chesia P.M.Silaen
13323035	:	Glen Rafael Situmorang
13323045	:	Dealova Zevanya Manurung

**PROGRAM STUDI DIII TEKNOLOGI KOMPUTER
FAKULTAS VOKASI
INSTITUT TEKNOLOGI DEL
2025**

DAFTAR ISI

HALAMAN SAMPUL LAPORAN PROYEK AKHIR	1
HALAMAN JUDUL LAPORAN PROYEK AKHIR	2
DAFTAR TABEL	6
DAFTAR GAMBAR.....	7
BAB I PRODUCT REQUIREMENT SPECIFICATION.....	8
 1.1 PENDAHULUAN	8
 1.1.1 Tujuan Penulisan Dokumen.....	8
 1.1.2 Latar Belakang Produk.....	8
 1.1.3 Tujuan Produk	9
 1.1.4 Ruang Lingkup Produk.....	10
 1.1.5 Manfaat Produk	11
 1.1.6 Definisi dan Singkatan.....	12
 1.1.7 Referensi	13
 1.2 DESKRIPSI UMUM PRODUK	13
 1.2.1 Permasalahan dan Persoalan	14
 1.2.2 Produk yang menjadi Inspirasi	14
 1.2.3 Produk yang akan dibangun.....	15
 1.2.4 Konteks	16
 1.2.5 Deskripsi Kebutuhan Produk	17
 1.2.6 Environment Hardware dan Software.....	18
 1.2.7 Metodologi dan Tools Pengembangan	20
 BAB II PROJECT PLANNING (PP).....	22
 2.1 PENDAHULUAN	22
 2.2 DESKRIPSI PROYEK	22
 2.2.1 Project Organization.....	23
 2.2.2 Work Breakdown Structure (WBS).....	23
 2.2.3 Budget	24
 2.2.4 Tools	26
 2.2.5 Resiko dan Hambatan	26
 BAB III PRODUCT DESIGN (PD) (DESAIN PENGEMBANGAN PRODUK)	29
 3.1 PENDAHULUAN	29

3.2 DESKRIPSI PRODUK DESIGN	30
3.2.1 Proses Bisnis Target System.....	30
3.2.2 Arsitektur Sistem [Hardware]	31
3.2.3 User Interface Layout Software	32
3.2.4 Desain Rangkaian Elektronik/Skematik Desain dan Desain PCB [Hardware]	34
3.2.5 Mekanisme Komunikasi Data [Hardware].....	36
3.2.5.1 Panel Surya.....	37
3.2.5.2 Drum	38
3.2.5.3 Rangka	39
3.2.5.4 Pipa.....	40
3.2.5.5 Box.....	41
3.2.5.6 Turbin	41
3.2.5.7 Roda	42
3.2.5.8 Desain Keseluruhan	43
4.1 PENDAHULUAN	45
4.2 DESKRIPSI	45
4.2.1 Prinsip Implementasi	45
4.2.2 Lingkungan Pengembangan Terintegrasi (IDE).....	46
4.2.3 Implementasi Desain PCB	46
4.2.4 Implementasi Desain Fisik/Desain Case Hardware	48
4.2.5 Desain Fitur Notifikasi Telegram	50
4.2.6 Integrasi Hardware dan Software	51
BAB V PRODUCT TESTING (PT)	65
5.1 PENDAHULUAN	65
5.2 DESKRIPSI PENGUJIAN.....	65
5.2.1 Pengujian motor DC	66
5.2.2 PENGUJIAN KETAHANAN BATERAI	67
5.2.3 Pengujian Sensor Ultrasonik.....	68
5.2.4 Pengujian Relay Module	69
5.2.5 Pengujian Modul Step-Down	69
5.2.6 Pengujian Turbin (Mekanisme Distribusi Pakan)	69
5.2.7 Pengujian Mikrokontroler dan Algoritma Kontrol	70
5.2.8 Pengujian Servo Motor (Buka/Tutup Saluran).....	70
5.2.9 Pengujian Panel Surya (MAYSUN MS120M-36 – 120W)	71
5.2.10 TOOLS PENGUJIAN.....	71

5.3 METODE PENGUJIAN.....	72
5.3.1 Pengujian Fungsional	72
5.3.2 Pengujian Non Fungsional	72
5.3.3 Pengujian Hardware.....	72
5.3.4 Pengujian Integrasi Software dan Hardware.....	72
5.3.5 Pengujian Prototipe (Prototype Testing)	73
BAB VI PRODUCT RELEASE (PR).....	75
6.1 PENDAHULUAN	75
6.2 DESKRIPSI.....	75
6.3 DAYA GUNA PRODUK.....	76
6.4 POSTER PRODUK	76
6.5 PERILISAN PRODUK (OPSIONAL)	78
6.5.1. TUJUAN PAMERAN PRODUK.....	78
6.6 EVALUASI DAN KEKURANGAN PRODUK.....	78
DAFTAR PUSTAKA.....	80

DAFTAR TABEL

Tabel 1. 1 Daftar Definisi dan Singkatan.....	12
Tabel 1. 2 Environment Hardware	18
Tabel 1. 3 Enviroment sofware	19
Tabel 2. 1 Budget yang diperlukan	25
Tabel 2. 2 Tabel	26
Tabel 2. 3 Resiko Utama dan Stategi Mitigasi.....	27
Tabel 5. 1 Hasil pengujian kecepatan Motor DC.....	67
Tabel 5. 2 Pengujian motor dc distribusi pakan.....	67
Tabel 5. 3 Hasil Pengujian Ketahanan Baterai	Error! Bookmark not defined.
Tabel 5. 4 Pengujian Sensor Ultrasonik.....	68
Tabel 5. 5 Pengujian Relay Module	69
Tabel 5. 6 Pengujian Modul Step-Down.....	69
Tabel 5. 7 Pengujian Turbin (Mekanisme Distribusi Pakan)	69
Tabel 5. 8 Pengujian Mikrokontroler dan Algoritma Kontrol	70
Tabel 5. 9 Pengujian Servo Motor (Buka/Tutup Saluran)	70

DAFTAR GAMBAR

Gambar 2. 1 Project Oraganization.....	23
Gambar 2. 2 Work Breakdown Structure.....	24
Gambar 2. 3 Work Breakdown structure Part 2.....	24
Gambar 3. 1 Proses Bisnis Target System	30
Gambar 3. 2 Blok Diagram Arsitektur Sistem.....	32
Gambar 3. 3 Desain rangkaian Elektronik	34
Gambar 3. 4 Desain rangkaian Skematik.....	35
Gambar 3. 5 Desain PCB	36
Gambar 3. 6 Panel Surya	37
Gambar 3. 7 Drum	38
Gambar 3. 8 Rangka	39
Gambar 3. 9 Pipa.....	40
Gambar 3. 10 Box	41
Gambar 3. 11 Turbin.....	42
Gambar 3. 12 Roda	43
Gambar 3. 13 Desain Keseluruhan	44
Gambar 4. 1 Implementasi pada PCB	47
Gambar 4. 2 Implementasi Desain Fisik.....	49
Gambar 4. 3.....	50
Gambar 6. 1 Poster.....	77

BAB I

PRODUCT REQUIREMENT SPECIFICATION

1.1 PENDAHULUAN

Dokumen ini disusun sebagai pedoman utama dalam pengembangan prototipe sistem eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram. Pembuatan dokumen ini mengikuti siklus hidup pengembangan produk yang terstruktur, mulai dari tahap perencanaan hingga evaluasi, untuk memastikan proses pengembangan berjalan secara sistematis dan efisien. Tujuan utama dari penyusunan dokumen ini adalah agar perancangan dan implementasi *eFeeder* dapat dilakukan sesuai dengan kebutuhan pengguna serta mendukung terciptanya solusi otomatisasi yang andal dan terintegrasi.

1.1.1 Tujuan Penulisan Dokumen

Dokumen ini di buat dengan tujuan sebagai berikut:

1. Menjelaskan kebutuhan sistem *eFeeder* sebagai dasar dalam membuat prototipe agar pengembangannya terarah dan jelas.
2. Memberikan panduan bagi tim pengembang dalam membuat, merakit, dan menguji sistem, baik bagian alat maupun aplikasinya.
3. Membantu semua pihak yang terlibat, seperti tim teknis, pengguna, dan pemilik proyek, agar punya pemahaman yang sama tentang tujuan dan cara kerja sistem.
4. Memastikan semua tahapan dalam proses pengembangan dilakukan secara berurutan dan rapi, dari perencanaan sampai evaluasi.
5. Menjadi acuan untuk mengecek apakah sistem yang dibuat sudah sesuai dengan kebutuhan dan bisa berjalan dengan baik.

1.1.2 Latar Belakang Produk

Dalam kegiatan budidaya ternak unggas seperti ayam dan bebek, serta budidaya ikan air tawar di keramba, pemberian pakan merupakan salah satu aktivitas paling penting dan rutin yang secara langsung memengaruhi pertumbuhan, kesehatan, serta produktivitas hewan ternak. Namun, dalam praktiknya, banyak peternak menghadapi kendala dalam menjaga konsistensi waktu dan jumlah pemberian pakan akibat keterbatasan tenaga kerja, jarak lokasi, atau aktivitas lain yang menyita waktu. Ketergantungan pada proses manual juga meningkatkan risiko

keterlambatan atau ketidakteraturan dalam pemberian pakan, yang pada akhirnya dapat berdampak pada penurunan hasil panen.

Jika permasalahan ini tidak segera ditangani, dampaknya bisa cukup serius. Ketidakteraturan dalam pemberian pakan dapat menyebabkan stres pada hewan, menghambat pertumbuhan, meningkatkan risiko penyakit, dan menyebabkan pemborosan pakan. Dalam skala besar, hal ini bisa berujung pada kerugian finansial bagi peternak dan menurunkan efisiensi operasional. Terlebih lagi, bagi peternak yang mengelola keramba di area yang sulit dijangkau setiap saat, kontrol manual menjadi tidak efisien dan tidak praktis.

Melihat tantangan tersebut, muncul peluang untuk mengembangkan solusi berbasis teknologi yang dapat membantu peternak dalam mengotomatisasi proses pemberian pakan. Dengan memanfaatkan *Internet of Things* (IoT) dan teknologi komunikasi digital, sistem pemberi pakan otomatis dapat dirancang untuk menjadwalkan dan mengontrol distribusi pakan secara akurat dan konsisten, bahkan dari jarak jauh. Hal ini akan sangat berguna khususnya untuk peternakan unggas dan perikanan air tawar yang dikelola di lokasi seperti keramba terapung, yang tidak selalu mudah diakses.

Oleh karena itu, dikembangkanlah Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram sebagai solusi yang dapat diandalkan untuk meningkatkan efisiensi dan *efektivitas* pemberian pakan. Sistem ini memungkinkan pengguna untuk mengatur waktu dan jumlah pakan melalui aplikasi Telegram, memberikan kemudahan pemantauan dan pengendalian kapan saja dan di mana saja. Diharapkan, dengan adanya sistem ini, para peternak dapat mengelola usaha mereka dengan lebih baik, mengurangi risiko human error, serta meningkatkan produktivitas secara keseluruhan.

1.1.3 Tujuan Produk

Tujuan dari pengembangan produk eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram ini adalah:

1. Memberi Pakan dengan sistem otomatis yang dapat memberikan pakan ternak sesuai jadwal yang ditentukan.
2. Mengurangi ketergantungan pada pemberian pakan manual untuk meningkatkan peternak dalam pemberian pakan.

3. Menyediakan sistem pemantauan stok pakan dan notifikasi saat pakan masih ada atau hampir habis, yang bertujuan untuk menghindari kelupaan peternak akan pakan ternaknya.
4. Alat ini juga sudah dilengkapi pemberian pakan manual yang dapat digunakan kapan saja, jika kita ingin memberikannya diluar jadwal yang sudah kita tentukan.

1.1.4 Ruang Lingkup Produk

Ruang lingkup dalam pengembangan eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram mencakup berbagai aspek yang bertujuan untuk mempermudah peternak dalam memberikan pakan secara terjadwal. Berikut adalah cakupan utama dari sistem yang akan dikembangkan:

1. Sistem otomatis pemberian pakan.

Sistem ini dirancang untuk mengatur jadwal pakan ternak secara otomatis, sehingga peternak tidak perlu melakukan pemberian pakan secara manual setiap hari. Dengan adanya fitur ini, pakan dapat disalurkan secara lebih teratur dan sesuai dengan kebutuhan ternak.

2. Notifikasi stok pakan

Sistem akan memiliki fitur notifikasi yang memberikan peringatan kepada peternak ketika stok pakan hampir habis. Dengan adanya fitur ini, peternak dapat segera melakukan pengisian ulang pakan sebelum benar-benar habis, sehingga pemberian pakan tetap berjalan lancar tanpa ada keterlambatan.

3. Mekanisme motor untuk mendistribusikan pakan

Sistem ini akan menggunakan motor sebagai komponen utama dalam mendistribusikan pakan ke area ternak. Motor akan bekerja berdasarkan jadwal yang telah diatur dan dapat dikonfigurasi untuk menyesuaikan jumlah serta frekuensi penyebaran pakan sesuai kebutuhan.

4. Pengguna utama system

Pengguna utama dari sistem ini adalah peternak yang ingin meningkatkan sistem dalam pemberian pakan ternak. Sistem ini dapat digunakan oleh peternak dengan skala kecil hingga besar yang ingin memastikan pakan ternak diberikan secara terjadwal tanpa perlu pengawasan langsung setiap waktu.

5. Batasan teknis system

Dari sisi teknis, sistem ini akan bekerja dengan mengandalkan konektivitas antara perangkat keras dan kontrol via telegram. Sistem harus memiliki daya yang cukup untuk menjalankan motor, sensor, serta modul komunikasi agar dapat berfungsi dengan optimal dalam kondisi lingkungan peternakan yang beragam.

6. Batasan non-teknis system

Dalam penggunaannya, sistem ini memerlukan kesiapan dan pemahaman dari peternak terkait cara pengoperasiannya. Peternak perlu mengetahui cara mengatur jadwal pakan, memahami bagaimana sistem memberikan notifikasi, serta memastikan perangkat tetap dalam kondisi baik agar dapat berfungsi secara optimal dan berkelanjutan.

1.1.5 Manfaat Produk

Manfaat yang diharapkan dari produk ini meliputi:

1. Teknis

Memastikan pemberian pakan dilakukan secara otomatis, tepat waktu, dan sesuai kebutuhan untuk ternak unggas konsumsi seperti ayam dan bebek (tidak termasuk burung peliharaan atau hias) serta ikan air tawar yang dibudidayakan di keramba atau kolam.

2. Ekonomis

Mengurangi biaya operasional dengan menekan ketergantungan pada tenaga kerja harian dalam aktivitas pemberian pakan, serta meminimalkan pemborosan pakan karena sistem yang lebih terkontrol.

3. Lingkungan

Mengoptimalkan distribusi pakan untuk menghindari limbah organik dari sisa pakan yang tidak termakan, sekaligus membantu menjaga kualitas air di keramba atau sanitasi di kandang.

4. Pengguna

Memberikan kemudahan bagi peternak ayam, bebek, dan pembudidaya ikan air tawar dalam mengontrol dan memantau proses pemberian pakan dari jarak jauh melalui aplikasi Telegram, tanpa perlu hadir langsung di lokasi.

1.1.6 Definisi dan Singkatan

Bagian ini memuat istilah teknis atau singkatan yang digunakan dalam dokumen, lengkap dengan arti atau kepanjangannya agar memudahkan pemahaman

Tabel 1. 1 Daftar Definisi dan Singkatan

No	Definisi/Singkatan	Keterangan
1	IoT	<i>Internet of Things</i> , konsep konektivitas perangkat melalui internet untuk pengawasan.
2	E-Feeder	Sistem otomatis untuk pemberian pakan ternak berdasarkan jadwal yang ditentukan.
3	SCC	<i>Solar Charge Controller</i> , alat pengatur pengisian daya dari panel surya ke baterai agar tidak terjadi overcharge atau overcharge atau overdischarge.
4	API	<i>Application Programming Interface</i> , antarmuka yang memungkinkan komunikasi antara sistem dengan aplikasi Telegram.
5	Trello	Aplikasi manajemen proyek berbasis papan dan kartu yang membantu tim mengatur dan memantau tugas secara visual.
6	Jira	<i>Platform</i> pelacakan proyek dan bug yang digunakan untuk merencanakan, memantau, dan merilis perangkat lunak secara terstruktur.
7	Asana	Alat kolaborasi kerja yang memudahkan tim dalam mengelola tugas, menetapkan tenggat waktu, dan melacak progres proyek.
8	Slack	Aplikasi komunikasi tim yang menyediakan fitur pesan instan, saluran diskusi, dan integrasi dengan berbagai alat kerja.
9	Tinkercad	Aplikasi desain 3D dan simulasi elektronik berbasis web yang mudah digunakan untuk pemula dan pembuat prototipe.

No	Definisi/Singkatan	Keterangan
10	Multisim	Perangkat lunak simulasi sirkuit elektronik yang digunakan untuk mendesain, menganalisis, dan menguji rangkaian secara virtual.

1.1.7 Referensi

Sumarna, A. (2019). Internet of Things (IoT) ala eFishery. Website Desa Ciburial. Diakses pada 15 Februari 2025, dari <https://ciburial.desa.id/internet-of-things-iot-ala-efishery/>

eFishery. (2023). Percepat Pengembangan Teknologi untuk Akuakultur, eFishery Rekrut Pakar IoT Indonesia. eFishery. Diakses pada 15 Februari 2025, dari <https://efishery.com/id/resources/percepat-pengembangan-teknologi-untuk-akuakulturefishery-rekrut-pakar-iot-indonesia/>

Universitas Medan Area. (2023). Tingkatkan Budi Daya Perairan, Teknologi Berbasis IoT Menjadi Solusi Efektif. Universitas Medan Area. Diakses pada 15 Februari 2025, dari <https://teknik.uma.ac.id/2023/09/29/tingkatkan-budi-daya-perairan-teknologi->

Media Startup. (2019). eFishery: Startup Perikanan Asal Kota Bandung. Media Startup. Diakses pada 15 Februari 2025, dari <https://www.mediasstartup.id/2019/02/21/efishery-startup-perikanan-asal-kota-bandung/>

Jurnal Teknologi Informasi dan Aplikasi (JTIA). (Tahun). Judul Artikel. JTIA Polinema. Diakses pada 15 Februari 2025, dari <http://jtia.polinema.ac.id/index.php/jtia/article/view/43>

1.2 DESKRIPSI UMUM PRODUK

Bagian ini menjelaskan gambaran umum dari produk eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram, termasuk fungsi utama sistem, komponen perangkat keras dan perangkat lunak yang digunakan, lingkungan operasional, serta batasan dan asumsi yang berlaku. Deskripsi umum ini disusun untuk memberikan pemahaman awal mengenai ruang lingkup produk, cara kerjanya secara garis besar, serta interaksi antara pengguna, perangkat, dan aplikasi dalam sistem. Dengan memahami bagian ini, pembaca dapat mengetahui struktur dasar sistem sebelum masuk ke detail teknis pada bagian berikutnya.

1.2.1 Permasalahan dan Persoalan

Peternak sering menghadapi kendala dalam pemberian pakan ternak secara terjadwal terutama saat mereka memiliki kesibukan lain atau harus meninggalkan ternak dalam waktu tertentu. Pemberian pakan yang tidak konsisten dapat mempengaruhi pertumbuhan ternak yang menyebabkan ketidakseimbangan nutrisi, dan berdampak pada produktivitas. Selain itu, metode manual yang dilakukan saat ini memerlukan pengawasan terus-menerus yang dapat menyita waktu dan tenaga.

Persoalan:

1. Keterlambatan pemberian pakan akibat kesibukan peternak yang tidak selalu bisa hadir tepat waktu, mengganggu pola makan dan pertumbuhan ternak.
2. Kesulitan memantau ketersediaan pakan, menyebabkan keterlambatan pengisian ulang dan berisiko menghambat pemberian makan.
3. Pemberian pakan manual memerlukan banyak waktu dan tenaga, kurang efisien bagi peternak dengan banyak ternak atau kesibukan lain.
4. Tidak adanya sistem otomatis yang membantu mengatur jadwal sehingga membuat peternak bergantung pada metode konvensional yang kurang efektif.

Untuk mengatasi permasalahan tersebut, dikembangkan eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram yang memungkinkan pemberian pakan secara terjadwal, sistem ini juga dirancang untuk mengatur jadwal pemberian pakan secara otomatis, memantau ketersediaan pakan secara *real-time*. Dengan adanya teknologi ini, peternak dapat menghemat tenaga dan waktu serta mengurangi risiko keterlambatan pemberian pakan.

1.2.2 Produk yang menjadi Inspirasi

Dalam pengembangan eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram, merupakan salah satu produk yang menjadi inspirasi adalah *eFishery*, sebuah sistem pemberi pakan otomatis berbasis IoT yang telah digunakan secara luas dalam industri perikanan.

eFishery adalah perangkat IoT yang dirancang untuk memberi pakan ternak secara otomatis. Sistem ini dapat dikontrol melalui aplikasi mobile dan dilengkapi dengan fitur pemantauan serta analisis pola konsumsi pakan. Dengan teknologi ini, petani ikan dapat mengoptimalkan pemberian pakan agar lebih efisien dan terjadwal.

Kelebihan *eFishery*:

1. Memberikan pakan secara otomatis sesuai jadwal yang telah ditentukan, sehingga membantu menjaga pola makan ternak.
2. Memiliki fitur analisis berbasis kecerdasan buatan (AI) yang dapat menyesuaikan pola pemberian pakan berdasarkan kebutuhan ternak.

Kekurangan eFishery:

1. Dirancang untuk skala industri perikanan, sehingga kurang *fleksibel* untuk peternakan kecil atau rumahan.
2. Masih terbatas pada jenis ternak tertentu seperti ikan dan udang, sehingga tidak secara langsung dapat diterapkan untuk jenis ternak lainnya
3. Bergantung pada koneksi internet yang stabil untuk dapat berfungsi dengan optimal, yang bisa menjadi kendala di daerah dengan jaringan terbatas, dan belum dapat di kontrol.

Hal yang Dapat diadaptasi untuk E-Feeder:

1. Otomatisasi Pakan Berbasis IoT, sama seperti *eFishery* sistem *E-Feeder* akan menggunakan teknologi otomatisasi dalam pemberian pakan sesuai jadwal yang telah ditentukan.
2. Fleksibilitas untuk Berbagai Jenis Ternak, tidak terbatas hanya pada ikan dan udang, *E-Feeder* akan dikembangkan agar dapat digunakan untuk berbagai jenis ternak.

Dengan mengadaptasi beberapa keunggulan dari *eFishery* dan menyesuaikannya dengan kebutuhan peternakan darat, eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram diharapkan menjadi solusi yang lebih fleksibel, efisien, dan mudah digunakan bagi para peternak.

1.2.3 Produk yang akan dibangun

Produk yang akan dibangun adalah eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram, sebuah sistem yang dirancang untuk memberikan pakan secara otomatis dan terjadwal kepada ternak unggas konsumsi seperti ayam dan bebek (tidak termasuk burung peliharaan) serta ikan air tawar, khususnya yang dipelihara di lingkungan seperti kandang dan keramba. Sistem ini bertujuan untuk meningkatkan *konsistensi* dan efisiensi dalam pemberian pakan, sekaligus mengurangi beban kerja peternak. Dengan memanfaatkan

teknologi IoT dan antarmuka Telegram, sistem ini memungkinkan pengelolaan pakan dilakukan dari jarak jauh tanpa perlu pemberian manual secara terus-menerus.

Konsep utama dari produk ini adalah otomatisasi dalam pemberian pakan ternak dan pemantauan sistem secara *real-time*, yang semuanya dapat dikendalikan melalui aplikasi Telegram. Beberapa fitur utama dari sistem ini meliputi:

1. Pemberian pakan otomatis berdasarkan jadwal yang telah ditentukan, memastikan bahwa ayam, bebek, atau ikan air tawar mendapatkan pakan secara konsisten sesuai dengan kebutuhan harian mereka, tanpa ketergantungan pada kehadiran peternak.
2. Kontrol berbasis aplikasi Telegram yang terintegrasi dengan sistem IoT, memungkinkan peternak untuk mengatur jadwal pakan, memantau status sistem, dan memberikan perintah pemberian pakan kapan saja dari jarak jauh.
3. Sistem notifikasi otomatis, yang akan mengirimkan peringatan ke Telegram jika terjadi gangguan seperti pakan tidak keluar, stok hampir habis, atau alat tidak aktif, sehingga tindakan cepat dapat diambil.
4. Desain antarmuka yang sederhana dan ramah pengguna, sehingga peternak dapat dengan mudah menyesuaikan jadwal dan durasi pemberian pakan sesuai jenis ternak serta kondisi lingkungan seperti di keramba.
5. Pengelolaan pakan yang efisien dan sistematis, yang tidak hanya menghemat waktu dan tenaga, tetapi juga membantu menjaga kondisi kesehatan ternak dengan pemberian pakan yang lebih tepat dan terkontrol.

Nilai tambah dari *eFeeder* dibandingkan sistem pemberian pakan manual terletak pada kemampuannya dalam memberikan pakan secara otomatis, terjadwal, dan dapat dipantau serta dikendalikan dari jarak jauh. Fokus penggunaan untuk unggas konsumsi dan ikan tawar—bukan hewan peliharaan seperti burung—serta penerapan langsung di lingkungan seperti kandang dan keramba membuat sistem ini lebih relevan dan aplikatif bagi peternak skala kecil hingga menengah di berbagai wilayah.

1.2.4 Konteks

Sistem eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram akan diterapkan pada peternakan skala kecil hingga menengah, seperti peternak ikan, unggas, dll untuk mengotomatiskan pemberian pakan secara efisien. Sistem ini terdiri dari modul

kontrol otomatis yang mengatur jadwal pemberian pakan, sensor pemantauan untuk memastikan proses berjalan lancar, serta sistem distribusi pakan yang menyalurkan pakan secara merata. Peternak dapat memantau dan mengontrol sistem melalui aplikasi telegram berbasis IoT yang memberikan notifikasi. Selain itu, sistem ini dapat diintegrasikan dengan sumber daya seperti panel surya untuk meningkatkan energi, terutama di daerah yang memiliki keterbatasan listrik. Dengan adanya otomatisasi ini, peternak tidak hanya menghemat waktu dan tenaga tetapi juga dapat meningkatkan produktivitas serta kesejahteraan ternak secara keseluruhan.

1.2.5 Deskripsi Kebutuhan Produk

eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram dirancang untuk membantu peternak dalam mengelola pemberian pakan secara lebih efisien dan terjadwal. Sistem ini akan memastikan ternak mendapatkan pakan pada waktu yang sesuai tanpa memerlukan intervensi manual yang berulang. Dengan adanya sistem otomatisasi ini, diharapkan dapat mengurangi kesalahan manusia dalam pemberian pakan serta meningkatkan produktivitas dan kesejahteraan ternak. Berikut adalah kebutuhan fungsional dan non-fungsional yang harus dipenuhi dalam pengembangan sistem ini.

Kebutuhan Fungsional:

1. Otomatisasi pemberian pakan sesuai jadwal yang telah ditentukan, sehingga ternak dapat menerima pakan secara teratur tanpa keterlambatan.
2. Pemantauan dan notifikasi melalui aplikasi telegram berbasis IoT, sehingga peternak dapat menerima informasi secara real-time mengenai status pemberian pakan dan kondisi sistem.
3. Kontrol manual melalui aplikasi telegram untuk pemberian pakan di luar jadwal, memungkinkan fleksibilitas bagi peternak jika ada kebutuhan khusus dalam pengelolaan ternaknya.
4. Sensor pemantauan untuk mendeteksi keberhasilan distribusi pakan, memastikan bahwa pakan benar-benar dikeluarkan dan dikonsumsi oleh ternak tanpa terbuang sia-sia.
5. Kemampuan penyimpanan data terkait jadwal pemberian pakan dan pola makan ternak, yang dapat digunakan untuk analisis dan evaluasi sistem pemberian pakan.

Kebutuhan Non-Fungsional:

1. Respon sistem kurang dari 5 detik saat menerima perintah dari aplikasi, sehingga pengguna mendapatkan pengalaman yang responsif dan tidak ada keterlambatan dalam operasional sistem.
2. Sistem tahan terhadap debu, suhu ekstrem, dan kondisi lingkungan peternakan yang mungkin lembap atau kotor, sehingga perangkat dapat berfungsi secara optimal dalam jangka waktu lama.
3. Konsumsi daya rendah untuk energi, agar sistem dapat berjalan secara terus-menerus tanpa menghabiskan terlalu banyak daya listrik atau baterai.
4. Keamanan data pengguna pada aplikasi terjamin, dengan enkripsi dan sistem otorisasi yang membatasi akses hanya kepada pengguna yang berwenang.
5. Skalabilitas sistem untuk mendukung berbagai jenis peternakan, baik dalam skala kecil maupun besar, dengan kemampuan penyesuaian berdasarkan jumlah dan jenis ternak yang dipelihara.
6. Kemudahan instalasi dan konfigurasi, sehingga peternak dapat dengan cepat mengintegrasikan sistem ini ke dalam operasional mereka tanpa memerlukan keahlian teknis yang tinggi.
7. Kompatibilitas dengan berbagai perangkat, baik smartphone maupun komputer, agar pengguna dapat mengakses dan mengontrol sistem dari mana saja dengan koneksi internet.

Dengan memenuhi kebutuhan-kebutuhan tersebut, eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram diharapkan mampu menjadi solusi yang praktis, efisien, bagi peternak dalam mengelola pemberian pakan ternak mereka. Sistem ini akan membantu meningkatkan efektivitas operasional peternakan sekaligus memastikan kesejahteraan ternak terjaga dengan baik.

1.2.6 Environment Hardware dan Software

Lingkungan perangkat keras dan perangkat lunak yang digunakan dalam pengembangan, pengujian, dan implementasi sistem eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram terdiri dari berbagai komponen yang mendukung otomatisasi pemberian pakan dan integrasi dengan sistem berbasis IoT. Berikut adalah tabel yang merinci kebutuhan *hardware* dan *software* yang digunakan dalam sistem ini:

Tabel 1. 2 Environment Hardware

No	Hardware	Spesifikasi atau Keterangan
1	Mikrokontroler (Arduino/ESP32)	Mengontrol sensor dan aktuator serta mengirimkan data ke server atau aplikasi.
2	Motor DC	Sebagai penggerak utama dalam mekanisme pelontar pakan, memungkinkan kontrol kecepatan dan jarak lemparan.
3	SCC	Mengubah energi sinar matahari dari panel surya menjadi energi listrik yang stabil dan aman untuk disimpan dalam baterai.
4	Motor servo	Digunakan untuk membuka dan menutup katup pemberi pakan jika diperlukan mekanisme tambahan.
5	Panel Surya	Sumber energi alternatif yang mengubah sinar matahari menjadi listrik untuk mengoperasikan sistem.
6	Aki kering	Menyimpan energi listrik sebagai sumber daya utama untuk menjalankan <i>E-Feeder</i> , sistem pemberi pakan ternak otomatis Via Telegram.
7	Relay Module	Mengontrol arus listrik yang mengaktifkan motor DC saat pemberian pakan berlangsung.
8	Kabel silicon	Penghantar listrik yang fleksibel dan tahan panas, memastikan aliran daya yang stabil.

Tabel 1. 3 Enviroment sofware

1	Aplikasi Telegram	Digunakan untuk mengontrol dan memantau sistem secara real-time melalui koneksi internet.
2	Arduino IDE	Mengembangkan dan mengunggah kode ke ESP32.

Dengan menggunakan perangkat keras dan perangkat lunak di atas, sistem eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram dapat berfungsi secara optimal dalam memberikan pakan sesuai jadwal, memantau kondisi sistem, serta memungkinkan pemilik ternak untuk mengontrol dan mengawasi pemberian pakan dengan mudah dan efisien

1.2.7 Metodologi dan Tools Pengembangan

1.2.7.1 Metodologi

Metode pengembangan yang digunakan dalam proyek eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram adalah metode *Agile*. *Agile* dipilih karena fleksibilitasnya dalam menangani perubahan kebutuhan dan kemampuannya dalam menghasilkan sistem yang lebih adaptif terhadap umpan balik pengguna. Dengan metode ini, pengembangan dilakukan secara iteratif dan bertahap, memungkinkan evaluasi serta peningkatan kualitas sistem secara berkelanjutan.

Tahapan utama dalam metodologi *Agile* yang diterapkan pada proyek ini meliputi:

1. Perencanaan kebutuhan: Mengidentifikasi kebutuhan sistem, termasuk fitur utama seperti pemberian pakan otomatis, pemantauan pakan, dan integrasi IoT.
2. Desain prototipe awal: Membuat rancangan awal perangkat keras dan perangkat lunak yang akan digunakan dalam sistem.
3. Pembuatan kode program: Mengembangkan kode untuk mengontrol mikrokontroler, sensor, serta membangun aplikasi telegram untuk monitoring dan kontrol sistem.
4. Pengujian dan evaluasi: Menguji sistem untuk memastikan fungsionalitas bekerja dengan baik, termasuk uji pemberian pakan, kestabilan koneksi, dan notifikasi ke aplikasi.
5. Iterasi perbaikan berdasarkan *feedback*: Melakukan perbaikan atau pengoptimalan sistem berdasarkan hasil uji coba dan umpan balik dari pengguna untuk meningkatkan kinerja serta keandalan sistem.

Dengan menggunakan metodologi *Agile*, pengembangan eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram dapat dilakukan secara lebih efisien dan responsif terhadap perubahan kebutuhan serta masukan dari pengguna, sehingga menghasilkan sistem yang lebih optimal dalam memberikan pakan secara otomatis dan meningkatkan manajemen pakan ternak.

1.2.7.2 Tools Pengembangan

Dalam pengembangan eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram, berbagai *tools* digunakan untuk merancang, membangun, menguji, dan mengelola proyek ini. *Tools* yang digunakan meliputi perangkat lunak untuk desain rangkaian

elektronik, pemrograman *mikrokontroler*, pengembangan aplikasi berbasis IoT, serta manajemen versi dan dokumentasi proyek.

Berikut adalah *tools* utama yang digunakan dalam pengembangan sistem ini:

1. Fritzing/Proteus: Digunakan untuk desain dan simulasi rangkaian elektronik sebelum dilakukan perakitan pada perangkat fisik.
2. Arduino IDE: Digunakan untuk menulis, mengunggah, dan menguji kode program pada mikrokontroler seperti Arduino atau ESP32.
3. GitHub/GitLab: Digunakan untuk penyimpanan kode program, manajemen versi, serta kolaborasi tim selama pengembangan.
4. Bizagi: Digunakan untuk membuat diagram alur proses (*business process model*) dari sistem *eFeeder*, seperti alur pemberian pakan otomatis, pemantauan stok, dan pengiriman notifikasi melalui Telegram.
5. Solidworks: Digunakan untuk membuat desain 3D komponen *eFeeder* seperti wadah pakan, dudukan motor, dan rangka sistem. Desain ini membantu visualisasi bentuk fisik perangkat sebelum proses pembuatan.

BAB II

PROJECT PLANNING (PP)

2.1 PENDAHULUAN

Pemberian pakan yang tepat waktu dan efisien sangat penting untuk menjaga kesehatan ternak. Namun, banyak peternak masih mengalami kendala seperti jadwal pemberian pakan yang tidak teratur, masih bergantung pada tenaga kerja manual, dan kurangnya pemantauan terhadap pakan yang diberikan. Masalah-masalah ini dapat menyebabkan pemborosan pakan yang tidak efisien, bahkan bisa berdampak negatif pada kesehatan ternak.

Untuk menjawab permasalahan tersebut, dibuatlah proyek eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram. Sistem ini dirancang untuk membantu peternak dalam memberikan pakan secara otomatis, terjadwal, dan bisa dikontrol dari *smartphone*. Dengan bantuan teknologi *Internet of Things* (IoT), sistem ini diharapkan dapat membuat proses pemberian pakan jadi lebih mudah, dan efisien.

Tujuan dari proyek ini adalah untuk menciptakan solusi yang bisa mengurangi ketergantungan pada tenaga kerja manual, meningkatkan manajemen pakan, serta memberikan kemudahan bagi peternak dalam mengelola ternaknya. Dengan perencanaan yang baik dan pemanfaatan teknologi yang sesuai, *eFeeder* diharapkan bisa menjadi inovasi yang bermanfaat dalam peternakan.

2.2 DESKRIPSI PROYEK

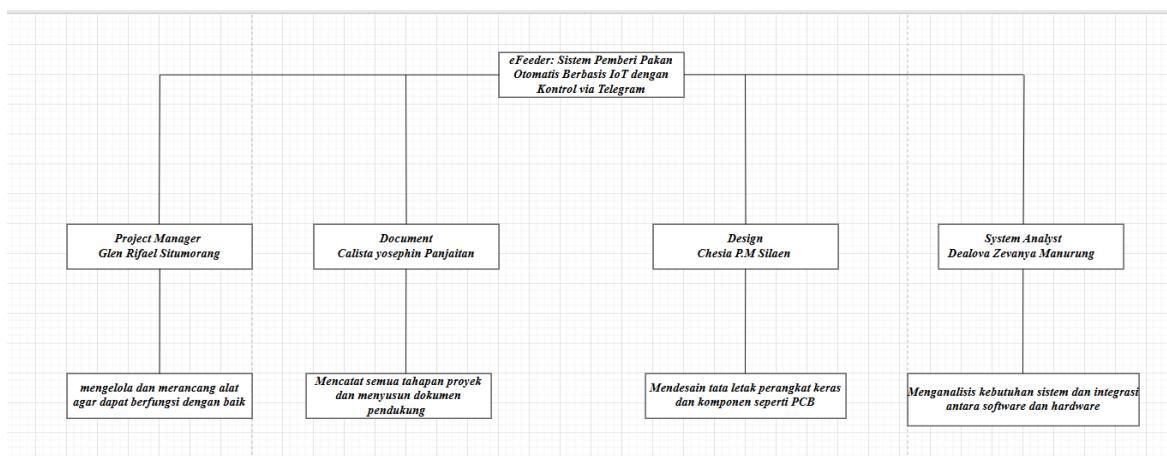
eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram adalah sebuah perangkat yang dirancang untuk membantu peternak dalam memberikan pakan kepada ternak secara otomatis dan terjadwal. Sistem ini terdiri dari komponen *hardware* seperti *mikrokontroler*, motor penggerak, sensor, serta wadah penyimpanan pakan yang saling terintegrasi.

Melalui pemanfaatan teknologi *Internet of Things* (IoT), sistem ini dapat dikendalikan dan dipantau menggunakan aplikasi Telegram. Pengguna cukup mengirimkan perintah melalui Telegram, dan sistem akan merespon untuk mengaktifkan pemberian pakan sesuai instruksi yang diterima.

eFeeder dirancang agar mudah digunakan dan dapat disesuaikan dengan kebutuhan pakan ternak sehari-hari. Dengan adanya sistem ini, proses pemberian pakan menjadi lebih efisien, dan praktis.

2.2.1 Project Organization

Dalam proyek eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram, setiap anggota tim memiliki tugas penting untuk mendukung keberhasilan sistem. Tugas pengelola proyek adalah memastikan alat dapat dirancang dan berfungsi dengan baik. Pengelola ini bertanggung jawab mengatur jalannya proyek dan mengawasi setiap proses pelaksanaan alat. Selain itu, ada juga yang bertugas mencatat semua tahapan selama proyek berlangsung. Pencatatan ini meliputi pembuatan laporan dan dokumen lain yang berisi penjelasan tentang cara kerja sistem serta langkah-langkah yang telah dilakukan selama pengembangan alat.



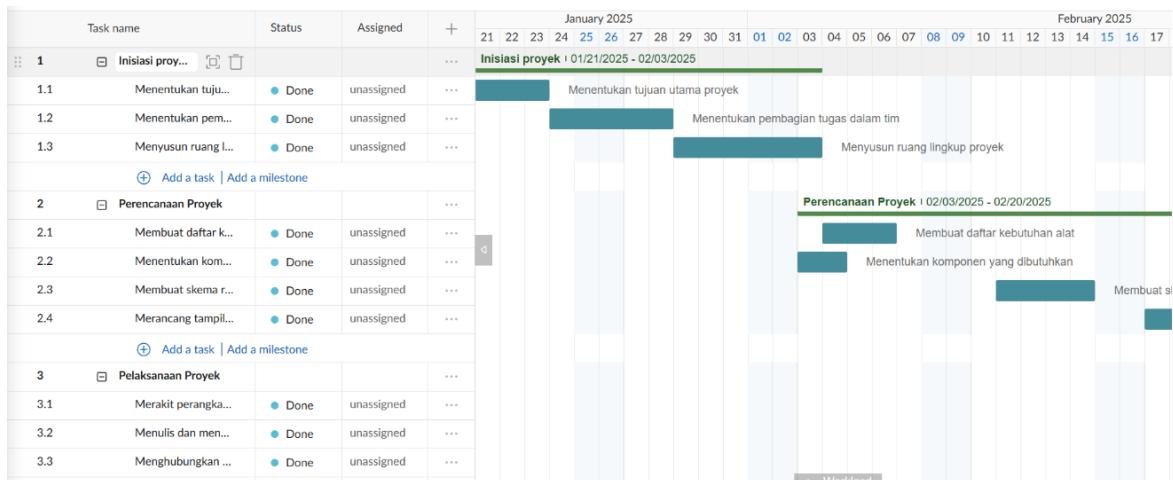
Gambar 2. 1 Project Organization

Desain sistem ditangani oleh desainer yang bertugas menyusun tata letak perangkat keras, termasuk posisi komponen seperti PCB agar sistem dapat dirakit dengan rapi dan efisien. Sementara itu, analis sistem bertugas menganalisis kebutuhan baik dari sisi perangkat lunak maupun perangkat keras, serta memastikan integrasi antar keduanya berjalan lancar. Analis juga memastikan fitur-fitur seperti kontrol pemberian pakan melalui Telegram dan sistem notifikasi dapat berfungsi dengan baik sesuai kebutuhan pengguna.

2.2.2 Work Breakdown Structure (WBS)

WBS adalah metode untuk membagi proyek menjadi bagian-bagian kecil agar lebih mudah dikelola. Bagian ini menjelaskan:

1. Proyek dibagi jadi komponen utama, sub-komponen, dan tugas-tugas kecil.
2. Tiap tugas dijadwalkan sesuai prioritas (timeline).
3. Sumber daya dialokasikan untuk tiap pekerjaan.



Gambar 2. 2 Work Breakdown Structure

Task name	Start date	Assigned	Status	Time log	+
Menentukan tujuan utama proyek	01/01/2025	unassigned	Done	0	⋮
Menentukan pembagian tugas dalam tim	01/02/2025	unassigned	Done	0	⋮
Menyusun ruang lingkup proyek	01/03/2025	unassigned	Done	0	⋮
Membuat daftar kebutuhan alat	01/04/2025	unassigned	Done	0	⋮
Menentukan komponen yang dibutuhkan	01/05/2025	unassigned	Done	0	⋮
Membuat skema rangkaian elektronik	01/11/2025	unassigned	Done	0	⋮
Merancang tampilan kontrol di Telegram	01/17/2025	unassigned	Done	0	⋮
Merakit perangkat keras (hardware)	01/21/2025	unassigned	Done	0	⋮
Menulis dan menguji program mikrokontroler	03/19/2025	unassigned	Done	0	⋮
Menghubungkan sistem dengan Telegram	04/07/2025	unassigned	Done	0	⋮
Uji coba pemberian pakan otomatis	04/22/2025	unassigned	Done	0	⋮
Mengevaluasi hasil uji coba	05/12/2025	unassigned	Done	0	⋮
Memperbaiki kekurangan yang ada	05/15/2025	unassigned	Done	0	⋮
Menyerahkan proyek akhir	05/27/2025	unassigned	In progress	0	⋮

Gambar 2. 3 Work Breakdown structure Part 2

2.2.3 Budget

Bagian ini menjelaskan alokasi anggaran untuk proyek, meliputi:

1. Rincian biaya tiap fase proyek: desain, pengembangan, pengujian, hingga implementasi.

2. Biaya tenaga kerja, perangkat keras, perangkat lunak, dan operasional lainnya.
3. Dana cadangan disiapkan untuk risiko tak terduga.
4. Total anggaran disajikan dalam tabel atau diagram agar mudah dipahami.

Tabel 2. 1 Budget yang diperlukan

NO	Komponen	Jumlah	Harga Satuan (IDR)	Harga Satuan (IDR)
1	Mikrokontroler (ESP32)	1	80.000	80.000
2	Motor DC atau Dinamo	1	80.000	80.000
3	Motor DC atau Dinamo	1	100.000	100.000
4	SCC (Solar Charge Controller)	1	150.000	150.000
5	Servo Motor	1	75.000	75.000
6	Aki Kering	1	200.000	200.000
7	Relay Module	1	45.000	45.000
8	Kabel Silicon	3m	20.000	60.000
9	Papan PCB	2	25.000	50.000
10	Drum	1	150.000	150.000
11	Panel Surya	1	300.000	300.00
12	RTC	1	40.000	40.000
13	3DC Printing Casing	1	110.000	110.000
14	Stepdown	1	20.000	20.000
Total Biaya			1.375.000	
Perangkat				
Keras				

Dengan total anggaran sebesar **Rp1.375.000**, proyek *eFeeder* diharapkan dapat berjalan dengan lancar mulai dari tahap desain hingga implementasi. Dana cadangan disiapkan untuk menutupi kemungkinan risiko yang muncul selama proses pengembangan.

2.2.4 Tools

Bagian ini menjelaskan alat dan teknologi yang digunakan dalam proyek, seperti perangkat keras dan lunak, *tools* manajemen proyek (Trello, Jira), alat kolaborasi tim (*Google Workspace*, *Slack*), serta *tools* untuk desain, simulasi, atau pengujian. Pemilihan *tools* disesuaikan dengan kebutuhan proyek.

Tabel 2. 2 Tabel

No	Hardware	Software	Tool managemen proyek	Alat bantu untuk kolaborasi tim	Tools untuk simulasi, desain, pengujian	Alasan pemilihan tools
1.	Komputer:	Visual Studio Code	Trello	Github	Proteus, Frizing	Digunakan untuk pemrograman mikrokontroler ESP32 dan desain skematik rangkaian.
2	Mikrokontroler ESP32	VS Code + PlatformIO	Jira	Google Drive	Tinkercad	Digunakan untuk coding dan melaporkan hasil uji coba kepada Dosen Pembimbing
3	Panel Surya	SolidWorks	Asana	Slack	Multisim	Digunakan untuk desain hardware dan simulasi sistem tenaga surya

2.2.5 Resiko dan Hambatan

Bagian ini mengidentifikasi risiko dan hambatan yang mungkin terjadi, seperti keterlambatan jadwal, pembengkakan anggaran, atau kegagalan teknologi. Disertakan penilaian dampak dan

kemungkinan, serta strategi mitigasi untuk mengatasinya. Hambatan eksternal seperti regulasi atau gangguan suplai juga dibahas beserta langkah antisipasinya.

Tabel 2. 3 Resiko Utama dan Stategi Mitigasi

No	Jenis Resiko	Dampak	Kemungkinan Terjadi	Strategi Mitigasi
1	Keterlambatan Jadwal	Proyek tidak selesai sesuai target	Sedang	Buat jadwal yang masuk akal dan cek kemajuan secara rutin
2	Anggaran yang Membengkak	Uang tidak cukup, proyek bisa terhenti	Tinggi	Rencanakan anggaran dengan baik, cari barang yang lebih murah, siapkan cadangan
3	Kegagalan Teknologi	Alat tidak jalan atau sering error	Tinggi	Melakukan uji coba berkala, memastikan koneksi internet yang stabil, dan memiliki cadangan perangkat keras utama seperti ESP32.
4	Keamanan Data dan Sistem	Potensi serangan siber atau akses tidak sah yang dapat mengganggu kinerja sistem atau membahayakan data pengguna	Sedang	Mengimplementasikan enkripsi dalam komunikasi data, menggunakan autentikasi yang kuat pada bot Telegram, dan memperbarui firmware serta perangkat lunak secara berkala
5	Kesalahan dalam	Sistem kendali tidak berfungsi, sehingga	Sedang	Tes kode sedikit demi sedikit dan

No	Jenis Resiko	Dampak	Kemungkinan Terjadi	Strategi Mitigasi
	Pemrograman Web-Based Control	pengguna tidak dapat mengontrol eFeeder melalui Telegram.		perbaiki jika ada error
6	Kendala Lingkungan	Alat cepat rusak karena panas atau air	Sedang	Gunakan bahan tahan air dan lindungi komponen elektronik
7	Kesulitan dalam Integrasi <i>Hardware</i> dan <i>Software</i>	Sistem tidak berjalan sesuai desain karena ketidaksesuaian antara perangkat keras dan perangkat lunak.	Sedang	Buat sistem secara bertahap dan tes tiap bagian

BAB III

PRODUCT DESIGN (PD)

(DESAIN PENGEMBANGAN PRODUK)

3.1 PENDAHULUAN

Pada tahap ini, dilakukan perancangan sistem eFeeder: Sistem Pemberi Pakan Otomatis berbasis IoT dengan kontrol melalui Telegram. Tujuan perancangan adalah memastikan sistem dapat bekerja dengan baik dan sesuai kebutuhan pengguna. Sistem dirancang untuk memanfaatkan energi surya sebagai sumber daya utama. Energi listrik yang dihasilkan dari panel surya disimpan dalam aki kering untuk menjaga ketersediaan daya saat dibutuhkan. Energi ini kemudian digunakan untuk menggerakkan motor DC yang berfungsi sebagai penggerak utama turbin.

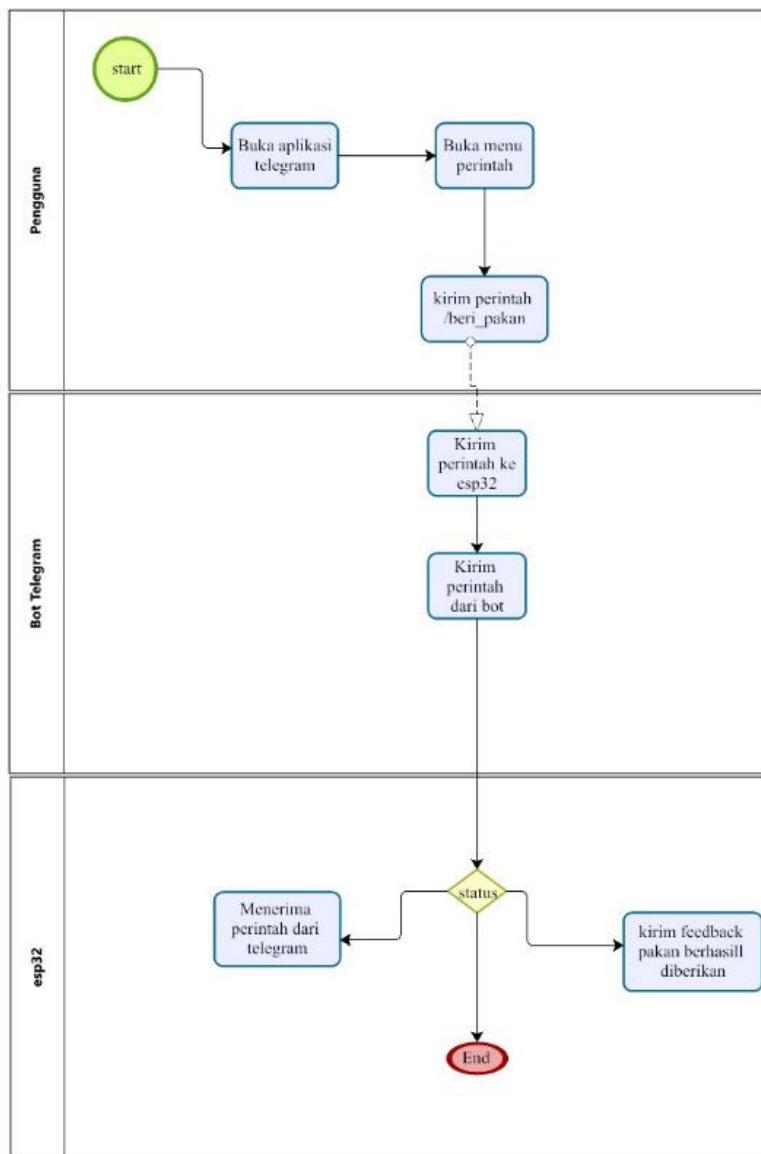
Sistem juga dilengkapi dengan drum sebagai wadah pakan, yang bisa dipindahkan menggunakan troli besi untuk mempermudah mobilitas. Aliran listrik dan gerakan mekanis dalam sistem dihubungkan melalui kabel dan pipa yang menyambungkan komponen-komponen seperti dinamo dan motor DC. Dengan desain ini, sistem *eFeeder* dirancang agar dapat digunakan secara mandiri dan efisien, baik untuk kebutuhan industri kecil, pertanian, maupun aplikasi lain yang membutuhkan pemberian pakan otomatis berbasis energi terbarukan.

Sistem ini menggunakan panel surya sebagai sumber utama energi, yang kemudian dikonversi menjadi listrik. Energi listrik tersebut disimpan dalam aki kering untuk memastikan ketersediaan daya saat dibutuhkan. Untuk menampung dan mengatur aliran energi atau cairan, digunakan drum yang dapat dipindahkan dengan bantuan troli besi agar lebih *fleksibel* dalam pengoperasian. Energi atau cairan dialirkan melalui pipa ke berbagai bagian sistem. Komponen seperti box dinamo berfungsi melindungi dinamo, sementara motor DC mengubah energi listrik menjadi gerakan mekanik. Gerakan ini kemudian digunakan untuk memutar turbin sebagai bagian akhir dari proses kerja sistem.

3.2 DESKRIPSI PRODUK DESIGN

3.2.1 Proses Bisnis Target System

Bagian ini menjelaskan alur kerja utama yang akan didukung oleh sistem. Proses digambarkan dalam bentuk *flowchart* atau BPMN untuk memvisualisasikan aktivitas yang akan diotomasi atau dioptimasi. Visualisasi ini membantu menjelaskan bagaimana sistem berinteraksi dengan pengguna maupun proses lain di dunia nyata.



Gambar 3. 1 Proses Bisnis Target System

Diagram ini menunjukkan alur kerja sistem otomatisasi pemberian pakan yang dikendalikan melalui aplikasi Telegram dengan bantuan *mikrokontroler* ESP32. Proses diawali oleh pengguna yang membuka aplikasi Telegram, mengakses menu perintah, dan mengirimkan

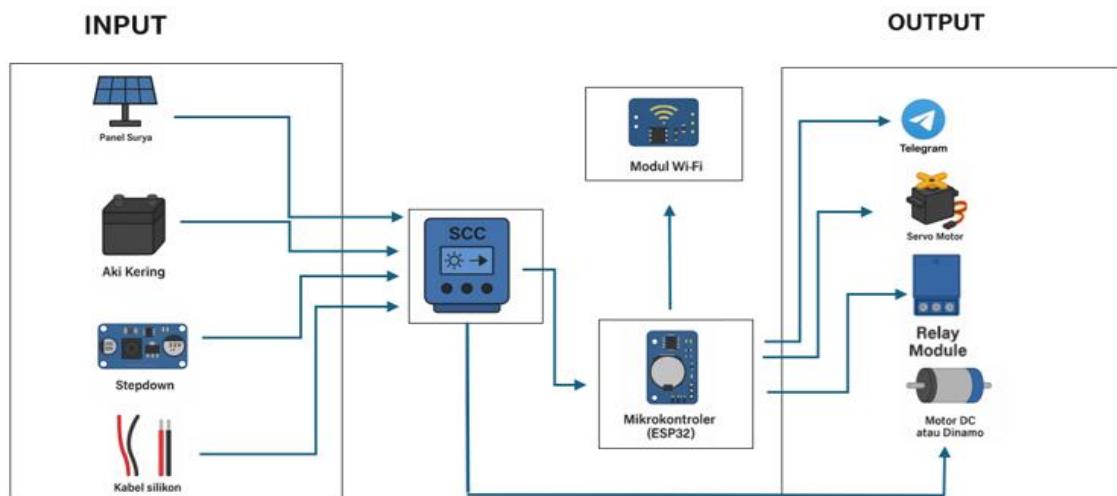
perintah "/beri_pakan" melalui bot Telegram. Perintah ini kemudian diterima oleh bot dan diteruskan ke ESP32 sebagai perangkat pengendali untuk menjalankan mekanisme pemberian pakan.

Setelah ESP32 menerima perintah, perangkat akan mengeksekusi instruksi dan memeriksa status pelaksanaannya. Jika proses pemberian pakan berhasil, ESP32 akan mengirimkan umpan balik melalui bot Telegram berupa notifikasi bahwa pakan telah berhasil diberikan. Namun jika terjadi kegagalan, sistem akan menunggu perintah ulang dari pengguna. Dengan demikian, sistem ini memungkinkan proses pemberian pakan dilakukan secara otomatis dan lebih praktis melalui interaksi sederhana di aplikasi Telegram.

3.2.2 Arsitektur Sistem [Hardware]

Energi matahari ditangkap oleh panel surya dan dikonversi menjadi energi listrik. Energi ini kemudian disimpan dalam aki kering agar daya tetap tersedia meskipun kondisi cahaya matahari berkurang. Panel surya juga terhubung ke drum penampung, yang berfungsi sebagai wadah utama bahan pakan atau cairan.

Drum ini dapat dipindahkan menggunakan troli besi untuk memudahkan pengisian atau pengaturan posisi. Dari drum, aliran diarahkan ke pipa yang berfungsi sebagai saluran distribusi. Aliran ini dikendalikan oleh sistem servo, yang kemudian mengaktifkan motor DC. Motor tersebut berperan sebagai penggerak utama yang mengubah energi listrik menjadi gerakan mekanik, dan pada akhirnya menggerakkan turbin atau mekanisme distribusi pakan. Diagram ini menyederhanakan fungsi dan alur kerja sistem, serta membantu dalam memahami keterkaitan antar komponen dalam mendukung operasional alat secara otomatis dan efisien. Blok diagram pada Gambar 3.2 menunjukkan alur kerja dari sistem pemberi pakan otomatis berbasis energi surya. Diagram ini menggambarkan hubungan antar komponen utama yang saling terintegrasi untuk mendukung kinerja sistem secara keseluruhan.



Gambar 3. 2 Blok Diagram Arsitektur Sistem

1. INPUT (*Sumber Energi dan Komponen Penunjang*)
 - a. Panel Surya
Menyerap energi matahari dan mengubahnya menjadi listrik DC.
 - b. Aki Kering
Menyimpan energi listrik agar sistem tetap bisa bekerja saat tidak ada matahari.
 - c. Stepdown
Mengurangi tegangan agar sesuai dengan kebutuhan komponen elektronik.
 - d. Kabel Silikon
Penghubung *fleksibel* dan tahan panas antara komponen-komponen listrik.
Semua input ini terhubung ke *Solar Charge Controller* (SCC).
2. SCC (Solar Charge Controller)
 - a. Mengatur aliran listrik dari panel surya dan aki agar aman.
 - b. Menyalurkan listrik ke Mikrokontroler (ESP32).
3. MIKROKONTROLER (ESP32) (*Pusat Kendali*)
 - a. Menerima daya dari aki, lalu disalurkan oleh SCC.
 - b. Menghubungkan ke Modul Wi-Fi agar terkoneksi dengan internet.
 - c. Menerima perintah atau mengirim notifikasi ke Telegram.
 - d. Mengontrol keluaran seperti:
 - i. Servo Motor: Membuka-tutup tempat pakan.
 - ii. Relay Module: Mengatur arus ke Motor DC.
4. OUTPUT (*Aksi atau Tindakan dari Sistem*)
 1. Telegram: Media untuk kirim perintah (ON/OFF pakan) atau menerima notifikasi (jadwal, status).
 2. Servo Motor: Digunakan untuk membuka tutup katup/penutup tempat pakan.
 3. Relay Module: Saklar elektronik yang memungkinkan ESP32 mengaktifkan Motor DC.
 4. Motor DC atau Dinamo: Memutar roda atau mekanisme pengalir pakan ke ternak.

3.2.3 User Interface Layout Software

Antarmuka pengguna sistem *eFeeder* dirancang berbasis bot Telegram agar sederhana dan mudah digunakan oleh siapa pun, termasuk peternak tanpa latar belakang teknis. Melalui bot

ini, pengguna cukup mengetikkan perintah atau memilih tombol yang tersedia untuk mengontrol dan memantau sistem pemberi pakan otomatis.

Beberapa perintah utama yang tersedia meliputi:

1. /berikan_pakan_ikan :untuk memberi pakan secara manual.
2. /status:menampilkan status perangkat seperti IP, kondisi servo, relay, dan jadwal pakan.
3. /jumlah_pakan:menampilkan sisa pakan.
4. /berikan_pakan_automatis:menampilkan jadwal otomatis dan log pemberian pakan.

Saat pengguna mengetikkan /status, sistem menampilkan informasi detail seperti:

1. Nama WiFi yang terhubung (misalnya: Efeeder),
2. IP lokal perangkat (contoh: 192.168.186.176),
3. Status servo (misal: Tertutup),
4. Status relay (ON/OFF),
5. Waktu saat ini, serta daftar waktu pemberian pakan otomatis beserta indikator apakah pakan sudah diberikan pada waktu tersebut.

Setelah mengetik /jumlah_pakan, sistem akan membalas dengan status ketersediaan pakan, seperti " Pakan masih banyak."

Perintah /berikan_pakan_automatis menampilkan jadwal otomatis (misalnya: 07:00, 12:00, 17:00, dan 09:45 TEST), serta riwayat pemberian pakan, baik yang dilakukan secara manual maupun otomatis. Misalnya, sistem mencatat:

1. 09:37:34 (Manual)
2. Pakan diberikan pada 09:45:01 (Otomatis TEST)

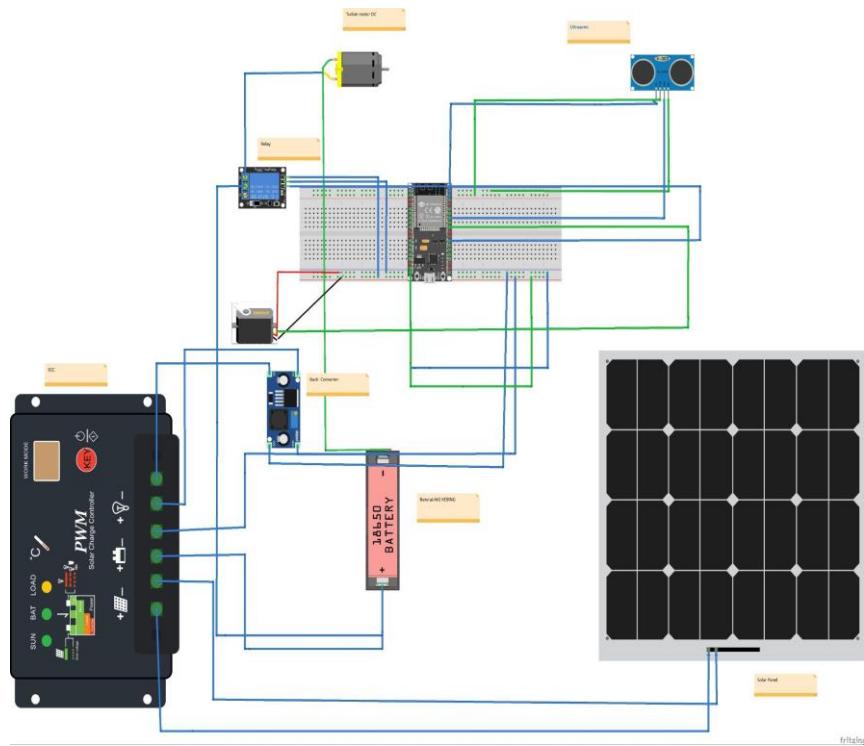
Beberapa perintah dapat menghasilkan respons ganda saat sistem aktif berkali-kali dalam waktu berdekatan, seperti terlihat pada log pengiriman pakan otomatis yang terjadi tiga kali pada pukul 09:45:15.

Sistem juga mencatat log aktivitas secara otomatis, baik untuk pemberian pakan manual maupun otomatis. Riwayat ini ditampilkan dengan format yang mudah dipahami, mencakup waktu pemberian pakan serta metode pemicunya (manual atau otomatis). Selain itu, *eFeeder* juga dilengkapi fitur notifikasi otomatis yang akan menginformasikan pengguna bila terjadi kondisi tertentu, seperti pakan hampir habis, gangguan sistem, atau konfirmasi bahwa pakan telah berhasil diberikan. Dengan kombinasi fitur-fitur tersebut, pengguna dapat mengelola dan memantau sistem pemberian pakan secara efisien dan hanya melalui aplikasi Telegram.

3.2.4 Desain Rangkaian Elektronik/Skematik Desain dan Desain PCB [Hardware]

Bagian ini menyajikan diagram yang menunjukkan alur komunikasi atau interaksi antar objek dalam sistem secara berurutan. Diagram ini digunakan untuk memvisualisasikan cara kerja sistem dalam satu skenario tertentu, seperti pengolahan data sensor atau pengendalian aktuator.

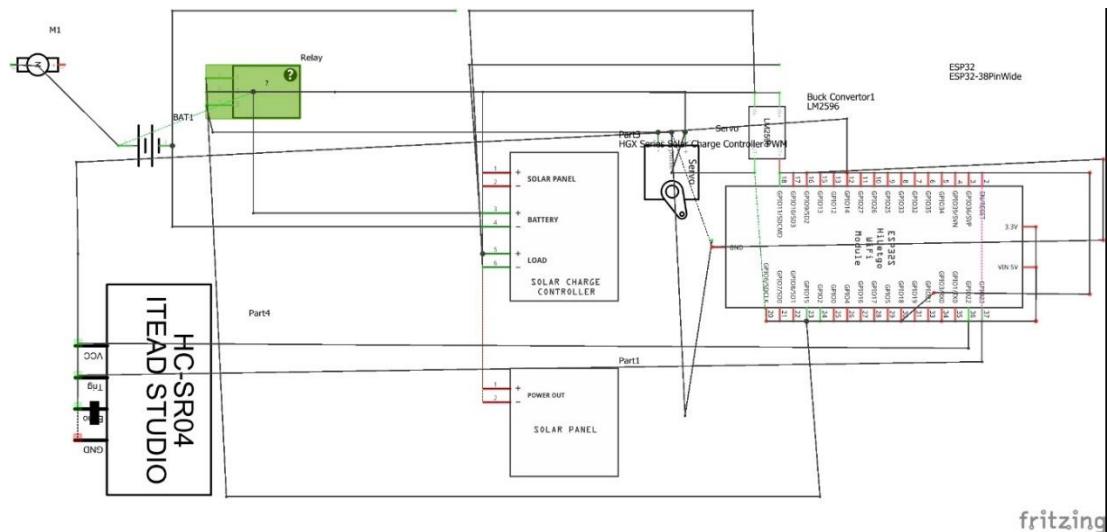
Tujuannya adalah untuk memahami urutan logika dan proses yang terjadi dalam sistem.



Gambar 3. 3 Desain rangkaian Elektronik

Rangkaian ini merupakan sistem pemberi pakan otomatis berbasis IoT yang memanfaatkan tenaga surya. Sumber daya utama berasal dari panel surya yang mengisi baterai 18650 melalui *solar charge controller*. Energi dari baterai kemudian disalurkan ke berbagai komponen elektronik di sistem. *Mikrokontroler* ESP32 menjadi pusat kendali dari seluruh sistem ini. ESP32 terhubung dengan beberapa komponen: sensor ultrasonik (HC-SR04) untuk mendeteksi keberadaan atau jarak objek (misalnya pakan), layar OLED untuk menampilkan informasi, dan relay yang mengendalikan motor DC untuk proses pemberian pakan. Relay diaktifkan oleh ESP32 sesuai dengan perintah yang diterima, misalnya melalui Telegram. Ada juga modul ESP32 yang terkoneksi dengan modul Wi-Fi.

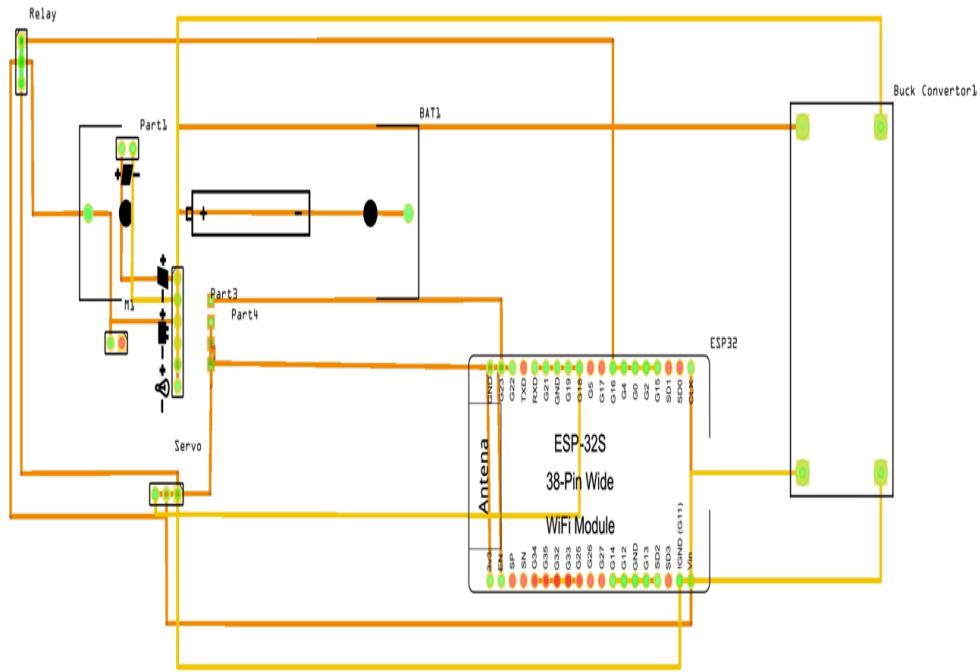
Desain Skematik



Gambar 3. 4 Desain rangkaian Skematik

Rangkaian skematik di atas menunjukkan sistem otomasi pemberi pakan berbasis ESP32 dengan sumber daya dari panel surya. Panel surya menyalurkan energi ke *Solar Charge Controller* (SCC), yang kemudian mengisi aki (BAT1) dan menyuplai daya ke beban (LOAD). Aki menjadi sumber utama untuk menggerakkan motor DC (M1) melalui modul relay, yang dikendalikan oleh sinyal digital dari ESP32. Untuk konversi tegangan agar sesuai dengan kebutuhan ESP32 dan servo motor, digunakan modul buck converter LM2596. Sensor ultrasonik HC-SR04 terhubung ke ESP32 melalui pin trigger dan echo untuk mendeteksi ketinggian pakan di dalam drum. Servo motor dikendalikan langsung oleh ESP32 dan berperan dalam membuka-tutup saluran pakan. Sistem ini merupakan integrasi antara energi terbarukan, kontrol *mikrokontroler*, dan aktuator mekanik untuk menghasilkan pemberi pakan otomatis yang efisien dan dapat dipantau dari jarak jauh.

Desain PCB



Gambar 3. 5 Desain PCB

Rangkaian *eFeeder* menggunakan *mikrokontroler* ESP32 sebagai pusat pengendali. ESP32 mendapat suplai daya dari baterai (BAT1) yang disalurkan melalui *buck converter* untuk menurunkan tegangan menjadi 5V. Tegangan 5V ini digunakan untuk menyalaikan ESP32, servo motor, dan modul relay. Servo motor disambungkan ke pin GPIO13 pada ESP32 dan berfungsi untuk membuka atau menutup tempat pakan. Modul relay disambungkan ke pin GPIO12 dan digunakan untuk mengontrol perangkat tambahan seperti motor pengaduk atau sistem lainnya.

Selain itu, terdapat sensor atau tombol yang terhubung ke pin GPIO34 untuk membaca kondisi tertentu, seperti apakah tempat pakan masih penuh atau sudah hampir habis. Semua komponen memiliki koneksi *ground* (GND) yang terhubung bersama agar sistem bekerja dengan baik. Sistem ini juga dilengkapi dengan koneksi Wi-Fi yang memungkinkan ESP32 menerima perintah dari Telegram, sehingga peternak bisa mengendalikan alat ini menggunakan aplikasi Telegram

3.2.5 Mekanisme Komunikasi Data [*Hardware*]

Pada sistem *eFeeder*: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram, komunikasi data dilakukan secara langsung antara *mikrokontroler* ESP32 dan bot

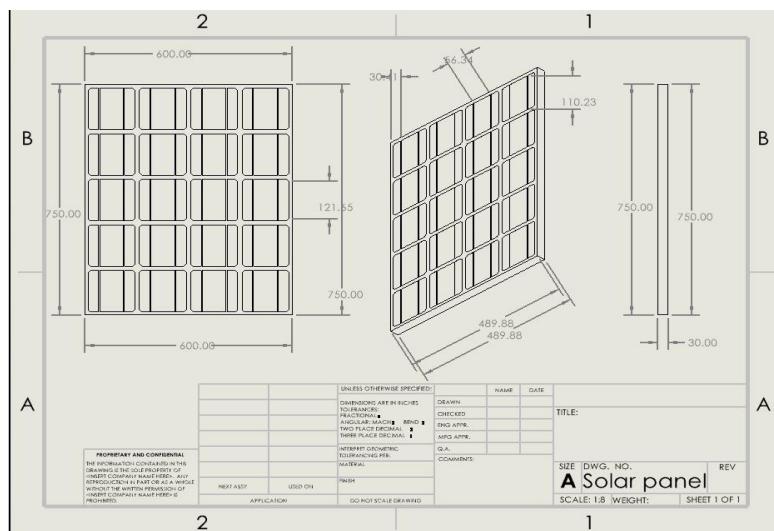
Telegram melalui koneksi internet (Wi-Fi). ESP32 terhubung ke jaringan Wi-Fi dan menggunakan API Telegram untuk mengirim dan menerima pesan.

ESP32 berfungsi sebagai pengendali utama, menerima perintah dari pengguna melalui bot Telegram, seperti memberi pakan secara manual, memeriksa status pakan, atau melihat riwayat pemberian pakan. Selain itu, ESP32 juga dapat mengirim notifikasi otomatis ke Telegram, misalnya saat pakan hampir habis atau ketika perangkat mengalami gangguan.

Komunikasi ini dilakukan menggunakan protokol HTTP melalui permintaan (*request*) ke server Telegram API. Metode ini cukup sederhana dan efektif untuk sistem yang tidak memerlukan komunikasi *real-time* yang kompleks.

3.2.5.1 Panel Surya

Desain fisik *prototipe eFeeder* menggunakan konsep modular, di mana komponen seperti wadah pakan, motor penggerak, ESP32, sensor ultrasonik, modul Wi-Fi, dan panel surya disusun terpisah namun saling terhubung membentuk satu sistem utuh. Pendekatan ini mempermudah perakitan, perawatan, dan pengembangan sistem. Panel surya digunakan sebagai sumber daya utama agar eFeeder dapat beroperasi secara mandiri tanpa listrik PLN, cocok untuk lokasi terpencil. Semua komponen dipasang rapi di atas rangka kokoh agar sistem tetap stabil dan aman saat beroperasi.



Gambar 3. 6 Panel Surya

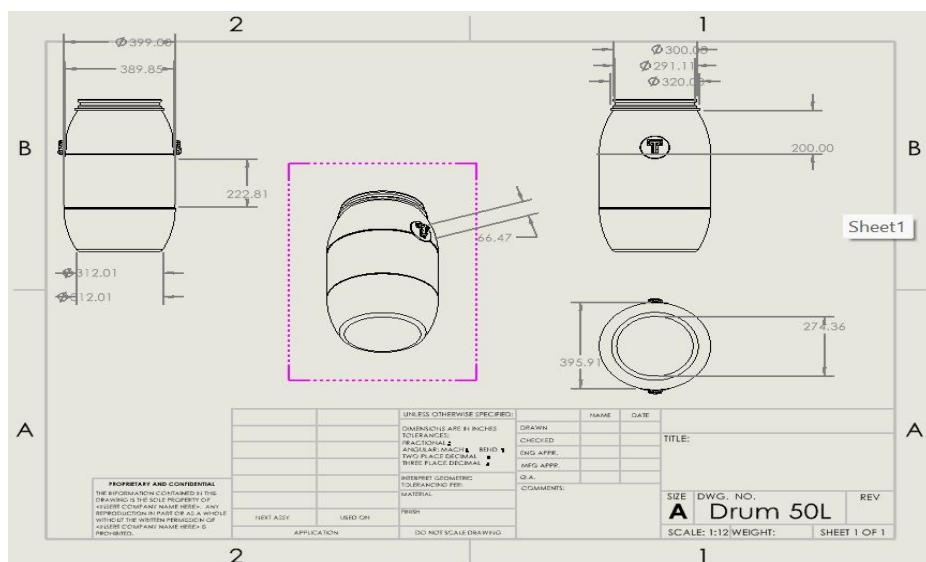
Dalam sistem eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram, panel surya (solar panel) digunakan sebagai sumber listrik utama. Panel surya ini berfungsi untuk menangkap sinar matahari dan mengubahnya menjadi energi listrik. Energi ini kemudian disimpan di dalam baterai dan digunakan untuk menghidupkan semua bagian sistem,

seperti mikrokontroler ESP32, motor penggerak pakan, dan koneksi ke Telegram. Dengan begitu, *eFeeder* bisa tetap bekerja meskipun tidak terhubung ke listrik.

Panel surya yang digunakan memiliki ukuran sekitar 60 cm x 75 cm, cukup besar untuk menghasilkan listrik yang dibutuhkan sistem. Ukuran ini bisa menghasilkan daya sekitar 30 hingga 50 watt. Penggunaan panel surya membuat sistem *eFeeder* lebih hemat biaya dan bisa digunakan di daerah yang tidak punya jaringan listrik. Selain itu, sistem ini juga ramah lingkungan karena menggunakan energi matahari. Dengan bantuan panel surya, *eFeeder* dapat bekerja secara otomatis tanpa perlu khawatir kehabisan daya.

3.2.5.2 Drum

Desain fisik *prototipe eFeeder* dibuat modular dengan menyusun komponen secara terpisah namun terhubung sebagai satu sistem. Komponen utamanya meliputi drum sebagai wadah pakan, motor penggerak, mikrokontroler ESP32, sensor ultrasonik, modul Wi-Fi untuk koneksi Telegram, serta panel surya sebagai sumber daya utama. Pendekatan ini memudahkan proses perakitan, perawatan, dan pengembangan sistem. Semua komponen disusun rapi di atas rangka yang kokoh agar sistem tetap stabil saat beroperasi, serta memungkinkan *eFeeder* digunakan secara mandiri di lokasi tanpa listrik PLN.



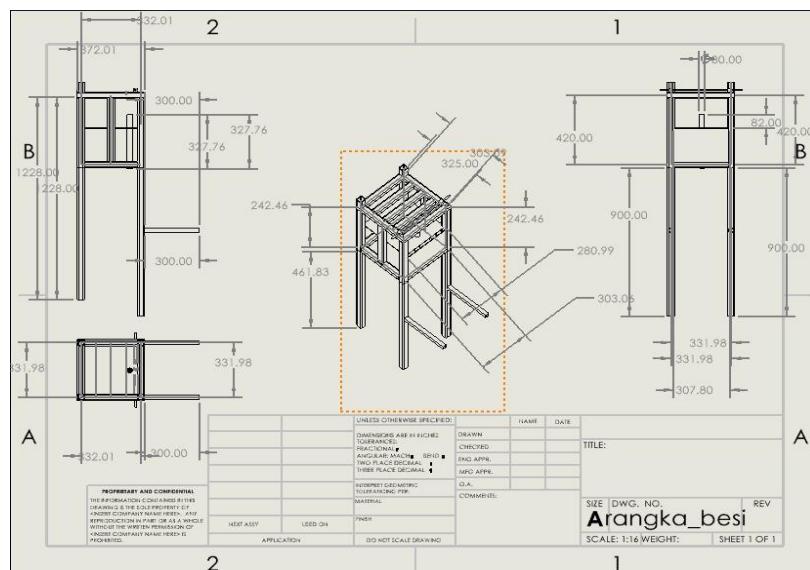
Gambar 3. 7 Drum

Drum berukuran 80 liter digunakan sebagai tempat utama untuk menyimpan pakan dalam sistem *eFeeder*: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram. Karena kapasitasnya lebih besar dari drum 50 liter, drum ini bisa menampung lebih banyak pakan. Ini sangat cocok untuk kolam ikan atau kandang ternak yang jumlah hewannya banyak, atau jika ingin sistem berjalan lebih lama tanpa harus sering diisi ulang.

Drum ini berbentuk tabung dan bagian bawah sedikit mengecil agar pakan mudah turun ke bawah. Di bagian bawahnya ada lubang untuk mengeluarkan pakan, yang akan dihubungkan ke motor atau alat pengatur pakan otomatis. Bagian atas drum memiliki tutup yang bisa dibuka agar pakan bisa diisi dengan mudah.

3.2.5.3 Rangka

Desain fisik *prototipe eFeeder* menggunakan konsep modular, dengan komponen utama seperti drum sebagai wadah pakan, motor penggerak, ESP32, sensor ultrasonik, modul Wi-Fi, dan panel surya. Drum ditempatkan pada rangka khusus yang dirancang untuk menopang beban pakan, menggunakan bahan logam ringan namun kuat agar tetap stabil saat pakan didistribusikan. Rangka ini juga menjadi dasar pemasangan komponen lain, sehingga seluruh sistem tersusun rapi, kokoh, dan mudah dirawat. Pendekatan ini memungkinkan *eFeeder* berfungsi secara efisien dan mandiri di berbagai kondisi lapangan.



Gambar 3. 8 Rangka

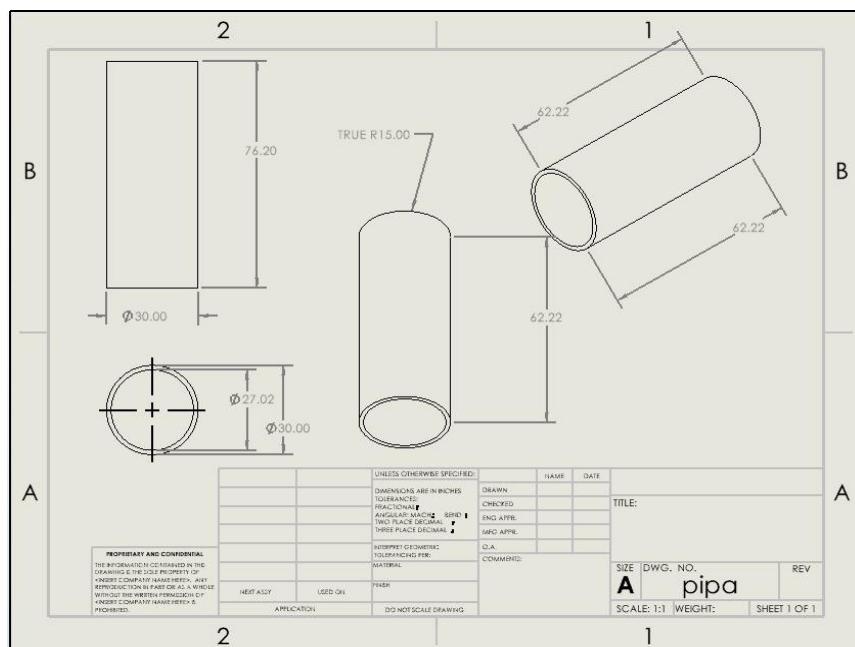
Rangka besi dalam sistem *eFeeder*: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram berfungsi sebagai penopang utama dari seluruh komponen, seperti drum pakan, motor penggerak, dan perangkat elektronik. Rangka ini dibuat dari bahan besi yang kokoh dan tahan terhadap cuaca, sehingga aman digunakan di luar ruangan seperti di pinggir kolam atau kandang ternak. Rangka ini menjaga agar sistem tetap stabil dan tidak mudah roboh saat digunakan.

Berdasarkan gambar teknik, rangka besi ini memiliki tinggi total sekitar 1228 mm, dengan lebar bagian atas 420 mm dan lebar bagian bawah 331.98 mm. Jarak antar kaki bagian bawah

adalah 307.80 mm, sementara penopang tengah memiliki panjang 300 mm. Dudukan untuk menempatkan drum berada di atas dengan kemiringan sekitar 369.90 mm secara diagonal, dan tingginya dari atas rangka ke penyangga dudukan sekitar 242.46 mm. Jarak antara rangka bagian atas ke penopang bawah sekitar 900 mm, sehingga memberikan ruang yang cukup untuk instalasi motor atau saluran keluar pakan di bagian bawah drum.

3.2.5.4 Pipa

Gambar teknik di atas menunjukkan salah satu bagian penting dari sistem *eFeeder*, yaitu sebuah pipa silinder yang digunakan sebagai wadah penampung pakan. Pipa ini memiliki panjang 76,20 mm dengan diameter luar 30,00 mm dan diameter dalam 27,02 mm. Artinya, pipa ini memiliki dinding yang cukup tebal untuk menampung pakan. Bagian ujung atasnya memiliki lengkungan dengan jari-jari 15,00 mm, membuat tampilannya lebih halus. Ukuran-ukuran ini dibuat dengan tepat agar mudah dipasang dan digunakan dalam sistem.

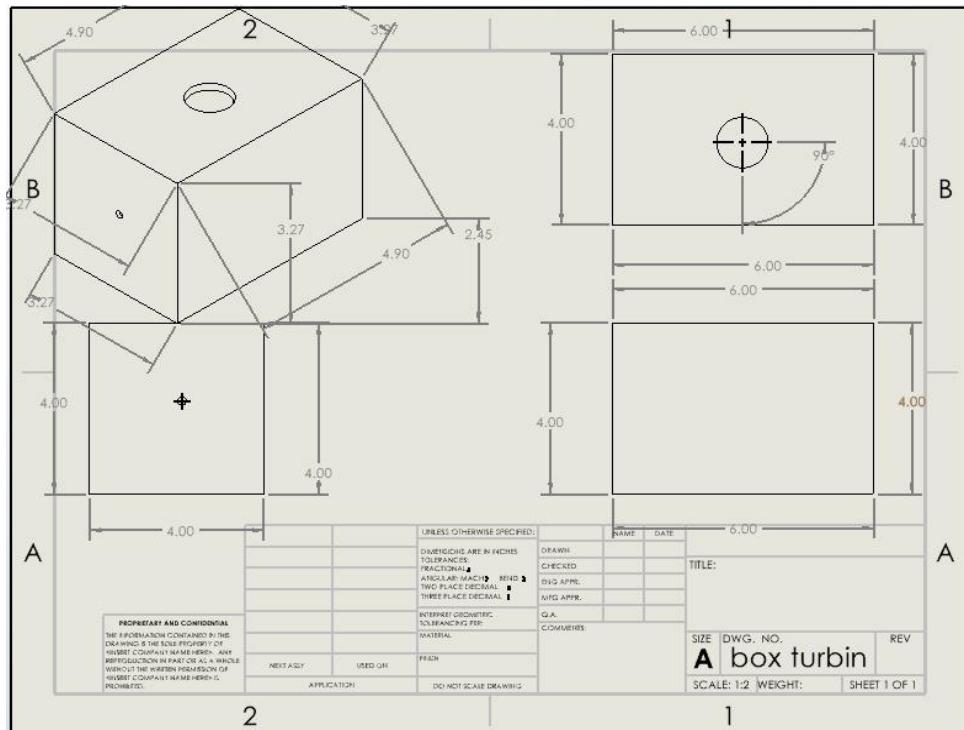


Gambar 3. 9 Pipa

Dalam sistem *eFeeder*, pipa ini akan menampung pakan sebelum dikeluarkan melalui bagian bawahnya. Bagian bawah pipa dipasang motor servo yang berfungsi untuk membuka dan menutup lubang pengeluaran pakan. Semua proses ini dikendalikan oleh *mikrokontroler* ESP32 yang terhubung dengan internet. Pengguna bisa mengirim perintah lewat Telegram Bot untuk mengatur kapan pakan dikeluarkan, baik secara langsung maupun otomatis sesuai jadwal.

3.2.5.5 Box

Desain fisik *prototipe eFeeder* dibuat modular dengan komponen seperti drum sebagai wadah pakan, motor penggerak, ESP32, sensor ultrasonik, modul Wi-Fi, panel surya, dan pipa saluran pakan. Untuk melindungi komponen elektronik dari debu, air, dan panas, digunakan box khusus tahan cuaca yang menampung ESP32, modul Wi-Fi, dan sistem kelistrikan lainnya.



Gambar 3. 10 Box

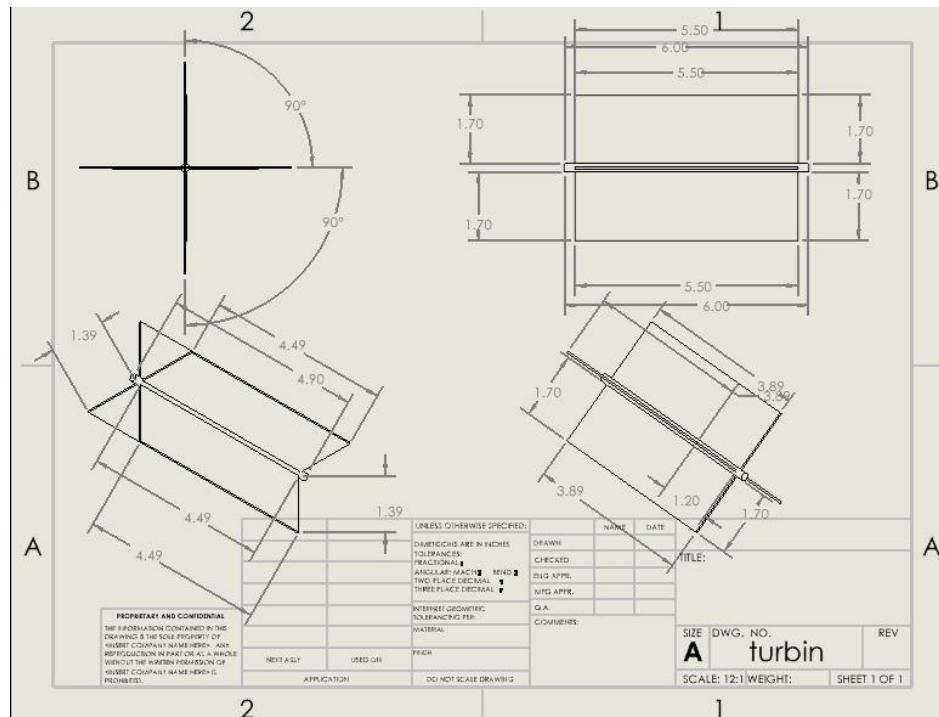
Gambar teknik di atas menunjukkan desain dari komponen box turbin, yang merupakan bagian dari sistem eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram. Box ini berfungsi sebagai rumah atau pelindung untuk turbin yang digunakan untuk menggerakkan mekanisme pemberian pakan. Box ini memiliki bentuk balok dengan dimensi panjang 6,00 inci, lebar 4,00 inci, dan tinggi 4,00 inci.

Desain box ini dibuat simetris dan agar mudah dipasang dan disesuaikan dengan komponen lainnya. Kotak ini akan dipasang dalam sistem IoT yang dikontrol oleh *mikrokontroler* ESP32. Saat pengguna mengirim perintah dari Telegram Bot, ESP32 akan mengaktifkan turbin yang berada di dalam box ini untuk menjalankan sistem pemberian pakan secara otomatis.

3.2.5.6 Turbin

Desain fisik *prototipe eFeeder* mengusung konsep modular, dengan komponen utama seperti drum sebagai wadah pakan, motor penggerak, pipa saluran, ESP32, sensor ultrasonik, modul

Wi-Fi, panel surya, box pelindung elektronik, dan tambahan turbin kecil sebagai sumber energi alternatif saat cahaya matahari terbatas, seperti saat hujan atau mendung. Turbin ini memanfaatkan angin untuk membantu mengisi daya baterai, sehingga sistem tetap dapat beroperasi secara mandiri. Semua komponen disusun di atas rangka kokoh untuk memastikan stabilitas dan kemudahan perawatan di lingkungan luar ruangan.

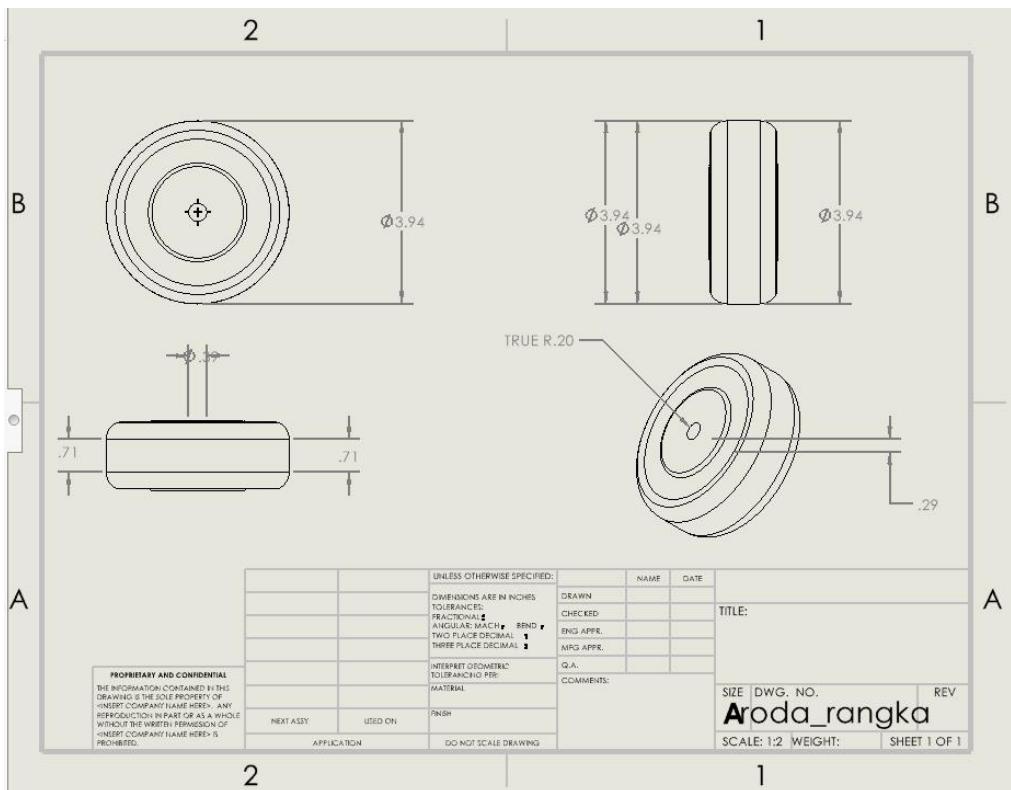


Gambar 3. 11 Turbin

Turbin memiliki bentuk seperti silinder dengan sudut bilah sebesar 90 derajat, sehingga mampu memutar dan mendorong pakan saat berputar. Ukuran keseluruhan turbin adalah panjang 6 inci dan diameter sekitar 4,49 inci, dengan bilah-bilah yang dipasang pada poros tengah. Bilah ini membantu mengatur aliran pakan ketika turbin bergerak. Struktur ini dibuat dengan mempertimbangkan kekuatan dan kestabilan agar mampu bekerja terus-menerus dalam kondisi lembap atau berdebu, sesuai dengan lingkungan tempat feeder digunakan.

3.2.5.7 Roda

Gambar ini menunjukkan desain roda yang digunakan dalam proyek eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram. Roda ini berfungsi untuk mempermudah pemindahan alat dari satu lokasi ke lokasi lain, terutama ketika alat perlu dipindahkan untuk perawatan, pengisian ulang pakan, atau penyesuaian posisi.



Gambar 3. 12 Roda

Desain roda disesuaikan dengan kebutuhan lapangan, sehingga alat tetap stabil saat digunakan namun mudah dipindahkan saat diperlukan. Dengan adanya roda, mobilitas sistem *eFeeder* menjadi lebih efisien, terutama di lingkungan peternakan atau kolam yang luas dan memiliki permukaan tanah yang bervariasi.

3.2.5.8 Desain Keseluruhan

Tampilan keseluruhan dari sistem prototipe menunjukkan bahwa *eFeeder* memiliki susunan yang sederhana namun fungsional. Komponen elektronik seperti ESP32, relay, dan diletakkan dalam box pelindung untuk menjaga dari debu atau percikan air. Sementara itu, modul sensor dan aktuator seperti servo dan motor diletakkan pada bagian yang mudah diakses untuk mempermudah proses uji coba atau perbaikan. Dengan desain fisik seperti ini, *eFeeder* tidak hanya siap untuk demonstrasi, tetapi juga berpotensi untuk diterapkan di lapangan secara langsung.



Gambar 3. 13 Desain Keseluruhan

Dari gambar tersebut, terlihat bahwa drum pada sistem ini digunakan sebagai tempat penyimpanan utama pakan ternak. Drum dilengkapi dengan tutup yang bisa dibuka, sehingga proses pengisian ulang pakan jadi lebih praktis. Di bagian bawah drum terdapat sistem pengeluaran otomatis yang mengatur aliran pakan sesuai kebutuhan, tanpa harus dilakukan secara manual. Drum ini ditopang oleh rangka besi atau baja yang dirancang kokoh dan stabil agar mampu menahan beban saat drum penuh. Di bagian bawah rangka disediakan ruang untuk sistem distribusi, yang akan mengarahkan pakan langsung ke tempat makan hewan. Pengeluaran pakan dikontrol oleh motor penggerak, seperti servo atau solenoid, yang akan membuka atau menutup saluran pakan. Mekanismenya bisa dikendalikan dari menggunakan perintah lewat bot Telegram. Seluruh sistem ditenagai oleh panel surya yang berfungsi sebagai sumber listrik utama. Energi yang ditangkap panel disimpan ke dalam aki, dan digunakan untuk menjalankan motor, sensor, serta mikrokontroler ESP32.

BAB IV

PRODUCT IMPLEMENTATION (PI)

4.1 PENDAHULUAN

Dokumen implementasi produk ini disusun dengan mengacu pada dokumen desain pengembangan produk yang telah dibuat sebelumnya. Desain tersebut menjadi dasar dalam proses pengembangan, sehingga setiap langkah dalam implementasi tetap sesuai dengan perencanaan awal yang telah dirancang secara sistematis.

Bagian ini menjelaskan bagaimana proses implementasi dilakukan berdasarkan dokumen desain. Dengan begitu, setiap komponen yang dikembangkan, baik perangkat keras maupun perangkat lunak, mengikuti struktur dan alur kerja yang telah ditentukan sebelumnya dalam dokumen desain tersebut.

Isi dari dokumen ini mencakup tahapan-tahapan penting selama implementasi. Beberapa tahapan tersebut meliputi proses pengkodean untuk mengatur fungsi sistem, pengujian setiap komponen agar berjalan sesuai harapan, serta integrasi antara bagian-bagian sistem agar bekerja secara menyatu.

Melalui bagian ini, pembaca dapat memahami bagaimana rancangan sistem diterapkan ke dalam bentuk nyata secara bertahap. Penjelasan ini memberikan gambaran umum mengenai pelaksanaan teknis dari sistem yang dikembangkan berdasarkan rancangan yang telah ditetapkan sebelumnya.

4.2 DESKRIPSI

4.2.1 Prinsip Implementasi

Pengembangan eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram dilakukan dengan pendekatan metodologi *Agile* untuk memastikan fleksibilitas dan adaptabilitas selama proses implementasi. Metodologi ini dipilih karena memungkinkan pengembangan sistem secara *iteratif* dengan pembaruan yang cepat berdasarkan umpan balik dari pengguna. Proses pengembangan melibatkan tahapan berikut:

1. Perencanaan: Menentukan kebutuhan sistem dan merancang bagaimana alat akan bekerja.
2. Pengembangan Bertahap: Membuat fitur utama satu per satu dalam waktu singkat (*sprint*), seperti kontrol motor, sensor pakan, dan pengiriman pesan lewat Telegram.

3. Pengujian: Menguji setiap modul secara individual dan melakukan integrasi sistem untuk memastikan keandalan.
4. Peluncuran dan Pemantauan: Menerapkan sistem di tempat sebenarnya, lalu mengamati hasilnya untuk mengetahui apakah perlu diperbaiki lagi.

4.2.2 Lingkungan Pengembangan Terintegrasi (IDE)

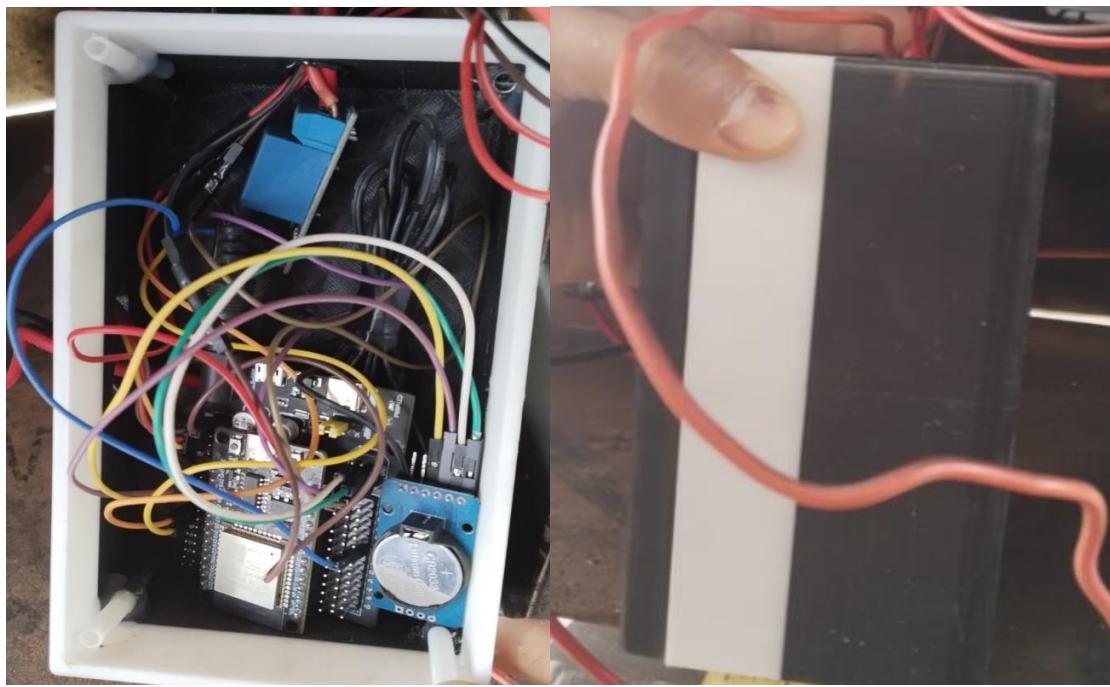
Untuk memastikan pengembangan sistem, berikut adalah perangkat lunak dan alat yang digunakan selama proses implementasi:

1. Arduino IDE: Digunakan untuk pemrograman *mikrokontroler* yang mengendalikan motor dan sensor dalam sistem *eFeeder*.
2. Visual Studio Code: Digunakan untuk pengembangan skrip backend dan komunikasi dengan Telegram API.
3. Fritzing: Digunakan untuk perancangan dan simulasi rangkaian elektronik sebelum implementasi fisik.
4. PlatformIO: Sebagai alternatif IDE yang mendukung pengembangan berbasis mikrokontroler dengan integrasi library yang lebih luas.
5. SolidWorks: Digunakan untuk merancang model 3D dari perangkat keras, seperti wadah pakan, dudukan motor, atau casing perangkat elektronik. Desain ini membantu dalam pembuatan fisik yang presisi.

Pemilihan perangkat lunak tersebut didasarkan pada kompatibilitas dengan hardware yang digunakan serta fitur yang mendukung pengembangan secara efisien.

4.2.3 Implementasi Desain PCB

Implementasi desain PCB pada sistem *eFeeder* bertujuan untuk menghubungkan komponen utama seperti ESP32, sensor ultrasonik, dan motor ke dalam satu rangkaian elektronik yang tertata rapi dan efisien. Dengan menggunakan papan PCB, sistem menjadi lebih stabil, praktis, dan aman digunakan di lingkungan peternakan.



Gambar 4. 1 Implementasi pada PCB

Perancangan PCB

1. *Software* yang Digunakan
 - a. Untuk merancang PCB, digunakan perangkat lunak solidwork.
 - b. Solidwork dipilih karena memiliki antarmuka yang ramah pengguna dan cocok digunakan untuk pengembangan prototipe secara cepat. Solidwork memungkinkan pembuatan skematik rangkaian, tata letak PCB, dan ekspor file.
2. Langkah-langkah perancangan:
 - a. Membuat skematik rangkaian elektronik yang menghubungkan ESP32 dengan sensor dan aktuator.
 - b. Menyusun tata letak komponen pada papan PCB sesuai kebutuhan ukuran dan fungsi.
 - c. Mengatur jalur koneksi antar komponen
 - d. Mengeksport desain dalam format Gerber File untuk proses fabrikasi.

Proses Fabrikasi PCB

1. Langkah-langkah fabrikasi:
 - a. Mengunggah Gerber File ke layanan fabrikasi seperti JLCPCB atau PCBWay.
 - b. Menentukan spesifikasi PCB, termasuk jumlah layer, ketebalan, jenis bahan (FR4), dan warna solder mask.

- c. Melakukan pemesanan dan menunggu proses produksi, yang biasanya memakan waktu 5-7 hari kerja.
 - d. Menerima dan melakukan inspeksi terhadap PCB untuk memastikan tidak ada kesalahan produksi.
2. Pemasangan Komponen
1. Manual Soldering
 - Beberapa komponen seperti ESP32, sensor berat (load cell), dan relay module dipasang menggunakan teknik manual soldering dengan solder konvensional.
 2. Reflow Soldering (Opsional)
 - a. Jika terdapat komponen *Surface Mount Device (SMD)*, teknik *reflow soldering* dapat digunakan untuk mempermudah pemasangan.
 - b. Langkah-langkah *reflow soldering*:
 - i. Mengaplikasikan solder paste pada pad PCB.
 - ii. Meletakkan komponen SMD sesuai posisi yang ditentukan.
 - iii. Memasukkan PCB ke dalam *reflow oven* untuk melelehkan solder dan menghubungkan komponen dengan PCB.

Pengujian dan Validasi

1. Pemeriksaan jalur listrik menggunakan multimeter untuk memastikan tidak ada hubungan pendek.
2. Pengujian fungsional dengan menghubungkan PCB ke ESP32 dan memverifikasi koneksi sensor serta aktuator.
3. Simulasi kontrol via Telegram untuk memastikan sistem dapat menerima perintah dan menjalankan fungsinya dengan benar.

4.2.4 Implementasi Desain Fisik/Desain Case Hardware

Proses pembuatan casing dimulai dengan membuat desain menggunakan perangkat lunak *SolidWorks* yang disesuaikan dengan ukuran dan bentuk komponen alat. Setelah desain selesai, bagian casing pelindung PCB dicetak menggunakan teknik 3D printing dengan bahan PLA+. Sementara itu, casing utama dibuat dari besi dengan proses pemotongan dan pembentukan sesuai desain. Setelah semua bagian casing selesai dibuat, dilakukan proses perakitan dengan memasang casing pelindung PCB ke dalam casing utama. Selanjutnya, casing besi menjalani tahap finishing berupa pengamplasan dan pengecatan untuk memperhalus permukaan dan memberikan tampilan yang rapi. Casing juga dilengkapi dengan ventilasi udara dan pintu atas agar mudah dalam pengamatan kondisi perangkat di dalamnya.

Desain fisik atau casing untuk eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram bertujuan untuk melindungi komponen elektronik dan mekanis dari faktor lingkungan seperti debu, air, dan benturan. Proses pembuatan casing dilakukan dengan teknologi 3D printing menggunakan bahan filamen yang tahan terhadap kondisi lembap, seperti PLA+ atau PETG, yang kemudian dirancang presisi untuk melindungi komponen elektronik

Material yang digunakan yaitu:

1. Filamen PLA+ atau PETG hasil cetak 3D untuk casing elektronik.
2. Aluminium hollow sebagai kerangka dasar.

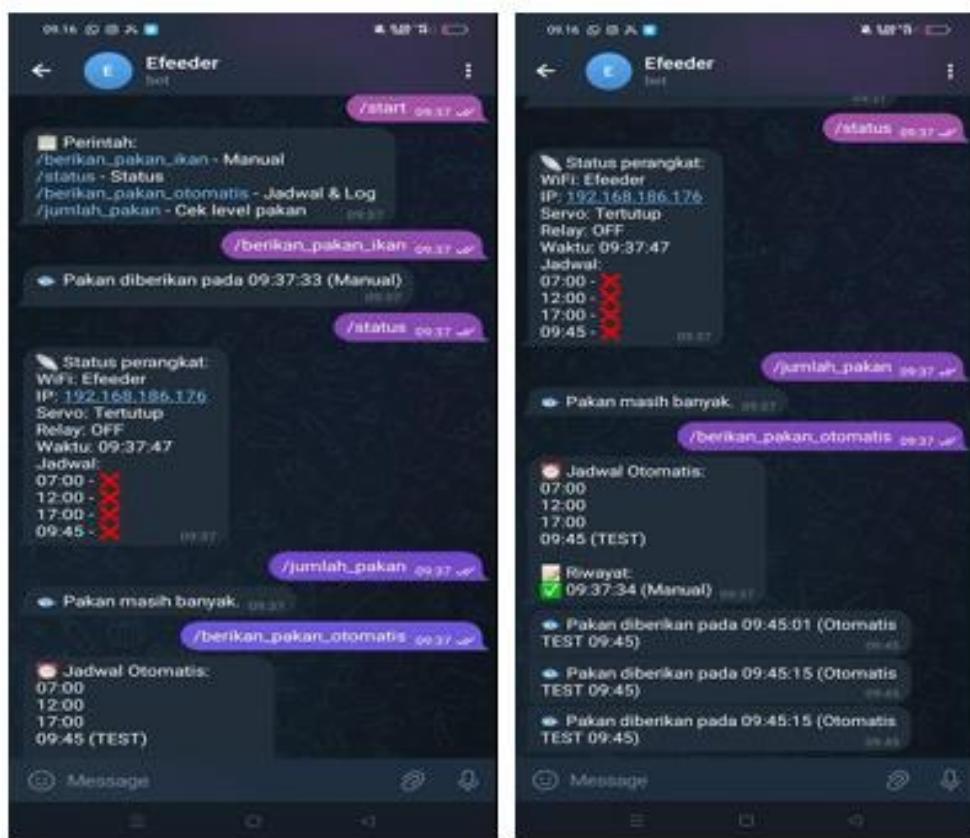


Gambar 4. 2 Implementasi Desain Fisik

Berdasarkan gambar implementasi fisik prototipe, sistem ini dirancang secara vertikal dan efisien dalam penggunaan ruang. Pada bagian paling atas, terdapat panel surya yang berfungsi sebagai sumber utama energi listrik. Panel ini dipasang dengan kemiringan yang optimal agar dapat menangkap sinar matahari secara maksimal sepanjang hari. Tepat di bawah panel, terdapat kerangka besi yang menyangga drum biru, yang berfungsi sebagai tempat penyimpanan pakan. Drum dilengkapi dengan tutup di bagian atas untuk memudahkan proses pengisian ulang pakan, dan memiliki pegangan di sisi kanan-kiri untuk kemudahan pemindahan atau pengangkatan.

Bagian tengah ke bawah dari sistem ini berisi modul distribusi pakan dan elektronik pengendali. Tepat di bawah drum, terdapat ruang bening transparan (kemungkinan dari akrilik) yang memuat komponen mekanik seperti servo atau motor DC, yang mengatur aliran pakan dari drum ke saluran bawah. Di bagian paling bawah, terdapat modul motor dan kotak distribusi pakan, yang didukung oleh roda, memungkinkan sistem untuk dipindahkan secara manual. Kabel dan komponen seperti aki, *solar charge controller* (SCC), dan relay kemungkinan besar berada di dalam atau dekat dengan kotak dasar tersebut. Implementasi ini menunjukkan bahwa sistem sudah cukup siap untuk pengujian fungsional, dengan struktur yang kuat dan penggunaan material yang sesuai untuk lingkungan luar.

4.2.5 Desain Fitur Notifikasi Telegram



Gambar 4. 3

Beberapa perintah utama yang tersedia meliputi:

1. /berikan_pakan_ikan :untuk memberi pakan secara manual.
2. /status:menampilkan status perangkat seperti IP, kondisi servo, relay, dan jadwal pakan.
3. /jumlah_pakan:menampilkan sisa pakan.
4. /berikan_pakan_automatis:menampilkan jadwal otomatis dan log pemberian pakan.

Saat pengguna mengetikkan /status, sistem menampilkan informasi detail seperti:

1. Nama WiFi yang terhubung (misalnya: Efeeder),
2. IP lokal perangkat (contoh: 192.168.186.176),
3. Status servo (misal: Tertutup),
4. Status relay (ON/OFF),
5. Waktu saat ini, serta daftar waktu pemberian pakan otomatis beserta indikator apakah pakan sudah diberikan pada waktu tersebut.

Setelah mengetik /jumlah_pakan, sistem akan membalas dengan status ketersediaan pakan, seperti "Pakan masih banyak."

Perintah /berikan_pakan_otomatis menampilkan jadwal otomatis (misalnya: 07:00, 12:00, 17:00, dan 09:45 TEST), serta riwayat pemberian pakan, baik yang dilakukan secara manual maupun otomatis. Misalnya, sistem mencatat:

1. 09:37:34 (Manual)
2. Pakan diberikan pada 09:45:01 (Otomatis TEST)

Beberapa perintah dapat menghasilkan respons ganda saat sistem aktif berkali-kali dalam waktu berdekatan, seperti terlihat pada log pengiriman pakan otomatis yang terjadi tiga kali pada pukul 09:45:15.

Sistem juga mencatat log aktivitas secara otomatis, baik untuk pemberian pakan manual maupun otomatis. Riwayat ini ditampilkan dengan format yang mudah dipahami, mencakup waktu pemberian pakan serta metode pemicunya (manual atau otomatis). Selain itu, *eFeeder* juga dilengkapi fitur notifikasi otomatis yang akan menginformasikan pengguna bila terjadi kondisi tertentu, seperti pakan hampir habis, gangguan sistem, atau konfirmasi bahwa pakan telah berhasil diberikan. Dengan kombinasi fitur-fitur tersebut, pengguna dapat mengelola dan memantau sistem pemberian pakan secara efisien dan hanya melalui aplikasi Telegram.

4.2.6 Integrasi Hardware dan Software

Hardware dan *Software* dilakukan secara bertahap untuk memastikan seluruh komponen bekerja secara optimal.

Berikut langkah-langkahnya:

1. Pemrograman Mikrokontroler (ESP32)

2. Integrasi Sensor dan Aktuator
3. Sensor pakan diprogram untuk mendeteksi jumlah pakan yang tersedia.
4. Motor DC dan Servo Motor dikendalikan berdasarkan sinyal dari ESP32 untuk membuka dan menutup saluran pemberian pakan.
5. Relay module digunakan untuk mengontrol arus listrik ke motor DC.
6. Bot Telegram dibuat menggunakan BotFather untuk menerima perintah dari pengguna.
7. API Telegram diimplementasikan dalam kode ESP32 agar dapat mengirim dan menerima pesan perintah pemberian pakan.
8. Perintah seperti pemberi pakan diprogram untuk mengaktifkan motor DC dan mengeluarkan pakan.

4.2.6.1 Pengujian Awal

Sebelum implementasi penuh, dilakukan beberapa pengujian untuk memastikan integrasi berjalan dengan baik:

1. Pengujian Komunikasi Wi-Fi: Memastikan ESP32 dapat terhubung ke jaringan dan menerima perintah dari Telegram.
2. Pengujian Sensor Pakan: Mengecek apakah sensor dapat membaca tingkat ketersediaan pakan dengan akurat.
3. Pengujian Aktuator (Motor DC & Servo Motor): Memastikan motor beroperasi sesuai dengan perintah yang diberikan dari aplikasi.
4. Pengujian Keseluruhan Sistem: Mengintegrasikan seluruh komponen untuk memastikan proses pemberian pakan berjalan otomatis berdasarkan jadwal yang ditentukan.

4.2.6.2 Debugging Masalah Integrasi

Selama proses integrasi, beberapa kendala yang mungkin terjadi antara lain:

1. Koneksi Wi-Fi tidak stabil dengan Solusi: Menambahkan mekanisme auto-reconnect pada kode ESP32.
2. Sensor tidak memberikan data akurat dengan Solusi: Kalibrasi sensor dan melakukan pengecekan kabel koneksi.
3. Motor DC tidak berfungsi dengan Solusi: Memeriksa relay module dan tegangan daya motor.
4. Perintah dari Telegram tidak dieksekusi Solusi: Mengecek token API Telegram dan memastikan bot berjalan dengan benar.

Berikut adalah kode implementasi kontrol motor menggunakan Arduino IDE:

```
#include <WiFi.h>

#include <WiFiClientSecure.h>

#include <WiFiUdp.h>

#include <UniversalTelegramBot.h>

#include <ESP32Servo.h>

#include <Wire.h>

#include <RTCLib.h>

#include <TimeLib.h>

// --- Konfigurasi WiFi & Telegram ---

#define WIFI_SSID "Efeeder"

#define WIFI_PASSWORD "11111111"

#define

BOT_TOKEN "7934501090:AAGMwSIB7ITXIZPOGIaMezi1XMAOYM5Ss-g"

#define CHAT_ID "7530731137"

// --- Waktu dan RTC ---

WiFiUDP udp;

const unsigned int localPort = 2390;

const char* ntpServer = "pool.ntp.org";

const int timeZone = 7; // WIB (UTC+7)

byte packetBuffer[48];

RTC_DS3231 rtc;

// --- Telegram bot ---

WiFiClientSecure secured_client;
```

```

UniversalTelegramBot bot(BOT_TOKEN, secured_client);

// --- Servo & Relay ---

Servo servo;

#define SERVO_PIN 21

#define RELAY_PIN 25

#define RELAY_ON LOW

#define RELAY_OFF HIGH


// --- Ultrasonik ---

#define TRIG_PIN 5

#define ECHO_PIN 18

// --- Timer & Status ---

bool servoIsOpen = false;

unsigned long servoOpenTime = 0;

const unsigned long SERVO_DURATION = 8000;

unsigned long bot_lasttime = 0;

const unsigned long BOT_MTBS = 1000;

bool alreadyFed[4] = {false, false, false, false};

bool hasResetToday = false;

int lastCheckedMinute = -1;

String feedingLog = "";


// --- NTP ---

void sendNTPpacket(IPAddress& address) {

    memset(packetBuffer, 0, 48);

```

```

packetBuffer[0] = 0b11100011;
packetBuffer[1] = 0;
packetBuffer[2] = 6;
packetBuffer[3] = 0xEC;
packetBuffer[12] = 49;
packetBuffer[13] = 0x4E;
packetBuffer[14] = 49;
packetBuffer[15] = 52;

udp.beginPacket(address, 123);
udp.write(packetBuffer, 48);
udp.endPacket();
}

bool getNTPTime() {
    IPAddress ntpServerIP;
    if (!WiFi.hostByName(ntpServer, ntpServerIP)) return false;
    sendNTPpacket(ntpServerIP);
    uint32_t start = millis();
    while (millis() - start < 1500) {
        int size = udp.parsePacket();
        if (size >= 48) {
            udp.read(packetBuffer, 48);
            unsigned long highWord = word(packetBuffer[40], packetBuffer[41]);
            unsigned long lowWord = word(packetBuffer[42], packetBuffer[43]);
        }
    }
}

```

```

        unsigned long secsSince1900 = (highWord << 16) | lowWord;

        const unsigned long seventyYears = 2208988800UL;

        unsigned long epoch = secsSince1900 - seventyYears;

        setTime(epoch + timeZone * 3600);

        return true;

    }

    delay(10);

}

return false;

}

void setRTCTimeFromTimeLib() {

    rtc.adjust(DateTime(year(), month(), day(), hour(), minute(), second()));

}

bool getRTCTime() {

    DateTime now = rtc.now();

    if (now.year() < 2020 || now.year() > 2099) return false;

    setTime(now.hour(), now.minute(), now.second(), now.day(), now.month(),
    now.year());

    return true;

}

void connectWiFi() {

    WiFi.mode(WIFI_STA);

    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

    int tries = 0;

    while (WiFi.status() != WL_CONNECTED && tries < 20) {

```

```

delay(500);

Serial.print(".");
tries++;

}

Serial.println();

if (WiFi.status() == WL_CONNECTED) {

    udp.begin(localPort);

    secured_client.setInsecure();

    Serial.println("WiFi connected.");

} else {

    Serial.println("WiFi connection failed.");

}

}

String getTimeString() {

    DateTime now = rtc.now();

    char buf[20];

    snprintf(buf,    sizeof(buf),    "%02d:%02d:%02d",    now.hour(),    now.minute(),
now.second());

    return String(buf);

}

void openFeeding(String source) {

    digitalWrite(RELAY_PIN, RELAY_ON);

    servo.write(90);

    servoIsOpen = true;

    servoOpenTime = millis(); // reset timer setiap kali dipanggil
}

```

```

String msg = "🕒 Pakan diberikan pada " + getTimeString() + " (" + source + ")";
bot.sendMessage(CHAT_ID, msg);

feedingLog += "☑ " + getTimeString() + " (" + source + ")\n";
}

void resetDailySchedule() {
    DateTime now = rtc.now();

    if (now.hour() == 0 && now.minute() == 0 && !hasResetToday) {

        for (int i = 0; i < 4; i++) alreadyFed[i] = false;

        feedingLog = "";

        hasResetToday = true;
    } else if (now.hour() != 0 || now.minute() != 0) {

        hasResetToday = false;
    }
}

void checkAutoFeeding() {
    DateTime now = rtc.now();

    int h = now.hour();

    int m = now.minute();

    if (m == lastCheckedMinute) return;

    lastCheckedMinute = m;

    const int jadwalJam[] = {7, 12, 17, 15};

    const int jadwalMenit[] = {0, 0, 0, 30}; // Ubah 15:00 jadi 15:30

    const int jadwalCount = 4;

    for (int i = 0; i < jadwalCount; i++) {

```

```

        if (h == jadwalJam[i] && m == jadwalMenit[i] && !alreadyFed[i]) {

            String waktuStr = String(jadwalJam[i]) + ":" + (jadwalMenit[i] < 10 ? "0" : "") +
String(jadwalMenit[i]);

            openFeeding("Otomatis " + waktuStr);

            alreadyFed[i] = true;

        }

    }

}

long readUltrasonicDistance() {

    digitalWrite(TRIG_PIN, LOW);

    delayMicroseconds(2);

    digitalWrite(TRIG_PIN, HIGH);

    delayMicroseconds(10);

    digitalWrite(TRIG_PIN, LOW);

    long duration = pulseIn(ECHO_PIN, HIGH, 30000);

    return (duration == 0) ? -1 : duration * 0.034 / 2;

}

void handleNewMessages(int n) {

    for (int i = 0; i < n; i++) {

        String text = bot.messages[i].text;

        String chat_id = bot.messages[i].chat_id;

        if (text == "/start") {

            String welcome = "👋 Halo Bosku, Selamat datang di eFeeder!\n\n";
            welcome += "Perintah yang tersedia:\n";
            welcome += "/berikan_pakan_ikan\n";
        }
    }
}

```

```

welcome += "/status\n";

welcome += "/berikan_pakan_otomatis\n";

welcome += "/jumlah_pakan\n";

bot.sendMessage(chat_id, welcome);

}

else if (text == "/berikan_pakan_ikan") {

openFeeding("Manual");

}

else if (text == "/status") {

DateTime now = rtc.now();

String s = "  Status perangkat:\n";

s += "WiFi: " + String(WiFi.SSID()) + "\n";

s += "IP: " + WiFi.localIP().toString() + "\n";

s += "Servo: " + String(servoIsOpen ? "Tertutup" : "Terbuka") + "\n";

s += "Relay: " + String(digitalRead(RELAY_PIN) == RELAY_ON ? "ON" : "OFF") + "\n";

s += "Waktu: " + getTimeString() + "\n";

s += "Jadwal:\n";

s += "07:00 - " + String(alreadyFed[0] ? "✓\n" : "✗\n");

s += "12:00 - " + String(alreadyFed[1] ? "✓\n" : "✗\n");

s += "17:00 - " + String(alreadyFed[2] ? "✓\n" : "✗\n");

s += "15:30 - " + String(alreadyFed[3] ? "✓\n" : "✗\n");

bot.sendMessage(chat_id, s);

}

```

```

else if (text == "/berikan_pakan_otomatis") {

    String msg = "⌚ Jadwal Otomatis:\n07:00\n12:00\n17:00\n15:30 (TEST)\n\n📝
Riwayat:\n";

    msg += (feedingLog.length() > 0) ? feedingLog : "Belum ada./";

    bot.sendMessage(chat_id, msg);

}

else if (text == "/jumlah_pakan") {

    long distance = readUltrasonicDistance();

    String msg;

    if (distance == -1) {

        msg = "🔍 Gagal membaca sensor./";

    } else if (distance <= 47) {

        msg = "Pakan masih cukup bosku./";

    } else if (distance >= 49) {

        msg = "Pakan sudah tinggal sedikit ini bosku./";

    } else {

        msg = "Pakan dalam kondisi normal ya bosku./";

    }

    bot.sendMessage(chat_id, msg);

}

void setup() {

    Serial.begin(115200);
}

```

```
pinMode(RELAY_PIN, OUTPUT);
digitalWrite(RELAY_PIN, RELAY_OFF);

pinMode(TRIG_PIN, OUTPUT);
pinMode(ECHO_PIN, INPUT);

servo.attach(SERVO_PIN);
servo.write(0);
connectWiFi();

if (!rtc.begin()) {
    Serial.println("RTC tidak ditemukan!");
    while (1) delay(10);
}

if (rtc.lostPower()) {
    if (getNTPTime()) setRTCTimeFromTimeLib();
} else {
    if (!getRTCTime()) {
        if (getNTPTime()) setRTCTimeFromTimeLib();
    }
}
secured_client.setInsecure();
bot.updateToken(BOT_TOKEN);
}

void loop() {
```

```
static unsigned long lastNtpSync = 0;

if (WiFi.status() == WL_CONNECTED) {

    if (millis() - lastNtpSync > 3600000) {

        if (getNTPTime()) {

            setRTCTimeFromTimeLib();

        }

        lastNtpSync = millis();

    }

}

static unsigned long lastRTCUpdate = 0;

if (millis() - lastRTCUpdate > 1000) {

    if (!getRTCTime()) {

        Serial.println("Gagal update waktu RTC");

    }

    lastRTCUpdate = millis();

}

if (millis() - bot_lasttime > BOT_MTBS) {

    int newMessages = bot.getUpdates(bot.last_message_received + 1);

    if (newMessages) {

        handleNewMessages(newMessages);

    }

    bot_lasttime = millis();

}

resetDailySchedule();
```

```
checkAutoFeeding();

if (servoIsOpen && millis() - servoOpenTime > SERVO_DURATION) {

    servo.write(0);

    digitalWrite(RELAY_PIN, RELAY_OFF);

    servoIsOpen = false;

}

}
```

Dengan mengikuti langkah-langkah ini, sistem *eFeeder* bisa dipasang dan dijalankan dengan baik. Setiap bagian dari alat dan aplikasi akan saling terhubung sehingga sistem bisa bekerja sesuai rencana.

Dengan begitu, pemberian pakan otomatis berbasis IoT jadi lebih mudah dan lancar. Sistem *eFeeder* bisa mengatur pemberian pakan secara otomatis, mengurangi kesalahan, dan membantu pekerjaan jadi lebih efisien.

BAB V

PRODUCT TESTING (PT)

5.1 PENDAHULUAN

Proses pengujian produk dilakukan untuk memastikan bahwa sistem eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram bekerja sesuai dengan rancangan yang telah dibuat sebelumnya. Pengujian ini sangat penting agar semua bagian sistem, bisa berfungsi dengan benar dan tidak mengalami kendala saat digunakan di lapangan.

Dalam pengujian ini, tim mengecek apakah semua fitur yang dirancang sudah berjalan sesuai dengan kebutuhan. Beberapa hal yang diuji antara lain adalah kecepatan alat merespon perintah, ketepatan waktu pemberian pakan, serta kemampuan sistem dalam menangani perintah dari aplikasi Telegram. Semua pengujian ini dilakukan dengan memperhatikan kenyamanan dan kebiasaan pengguna, terutama peternak.

Aspek perangkat keras yang diuji meliputi ESP32 sebagai otak sistem, motor servo untuk membuka dan menutup tempat keluarnya pakan, serta sensor ultrasonik untuk mendeteksi sisa pakan. Sedangkan dari sisi perangkat lunak, diuji bagaimana sistem merespons perintah dari Telegram, menjalankan jadwal pemberian pakan otomatis, dan mengirim notifikasi tentang kondisi stok pakan.

Pengujian dilakukan berdasarkan dokumen desain dan skenario yang telah direncanakan. Hasil dari pengujian ini diharapkan dapat memberikan gambaran yang jelas apakah sistem sudah siap digunakan. Selain itu, pengujian ini juga membantu menemukan dan memperbaiki kekurangan sebelum alat benar-benar dipakai oleh peternak dalam kegiatan sehari-hari.

Pengujian dilakukan mengacu pada dokumen desain dan skenario implementasi, serta disesuaikan dengan kebutuhan pengguna utama, yaitu peternak. Harapannya, *eFeeder* dapat:

1. Mengurangi ketergantungan pada tenaga kerja manual
2. Memberikan kemudahan kontrol melalui aplikasi Telegram
3. Meningkatkan sistem dalam manajemen pakan

5.2 DESKRIPSI PENGUJIAN

Pengujian dilakukan secara menyeluruh pada setiap bagian dari sistem *eFeeder*, baik pada perangkat keras seperti motor, sensor, dan modul komunikasi, maupun pada perangkat lunak seperti bot Telegram dan pemrograman ESP32. Setiap komponen diuji secara terpisah terlebih

dahulu untuk memastikan fungsinya berjalan dengan baik, kemudian dilakukan pengujian saat semua bagian telah diintegrasikan menjadi satu sistem.

Tujuan dari pengujian ini adalah untuk memastikan semua bagian dapat berfungsi secara sinkron dan tidak mengalami kendala saat dijalankan bersama. Selain itu, pengujian juga membantu menemukan kesalahan atau gangguan sejak dini agar dapat segera diperbaiki sebelum digunakan oleh pengguna. Dengan melakukan pengujian secara menyeluruh, diharapkan sistem *eFeeder* dapat berjalan dengan optimal, stabil, dan memberikan pengalaman yang mudah serta efektif dalam memberi pakan hewan secara otomatis melalui Telegram.

Ruang lingkup pengujian meliputi:

1. Pengujian Fungsional: Memastikan bahwa sistem dapat menerima dan menjalankan perintah yang dikirim melalui aplikasi Telegram, seperti perintah pemberian pakan dan pengecekan stok pakan.
2. Pengujian Integrasi: Menilai integrasi dan komunikasi antara komponen perangkat keras dan perangkat lunak, termasuk ESP32, motor servo, sensor ultrasonik, serta aplikasi Telegram.
3. Pengujian Hardware: Memverifikasi bahwa komponen seperti motor servo, sensor ultrasonik, dan mikrokontroler ESP32 berfungsi sesuai dengan spesifikasinya, termasuk respon, akurasi, dan kestabilan.
4. Pengujian Sistem: Menguji bagaimana sistem bekerja secara menyeluruh dalam skenario penggunaan nyata, seperti:
 - a. Penggunaan harian oleh peternak
 - b. Operasi dalam waktu lama
 - c. Respons terhadap kondisi darurat seperti pakan habis

5.2.1 Pengujian motor DC

Pengujian ini bertujuan untuk menganalisis pengaruh nilai *Pulse Width Modulation* (PWM) terhadap kecepatan putar (RPM) motor DC 12V dalam sistem eFeeder, yaitu alat pemberi pakan ikan otomatis. Motor digunakan untuk memutar turbin seberat 0,25 kg yang berfungsi melemparkan pakan ke kolam.

Pengujian dilakukan dalam dua kondisi:

1. Tanpa beban turbin (motor berputar bebas),
2. Dengan beban turbin (motor memutar turbin untuk melemparkan pakan).

Tujuannya adalah untuk mengetahui dampak pemasangan turbin terhadap performa motor, khususnya terhadap RPM yang dihasilkan pada berbagai level PWM.

Tabel 5. 1 Hasil pengujian kecepatan Motor DC

PWM (%)	RPM (tanpa beban turbin)	RPM (dengan beban turbin 0,25 kg)
30%	1000	820
50%	1600	1380
80%	2000	1780

Tabel 5. 2 Pengujian motor dc distribusi pakan

Skenario Pengujian	Langkah Pengujian	Kondisi Setelah Pengujian	Status
Motor aktif saat diberi daya	Aktifkan motor melalui mikrokontroler/relay	Motor menyala, mekanisme pengeluaran pakan bekerja	Berfungsi
Motor berhenti saat perintah stop	Kirim sinyal stop dari sistem kontrol	Motor menyala, mekanisme pengeluaran pakan bekerja	Berfungsi
Motor tidak bergerak tanpa perintah	Sistem dalam kondisi idlex	Motor tidak aktif, tidak ada gerakan	Berfungsi

5.2.2 PENGUJIAN KETAHANAN BATERAI

Tujuan dari pengujian ini bertujuan untuk mengetahui seberapa lama baterai dapat menyuplai daya untuk sistem *eFeeder* (terdiri dari motor DC, ESP32, relay, *step-down converter*, sensor ultrasonik, dan servo), serta memverifikasi kapasitas aktual aki dalam kondisi pemakaian riil.

Langkah Pengujian

3. Siapkan alat ukur:
 - Multimeter atau ammeter untuk mengukur arus beban.
 - Voltmeter untuk memantau tegangan aki selama operasi.
4. Jalankan sistem *eFeeder* seperti biasa (melakukan siklus pemberian pakan).
5. Ukur konsumsi arus pada berbagai kondisi:
 - Sistem idle (hanya ESP32 dan sensor aktif)
 - Saat motor servo aktif
 - Saat motor DC aktif (memutar turbin)

Hitung konsumsi daya rata-rata:

$$\text{Daya (W)} = \text{Tegangan (V)} \times \text{Arus (A)}$$

Estimasi daya tahan baterai:

$$\text{Durasi (jam)} = \text{Kapasitas baterai (Ah)} / \text{Arus rata-rata (A)}$$

Contoh Perhitungan:

- Kapasitas baterai = $40\text{Ah} \times 12\text{V} = 480\text{Wh}$
- Jika konsumsi sistem = $2\text{A} \rightarrow$ daya $\approx 24\text{W}$
- Maka durasi operasi $\approx 40\text{Ah} \div 2\text{A} = 20$ jam operasi terus-menerus

5.2.3 Pengujian Sensor Ultrasonik

Tabel 5. 3 Pengujian Sensor Ultrasonik

Skenario Pengujian	Langkah Pengujian	Kondisi Setelah Pengujian	Status
Sensor membaca saat drum hampir kosong	Kosongkan isi drum, ukur jarak sensor ke dasar ($\pm 50\text{ cm}$)	Sensor membaca jarak $\pm 48 - 50\text{ cm}$, sistem mengenali kondisi kosong	Berfungsi
Sensor membaca saat drum terisi setengah	Isi drum hingga setengah tinggi ($\pm 25\text{ cm}$ dari dasar)	Sensor membaca jarak $\pm 25\text{ cm}$, kondisi terbaca sebagai “cukup”	Berfungsi
Sensor membaca saat drum penuh	Isi drum hingga $\pm 5\text{ cm}$ dari tutup	Sensor membaca jarak $\pm 5 - 10\text{ cm}$, sistem mengenali kondisi penuh	Berfungsi
Sensor stabil meskipun ada getaran kecil	Guncangkan rangka ringan saat pembacaan sensor	Data tetap stabil, tidak ada lonjakan nilai signifikan	Berfungsi

5.2.4 Pengujian Relay Module

Tabel 5. 3 Pengujian Relay Module

Skenario Pengujian	Langkah Pengujian	Kondisi Setelah Pengujian	Status
Relay aktif saat diberi sinyal	Kirim sinyal HIGH dari ESP32	Klik relay terdengar, tegangan output berubah	Berfungsi
Relay memutus arus saat sinyal OFF	Kirim sinyal LOW dari ESP32	Klik relay terdengar, tegangan output berubah	Berfungsi
Beban motor dapat dikendalikan relay	Sambungkan motor DC ke output relay	Motor menyala dan mati sesuai logika kontrol	Berfungsi

5.2.5 Pengujian Modul Step-Down

Tabel 5. 4 Pengujian Modul Step-Down

Skenario Pengujian	Langkah Pengujian	Kondisi Setelah Pengujian	Status
Output sesuai kebutuhan ESP32 (3.3–5V)	Ukur tegangan output dengan multimeter	Tegangan output stabil di kisaran 5V	Berfungsi
Modul tidak panas berlebihan saat beban ringan	Jalankan sistem selama 15–30 menit dengan beban ringan	Modul tidak mengalami overheating	Berfungsi
Drop tegangan saat beban tinggi	Jalankan semua aktuator bersamaan	Tegangan tetap stabil, tidak drop signifikan	Berfungsi

5.2.6 Pengujian Turbin (Mekanisme Distribusi Pakan)

Tabel 5. 5 Pengujian Turbin (Mekanisme Distribusi Pakan)

Skenario Pengujian	Langkah Pengujian	Kondisi Setelah Pengujian	Status
Turbin berputar saat motor aktif	Aktifkan motor DC yang terhubung dengan turbin	Turbin berputar stabil dan mendorong pakan keluar	Berfungsi

Skenario Pengujian	Langkah Pengujian	Kondisi Setelah Pengujian	Status
Turbin berhenti saat motor off	Matikan motor	Turbin berhenti berputar	Berfungsi
Efisien aliran pakan	Uji dengan pakan berbagai ukuran	Pakan keluar lancar tanpa menyumbat turbin	Berfungsi

5.2.7 Pengujian Mikrokontroler dan Algoritma Kontrol

Tabel 5. 6 Pengujian Mikrokontroler dan Algoritma Kontrol

Skenario Pengujian	Langkah Pengujian	Kondisi Setelah Pengujian	Status
Sistem merespons waktu dari RTC	Atur waktu RTC dan amati respons mikrokontroler	Sistem memicu motor dan servo sesuai waktu yang dijadwalkan	Berfungsi
Logika kontrol motor berjalan normal	Jalankan program distribusi otomatis	Urutan motor dan servo berjalan sesuai logika	Berfungsi
Skenario Pengujian	Skenario Pengujian	Skenario Pengujian	Skenario Pengujian

5.2.8 Pengujian Servo Motor (Buka/Tutup Saluran)

Tabel 5. 7 Pengujian Servo Motor (Buka/Tutup Saluran)

Skenario Pengujian	Langkah Pengujian	Kondisi Setelah Pengujian	Status
Servo membuka saluran pakan	Kirim perintah "/beri_pakan" melalui sistem	Servo bergerak membuka saluran dengan lancar	Berfungsi
Servo menutup saluran pakan	Setelah ada notif "pakan berhasil diberikan"	Servo kembali ke posisi menutup, tanpa kendala	Berfungsi
Servo idle saat tidak diperintah	Sistem tidak mengirim sinyal	Servo tetap di posisi kembali	Berfungsi

5.2.9 Pengujian Panel Surya (MAYSUN MS120M-36 – 120W)

Menentukan kemampuan panel surya dalam mengisi baterai, berapa lama waktu yang dibutuhkan, dan apakah cukup untuk mencukupi kebutuhan harian sistem *eFeeder*.

Langkah Pengujian

1. Siapkan alat ukur:

- Wattmeter atau multimeter untuk mengukur tegangan dan arus dari panel.
- Charger controller (PWM/MPPT) untuk melindungi baterai dan mengatur arus masuk.

2. Sambungkan panel ke baterai melalui charge controller.

3. Ukur output panel pada waktu berbeda (pagi, siang, sore):

- Tegangan (V)
- Arus (A)

4. Hitung energi yang dihasilkan:

$$\text{Energi harian (Wh)} = \text{Daya rata-rata (W)} \times \text{lama penyinaran (jam)}$$

Contoh Perhitungan:

Panel 120W, sinar matahari efektif 4 jam $\rightarrow 120\text{W} \times 4 = 480\text{Wh/hari}$

Jika sistem mengonsumsi 50Wh/hari, maka, panel mampu mengisi cukup energi setiap hari + menyisakan cadangan untuk charging aki.

5.2.10 TOOLS PENGUJIAN

Berikut adalah alat-alat yang digunakan saat menguji:

1. Multimeter: Untuk mengukur tegangan listrik pada rangkaian.
2. *Software Telegram*: Digunakan untuk memberi perintah ke alat.
3. Serial Monitor (Arduino IDE): Melihat data dan pesan dari ESP32.
4. Voltmeter, untuk memantau tegangan aki selama operasi.
5. Solar Charge Controller, menagtur arus masuk.

5.3 METODE PENGUJIAN

5.3.1 Pengujian Fungsional

1. Pengguna mengirim perintah lewat Telegram.
2. Sistem harus menjalankan perintah (misalnya mengaktifkan motor).
4. Hasil diukur dari apakah motor benar-benar bekerja dan pakan keluar.

5.3.2 Pengujian Non Fungsional

1. Kinerja: Cek seberapa cepat alat merespon perintah.
2. Keandalan: Cek apakah alat tetap bekerja setelah beberapa kali digunakan.
3. Keamanan: Pastikan hanya pengguna tertentu yang bisa mengakses alat.

5.3.3 Pengujian Hardware

1. Kirim perintah dari Telegram, pastikan ESP32 menerimanya dan menggerakkan motor
2. Periksa apakah sensor bekerja dan mengirimkan data ke ESP32
4. Cek data sensor diteruskan ke Telegram sebagai notifikasi pengguna
5. Debug delay komunikasi, error parsing perintah, dan kesalahan aktivasi motor

5.3.4 Pengujian Integrasi Software dan Hardware

Pengujian integrasi dilakukan untuk memastikan bahwa perangkat lunak (Telegram Bot dan program ESP32) dapat bekerja secara sinkron dengan perangkat keras (ESP32, motor servo, sensor ultrasonik, dan aktuator). Tujuan dari pengujian ini adalah memastikan seluruh komponen berkomunikasi dan beroperasi sesuai fungsinya dalam satu kesatuan sistem.

Hal yang Diuji:

1. Komunikasi Data
 - a. ESP32 menerima perintah dari Telegram melalui koneksi internet.
 - b. Data sensor level pakan dikirim dan direspon secara real-time.
2. Respon Perintah ke *Hardware*
 - a. Saat pengguna mengirim perintah /beri_pakan, sistem harus langsung mengaktifkan motor servo.
 - b. Saat perintah /cek_stok dikirim, sistem membaca data sensor ultrasonik dan membalas status stok pakan ke Telegram.

3. Debugging Integrasi

- a. Mengidentifikasi keterlambatan eksekusi (latensi antara perintah dan respon).
- b. Memastikan tidak ada konflik sinyal antara sensor dan aktuator.

Mendeteksi gangguan koneksi Wi-Fi atau error parsing perintah.

5.3.5 Pengujian Prototipe (Prototype Testing)

Pengujian prototipe dilakukan setelah semua komponen *hardware* dan *software* berhasil diintegrasikan. Tujuannya adalah mengevaluasi fungsi, ketahanan, dan pengalaman pengguna sebelum produk digunakan dalam lingkungan nyata secara penuh.

Pengujian prototipe dilakukan setelah semua komponen *hardware* dan *software* berhasil diintegrasikan. Tujuannya adalah mengevaluasi fungsi, ketahanan, dan pengalaman pengguna sebelum produk digunakan dalam lingkungan nyata secara penuh.

1. Simulasi Pengguna Peternak:

- a. Mengirim perintah secara manual via Telegram (misal pagi dan sore).
- b. Sistem mengirim notifikasi keberhasilan setiap kali pakan diberikan.

2. Pengujian Jadwal Otomatis

- a. Pengguna mengatur waktu pemberian pakan otomatis.
- b. Sistem harus secara otomatis menjalankan perintah pada jam yang sudah ditentukan, tanpa perlu interaksi manual.

3. Kondisi Error atau Tidak Normal:

- a. Memutus koneksi Wi-Fi, sistem harus tetap stabil dan memberikan log error.
- b. Mengosongkan drum pakan, sistem memberi peringatan bahwa pakan habis

4. Pengujian Jangka Panjang (Daya Tahan):

- a. Sistem dijalankan selama minimal 3 hari dengan frekuensi 3 kali pemberian pakan per hari.
- b. Pantauan dilakukan untuk mendeteksi: *overheat*, *delay*, kesalahan mekanis, atau kegagalan komunikasi.

5. Evaluasi dari Umpan Balik Pengguna:

- a. Kemudahan penggunaan Telegram
- b. Kecepatan sistem dalam merespon
- c. Keakuratan sistem dalam mendeteksi sisa pakan
- d. Harapan pengembangan lanjutan dari pengguna (misal: notifikasi volume air, log data hari)

BAB VI

PRODUCT RELEASE (PR)

6.1 PENDAHULUAN

Peluncuran produk merupakan tahap akhir dari siklus pengembangan sistem *eFeeder*, yaitu sistem pemberi pakan otomatis berbasis IoT yang dapat dikontrol melalui Telegram. Tahap ini dilakukan setelah sistem berhasil dirancang, dibuat, dan diuji secara menyeluruh untuk memastikan semua fungsi berjalan dengan baik.

Dalam proyek ini, *eFeeder* telah melewati tahapan penting seperti analisis kebutuhan, perancangan perangkat keras dan lunak, pembuatan prototipe, serta pengujian modul RFID, *mikrokontroler*, motor servo, koneksi Wi-Fi, dan integrasi kontrol lewat bot Telegram. Semua tahapan ini telah dijelaskan pada bab sebelumnya.

Peluncuran produk dilakukan secara terbatas (*soft launch*) untuk memperkenalkan sistem kepada pihak-pihak terkait seperti dosen pembimbing, rekan satu tim, serta beberapa pengguna awal dari kalangan peternak atau pemerhati teknologi pertanian.

Hal ini bertujuan untuk mengumpulkan masukan awal yang berguna untuk pengembangan lebih lanjut. Peluncuran produk adalah tahap akhir dari siklus pengembangan yang bertujuan untuk memperkenalkan prototipe dan memperoleh masukan awal kemudian produk dinyatakan siap setelah melalui tahapan pengujian menyeluruh.

6.2 DESKRIPSI

Proses peluncuran produk dilakukan secara internal di lingkungan kampus Institut Teknologi Del, tepatnya di area Laboratorium Proyek Mahasiswa. Tempat ini dipilih karena mendukung kegiatan presentasi dan demonstrasi langsung di hadapan dosen pembimbing dan tim penilai. Peluncuran ini dilakukan pada tanggal 16 Mei 2025 sebagai bagian dari penilaian akhir mata kuliah Proyek Akhir.

Strategi peluncuran yang digunakan adalah presentasi langsung (*direct demonstration*). Dalam demonstrasi ini ditunjukkan cara kerja sistem *eFeeder*, mulai dari pemberian pakan otomatis menggunakan motor servo, pembacaan jadwal pakan berdasarkan data yang dikirim melalui bot Telegram, hingga kontrol sistem menggunakan aplikasi Telegram.

Seluruh proses demonstrasi dilakukan dalam jaringan lokal Wi-Fi yang telah dikonfigurasi, dan sistem mampu merespons perintah dari pengguna dengan baik. IP Address dari sistem lokal

digunakan sebagai sarana pemantauan status perangkat, meskipun kontrol utama tetap dilakukan melalui *platform* Telegram.

Untuk mendukung peluncuran, dokumentasi visual berupa video demonstrasi, tangkapan layar kontrol Telegram, dan poster produk juga disiapkan. Hal ini bertujuan agar informasi tentang produk *eFeeder* dapat dipahami dengan lebih mudah, terutama oleh pihak-pihak yang tidak bisa hadir langsung saat peluncuran, tim juga menyediakan dokumentasi berupa:

1. Menyusun dokumentasi teknis lengkap (user manual, diagram sistem, dan skema rangkaian).
2. Vidio simulasi penggunaan *eFeeder* oleh peternak
3. Panduan visual pemasangan *eFeeder*

6.3 DAYA GUNA PRODUK

Produk *eFeeder* memiliki berbagai manfaat yang memudahkan peternak:

1. Memberi pakan secara otomatis dan terjadwal
2. Dapat dikontrol dari melalui aplikasi Telegram
3. Mengurangi ketergantungan pada tenaga kerja manual
4. Efisien dan hemat energi karena menggunakan panel surya

6.4 POSTER PRODUK

Bagian ini mencakup desain visual untuk promosi produk, seperti poster, brosur, atau materi Untuk menarik minat pengguna dan memudahkan promosi, dibuat sebuah poster produk.



Gambar 6. 1 Poster

1. Judul Produk : "eFeeder: Sistem Pemberi Pakan Otomatis Berbasis IoT dengan Kontrol via Telegram".
2. Gambar Produk : Tampilkan unit prototipe dari berbagai sisi: tampak depan, atas, samping, serta proses perakitan.
3. Fitur Unggulan :
 - a. Kendali melalui Bot Telegram
 - b. Motor servo otomatis untuk membuka-tutup wadah pakan
 - c. Pemantauan stok pakan secara real-time dengan notifikasi jika hampir habis
 - d. Terintegrasi dengan Wi-Fi lokal
 - e. Jadwal pemberian pakan dapat dikontrol dan diubah langsung dari Telegram
4. Logo Institut Teknologi Del
5. Informasi Kontak : Nama tim dan NIM.
6. Warna dominan : Biru dan hijau, melambangkan air dan lingkungan.

7. Sasaran Audiens : Poster disesuaikan untuk menarik perhatian komunitas lingkungan, akademisi, dan calon pengguna teknologi ini.

Poin-poin penting:

1. Elemen desain: gambar produk, fitur, logo, tagline.
2. Informasi yang dicantumkan: kontak, nama produk, institusi.
3. Audiens target: komunitas peduli lingkungan dan akademisi.

6.5 PERILISAN PRODUK (OPSIONAL)

Pada tahap ini, produk belum dirilis secara resmi ke pasar atau ke pengguna akhir. Sebagai gantinya, produk dipamerkan dalam kegiatan Pameran IT Del. Berikut rincian proses tersebut:

Pameran Proyek Mahasiswa – IT Del

Waktu: Jumat 16 mei 2025

Tempat: Institut Teknologi Del, Laguboti

Tujuan: Menampilkan hasil inovasi teknologi mahasiswa dan memperoleh masukan awal dari Pengunjung dan Dosen.

6.5.1. TUJUAN PAMERAN PRODUK

Pameran dilakukan untuk:

1. Mendapatkan masukan awal dari calon pengguna, mitra potensial, dan pemangku kepentingan
2. Menunjukkan fungsi dan keunggulan produk secara langsung
3. Melakukan uji coba terbatas dalam lingkungan terkendal

6.6 EVALUASI DAN KEKURANGAN PRODUK

Produk *eFeeder* yang telah dikembangkan menunjukkan kemajuan signifikan dalam mengotomatisasi proses pemberian pakan ternak, terutama dengan dukungan sumber energi terbarukan dari panel surya dan sistem kontrol berbasis ESP32 yang dapat dioperasikan melalui Telegram. Namun, dari hasil evaluasi, terdapat beberapa kekurangan yang masih perlu diperbaiki. Salah satu kelemahan utama adalah sistem ini bersifat statis dan tidak dirancang untuk dapat berputar atau bergerak ke beberapa titik distribusi pakan. Hal ini menyebabkan

alat hanya dapat melayani satu lokasi pemberian pakan secara tetap, sehingga kurang fleksibel jika diterapkan pada skala peternakan yang lebih luas atau sistem kandang terpisah.

Selain itu, sistem belum dilengkapi dengan deteksi gangguan atau umpan balik kondisi secara menyeluruh. Misalnya, tidak ada sensor tambahan untuk memastikan bahwa pakan benar-benar mengalir ketika motor atau servo aktif. Ini bisa menyebabkan pakan macet tanpa terdeteksi. Tampilan data secara real-time juga belum tersedia, sehingga pengguna hanya bergantung pada perintah Telegram tanpa adanya informasi visual atau log status alat, seperti kapasitas pakan yang tersisa atau tegangan aki. Dari sisi desain, perlindungan terhadap cuaca dan keamanan kelistrikan juga masih dapat ditingkatkan, terutama mengingat alat ini akan dipasang di lingkungan terbuka yang rentan terhadap hujan dan debu.

DAFTAR PUSTAKA

- Juanda, J., Firdaus, M. R., & Zakaria, M. (2023). Pakan Ikan Otomatis Berbasis NodeMCU ESP8266 dengan Setting Waktu Telegram. Jurnal Sains dan Teknologi 4.0, 1(1), 54–60. Diakses dari <https://jurnal.komputasi.org/index.php/jst/article/view/15> Media Publikasi Journal + 4 jurnal.komputasi.org + 4 Jurnal ITB Semarang + 4*
- Saputra, T., Santoso, F., & Sunardi. (2023). Implementasi Alat Pemberi Pakan Ikan Otomatis Serta Deteksi Suhu Air Berbasis Mikrokontroler dengan Menggunakan Bot Telegram. G-Tech: Jurnal Teknologi Terapan, 7(4). <https://doi.org/10.33379/gtech.v7i4.3058> Uniramalang eJournal + 1 Media Publikasi Journal + 1*
- Bismi, N., & Munawaroh. (2023). Perancangan Sistem Pakan Ikan Berbasis Internet of Things Dengan Sistem Pemantauan Menggunakan Bot Telegram Untuk Peternakan Ikan. OKTAL: Jurnal Ilmu Komputer dan Sains, 2(01), 310–318. Diakses dari <https://journal.mediapublikasi.id/index.php/oktal/article/view/2113> Media Publikasi Journal + 1 Media Publikasi Journal + 1*
- Yasmin, M. P., Sarimuddin, S., & Al Jum'ah, M. N. (2024). Sistem Kendali Pemberi Pakan Ikan Otomatis Berbasis Internet Of Things. LAITO Journal: Leading Articles in Information Technology and Operations, 1(2), 75–81. Diakses dari <https://journal.usn.ac.id/index.php/LAITO/article/view/341> journal.usn.ac.id + 1 Media Publikasi Journal + 1*
- Samsumar, L. D., Hambali, H., & Zaenudin, Z. (2023). Sistem Pemberian Pakan Ikan Otomatis Berbasis IoT. Jurnal Penelitian Teknologi Informasi dan Sains, 1(2). <https://doi.org/10.54066/jptis.v1i2.1687> Jurnal ITB Semarang*