

Lampiran

1. Source code

Kode penghalau pakan hama

```
// #####Program Penghalau Hama#####
bool value1 = digitalRead(Receiver_PIN1);
bool value2 = digitalRead(Receiver_PIN2);
bool value3 = digitalRead(Receiver_PIN3);

Serial.print("Receiver 1: "); Serial.println(value1);
Serial.print("Receiver 2: "); Serial.println(value2);
Serial.print("Receiver 3: "); Serial.println(value3);

// Kontrol motor DC berdasarkan sensor
if (value1 == 1 || value2 == 1 || value3 == 1) {
    digitalWrite(PIN_RELAY_MotorDC, LOW);
    Serial.println("Motor DC OFF");
} else {
    digitalWrite(PIN_RELAY_MotorDC, HIGH);
    Serial.println("Motor DC ON");
}
```

Kode pakan ikan

```
// #####Program Pakan Ikan#####
// Sinkronisasi waktu dengan waktu sekarang
currentTime = timezone.dateTime("H:i");
Serial.print("Waktu saat ini => ");
Serial.println(currentTime);

int currentHour = timezone.hour();
int currentMinute = timezone.minute();

// Mengecek apakah salah satu tombol ditekan dan memberi pakan sesuai dengan delay yang ditentukan
if (button1Pressed) {
    feedFish(delayTime1);
    button1Pressed = false;
} else if (button2Pressed) {
    feedFish(delayTime2);
    button2Pressed = false;
} else if (button3Pressed) {
    feedFish(delayTime3);
    button3Pressed = false;
}

delay(1000);
```

Kode buzzer dan Ultrasonik

```

#####Program ultrasonik & Buzzer#####
long duration, distance;
digitalWrite(TRIG_PIN, LOW);
delayMicroseconds(2);

digitalWrite(TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);

duration = pulseIn(ECHO_PIN, HIGH);
distance = (duration * 0.034) / 2;

Serial.print("Jarak: ");
Serial.print(distance);
Serial.println(" cm");

Blynk.virtualWrite(V6, distance);

// Menghitung persentase pakan
int feedPercentage = map(distance, MIN_DISTANCE, MAX_DISTANCE, 100, 0);
feedPercentage = constrain(feedPercentage, 0, 100);

Serial.print("Persentase Pakan: ");
Serial.print(feedPercentage);
Serial.println("%");

Blynk.virtualWrite(V7, feedPercentage);

Blynk.virtualWrite(V7, feedPercentage);

// Mengaktifkan buzzer jika pakan habis
if (feedPercentage == 0) {
  Serial.println("PAKAN KOSONG");
  digitalWrite(BUZZER_PIN, HIGH);
} else {
  Serial.println("PAKAN TERSEDIA");
  digitalWrite(BUZZER_PIN, LOW);
}

delay(400);

```

Kode penjadwalan dan kontrol blynk

```
// #####Kontrol Daya Dan
// Laser#####
// Fungsi untuk kontrol relay dari aplikasi Blynk
BLYNK_WRITE(V1) {
    int pinValue = param.asInt();
    digitalWrite(PIN_RELAY_OnOff, pinValue);
}

// Fungsi untuk kontrol relay laser dari aplikasi Blynk
BLYNK_WRITE(V2) {
    int pinValue = param.asInt();
    if (pinValue == 1) {
        digitalWrite(PIN_RELAY_Laser, HIGH);
    } else {
        digitalWrite(PIN_RELAY_Laser, LOW);
    }
}
//
#####
#####

//#####Program Inputan Jadwal Pakan Dan
//Button Pilihan###

// Fungsi untuk kontrol tombol 5kg
BLYNK_WRITE(V3) {
    int pinValue = param.asInt();
    button1Pressed = (pinValue == 1);
    delayTime1 = 175170; // Waktu delay untuk 5kg
    if (button1Pressed) {
        digitalWrite(PIN_RELAY_BLOWER, LOW); // Nyalakan blower
        saat tombol ditekan
    }
}

// Fungsi untuk kontrol tombol 10kg
BLYNK_WRITE(V4) {
    int pinValue = param.asInt();
    button2Pressed = (pinValue == 1);
    delayTime2 = 350340; // Waktu delay untuk 10kg
    if (button2Pressed) {
        digitalWrite(PIN_RELAY_BLOWER, LOW); // Nyalakan blower
        saat tombol ditekan
    }
}

// Fungsi untuk kontrol tombol 20kg
```

```

BLYNK_WRITE(V5) {
  int pinValue = param.asInt();
  button3Pressed = (pinValue == 1);
  delayTime3 = 700680; // Waktu delay untuk 20kg
  if (button3Pressed) {
    digitalWrite(PIN_RELAY_BLOWER, LOW); // Nyalakan blower
    saat tombol ditekan
  }
}

// Fungsi untuk menggerakkan servo untuk memberi pakan
void feedFish(int delayTime) {
  digitalWrite(PIN_RELAY_BLOWER, LOW);

  unsigned long blowerStartTime = millis();
  while (millis() - blowerStartTime < 10000) {
    for (int pos = 0; pos <= 180; pos += 1) {
      myServo2.write(pos);
      delay(10);
    }
    for (int pos = 180; pos >= 0; pos -= 1) {
      myServo2.write(pos);
      delay(10);
    }
  }

  myServo.write(90);
  unsigned long valveOpenTime = millis();
  while (millis() - valveOpenTime < delayTime) {
    for (int pos = 0; pos <= 180; pos += 1) {
      myServo2.write(pos);
      delay(10);
    }
    for (int pos = 180; pos >= 0; pos -= 1) {
      myServo2.write(pos);
      delay(10);
    }
  }

  myServo.write(0);
  unsigned long valveCloseTime = millis();
  while (millis() - valveCloseTime < 10000) {
    for (int pos = 0; pos <= 180; pos += 1) {
      myServo2.write(pos);
      delay(10);
    }
    for (int pos = 180; pos >= 0; pos -= 1) {
      myServo2.write(pos);
      delay(10);
    }
  }
}

```

```
    }  
  }  
  
  digitalWrite(PIN_RELAY_BLOWER, HIGH);  
  myServo2.write(0);  
}
```