



**CYBER-PHYSICAL SYSTEM FINAL PROJECT REPORT
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITAS INDONESIA**

AQUAFEED

GROUP 17

YOEL DWI MIRYANO	2206059534
SHARIF FATIH ASAD MASYHUR	2206063014
SIHOMBING GIOVANO GERALDO	2206059566
MUHAMMAD LUTFI SETIADI	2206059805

PREFACE

Kami persembahkan laporan proyek ini yang berjudul AquaFeed yang dikembangkan sebagai proyek dari Praktikum Sistem Siber Fisik. Proyek ini adalah pemberi makan ikan otomatis yang memudahkan pengguna untuk memberi makan secara teratur. Dilengkapi dengan tiga tombol, pengguna dapat memilih tombol waktu pemberian makanan (2, 3, atau 4 detik) dan tombol interrupt untuk menghentikan berjalannya servo. Ketika timer mencapai nol, motor servo membuka wadah makanan sambil menyalakan LED dan buzzer sebagai indikator. Informasi pilihan waktu dan status operasi ditampilkan melalui Serial Monitor. Setelah selesai, motor servo kembali ke posisi awal. Proyek ini dirancang dengan penekanan pada kemudahan penggunaan, efisiensi, serta keamanan bagi ikan dan pemiliknya.

Pendekatan kami melibatkan riset yang mendalam mengenai sistem pemberi makan ikan otomatis agar dapat dengan cermat merencanakan dan mendesain segala fungsionalitas dan mekanisme pada proyek kami. Laporan proyek ini pun berfungsi sebagai dokumentasi singkat tentang perkembangan proyek kami, tantangan yang dihadapi, pembagian tugas dan tanggung jawab, serta solusi akhir yang ditemukan.

Kami ingin mengucapkan rasa terima kasih kepada seluruh tim asisten Digital Laboratorium dan teman-teman kami atas bimbingan dan dukungan yang mereka berikan. Proyek ini merupakan hasil kerja kolektif kami, dan komitmen kami dalam mengembangkan sistem pemberi makan ikan otomatis yang praktis dan efektif. Dukungan dan kerjasama dari berbagai pihak telah memungkinkan kami untuk menyelesaikan proyek ini dengan baik dan sesuai dengan tujuan yang telah ditetapkan.

Depok, May 18, 2024

Group 17

TABLE OF CONTENTS

CHAPTER 1	4
INTRODUCTION	4
1.1 PROBLEM STATEMENT	4
1.3 ACCEPTANCE CRITERIA	5
1.4 ROLES AND RESPONSIBILITIES	6
1.5 TIMELINE AND MILESTONES	6
CHAPTER 2	7
IMPLEMENTATION	7
2.1 HARDWARE DESIGN AND SCHEMATIC	7
2.2 SOFTWARE DEVELOPMENT	7
2.3 HARDWARE AND SOFTWARE INTEGRATION	9
CHAPTER 3	17
TESTING AND EVALUATION	19
3.1 TESTING	19
3.2 RESULT	19
3.3 EVALUATION	21
CHAPTER 4	22
CONCLUSION	22

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

Memberi makan ikan secara teratur dan tepat waktu sangat penting untuk kesejahteraan ikan hias, baik yang dipelihara di akuarium rumah maupun kolam komersial. Namun, tuntutan kehidupan sehari-hari seringkali menyulitkan pemilik ikan untuk menjaga jadwal pemberian makan yang konsisten. Keterlambatan atau ketidakteraturan dalam pemberian pakan dapat berdampak buruk pada kesehatan ikan, kualitas air, dan bahkan menyebabkan kematian ikan. Oleh karena itu, terdapat kebutuhan mendesak akan solusi otomatis yang dapat memastikan ikan diberi makan secara teratur tanpa memerlukan pengawasan terus-menerus dari pemiliknya.

Proyek ini bertujuan untuk mengatasi tantangan merancang sistem yang dapat mengatur waktu pemberian pakan ikan secara otomatis, memberikan informasi terkini operasional secara real-time kepada pengguna, dan memastikan mekanisme pemberian pakan berjalan lancar dan tepat. Sistem harus mudah digunakan, hemat energi, dan dapat diandalkan dalam jangka waktu lama. Selain itu, produk ini harus dapat disesuaikan dengan beragam kebutuhan pengguna dengan jadwal pemberian makan yang berbeda.

Inspirasi proyek ini muncul dari observasi pribadi, pengalaman, dan wawasan dari berbagai sumber literatur yang membahas pentingnya otomatisasi dalam budidaya ikan. Sejumlah penelitian dan artikel telah menjelaskan kesulitan yang dihadapi pemilik akuarium dalam menjaga jadwal pemberian makan yang konsisten. Menyadari adanya peluang untuk mengatasi masalah ini dan meningkatkan kualitas hidup ikan serta kenyamanan pemiliknya, kami mulai mengembangkan sistem pemberian pakan ikan otomatis berbasis Arduino.

1.2 PROPOSED SOLUTION

Menanggapi tantangan jadwal pemberian pakan ikan yang tidak konsisten, kami mengusulkan solusi inovatif berupa sistem pemberian pakan ikan otomatis berbasis Arduino. Sistem ini dirancang dengan cermat untuk mengeluarkan makanan ikan pada interval yang telah ditentukan, memastikan bahwa hewan peliharaan air Anda menerima makanan secara

teratur dan tepat waktu. Komponen utama dari proyek ini termasuk Arduino Uno sebagai mikrokontroler utama, motor servo yang bertanggung jawab untuk membuka dan menutup wadah makanan, lampu LED dan buzzer untuk umpan balik visual dan pendengaran, serta tombol tekan untuk memprogram waktu makan. Selain itu, integrasi resistor, breadboard, dan kabel jumper yang andal sangat penting untuk menciptakan sirkuit fungsional dan memberi daya pada Arduino.

Dalam hal pemrograman, kami memanfaatkan kemampuan Arduino IDE yang kuat, memberdayakan pengguna untuk mengembangkan, menguji, dan mengunggah kode ke sirkuit Arduino. Kode yang dibuat dengan cermat memungkinkan sistem menafsirkan input dari tombol dengan pilihan durasi waktu yang fleksibilitas, mengoperasikan motor servo, dan memberikan umpan balik real-time melalui LED dan buzzer terintegrasi. Selain itu, informasi penting mengenai pilihan waktu dan status operasional mudah diakses melalui serial monitor di Arduino IDE atau aplikasi software sejenisnya, memungkinkan pengguna untuk memantau dan memverifikasi fungsionalitas sistem yang sempurna. Dengan perpaduan kesederhanaan antara perangkat keras dan perangkat lunak yang dibuat dengan cermat, sistem pemberian pakan ikan otomatis beroperasi dengan lancar, sekaligus mengurangi tanggung jawab perawatan ikan dari pemiliknya.

1.3 ACCEPTANCE CRITERIA

The acceptance criteria of this project are as follows:

1. **Kegunaan:** Pengguna akan merasa mudah mengoperasikan alat ini. Memiliki tombol atau interface yang jelas dan intuitif akan meningkatkan keberhasilan proyek.
2. **Stabilitas dan Keandalan:** Dalam jangka waktu yang lama, kinerja alat harus stabil dan dapat diandalkan. Keberhasilan ditunjukkan jika gadget dapat beroperasi terus menerus tanpa memerlukan keterlibatan manusia yang sering.
3. **Konsumsi Energi:** Perangkat perlu menggunakan lebih sedikit energi. Proyek ini akan lebih sukses dan efisien jika alat tersebut dapat dijalankan dengan listrik yang lebih sedikit.
4. **Kemampuan Via Serial Monitor:** Perangkat harus menawarkan fitur untuk pemantauan, termasuk layar serial monitor. Keberhasilan proyek akan meningkat jika pengguna dapat memantau status pemberian makan.

1.4 ROLES AND RESPONSIBILITIES

The roles and responsibilities assigned to the group members are as follows:

Roles	Responsibilities	Person
Role 1	Membuat dan debugging program, mengimplementasi desain pada software dan hardware.	Yoel Dwi Miryano
Role 2	Membuat Github repo dan menyusun laporan proyek, Flowchart, PPT, serta README.md pada Github.	Sharif Fatih Asad Masyhur
Role 3	Memantau progress proyek, mendesain komponen hardware dan software serta menyusun algoritma program.	Sihombing Giovano Geraldo
Role 4	Memastikan laporan dan dokumentasi yang dibuat terkomprehensif dan rapih.	Muhammad Lutfi Setiadi

Table 1. Roles and Responsibilities

1.5 TIMELINE AND MILESTONES

Kegiatan	Mei																
	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Hardware Design completion																	
Software Development																	
Integration and Testing of Hardware and Software																	
Final Product Assembly and Testing																	

Table 2. Timeline and Milestones

CHAPTER 2

IMPLEMENTATION

2.1 HARDWARE DESIGN AND SCHEMATIC

Arduino Uno mengontrol waktu pengumpanan dan motor servo. Pengguna memilih durasi makan melalui tombol tekan. Motor servo membuka dan menutup tutup wadah makanan pada waktu yang ditentukan, sedangkan LED dan buzzer memberikan indikator visual dan pendengaran. Informasi ditampilkan melalui serial monitor.

Setelah jadwal pemberian makan dipilih, pengatur waktu internal akan menghitung mundur. Ketika pengatur waktu mencapai nol, motor servo membuka tutupnya, LED menyala, dan buzzer berbunyi. Setelah durasi pemberian makan, motor servo menutup wadah makanan, dan LED serta buzzer mati. Sistem dapat mengulangi proses pemberian makan tergantung pada button yang dipilih oleh pengguna (2 detik, 3 detik, 4 detik, dan interrupt button). Proyek ini mencakup penampilan informasi seperti durasi waktu pada layar serial monitor. Semua komponen memerlukan sumber tegangan dan output utamanya adalah pergerakan motor servo yang mengatur feeding sesuai jadwal yang telah diatur sebelumnya.

2.2 SOFTWARE DEVELOPMENT

Pengembangan software dari proyek AquaFeed terbentuk menjadi beberapa komponen dan inti program yang kita adaptasi dari modul praktikum dengan mengembangkannya melalui beberapa komponen seperti:

- Inti Program

Bagian ini mendefinisikan subrutin utama, yang mencakup inisialisasi untuk semua subrutin penting yang digunakan dalam proyek. Subrutin ini melibatkan tombol tekan, motor servo, LED, buzzer, dan serial monitor. Semua komponen akan diinisialisasi terlebih dahulu. Sistem kemudian akan memeriksa input dari tombol tekan, mengatur timer, dan mengontrol pengoperasian motor servo, LED, dan buzzer berdasarkan waktu yang dipilih.

- Inisialisasi Komponen

Pada bagian ini, semua komponen perangkat keras seperti tombol tekan, motor servo, LED, buzzer, dan serial monitor. Inisialisasi dilakukan dengan mengatur Data Direction Register (DDR) dan port yang sesuai.

- Memilih Durasi Pemberian Makan

Bagian ini menunggu masukan dari tombol tekan untuk memilih durasi pemberian makan. Subrutin memeriksa status setiap tombol dan menyimpan durasi yang dipilih dalam register.

- Motor Servo, LED, dan Kontrol Buzzer

Subrutin ini mengontrol motor servo untuk membuka dan menutup wadah makanan ikan, serta menyalakan dan mematikan LED dan buzzer berdasarkan waktu yang telah ditentukan. Pengatur waktu menghitung mundur menggunakan register dan loop penundaan.

- Pembacaan Inputan pada Serial Monitor

Pembacaan inputan button diubah dan ditampilkan pada Serial Monitor. Jika kondisi kurang ideal maka buzzer akan diaktifkan sebagai notifikasi.

Proyek dimulai dengan inisialisasi sistem, di mana DDR dan PORT untuk semua komponen ditetapkan. Kemudian serial monitor diinisialisasi untuk menampilkan pesan awal. Selama pemilihan pengatur waktu, sistem memeriksa status tombol tekan dan menyimpan pilihan durasi (2 detik, 3 detik, 4 detik) di register. Selanjutnya, pilihan durasi ditampilkan pada serial monitor, dan pengatur waktu mulai menghitung mundur dari durasi yang dipilih. Ketika pengatur waktu mencapai nol, motor servo diaktifkan untuk membuka wadah makanan. LED dan buzzer juga diaktifkan pada saat ini. Setelah durasi feeding selesai, motor servo dikembalikan ke posisi awal, dan LED serta buzzer dimatikan. Program perakitan terstruktur untuk proyek AquaFeed ini dapat diterapkan secara sistematis untuk memastikan bahwa setiap komponen berfungsi sesuai yang diharapkan. Berikut adalah skema dari proyek AquaFeed dalam bentuk flowchart:

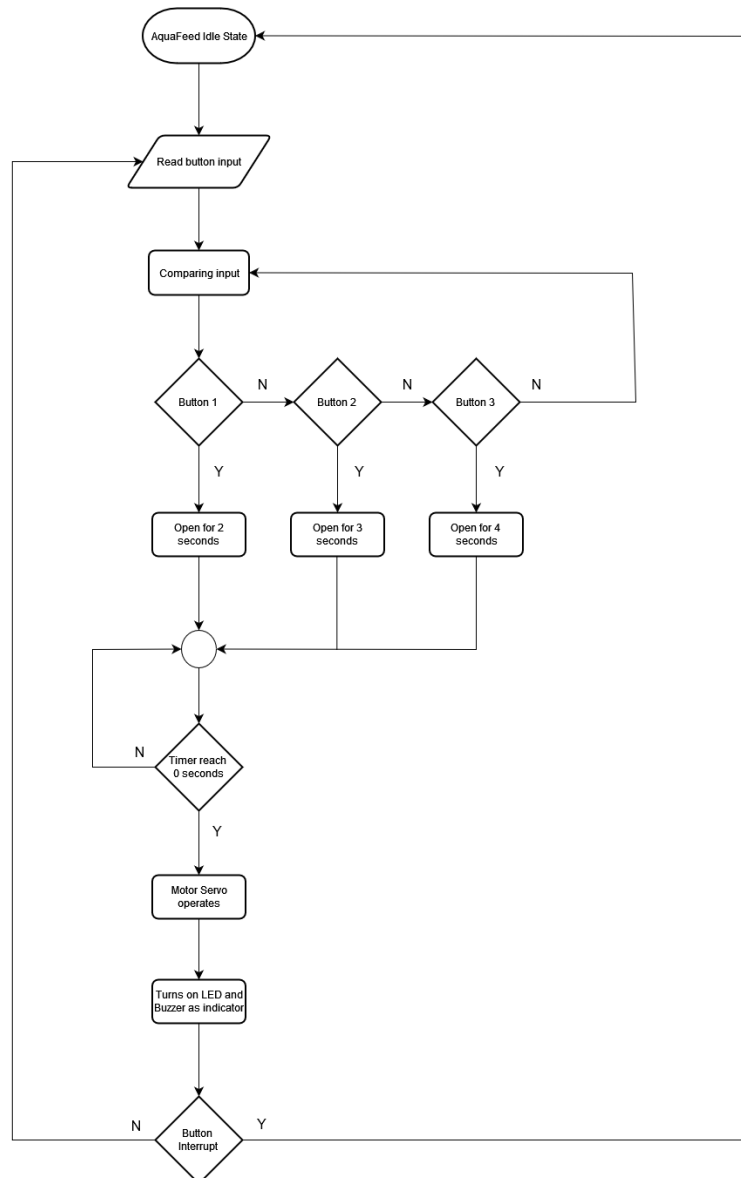


Fig 1. Flowchart

2.3 HARDWARE AND SOFTWARE INTEGRATION

Proyek AquaFeed mengintegrasikan komponen perangkat keras dan perangkat lunak untuk mencapai tujuan memberi makan ikan secara otomatis. Proyek ini menggabungkan tombol tekan, motor servo, LED, buzzer, dan serial monitor untuk memberikan pengalaman pemberian makan yang andal dan informatif. Perangkat lunak pada Arduino Uno mengontrol semua komponen perangkat keras melalui kode yang ditulis dalam bahasa Assembly dan C++ untuk interrupt, memastikan operasi sinkron sesuai kebutuhan pengguna.

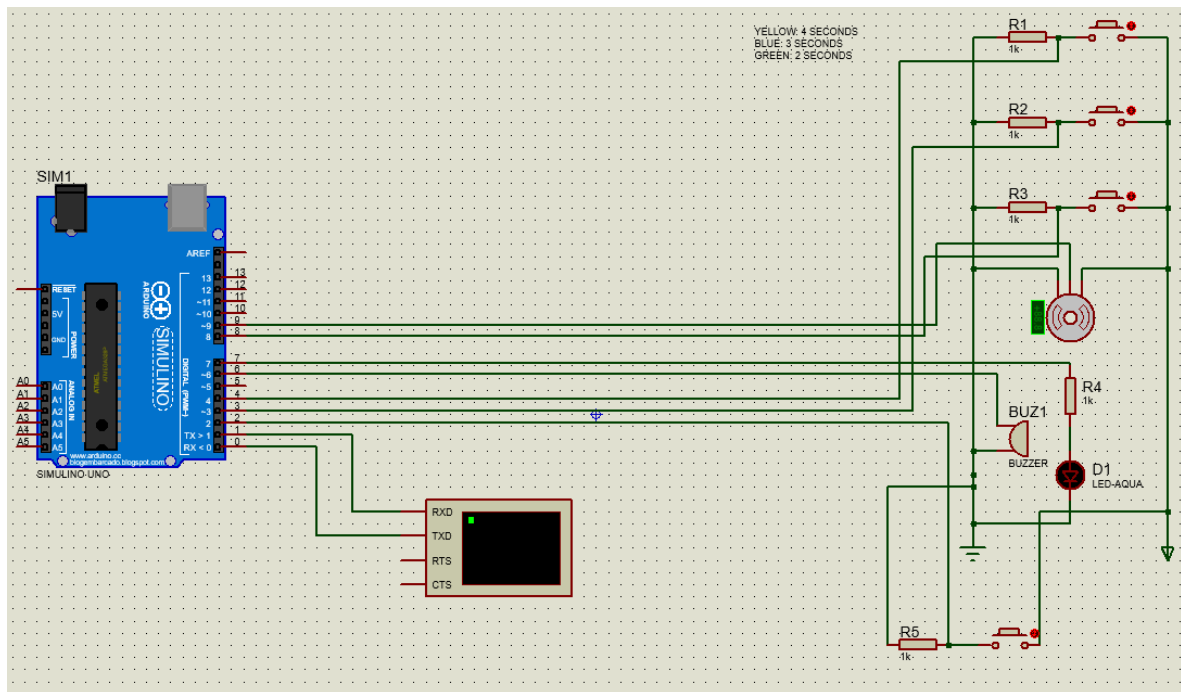


Fig 2. Rangkaian Proteus

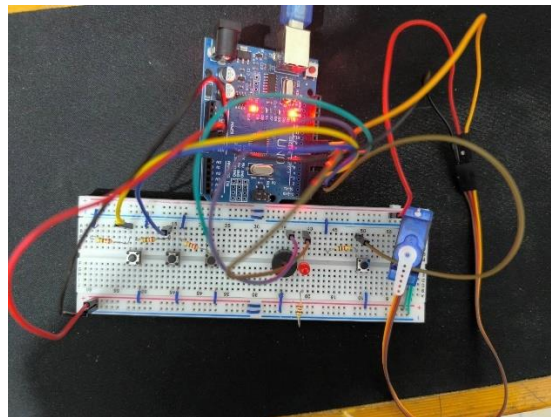


Fig 3. Rangkaian Asli

Berikut adalah source code dari proyek AquaFeed (file .S dan .ino):

```
#define __SFR_OFFSET 0x00
#include <avr/io.h>
#include <avr/interrupt.h>

.extern status

.global INIT

/*
GREEN      = 2 Seconds
BLUE       = 3 Seconds
YELLOW     = 4 Seconds
DELAY BUKA = 1 Second
```

```

*/

INIT:
    RCALL INIT_SERIAL

    CBI    DDRB, 0
    CBI    DDRD, 3
    CBI    DDRD, 4

//BUTTONS
BUTTON:
    SBIC   PINB, 0
    RCALL  GREEN

    SBIC   PIND, 3
    RCALL  BLUE

    SBIC   PIND, 4
    RCALL  YELLOW

    RET

//BUTTON ACTION
GREEN:
    LDI    R30, lo8(GREEN_MESSAGE)
    LDI    R31, hi8(GREEN_MESSAGE)
GREEN_AGAIN:
    LPM    R18, Z+
    CPI    R18, 0
    BREQ   GREEN_LOOP

    RCALL  LCD_BUFFER

    STS    UDR0, R18

    RJMP   GREEN_AGAIN

GREEN_MESSAGE:
    .ascii "Delay 2 seconds"
    .byte 10, 13, 0
GREEN_LOOP:
    RCALL  SERVO
    RCALL  GREEN_2S

    LDS    R29, status
    CPI    R29, 1
    BREQ   BUTTON

```

```

    RJMP    GREEN_LOOP

BLUE:
    LDI     R30, lo8(BLUE_MESSAGE)
    LDI     R31, hi8(BLUE_MESSAGE)
BLUE_AGAIN:
    LPM     R18, Z+
    CPI     R18, 0
    BREQ    BLUE_LOOP

    RCALL   LCD_BUFFER

    STS     UDR0, R18

    RJMP    BLUE_AGAIN

BLUE_MESSAGE:
    .ascii "Delay 3 seconds"
    .byte 10, 13, 0
BLUE_LOOP:
    RCALL   SERVO
    RCALL   BLUE_3S

    LDS     R29, status
    CPI     R29, 1
    BREQ    BUTTON

    RJMP    BLUE_LOOP

YELLOW:
    LDI     R30, lo8(YELLOW_MESSAGE)
    LDI     R31, hi8(YELLOW_MESSAGE)
YELLOW_AGAIN:
    LPM     R18, Z+
    CPI     R18, 0
    BREQ    YELLOW_LOOP

    RCALL   LCD_BUFFER

    STS     UDR0, R18

    RJMP    YELLOW_AGAIN

YELLOW_MESSAGE:
    .ascii "Delay 4 seconds"
    .byte 10, 13, 0
YELLOW_LOOP:
    RCALL   SERVO

```

```

RCALL YELLOW_4S

LDS    R29, status
CPI    R29, 1
BREQ   YELLOWFIN

RJMP   YELLOW_LOOP
YELLOWFIN:
RJMP   BUTTON

//SERIAL MONITOR
INIT_SERIAL:
CLR    R24
STS    UCSR0A, R24           ;clear UCSR0A register
STS    UBRR0H, R24           ;clear UBRR0H register
LDI    R24, 103               ;& store in UBRR0L 103 value
STS    UBRR0L, R24           ;to set baud rate 9600
LDI    R24, 1<<RXEN0 | 1<<TXEN0 ;enable RXB & TXB
STS    UCSR0B, R24
LDI    R24, 1<<UCSZ00 | 1<<UCSZ01 ;asynch, no parity, 1 stop, 8 bits
STS    UCSR0C, R24

RET

LCD_BUFFER:
LDS    R17, UCSR0A
SBRS   R17, UDRE0             ;test data buffer if data can be sent
RJMP   LCD_BUFFER

RET

//BUTTON DELAYS
GREEN_2S:
.EQU   value, 34286           ;2 Seconds

LDI    R20, hi8(value)
STS    TCNT1H, R20
LDI    R20, lo8(value)
STS    TCNT1L, R20

RCALL  TIMER_INIT

RET

BLUE_3S:
.EQU   value, 18661           ;3 Seconds

LDI    R20, hi8(value)

```

```

    STS    TCNT1H, R20
    LDI     R20, lo8(value)
    STS    TCNT1L, R20

    RCALL  TIMER_INIT

    RET

YELLOW_4S:
    .EQU   value, 3036           ;4 Seconds

    LDI     R20, hi8(value)
    STS    TCNT1H, R20
    LDI     R20, lo8(value)
    STS    TCNT1L, R20

    RCALL  TIMER_INIT

    RET

//SERVO
SERVO:
    SBI     DDRB, 1 ;SERVO
    SBI     DDRD, 6 ;BUZZER
    SBI     DDRD, 7

SERVO_AGAIN:
    LDI     R26, 2
    LDI     ZL, lo8(POSITION)
    LDI     ZH, hi8(POSITION)
SERVO_LOOP:
    LPM     R24, Z+
    RCALL  BUZZER
    SBI     PORTD, 7
    RCALL  DELAY_1S
    RCALL  ROTATE_SERVO
    DEC     R26
    CBI     PORTD, 7
    BRNE   SERVO_LOOP

    RET

POSITION:
    .byte  40, 90

ROTATE_SERVO:
    LDI     R20, 3
SERVO_LOOP2:
    SBI     PORTB, 1

```

```
RCALL SERVO_PULSE
CBI PORTB, 1
RCALL DELAY_1S
DEC R20
```

```
BRNE SERVO_LOOP2
```

```
RET
```

SERVO_PULSE:

```
CLR R21
OUT TCNT0, R21
MOV R21, R24
OUT OCR0A, R21
LDI R21, 0b00001100
OUT TCCR0B, R21 ;timer0: CTC mode, prescaler 256

LDI R21, (1<<TOV0)
OUT TIFR0, R21
```

SERVO_PULSE_LOOP:

```
IN R21, TIFR0
SBRS R21, OCF0A
RJMP SERVO_PULSE_LOOP
```

```
CLR R21
OUT TCCR0B, R21
```

```
LDI R21, (1<<OCF0A)
OUT TIFR0, R21
```

```
LDI R21, 0
OUT TCCR0A, R21
OUT TCCR0B, R21
LDI R17, 0
```

```
RET
```

//BUZZER

BUZZER:

```
LDI R20, 0b10001011 ; Timer0 in Fast PWM mode, prescaler 64
OUT TCCR0A, R20
LDI R20, 0b00000011 ; Start timer with prescaler 64
OUT TCCR0B, R20
```

```
RET
```

//DELAYS

```

DELAY_1S:
    LDI    R21, 255
14:
    LDI    R22, 255
15:
    LDI    R23, 15
16:
    DEC    R23
    BRNE   16
    DEC    R22
    BRNE   15
    DEC    R21
    BRNE   14

    RET

TIMER_INIT:
    CLR    R20
    STS    TCCR1A, R20
    LDI    R20, 0x05
    STS    TCCR1B, R20                ;timer 1, normal, 1024 prescaler

L1:
    IN     R20, TIFR1
    SBRS   R20, TOV1
    RJMP   L1

    LDI    R20, 1 << TOV1
    OUT    TIFR1, R20

    CLR    R20
    STS    TCCR1B, R20

    RET

```

File interrupt.ino:

```

uint8_t btn_pin = 2;
volatile uint8_t status = 0;

extern "C"{
    void INIT();
}

void setup(){
    DDRD &= ~(1 << btn_pin);
    PORTD != (1 << btn_pin);

    attachInterrupt(digitalPinToInterrupt(btn_pin), button, FALLING);
}

```



```

}
void loop(){
  INIT();
  status = 0;
}
void button(){
  status = 1;
}

```

Pada integrasi software menggunakan bahasa Assembly, kita dapat menganalisis fungsionalitas dari subroutine dan algoritmanya yang memainkan dibalik layar dari berjalannya proyek AquaFeed:

```

INIT:
  RCALL INIT_SERIAL

  CBI   DDRB, 0
  CBI   DDRD, 3
  CBI   DDRD, 4

```

Pada subroutine INIT, menggunakan identifikasi serial untuk menentukan status pesan. Konfigurasi ulang port input untuk ketiga tombol yang akan digunakan untuk menentukan durasi pemberian pakan.

```

BUTTON:
  SBIC  PINB, 0
  RCALL GREEN

  SBIC  PIND, 3
  RCALL BLUE

  SBIC  PIND, 4
  RCALL YELLOW

  RET

```

Subroutine Tombol (BUTTON) menentukan mana dari tiga tombol (hijau, biru, atau kuning) yang ditekan. Fungsionalitas dari subroutine ini mengurutkan subrutin menurut simbol yang ditentukan (HIJAU, BIRU, atau KUNING).

```

SERVO:
  SBI   DDRB, 1 ;SERVO
  SBI   DDRD, 6 ;BUZZER
  SBI   DDRD, 7

```

Servo Subrutinum (SERVO) berperan untuk mengatur aliran servo yang menggerakkan dayung. Subroutine Servo menyesuaikan dengan namanya Motor Servo yang akan menyalakan buzzer sebagai isyarat pendengaran.

```
BUZZER:
LDI    R20, 0b10001011          ; Timer0 in Fast PWM mode, prescaler 64
OUT    TCCR0A, R20
LDI    R20, 0b00000011          ; Start timer with prescaler 64
OUT    TCCR0B, R20

RET
```

Terakhir ada subroutine Buzzer yang akan mengaktifkan buzzer pada saat proses pembuatan pakan untuk ikan.

Secara keseluruhan, program ini memungkinkan penghitungan ikan secara otomatis dengan durasi yang dapat diatur menggunakan tombol, serta memberikan isyarat visual dan aural melalui komunikasi serial dan buzzer.

CHAPTER 3

TESTING AND EVALUATION

3.1 TESTING

Pada tahap pengujian, kami membagi proses menjadi beberapa kasus berdasarkan kriteria keberhasilan yang ditentukan pada bab pertama. Pertama, kami memastikan semua tombol terhubung dengan benar dan dapat mengatur durasi feeding. Pada Pengujian Motor Servo, kami memeriksa apakah motor servo membuka dan menutup wadah makanan sesuai durasi yang dipilih. Kita juga perlu memastikan bahwa sistem dapat beroperasi dalam durasi waktu yang ditentukan (2 detik, 3 detik, dan 4 detik). Setelah itu, LED menyala dan buzzer berbunyi saat motor servo membuka wadah makanan, dan keduanya mati setelah wadah ditutup. Berikutnya adalah Pengujian serial monitor, memastikan serial monitor menampilkan informasi yang benar tentang sisa waktu.

3.2 RESULT

Proyek AquaFeed berjalan dengan baik, aman, dan sesuai anggaran serta memberikan hasil yang diharapkan. Prosedur analisis kebutuhan yang menyeluruh menjamin bahwa setiap komponen penting dari sistem pemberian makan otomatis telah diperhitungkan. Hal ini mencakup durasi pemberian makan serta isyarat suara dan visual bahwa buzzer berfungsi dengan baik. Metodologi desain modular memfasilitasi pengembangan, debugging, dan perbaikan yang lebih mudah. Meskipun setiap bagian dari sistem terintegrasi dengan baik, sistem ini dimaksudkan untuk beroperasi secara independen. Fitur penyesuaian inputan button untuk pengguna memudahkan perubahan durasi pemberian makan, memberi makanan ikan yang dinamis dan fleksibel untuk penggunaan sehari-hari. Penambahan fungsi di masa depan seperti integrasi aplikasi seluler atau penilaian kualitas air dimungkinkan berkat arsitektur sistem.

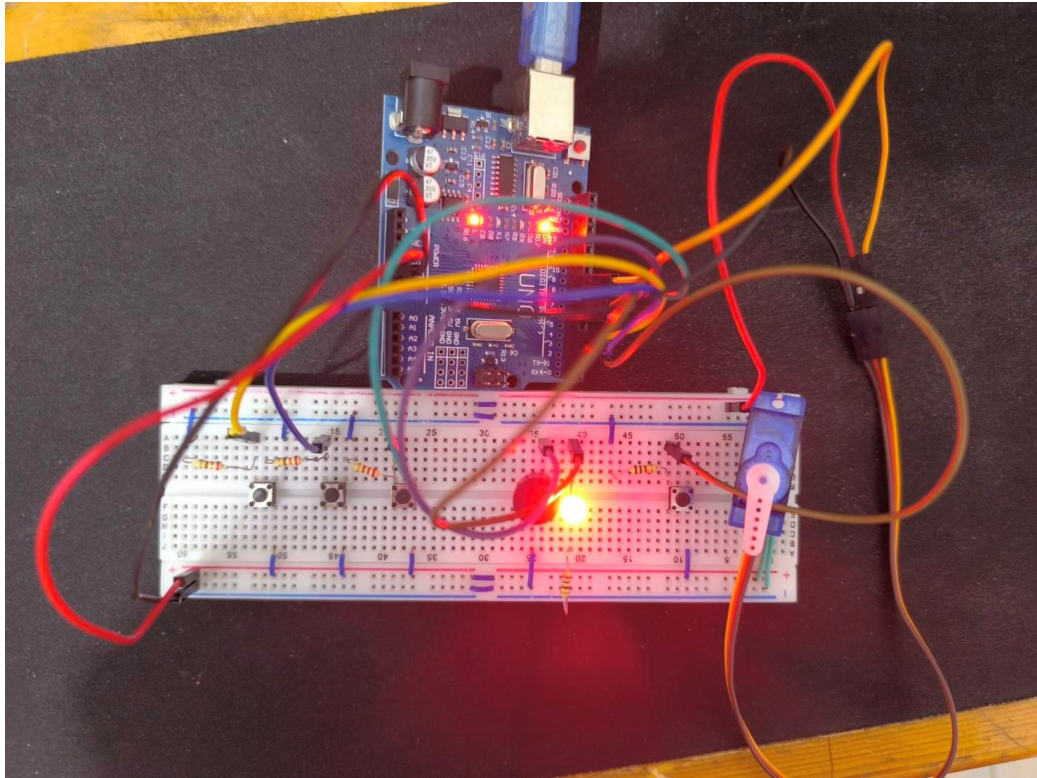


Fig 4. Hasil Testing pada Rangkaian Asli

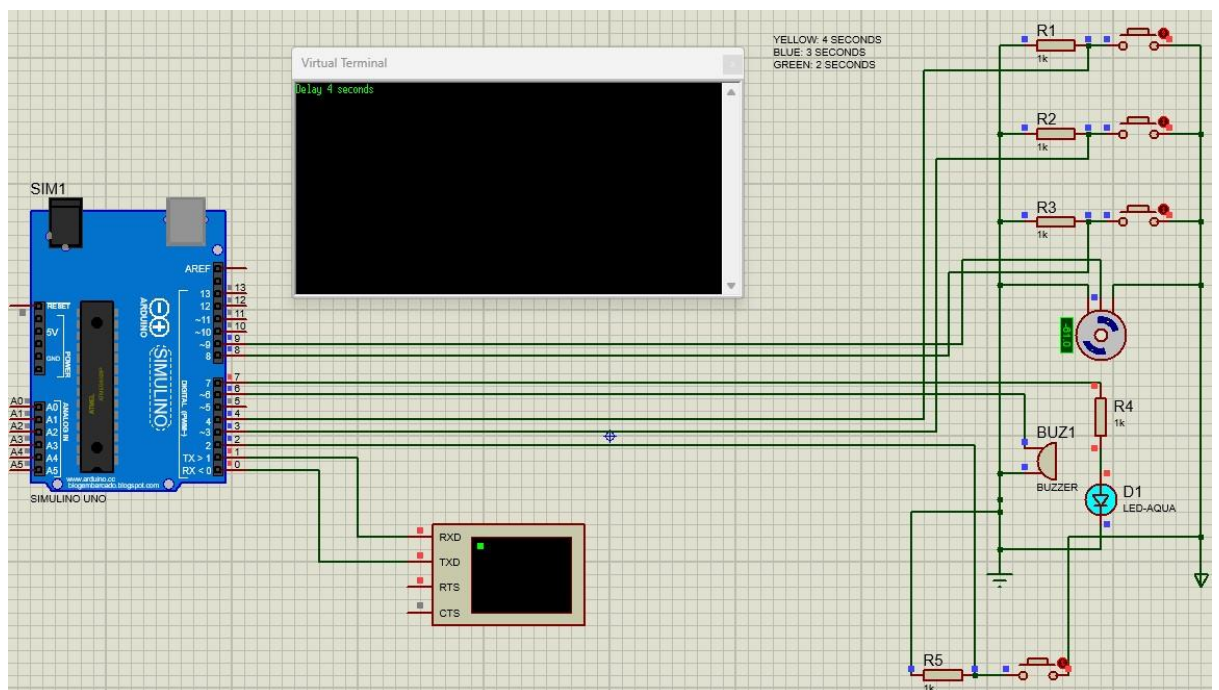


Fig 5. Output Setelah Inputan Tombol Delay 4s

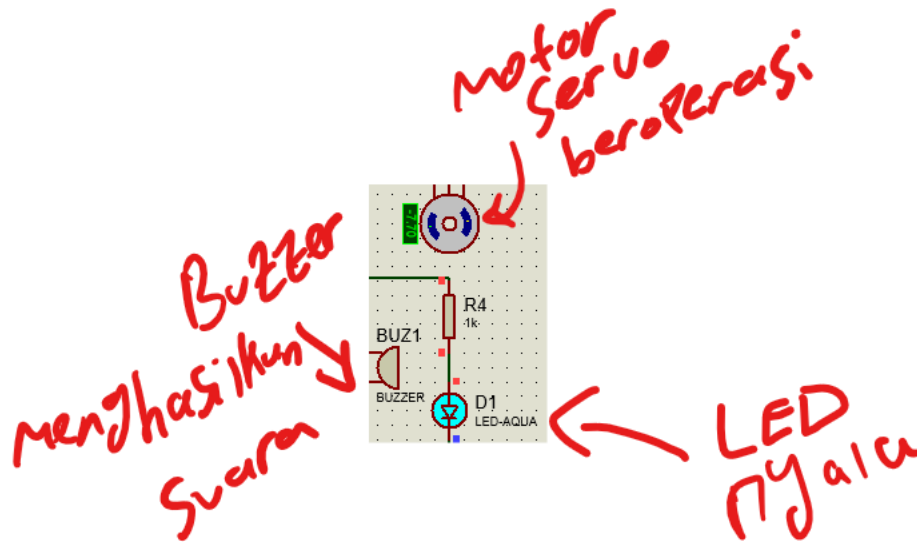


Fig. 6 Bukti hasil testing pada Proteus

Keberhasilan proyek AquaFeed dapat dikaitkan dengan kombinasi pengalaman pengguna yang kuat, pemilihan komponen berkualitas tinggi, implementasi perangkat lunak yang efisien, pelatihan pengguna yang komprehensif, sistem indikator yang fleksibel, dan dokumentasi yang kuat. Dengan gabungan semua faktor ini, AquaFeed dapat beroperasi secara otomatis, stabil, dan memberikan hasil yang berarti untuk penyiapan makan ikan, berbeda dengan metode manual atau sistem sebelumnya.

3.3 EVALUATION

Proyek AquaFeed berhasil berjalan dengan baik dan memenuhi kriterianya. Selama pengembangan, terdapat kesulitan dalam mengimplementasikan interrupt pada program Assembly. Interrupt diperlukan untuk menangani kejadian asinkron seperti penekanan tombol dan pembacaan secara efisien. Beralih dengan menggunakan bantuan file .ino memberikan solusi yang efektif untuk mengurangi kompleksitas dan meningkatkan efisiensi pengembangan. Sistem berfungsi dengan baik, memberikan pengalaman yang memuaskan bagi pengguna dalam pemberian makan ikan secara otomatis. Evaluasi ini menunjukkan pentingnya fleksibilitas dalam pendekatan pengembangan dan manfaat dari penggunaan bahasa pemrograman yang sesuai dengan kebutuhan proyek.

CHAPTER 4

CONCLUSION

Proyek AquaFeed telah berhasil diselesaikan sebagai sistem pemberian pakan ikan otomatis, yang memungkinkan pengguna dengan mudah menjadwalkan pemberian pakan secara teratur. Sistem ini secara efektif mengoperasikan motor servo untuk membuka dan menutup wadah makanan ikan berdasarkan durasi yang dipilih pengguna menggunakan tombol tekan. Pengguna dapat dengan mudah memilih durasi pemberian makan (2 detik, 3 detik, 4 detik) dan sistem akan meresponsnya. Integrasi komponen perangkat keras dengan perangkat lunak yang ditulis dalam bahasa Assembly dan C++ berhasil. Semua komponen bekerja sama secara mulus untuk memberikan pengalaman pengguna yang optimal.

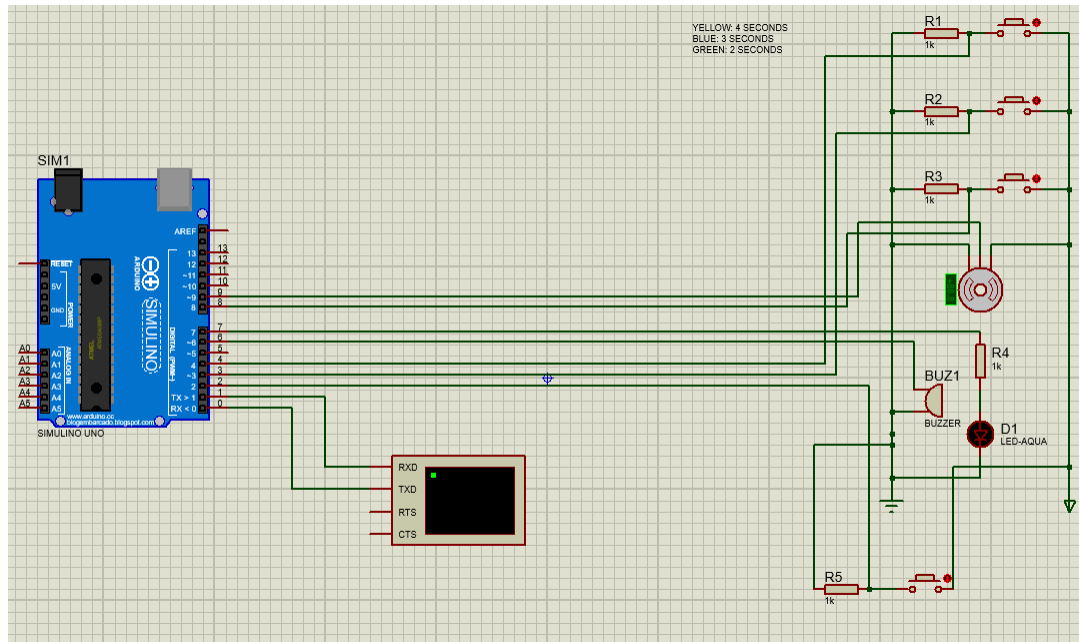
Secara keseluruhan, proyek AquaFeed mewakili kemajuan signifikan dalam otomatisasi budidaya ikan, menawarkan solusi yang efektif, efisien, dan andal kepada pengguna. Dengan kinerja yang kuat dari seluruh komponen dan integrasi yang kuat antara perangkat keras dan perangkat lunak, proyek ini telah mencapai semua tujuannya.

REFERENCES

- [1] syedzainnasir, "Arduino UNO Library for Proteus V3.0 - The Engineering Projects," *www.theengineeringprojects.com*, Nov. 06, 2023. [Online]. Available: <https://www.theengineeringprojects.com/2023/11/arduino-uno-library-for-proteus-v30.html> (accessed May 18, 2024).
- [2] S. Z. Nasir, "Servo Motor Control using Arduino," *The Engineering Projects*, May 22, 2017. [Online]. Available: <https://www.theengineeringprojects.com/2017/05/servo-motor-control-using-arduino.html> (accessed May 19, 2024).
- [3] Digital Laboratory, "Modul 8 SSF: SPI & I2C," *Google Docs*. [Online]. Available: <https://docs.google.com/document/d/1CsIbwLVUrsKjZ3YhyF0J-gsGWNUIRCQu7JTYMCx3cFY/edit> (accessed May 21, 2024).
- [4] Digital Laboratory, "Modul 3 SSF: Analog to Digital Converter," *Google Docs*. [Online]. Available: <https://docs.google.com/document/d/1arLt3fqXRw-WgkbqIP1RwYs-XJy9-QAfEcM21M3u44/edit> (accessed May 22, 2024).
- [5] Digital Laboratory, "Modul 4 SSF: Serial Port," *Google Docs*. [Online]. Available: https://docs.google.com/document/d/1rRWvBgL3Nsb_h10131A-1kiGQkrGGNLeYoeVIGR9zg/edit (accessed May 22, 2024).
- [6] A. Rofiq H., A. Shulfah Amir, A. Muchtar, and A. A. Rahmansyah, Eds., "Rancang Bangun Automatic Fish Feeder Berbasis Arduino," *core.ac.uk*, Jan. 01, 2020. [Online]. Available: <https://core.ac.uk/download/pdf/480739746.pdf> (accessed May 25, 2024).
- [7] Digital Laboratory. "Modul 7 SSF: Interrupt," *Google Docs*. [Online]. Available: https://docs.google.com/document/d/1VW7j3k_scKyOlzo72scSMFU58EgT-TLoiMOshQ48TUA/edit?usp=sharing (accessed May 26, 2024).

APPENDICES

Appendix A: Project Schematic



Appendix B: Documentation

