LAMPIRAN

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Servo.h>
#include <Ultrasonic.h>
#include <Adafruit_MLX90614.h>

#define SERVO_PIN                   9
#define LED_GREEN                   7
#define LED_RED                     6
#define PUSH_BUTTON_PIN             8
#define USTRIG_PIN                  12
#define USECHO_PIN                  13

#define START_RUN                   'a'
#define RESTART_ESP32               'r'
#define RESTART_SCAN                'n'

#define CLOSE_DOOR                  180
#define OPEN_DOOR                   90
#define DOOR_LOCK                   0
#define OPEN_DOOR_TIME              4000
#define CLOSE_DOOR_TIME             1000
#define DELAY_TEMP_SUCCESS          2000
#define RESTART_TIME                5000

#define ON                          0
#define OFF                         1

#define MASK_DETECT_TH              80 //in percent %
#define START_SCAN_COUNT            15
#define DISTANCE_ERROR_COUNT        10
#define ERROR_COUNT                 3
#define TEMP_MAX_ALLOWABLE          39
```

```
#define TEMP_THRESHOLD                35

#define MINIMUM_DISTANCE              35
#define MAXIMUM_DISTANCE              60

#define NO_OBJECT                     -1
#define TEMP_LOW                       0
#define TEMP_HIGH                      1
#define TEMP_ALLOWED                   2

#define SCAN_NO_OBJECT                 0
#define SCAN_PASS                      1
#define SCAN_ERROR                     2

#define DISP_IP_INDEX                  0
#define DISP_DISTANCE_ERROR_INDEX      1

LiquidCrystal_I2C lcd(0x27,16,2);  // set the LCD address to 0x27
for a 16 chars and 2 line display
Servo myServo;
Ultrasonic ultrasonic(USTRIG_PIN, USECHO_PIN);
Adafruit_MLX90614 mlx = Adafruit_MLX90614();

char gszIP_Add[20];
unsigned short gusMask_Detect = 0;
unsigned short gusLCD_Index = 0;
unsigned short gusDisp_Index = 0;
unsigned short gusIsNeedDisp = 1;
unsigned short gusIsNeed_Restart = 0;
unsigned short gusIsSend_Request = 0;

unsigned long gulStart_Timer_LCD = 0;
unsigned long gulRestart_Timer = 0;
unsigned long gulDistance_Timer = 0;
```

```
void setup()
{
  char szData[30];
  unsigned short usExit = 0;
  unsigned short usData_Len = 0;
  short a = 0;

  Serial.begin(9600);
  mlx.begin();

  lcd.init();                       // initialize the lcd
  lcd.backlight();

  pinMode(PUSH_BUTTON_PIN,INPUT_PULLUP);
  pinMode(LED_GREEN,OUTPUT);
  pinMode(LED_RED,OUTPUT);

  vLED_Control(SCAN_NO_OBJECT);
  vServo_Control(DOOR_LOCK);

  lcd.clear();
  lcd.print("Wifi");
  lcd.setCursor(0,1);
  lcd.print("connecting...");

  memset(szData, '\0', sizeof(szData));
  memset(gszIP_Add, '\0', sizeof(gszIP_Add));

  do
  {
    usData_Len = usRead_Serial_Data(szData, sizeof(szData));

    if(usData_Len > 0)
    {
      for(short i=0; i<usData_Len; i++)
```

```
      {
        if(szData[i] == '#')
        {
          i++;
          while(szData[i] != ',')
          {
            gszIP_Add[a] = szData[i++];
            a++;
          }
          usExit = 1;
          break;
        }
      }
    }
    else
    {
      if((millis() - gulRestart_Timer) > RESTART_TIME)
      {
        Serial.println(RESTART_SCAN);
        gulRestart_Timer = millis();
      }
    }
  }while(usExit == 0);

  vLCD_Disp_Ip(gszIP_Add);
  gusDisp_Index = DISP_IP_INDEX;
  gulStart_Timer_LCD = millis();
}

void loop()
{
  char szData[30];
  unsigned short usExit = 0;
  unsigned short usTempExit = 0;
  unsigned short usData_Len = 0;
```

```
unsigned short usTemp_Allowed = NO_OBJECT;
unsigned short usIsNeed_Rescan = 0;
unsigned short usFace_Distance = 0;
unsigned short usStart_Scanning_Count = 0;
unsigned short usDistance_Error = 0;
unsigned short usError_Count = 0;
short sMask_Percent = 0;
float fTemperature_Object = 0;

while(usTempExit == 0)
{
  fTemperature_Object = mlx.readObjectTempC();

  if(fTemperature_Object > TEMP_MAX_ALLOWABLE)
  {
    // temp high
    vLED_Control(SCAN_ERROR);
    usTemp_Allowed = TEMP_HIGH;
    gusIsNeedDisp = 1;
  }
  else if(fTemperature_Object > TEMP_THRESHOLD)
  {
    vLED_Control(SCAN_PASS);
    usTemp_Allowed = TEMP_ALLOWED;

    gusIsNeedDisp = 1;
    usTempExit = 1;
  }
  else
  {
    vLED_Control(SCAN_NO_OBJECT);
    usTemp_Allowed = NO_OBJECT;
  }

  if(gusIsNeedDisp == 1)
```

```
    {
      if(usTemp_Allowed != NO_OBJECT)
      {
        vDisp_Temp_Sensor(usTemp_Allowed, fTemperature_Object);
      }
      else if(usTemp_Allowed == NO_OBJECT)
      {
        vLCD_Disp_Ip(gszIP_Add);
        gusDisp_Index = DISP_IP_INDEX;
      }
      gusIsNeedDisp = 0;
    }
    vLCD_Disp_Timer_Index();
    sRead_But();
  }

  delay(DELAY_TEMP_SUCCESS);
  vLED_Control(SCAN_NO_OBJECT);

  while(usTempExit == 1)
  {
    usExit = 0;

    usFace_Distance = usRead_Distance();

    if((usFace_Distance > MINIMUM_DISTANCE) && (usFace_Distance <
MAXIMUM_DISTANCE))
    {
      usStart_Scanning_Count++;

      vDisp_Scanning();
    }
    else
    {
      usStart_Scanning_Count = 0;
```

```
    }

    if(usStart_Scanning_Count > START_SCAN_COUNT)
    {
      //optimal distance to scan face
      memset(szData, '\0', sizeof(szData));

      do
      {
        usFace_Distance = usRead_Distance();

        if((usFace_Distance      >      MINIMUM_DISTANCE)      &&
(usFace_Distance < MAXIMUM_DISTANCE))
        {
          if(gusIsSend_Request == 0)
          {
            Serial.println(START_RUN);
            gusIsSend_Request = 1;
          }

          usData_Len = usRead_Serial_Data(szData, sizeof(szData));
//Read data from ESP32-CAM
          if(usData_Len > 0)
          {
            if(szData[0] == '*')
            {
              sscanf(szData, "*%d,", &sMask_Percent);// ESP32-CAM
will return mask percent to Arduino

              usIsNeed_Rescan = 0;
              gusIsSend_Request = 0;
              usExit = 1;
            }
          }
        }
        else
```

```
      {
        usDistance_Error++;
      }

      if(usDistance_Error > DISTANCE_ERROR_COUNT)
      {
        usDistance_Error = 0;

        usIsNeed_Rescan = 1;
        usExit = 1;
      }
    }while(usExit == 0);


    if(usIsNeed_Rescan == 0)
    {
      vLCD_Disp_Status(sMask_Percent);

      if(sMask_Percent >= MASK_DETECT_TH) //if the percentage is
higher than MASK_DETECT_TH (80%), door will open
      {
        usTempExit = 0;
        vDoor_Control(ON);
      }
      else
      {
        vDoor_Control(OFF);
        usError_Count++;

        if(usError_Count >= ERROR_COUNT)
        {
          usTempExit = 0;
        }
      }
    }
```

```
      else if(usIsNeed_Rescan == 1)
      {
        usError_Count++;

        if(usError_Count >= ERROR_COUNT)
        {
          usTempExit = 0;
        }
        else
        {
          short a = 0;
          vLCD_Disp_Error_Scan();
          vLED_Control(SCAN_ERROR);
          memset(szData, '\0', sizeof(szData));
          Serial.println(RESTART_ESP32);
          gulRestart_Timer = millis();

          do
          {
            usData_Len        =        usRead_Serial_Data(szData,
sizeof(szData)); //Read data from ESP32-CAM, and dump previous
data.
            if(usData_Len > 0)
            {
              gusIsSend_Request = 0;
              for(short i=0; i<usData_Len; i++)
              {
                if(szData[i] == '#')
                {
                  i++;
                  while(szData[i] != ',')
                  {
                    gszIP_Add[a] = szData[i++];
                    a++;
                  }
```

```
                usExit = 0;
                break;
              }
            }
            else
            {
              if((millis() - gulRestart_Timer) > RESTART_TIME)
              {
                Serial.println(RESTART_SCAN);
                gulRestart_Timer = millis();
              }
            }
          }while(usExit == 1);
        }
      }
    }
    else
    {
      vLED_Control(SCAN_NO_OBJECT);

      if(gusIsNeedDisp == 1)
      {
        vLCD_Disp_Scan_Face(usFace_Distance);
        gusDisp_Index = DISP_DISTANCE_ERROR_INDEX;
        gusIsNeedDisp = 0;
      }
    }
    vLCD_Disp_Timer_Index();
    sRead_But();
  }
}
void vDisp_Scanning()
{
  lcd.clear();
```

```
  lcd.print("SCANNING...");
  lcd.setCursor(0,1);
  lcd.print("Pls wait & hold.");
}
//PERCENTAGE OF FACE DETECTED, MORE FACE DETECTED, LOW PERCENTAGE,
NO MASK
void vLCD_Disp_Status(short sMask_Percent)
{
  lcd.clear();
  lcd.print("Mask: ");
  lcd.print(sMask_Percent);
  lcd.print("%");
  lcd.setCursor(0,1);

  if(sMask_Percent >= MASK_DETECT_TH)
  {
    lcd.print("Enter Allowed.");
  }
  else
  {
    lcd.print("PLEASE WEAR MASK");
  }
}
void      vDisp_Temp_Sensor(short      usTemp_Allowed,      float
fTemperature_Object)
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Body Temp: ");
  lcd.print(fTemperature_Object);
  lcd.print(char(223));
  lcd.print("C");
  lcd.setCursor(0,1);

  if(usTemp_Allowed == TEMP_ALLOWED)
```

```
  {
    lcd.print("Body Temp OK");
  }
  else if(usTemp_Allowed == TEMP_HIGH)
  {
    lcd.print("Body Temp HIGH");
  }
  else if(usTemp_Allowed == TEMP_LOW)
  {
    lcd.print("Body Temp LOW");
  }
}
void vLCD_Disp_Ip(char *szIp)
{
  lcd.clear();

  if(gusLCD_Index == 0)
  {
    lcd.setCursor(2,0);
    lcd.print("PLEASE SCAN");
    lcd.setCursor(2,1);
    lcd.print("TEMPERATURE");
  }
  else
  {
    lcd.setCursor(1,0);
    lcd.print("CONNECTED TO:");
    lcd.setCursor(1,1);
    lcd.print(szIp);
  }
}
void vLCD_Disp_Scan_Face(unsigned short usFace_Distance)
{
  lcd.clear();
  if(gusLCD_Index == 0)
```

```
  {
    lcd.setCursor(0,0);
    lcd.print("Pls Scan ur face");
    lcd.setCursor(0,1);
    lcd.print(MINIMUM_DISTANCE);
    lcd.print("-");
    lcd.print(MAXIMUM_DISTANCE);
    lcd.print("cm from cam");
  }
  else if(gusLCD_Index == 1)
  {
    lcd.setCursor(0,0);
    lcd.print("youre ");
    lcd.print(usFace_Distance);
    lcd.print("cm away");

    lcd.setCursor(0,1);
    if(usFace_Distance > MAXIMUM_DISTANCE)
    {
      lcd.print("Move Closer.. ");
    }
    else if(usFace_Distance < MINIMUM_DISTANCE)
    {
      lcd.print("Take a step back..");
    }
  }
}
void vDisp_Face_Distance_Error(unsigned short usFace_Distance)
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("youre ");
  lcd.print(usFace_Distance);
  lcd.print("cm away");
```

```
  lcd.setCursor(0,1);
  if(usFace_Distance > MAXIMUM_DISTANCE)
  {
    lcd.print("Move Closer.. ");
  }
  else if(usFace_Distance < MINIMUM_DISTANCE)
  {
    lcd.print("Take a step back..");
  }
}
void vLCD_Disp_Error_Scan()
{
  lcd.clear();
  lcd.print("Take a step back");
  lcd.setCursor(0,1);
  lcd.print("and rescan again");
}
void vLED_Control(short sScan_Status)
{
  if(sScan_Status == SCAN_PASS)
  {
    digitalWrite(LED_GREEN,HIGH);
    digitalWrite(LED_RED,LOW);
  }
  else if(sScan_Status == SCAN_ERROR)
  {
    digitalWrite(LED_GREEN,LOW);
    digitalWrite(LED_RED,HIGH);
  }
  else
  {
    digitalWrite(LED_GREEN,LOW);
    digitalWrite(LED_RED,LOW);
  }
}
```

```
void vLCD_Disp_Timer_Index()
{
  if((millis() - gulStart_Timer_LCD) >= 1000)
  {
    gusLCD_Index++;

    if(gusDisp_Index == DISP_IP_INDEX)
    {
      if(gusLCD_Index > 1)
      {
        gusLCD_Index = 0;
      }
    }
    else if(gusDisp_Index == DISP_DISTANCE_ERROR_INDEX)
    {
      if(gusLCD_Index > 1)
      {
        gusLCD_Index = 0;
      }
    }

    gusIsNeedDisp = 1;
    gulStart_Timer_LCD = millis();
  }
}
short sRead_But()
{
  short sBut_Status = 0;

  sBut_Status = digitalRead(PUSH_BUTTON_PIN);

  if(sBut_Status == HIGH)
  {
    vDoor_Control(ON);
```

```
    gusLCD_Index = 0;
    gulStart_Timer_LCD = millis();
    gusIsNeedDisp = 1;
  }
}
void vDoor_Control(short sOnOff)
{
  if(sOnOff == ON)
  {
    vLED_Control(SCAN_PASS);
    vServo_Control(OPEN_DOOR);
    delay(OPEN_DOOR_TIME);

    vServo_Control(CLOSE_DOOR);
    delay(CLOSE_DOOR_TIME);
  }
  else
  {
    vServo_Control(DOOR_LOCK);
    vLED_Control(SCAN_ERROR);
    delay(OPEN_DOOR_TIME);
  }
}
unsigned short usRead_Distance()
{
  unsigned short usFace_Distance = 0;

  usFace_Distance = ultrasonic.read();
  delay(50);

  return usFace_Distance;
}
//CONTROL SERVO BASED ON RESULTS
void vServo_Control(int sDoor_Status)
{
```

```
  myServo.attach(SERVO_PIN);

  if(sDoor_Status == OPEN_DOOR) //open door
  {
    for(int i = CLOSE_DOOR; i > OPEN_DOOR; i--)
    {
      myServo.write(i);
      delay(20);
    }
  }
  else if(sDoor_Status == CLOSE_DOOR)
  {
    for(int i = OPEN_DOOR; i < CLOSE_DOOR; i++)
    {
      myServo.write(i);
      delay(20);
    }
  }
  else if(sDoor_Status == DOOR_LOCK)
  {
    myServo.write(CLOSE_DOOR);
  }

  myServo.detach();
}
//READ SERIAL DATA FROM ESP32-CAM
unsigned short usRead_Serial_Data(char *szData, short sDataSize)
{
  short i=0;

  if(Serial.available())
  {
    *(szData+i) = Serial.read();
    i++;
    delay(2);
```

```
    while(Serial.available())
    {
      *(szData+i) = Serial.read();
      i++;

      if(i >= sDataSize)
      {
        break;
      }
      delay(2);
    }
  }

  Serial.flush();

  return i;
}
```

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA JURUSAN
TEKNIK KOMPUTER
Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414
Website : www.polsri.ac.id E-mail : info@polsri.ac.id

REKOMENDASI UJIAN TUGAS AKHIR

Pembimbing Laporan Tugas Akhir, memberikan rekomendasi ujian laporan tugas akhir kepada,

Nama Mahasiswa	:	Muhammad Arayen
NIM	:	062030701735
Jurusan/Program Studi	:	Teknik Komputer/D3 Teknik Komputer
Judul Tugas Akhir	:	Sistem monitoring pengunjung toko dengan CCTV dilengkapi pendeteksi kebakaran berbasis IOT

Mahasiswa tersebut telah memenuhi persyaratan dan dapat mengikuti Ujian Laporan Tugas Akhir, pada Tahun Akademik 2023/2024

Disetujui oleh,
Pembimbing I

Indarto, S. T., M.Cs
NIP.197307062005011003

Palembang, 14 Agustus 2023
Pembimbing II

Ica Admirani, S.Kom, M.Kom
NIP.197903282005012001

Dipindai dengan CamScanner

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN
RISET DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 Fax. 0711-355918
Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id

LEMBAR BIMBINGAN LAPORAN TUGAS AKHIR

Nama Mahasiswa	:	Muhammad Arayen
NIM	:	062030701735
Jurusan/Program Studi	:	Teknik Komputer/DIII
Dosen Pembimbing	:	Indarto,ST.,M.Cs
Judul	:	Sistem monitoring pengunjung toko dengan CCTV dilnegkapi pendeteksi kebakaran berbasis IOT

NO	TANGGAL	URAIAN	PARAF PEMBIMBING

Palembang, 19 agustus 2023
Mengetahui,
Ketua Jurusan

Azwardi,S.T.,M.T.
NIP.197005232005011004

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN
RISET DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 Fax. 0711-355918
Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id

LEMBAR BIMBINGAN LAPORAN TUGAS AKHIR

Nama Mahasiswa	:	Muhammad Arayen
NIM	:	062030701735
Jurusan/Program Studi	:	Teknik Komputer/DIII
Dosen Pembimbing	:	Ica Admirani,S.Kom.,M.Kom
Judul	:	Sistem monitoring pengunjung toko dengan CCTV dilengkapi pendeteksi kebakaran berbasis IOT

NO	TANGGAL	URAIAN	PARAF PEMBIMBING
	9/8-2023	- Perbaiki Alat - Tambahkan skema rangkaian pd bab 3	
	14/8-2023	- Acc Alat - Acc Laporan	

Palembang, 19/8/2023

Mengetahui,
Ketua Jurusan

Azwardi,S.T.,M.T.
NIP.197005232005011004

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA
JURUSAN TEKNIK KOMPUTER
Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414
Website : www.polsri.ac.id E-mail : info@polsri.ac.id

REVISI UJIAN TUGAS AKHIR

Dosen Penguji	: Ahyar Supani, S.T., M.T
Nama Mahasiswa	: Muhammad Arayen
NIM	: 062030701735
Jurusan /Program Studi	: D3 Teknik Komputer
Judul LA/ Skripsi	: Sistem Monitoring Pengunjung Toko Dengan CCTV Dilengkapi Pendekteksi Kebakaran Berbasis IOT

No	Uraian	Paraf
	Total tulisan Laporan akhir direvisi, disesuaikan format lapo. akhir (bab 5/1. bab. 5).	

Palembang,
Dosen Penguji,

Ahyar Supani, S.T., M.T
NIP. 196802111992031002

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA
JURUSAN TEKNIK KOMPUTER
Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414
Website : www.polsri.ac.id E-mail : info@polsri.ac.id

REVISI UJIAN TUGAS AKHIR

Dosen Penguji : Herlambang Saputra, M.Kom.,Ph.D.
Nama Mahasiswa : Muhammad Arayen
NIM : 062030701735
Jurusan /Program Studi : D3 Teknik Komputer
Judul LA/ Skripsi : Sistem Monitoring Pengunjung Toko Dengan CCTV Dilengkapi Pendekteksi Kebakaran Berbasis IOT

No	Uraian	Paraf
1.	Perbaiki penulisan LA.	

Palembang, 6-8-23
Dosen Penguji,

Herlambang Saputra, M.Kom.,Ph.D.
NIP. 198103182008121002

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA
JURUSAN TEKNIK KOMPUTER
Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414
Website : www.polsri.ac.id E-mail : info@polsri.ac.id

REVISI UJIAN TUGAS AKHIR

Dosen Penguji : Mustaziri, ST.,M.Kom.
Nama Mahasiswa : Muhammad Arayen
NIM : 062030701735
Jurusan /Program Studi : D3 Teknik Komputer
Judul LA/ Skripsi : Sistem Monitoring Pengunjung Toko Dengan CCTV Dilengkapi Pendekteksi Kebakaran Berbasis IOT

No	Uraian	Paraf
1.	Tata tulis harus di perbaiki dan harus konsisten istilah asing di cetak miring	
2	Abstrak di perbaiki	
3	Latar belakang di perbaiki -	
4	Judul di perbaiki	
5	Rumusan, Batasan, Tujuan dan manfaat di perbaiki	
6	Bab II tambahkan teori asing. api panas dan kebakaran	
7	Blok Diagram di perbaiki	
8	Flow chard di perbaiki	
9	Pembahasan di perbaiki	
10	kesimpulan di perbaiki	

Palembang,
Dosen Penguji,

Mustaziri, ST.,M.Kom.
NIP. 196909282005011002

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA
JURUSAN TEKNIK KOMPUTER
Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414
Website : www.polsri.ac.id E-mail : info@polsri.ac.id

REVISI UJIAN TUGAS AKHIR

Dosen Penguji	: M. Miftakul Amin, S.Kom., M.Eng.
Nama Mahasiswa	: Muhammad Arayen
NIM	: 062030701735
Jurusan /Program Studi	: D3 Teknik Komputer
Judul LA/ Skripsi	: Sistem Monitoring Pengunjung Toko Dengan CCTV Dilengkapi Pendekteksi Kebakaran Berbasis IOT

No	Uraian	Paraf
	Sesuaikan dengan format Penulisan Laporan Akhir	

Palembang,
Dosen Penguji,

M. Miftakul Amin, S.Kom., M.Eng.
NIP. 197912172012121001

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA
JURUSAN TEKNIK KOMPUTER
Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414
Website : www.polsri.ac.id E-mail : info@polsri.ac.id

REVISI UJIAN TUGAS AKHIR

Dosen Penguji : Ervi Cofriyanti, S.Si., M.T.I
Nama Mahasiswa : Muhammad Arayen
NIM : 062030701735
Jurusan /Program Studi : D3 Teknik Komputer
Judul LA/ Skripsi : Sistem Monitoring Pengunjung Toko Dengan CCTV Dilengkapi Pendekteksi Kebakaran Berbasis IOT

No	Uraian	Paraf
1	Perbaiki judul.	✓ EF 25/9 2023
2	Tambahkan rancangan h/w (stemally), rancangan s/w, rancangan interface.	
3	Hal 8 → Hilangkan kata "saya" menjadi penulis.	✓ EF 25/9 2023
4	Judul tabel di atas, judul gambar di bawah.	✓ EF 25/9 2023

Palembang, 16/8/2023
Dosen Penguji,

Ervi Cofriyanti, S.Si., M.T.I
NIP. 198012222015042001

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA
JURUSAN TEKNIK KOMPUTER
Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414
Website : www.polsri.ac.id E-mail : info@polsri.ac.id

PELAKSANAAN REVISI UJIAN TUGAS AKHIR

Nama Mahasiswa : Muhammad Arayen
NIM : 062030701735
Jurusan /Program Studi : D3 Teknik Komputer
Judul LA/ Skripsi : Sistem Monitoring Pengunjung Toko Dengan CCTV Dilengkapi Pendekteksi Kebakaran Berbasis IOT

Telah melaksanakan revisi terhadap Laporan Tugas Akhir yang diujikan pada hari ..... tanggal ..... bulan .....tahun ..... Pelaksanaan revisi terhadap Laporan Tugas Akhir tersebut telah disetujui oleh Dosen Penguji yang memberikan revisi:

No	Komentar	Nama Dosen Penguji	Tanggal/ bulan	Tanda Tangan
1.	ACC	Ahyar Supani, S.T., M.T	18/9 2023	
2.	Acc	Herlambang Saputra, M.Kom., Ph.D.	21/10 -23	
3.	oke Acc	Mustaziri, ST.,M.Kom.	3/11 23	
4.	Acc	M.Miftakhul Amin,S.Kom,M.Eng.	19/9 2023	
5.	Acc	Ervi Cofriyanti, ~~S.Kom~~, M.Ti S.Si	25/9 2023	

Palembang, Agustus 2023
Ketua Penguji,

Ahyar Supani, S.T., M.T
NIP. 196802111992031002