

BAB IV

PEMBUATAN ALAT

4.1 Pembuatan Perangkat Keras

4.1.1 Alat dan Bahan

Pada pelaksanaan tugas proyek ini, terdapat berbagai alat-alat sebagai pendukung mempermudah perakitan pembuatan prototype smart feeder ikan lele dapat dilihat pada table 4-1.

Tabel 4- 1 Alat-alat yang digunakan

No	Nama Alat	Jumlah
1	Mesin Gerinda	1 Buah
2	Mesin Bor	1 Buah
3	Alat Solder	1 Buah
4	Gunting	1 Buah
5	Meteran	1 Buah
6	Mesin Heater Press	1 Buah
7	Kunci Pas	1 Buah
8	Tang Kombinasi	1 Buah
9	Multimeter	1 Buah
10	Obeng type plus	1 Buah
11	Obeng type minus	1 Buah
12	Alat tulis	1 Buah
13	Timbangan	1 Buah
14	Tang Rivet	1 Buah

Pada Tabel 4- 2 merupakan bahan-bahan yang digunakan dalam pembuatan *prototype smart feeder* ikan lele.

Tabel 4- 2 bahan-bahan yang digunakan

No	Nama Bahan	Jumlah
1	ESP32	1 Buah
2	Sensor Load Cell	1 Buah
3	LCD I2C	1 Buah
4	Keypad I2C	1 Buah
5	Modul Step-Down DC-DC	1 Buah
6	Power Supply 12V 30A	1 Buah
7	MOSFET IRLZ44N	1 Buah
8	Kabel Jumper	Secukupnya
9	Kabel Power	Secukupnya
10	Holo Galvanis	3 Buah
11	Motor Servo MG996R	2 Buah
12	Ember	1 Buah
13	Corong	1 Buah
14	Rivet	150 Buah
15	PCB	1 Buah
16	Terminal Block	2 Buah
17	Project Box	1 Buah

4.1.2 Pembuatan Perangkat Elektrik

Pembuatan perangkat elektrik merupakan tahap utama dalam proses perakitan sistem Smart Feeder ikan lele berbasis Internet of Things (IoT) menggunakan mikrokontroler ESP32. Pada tahap ini dilakukan perancangan dan integrasi seluruh komponen elektronik agar sistem dapat beroperasi sesuai dengan fungsi yang telah dirancang. Adapun langkah-langkah pembuatan perangkat elektrik adalah sebagai berikut.

1. Perancangan skematik dan layout PCB

Pada tahap ini, seluruh komponen sistem seperti ESP32, RTC DS3231, modul HX711, LCD 16×2 I2C, keypad 4×4 I2C, konektor motor servo MG996R, konektor

motor DC 775, MOSFET IRF9540, serta modul buck converter XL4015 dirancang dan dihubungkan sesuai dengan kebutuhan sistem. Perancangan skematik dilakukan untuk memastikan setiap komponen memiliki jalur koneksi yang benar sehingga dapat berkomunikasi dan bekerja sesuai fungsi masing-masing.

Setelah proses perancangan skematik selesai, tahap berikutnya adalah pembuatan layout PCB. Pada tahap ini dilakukan penempatan komponen serta pengaturan jalur PCB agar rangkaian menjadi lebih rapi, mudah dirakit, dan memiliki kualitas koneksi yang baik. Hasil desain PCB kemudian digunakan sebagai acuan dalam proses pembuatan perangkat elektrik pada sistem Smart Feeder.

Selain perancangan skematik dan PCB, visualisasi hubungan antar komponen juga dibuat menggunakan software Circuit Designer. Software ini digunakan untuk membuat diagram wiring tiga dimensi (3D Wiring Diagram) yang menampilkan hubungan antar komponen secara lebih jelas dan mudah dipahami. Diagram wiring tersebut membantu dalam proses perakitan perangkat serta mempermudah identifikasi jalur koneksi antara ESP32, sensor, aktuator, dan modul pendukung lainnya.

2. Proses Pencetakan dan Penyolderan PCB

Proses pencetakan PCB diawali dengan mencetak layout jalur rangkaian yang telah dirancang menggunakan software EasyEDA ke media kertas CTS (Coated Transfer Sheet) menggunakan printer laser. Kertas CTS dipilih karena mampu menghasilkan transfer toner yang baik sehingga pola jalur PCB dapat tercetak dengan lebih jelas dan presisi pada papan tembaga.

Sebelum proses transfer dilakukan, permukaan papan tembaga dibersihkan terlebih dahulu untuk menghilangkan debu, kotoran, dan lapisan oksidasi yang dapat memengaruhi kualitas hasil pencetakan. Selanjutnya, kertas CTS yang telah dicetak ditempatkan pada permukaan papan tembaga, kemudian dilakukan proses transfer menggunakan mesin heat press. Proses ini bertujuan untuk memindahkan toner dari kertas CTS ke permukaan tembaga secara merata sehingga pola jalur rangkaian dapat terbentuk sesuai desain yang telah dibuat.

Setelah proses transfer selesai, papan PCB direndam dalam air untuk

mempermudah pelepasan kertas CTS tanpa merusak jalur yang telah terbentuk. Tahap berikutnya adalah proses etching, yaitu merendam papan PCB ke dalam larutan ferric chloride (FeCl_3) untuk melarutkan bagian tembaga yang tidak tertutup toner. Dengan proses tersebut, jalur rangkaian PCB akan terbentuk sesuai dengan layout yang telah dirancang sebelumnya.

Setelah proses etching selesai, sisa toner pada permukaan PCB dibersihkan menggunakan thinner atau alkohol hingga jalur tembaga terlihat dengan jelas. Tahap terakhir adalah proses pengeboran pada titik-titik yang telah ditentukan untuk pemasangan komponen dan konektor. Hasil dari proses pencetakan, etching, dan pengeboran PCB ditunjukkan pada Gambar 4.1.



Gambar 4. 1 Proses Pengeboran PCB

Proses selanjutnya dilakukan setelah tahap pencetakan dan pengeboran PCB selesai. Pada tahap ini dilakukan pemasangan serta penyolderan komponen elektronik sesuai dengan layout yang telah dirancang pada PCB. Setiap komponen dipasang pada posisi yang telah ditentukan, kemudian kaki-kaki komponen disolder menggunakan timah solder dan soldering iron untuk membentuk sambungan listrik yang kuat dan stabil.

Komponen yang dipasang pada PCB meliputi ESP32, RTC DS3231, modul HX711, MOSFET IRF9540, terminal catu daya, serta beberapa komponen pendukung lainnya yang digunakan dalam sistem Smart Feeder. Sementara itu, komponen yang berada di luar PCB seperti LCD 16×2 I2C, keypad 4×4 I2C, motor servo MG996R, motor DC 775, sensor load cell, dan modul buck converter XL4015

dihubungkan menggunakan pin header, terminal sekrup, dan konektor JST-XH agar memudahkan proses pemasangan, perawatan, serta penggantian komponen apabila diperlukan.

Setelah seluruh komponen terpasang, dilakukan pemeriksaan ulang terhadap hasil penyolderan untuk memastikan tidak terdapat hubungan singkat (short circuit), jalur yang terputus, maupun sambungan solder yang kurang baik. Pemeriksaan ini bertujuan untuk memastikan seluruh rangkaian dapat bekerja dengan baik sebelum dilakukan tahap pengujian dan pemrograman sistem. Hasil perakitan dan penyolderan komponen pada PCB ditunjukkan pada Gambar 4.2.



Gambar 4. 2 Proses Solder PCB

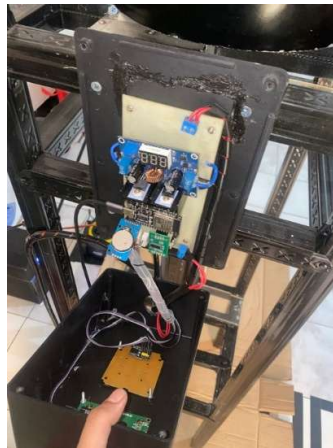
3. Integrasi Sistem

Dalam proses perakitan perangkat keras sistem Smart Feeder ikan lele berbasis ESP32, interkoneksi antarperiferal dirancang menggunakan metode sambungan lepas-pasang (detachable connection) untuk menghindari penyolderan permanen pada papan rangkaian utama (Printed Circuit Board / PCB). Modul penampil LCD 16×2, keypad 4×4, motor servo, dan sensor load cell (beserta modul HX711) dihubungkan menggunakan konektor JST (Japan Solderless Terminal), sedangkan aktuator berdaya tinggi seperti motor DC 775 diintegrasikan menggunakan terminal block.

Pemilihan jenis konektor ini bertujuan untuk memberikan fleksibilitas spasial yang tinggi selama proses integrasi mekanis ke dalam kerangka prototipe (casing). Fleksibilitas ini sangat krusial mengingat penempatan aktuator (motor servo dan

motor DC 775) serta sensor berat (load cell) memerlukan penyesuaian posisi yang presisi pada wadah penimbang (hopper) dan mekanisme pelontar agar akurasi takaran serta jangkauan sebaran pakan tercapai secara optimal. Di sisi lain, modul antarmuka pengguna seperti LCD dan keypad harus diletakkan pada panel kontrol luar agar mudah diakses serta diamati secara visual oleh peternak.

Selain mempermudah pengaturan tata letak, penggunaan konektor JST dan terminal block memberikan keunggulan signifikan dalam aspek pemeliharaan (maintainability) dan perbaikan (troubleshooting). Apabila terjadi kendala atau kerusakan pada komponen tertentu, modul dapat langsung dilepas tanpa memerlukan proses pelepasan timah atau penyolderan ulang (desoldering) pada PCB utama. Pendekatan modular ini sangat esensial dalam tahap pengembangan prototipe, sebab memberikan kemudahan bagi pengembang untuk melakukan modifikasi desain, pengujian komponen, atau penyesuaian mekanis secara cepat dan efisien. Wujud fisik dari keseluruhan komponen yang telah terinterkoneksi dengan papan rangkaian utama sistem Smart Feeder ditunjukkan pada Gambar 4.3.



Gambar 4. 3 Integrasi Sistem

4.1.3 Pembuatan Perangkat Mekanik

Pembuatan perangkat mekanik dilakukan untuk mendukung kinerja sistem Smart Feeder ikan lele berbasis Internet of Things (IoT). Perancangan mekanik bertujuan untuk menghasilkan struktur yang kokoh, stabil, dan mampu menopang seluruh komponen mekanik maupun elektronik yang digunakan pada sistem. Desain mekanik dibuat menggunakan perangkat lunak SolidWorks untuk

mempermudah proses visualisasi, perencanaan ukuran, serta integrasi antar komponen sebelum proses fabrikasi dilakukan.

Pada sistem ini, perangkat mekanik dirancang untuk mendukung proses penyimpanan pakan, penakaran massa pakan menggunakan sensor load cell, serta pendistribusian pakan secara otomatis sesuai jadwal yang telah ditentukan. Selain itu, rangka mekanik juga berfungsi sebagaiudukan komponen elektronik seperti ESP32, modul RTC DS3231, sensor load cell, motor servo, motor DC, dan panel kontrol.

Secara keseluruhan, sistem mekanik Smart Feeder terdiri dari beberapa komponen utama sebagai berikut:

1. Rangka Utama

Berfungsi sebagai struktur utama yang menopang seluruh komponen sistem. Rangka dibuat menggunakan material holo galvanis berbentuk persegi sehingga memiliki kekuatan yang baik terhadap beban pakan dan komponen yang dipasang pada alat.

2. Tabung Penampung Pakan

Berfungsi sebagai tempat penyimpanan pakan ikan sebelum proses penakaran dilakukan. Tabung ditempatkan pada bagian atas rangka untuk memanfaatkan gaya gravitasi sehingga pakan dapat mengalir menuju wadah penimbangan.

3. Corong Penyalur Pakan

Berfungsi mengarahkan pakan dari tabung penyimpanan menuju wadah penimbangan. Bentuk corong dibuat meruncing untuk mempermudah aliran pakan menuju mekanisme katup servo.

4. Wadah Penimbangan Pakan

Berfungsi sebagai tempat penakaran pakan yang terhubung dengan sensor load cell. Wadah ini digunakan untuk mengukur massa pakan secara real-time sebelum didistribusikan ke kolam.

5. Dudukan Motor Servo dan Mekanisme Katup

Berfungsi sebagai mekanisme buka-tutup saluran pakan. Motor servo digunakan untuk mengatur aliran pakan dari penampung utama menuju wadah penimbangan serta dari wadah penimbangan menuju sistem pelontar pakan.

6. Dudukan Motor DC dan Pelontar Pakan

Berfungsi sebagai tempat pemasangan motor DC 775 yang digunakan untuk memutar baling-baling pelontar pakan. Mekanisme ini membantu menyebarkan pakan ke

area yang lebih luas sehingga distribusi pakan tidak terpusat pada satu titik.

7. Box Panel Kontrol

Berfungsi sebagai tempat penyimpanan komponen elektronik seperti ESP32, modul catu daya, modul RTC, dan rangkaian kontrol lainnya. Box panel dipasang pada sisi rangka agar mudah diakses saat proses pengoperasian maupun pemeliharaan sistem.

4.2 Pembuatan Perangkat Lunak

Pemrograman mikrokontroler ESP32 dilakukan menggunakan perangkat lunak Arduino IDE dengan bahasa pemrograman C++. Program ini merupakan bagian utama dari sistem karena mengatur kerja seluruh komponen perangkat keras, mulai dari modul I2C (LCD 16×2, Keypad 4×4, RTC DS3231), sensor Load Cell (HX711), aktuator (Servo 1, Servo 2, Motor DC 775), hingga modul komunikasi nirkabel (Wi-Fi, Telegram Bot, dan Google Spreadsheet). Struktur kode disusun secara modular menggunakan pendekatan State Machine untuk memudahkan pengembangan, pengujian, dan perawatan sistem.

Pembagian program yang dikembangkan meliputi beberapa bagian utama, yaitu:

1. Program Inisialisasi dan Konfigurasi Awal Sistem

Program untuk mendefinisikan pin Input/Output ESP32, menginisialisasi protokol komunikasi I2C, serta mengaktifkan pustaka pendukung perangkat keras seperti LiquidCrystal_I2C, I2CKeyPad, RTCLib, HX711, dan ESP32Servo.

2. Program Konektivitas Wi-Fi dan Sinkronisasi Waktu (NTP)

Program untuk menghubungkan ESP32 ke jaringan internet (Wi-Fi) serta menyinkronkan waktu internal modul Real Time Clock (RTC DS3231) secara otomatis dengan waktu standar internet menggunakan server Network Time Protocol (NTP) zona waktu WIB (UTC+7).

3. Program Kontrol Lokal via Keypad 4×4

Program untuk mendeteksi penekanan tombol pada Keypad I2C secara interaktif, yang memungkinkan pengguna mengubah tampilan menu (slide jadwal), mengatur waktu jadwal pakan, serta menetapkan target massa pakan secara langsung (offline) di lokasi alat.

4. Program Kontrol Jarak Jauh via Telegram Bot

Program untuk memeriksa, menerima, dan memproses perintah teks dari pengguna melalui aplikasi Telegram (seperti perintah /status, /setjam,

/setmassa, /feednow, dan /reset) secara asinkron dengan mekanisme jeda (delay) pengamanan interaksi pengguna.

5. Program Pembacaan Waktu dan Penjadwalan Otomatis (Auto-Feeding)
Program looping untuk membaca data detik, menit, dan jam dari RTC DS3231 secara kontinu, kemudian mencocokkannya dengan array data jadwal aktif (hingga 10 jadwal) untuk memicu proses pemberian pakan secara otomatis (State: STATE_IDLE).
6. Program Pengukuran Massa Pakan dan Kontrol Katup Timbang (State: STATE_FEEDING_FILL)
Program untuk melakukan tare (reset skala) pada sensor load cell, mengendalikan pergerakan Servo 1 untuk membuka jalur pakan dari wadah utama, serta membaca berat pakan secara real-time melalui modul HX711 hingga target massa tercapai (closed-loop control).
7. Program Pendistribusian Pakan (State: STATE_FEEDING_DISPENSE)
Program untuk mengaktifkan Motor DC 775 melalui sakelar elektronik MOSFET N-Channel, menggerakkan Servo 2 guna membuka katup bawah wadah penimbang, serta memastikan pakan habis terlempar sebelum aktuator kembali dimatikan.
8. Program Deteksi Kesalahan Pakan Habis (State: STATE_ERROR_EMPTY)
Program pengaman sistem (safety check) untuk mendeteksi apabila massa aktual pakan kurang dari 80% target (pakan di penampung utama habis), yang kemudian memicu pengiriman pesan peringatan instan (alert) ke Telegram Bot.
9. Program Pelaporan Data (Cloud Logging) ke Google Spreadsheet
Program menggunakan pustaka ESP_Google_Sheet_Client (Mobizt) untuk menyusun dan mengirimkan rekam jejak operasional secara otomatis—meliputi cap waktu (timestamp), target massa, massa aktual, pemicu (trigger), dan status keberhasilan—ke basis data awan (cloud) Google Spreadsheet.
10. Program Penyegaran Antarmuka Visual (LCD 16×2)
Program untuk memperbarui informasi status operasional sistem secara real-time ke layar LCD, mencakup status jadwal aktif, berat pakan dinamis, waktu WIB, serta pesan notifikasi pengaturan.

4.2.1 Inisialisasi Sistem

ESP32 merupakan mikrokontroler utama yang digunakan sebagai pusat pengendali pada sistem Smart Feeder ikan lele berbasis IoT. Pada tahap awal program, dilakukan proses inisialisasi pustaka (library), penentuan alamat I2C (0x27 untuk LCD dan 0x21 untuk Keypad), serta konfigurasi pin ESP32 agar seluruh komponen perangkat keras dapat bekerja dengan baik.

Pada proses inisialisasi, program mendefinisikan pin-pin General Purpose Input/Output (GPIO) yang digunakan untuk menghubungkan modul I2C (SDA 21, SCL 23), sensor Load Cell HX711 (DT 34, SCK 32), aktuator motor servo (Servo 1 pin 16, Servo 2 pin 17), dan MOSFET (pin 33). Mengingat posisi pemasangan perangkat keras Keypad yang terbalik, program menerapkan penyesuaian pemetaan tombol secara perangkat lunak menjadi "D#0*C987B654A321NF" agar input tetap sesuai. Motor DC dikendalikan menggunakan logika N-Channel MOSFET dengan kondisi awal diatur ke LOW (mati).

Fungsi setup() juga menangani koneksi jaringan Wi-Fi, inisialisasi modul RTC DS3231, serta penyiapan klien Google Spreadsheet dan Telegram Bot. Guna meningkatkan keandalan sistem saat terjadi gangguan jaringan atau kedaluwarsa sertifikat, ditambahkan variabel enableGSheet yang memungkinkan pengguna menonaktifkan pengiriman data ke cloud secara opsional tanpa menghentikan fungsi utama alat.

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <I2CKeyPad.h>
#include <RTCLib.h>
#include <HX711.h>
#include <ESP32Servo.h>
#include <WiFi.h>
#include <time.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ESP_Google_Sheet_Client.h>
#define I2C_SDA 21
#define I2C_SCL 23
#define HX711_DT 34
#define HX711_SCK 32
#define SERVO1_PIN 16
#define SERVO2_PIN 17
#define MOTOR_DC_PIN 33
void setup() {
  Serial.begin(115200);
  Wire.begin(I2C_SDA, I2C_SCL);
  lcd.init();
  lcd.backlight();
  scale.begin(HX711_DT, HX711_SCK);
  scale.set_scale(calibration_factor);
  scale.tare();
  servo1.attach(SERVO1_PIN, 500, 2400);
  servo2.attach(SERVO2_PIN, 500, 2400);
  servo1.write(0);
  servo2.write(0);
  pinMode(MOTOR_DC_PIN, OUTPUT);
  digitalWrite(MOTOR_DC_PIN, LOW);
  WiFi.begin(WIFI_SSID, WIFI_PASS);
}

```

Gambar 4. 4 Konfigurasi Fungsi Setup dan Inisialisasi Hardware

4.2.2 Arsitektur Program Utama (*Main Loop*) dan Finite State Machine

Setelah proses inisialisasi selesai, mikrokontroler akan menjalankan fungsi `loop()` yang beroperasi secara terus-menerus. Sistem menerapkan teknik non-blocking delay menggunakan fungsi `millis()` yang dipadukan dengan arsitektur Finite State Machine (FSM). Arsitektur FSM membagi siklus operasional ke dalam empat kondisi diskret: `STATE_IDLE` (siaga dan evaluasi jadwal), `STATE_FEEDING_FILL` (penakaran massa pakan), `STATE_FEEDING_DISPENSE` (distribusi sentrifugal), dan `STATE_ERROR_EMPTY` (mitigasi galat/pakan habis).

Pada fungsi utama ini, diterapkan mekanisme interlock temporal untuk mencegah tabrakan interupsi (race condition) antara perangkat antarmuka. Sistem secara otomatis akan menunda penarikan pesan dari server Telegram selama 15 detik terakhir terhitung sejak pengguna melakukan interaksi fisik (menekan tombol) pada Keypad.

```

void loop() {
  if (enableGSheet) {
    GSheet.ready();
  }
  DateTime now = rtc.now();
  if (millis() - lastKeypadInteraction > 15000) {
    if (millis() - lastTelegramCheck > telegramInterval) {
      handleTelegramMessages();
      lastTelegramCheck = millis();
    }
  }
  handleKeypad(now);
  if (millis() - lastLcdUpdate > 500) {
    updateLCD(now);
    lastLcdUpdate = millis();
  }
  switch (currentState) {
    case STATE_IDLE:
      checkSchedule(now);
      break;
    case STATE_FEEDING_FILL:
      processFeedingFill();
      break;
    case STATE_FEEDING_DISPENSE:
      processFeedingDispense();
      break;
    case STATE_ERROR_EMPTY:
      handleErrorEmpty();
      break;
  }
}

```

Gambar 4.5 Arsitektur State Machine dan Multitasking Asinkron

4.2.3 Kendali Lingkar Tertutup (*Closed Loop*) Penakaran Pakan

Apabila waktu dari RTC telah sesuai dengan jadwal pakan atau terdapat pemicu manual, sistem akan bertransisi ke STATE_FEEDING_FILL. Pada fase ini, sistem menjalankan algoritma kendali lingkaran tertutup (closed-loop control) tingkat lanjut. Sebelum penakaran dimulai, sistem mengeksekusi Kalibrasi Cerdas (Multi-Point Calibration) yang secara otomatis menyesuaikan faktor kalibrasi (rentang 137.43 hingga 298.71) berdasarkan target massa pakan untuk menjaga kepresisian pembacaan sensor.

Proses pengisian pakan tidak dilakukan sekali jalan, melainkan melalui perulangan (loop) maksimal lima putaran. Pada putaran 1 hingga 4, ESP32 membuka Servo 1 secara perlahan (1 hingga 90 derajat) dan menerapkan variabel pengereman awal (fallImpactOffset). Hal ini bertujuan untuk menghentikan aliran pakan sedikit sebelum target tercapai guna mengkompensasi beban pakan yang masih melayang di udara.

Setiap kali katup ditutup, sistem melakukan jeda penstabilan selama 5 detik

untuk mengevaluasi susut mekanis (drift). Jika beban masih kurang, putaran selanjutnya otomatis dilakukan. Jika pada putaran ke-5 target belum tercapai, sistem akan memaksa katup terbuka penuh hingga 90 derajat berlawanan arah jarum jam selama 8 detik. Jika beban tetap tidak bertambah, alat secara mutlak memvonis bahwa tong pakan kosong, lalu masuk ke STATE_ERROR_EMPTY dengan mengaktifkan bendera penahan feedLeftoverInBucket.

```
void processFeedingFill() {
    int attemptCounter = 0;
    while (currentMass < (currentTargetMass - 1.5)) {
        attemptCounter++;
        float activeOffset = (attemptCounter < 5) ? fallImpactOffset : 0.0;
        for (int pos = 1; pos <= 90; pos++) {
            servo1.write(pos);
            unsigned long stepStart = millis();
            while (millis() - stepStart < 500) {
                if (scale.is_ready()) {
                    float rawRead = scale.get_units(1);
                    currentMass = (currentMass == 0.0) ? rawRead : (0.7 * currentMass) + (0.3 * rawRead);
                    if (currentMass >= (currentTargetMass - activeOffset)) break;
                }
                delay(10);
            }
            if (currentMass >= (currentTargetMass - activeOffset)) break;
        }
        servo1.write(0);
    }
    if (!pakanBenarHabis) {
        currentState = STATE_FEEDING_DISPENSE;
    } else {
        feedLeftoverInBucket = true;
        currentState = STATE_ERROR_EMPTY;
    }
}
```

Gambar 4. 6 Algoritma Kendali Lingkari Tertutup Massa Pakan

4.2.4 Mekanisme Distribusi Pakan

Setelah proses penakaran berhasil divalidasi dan wadah tidak terdeteksi kosong, sistem beralih ke STATE_FEEDING_DISPENSE. Algoritma dirancang untuk memastikan penyebaran pakan yang optimal di area kolam. ESP32 terlebih dahulu memberikan sinyal logika HIGH pada gate MOSFET (MOTOR_DC_PIN) untuk menyalakan motor DC pelontar. Terdapat injeksi latensi buatan selama 2000 milidetik sebelum katup penampung di bawahnya dibuka; hal ini krusial untuk memberikan momentum bagi impeler agar mencapai kecepatan putar maksimumnya sebelum dibebani oleh pelet pakan.

Berbeda dengan Servo 1, pembukaan Servo 2 juga dilakukan secara inkremental (bertahap 0 hingga 90 derajat dengan jeda 15 milidetik per derajat) untuk mencegah pakan tumpah secara drastis yang dapat memicu kemacetan pada

impeler pelontar. Perulangan pemantauan kemudian digunakan untuk menahan Motor DC dan Servo 2 agar tetap aktif hingga sensor Load Cell menunjukkan sisa beban di bawah 5 gram, atau akan dihentikan secara otomatis apabila telah melewati batas waktu tunggu (timeout) maksimal selama 15 detik.

```
void processFeedingDispense() {
    digitalWrite(MOTOR_DC_PIN, HIGH);
    delay(2000);
    for (int pos = 0; pos <= 90; pos++) {
        servo2.write(pos);
        delay(15);
    }
    unsigned long startWait = millis();
    while(true) {
        float s2_mass = scale.get_units(3);
        if (s2_mass < 5.0) break;
        if (millis() - startWait > 15000) break;
    }
    servo2.write(0);
    delay(1000);
    digitalWrite(MOTOR_DC_PIN, LOW);
    lastFeedSuccess = true;
    feedLeftoverInBucket = false;
    sendToGoogleSheets("BERHASIL");
    autoSwitchToNextSchedule();
    currentState = STATE_IDLE;
}
```

Gambar 4. 7 Algoritma Mekanisme Distribusi Pakan

4.2.5 Integrasi Antarmuka Komunikasi (Telegram dan Keypad)

Fungsionalitas kontrol sistem diperluas dengan penerapan antarmuka luring via Keypad I2C dan antarmuka daring via Telegram Bot API. Modul Keypad diolah oleh fungsi handleKeypad() untuk mengubah mode antarmuka LCD (uiMode), menetapkan variabel jam, dan massa pakan secara langsung tanpa internet.

Sementara itu, komunikasi Telegram dikelola melalui fungsi handleTelegramMessages(). Sistem menerapkan teknik parsing data untuk membedah teks perintah dari pengguna. Algoritma menggunakan sub-state Telegram (seperti TELEGRAM_WAIT_JAM dan TELEGRAM_WAIT_MASSA) untuk memandu pengguna memasukkan nilai parameter yang presisi secara bertahap, serta menjalankan perintah pakan seketika (/feednow) yang dapat menginterupsi kondisi STATE_IDLE untuk langsung melompat ke fase penakaran pakan.

4.2.6 Sistem Rekam Jejak Historis (*Cloud Datalogging*)

Sebagai penyempurnaan implementasi Internet of Things (IoT), sistem dilengkapi dengan fitur pelaporan telemetri persisten menggunakan pustaka ESP_Google_Sheet_Client. Setiap kali siklus pakan selesai (baik sukses maupun gagal karena pakan habis), ESP32 menyusun paket data berformat JSON. Paket data ini merangkum parameter komprehensif yang meliputi: cap waktu kejadian (berdasarkan RTC), target massa, massa aktual, jenis pemicu (contoh: "Auto" atau "Telegram"), serta status keberhasilan. Data tersebut kemudian ditransmisikan melalui metode REST API (Hypertext Transfer Protocol Secure) langsung ke baris basis data Google Spreadsheet, menjamin keberadaan jejak audit operasional (audit trail) guna evaluasi manajemen pakan lele yang berkelanjutan.