

Lampiran 3 Pseudocode

```

Deklarasi :
char ssid
char password
char host
int httpsPort
char token
char chat_id
char boundry

WiFiClientSecure client; //library untuk koneksi wifi

UniversalTelegramBot bot(token, client); //library untuk koneksi
aplikasi telegram bot

int Bot_mtbs
long Bot_lasttime

int chipSelect

SoftwareSerial cameraconnection = SoftwareSerial//untuk
menyesuaikan port pada kamera yang di
pasang di nodemcu

Adafruit_VC0706 cam = Adafruit_VC0706(&cameraconnection) //untuk
koneksi ke camera

Deskripsi :

void setup()
{
  pinMode(PIR_MOTION_SENSOR, INPUT)// koneksi ke sensor pir untuk
  pin terdapat pada library untuk
  lebi lengkap saya melampir kan
  code di lampiran

  pinMode(16, OUTPUT)//pada metode ini pinmode untuk koneksi ke
  microsd

  Serial.read(9600); // untuk bit rate pada serial monitor di
  arduino ide maksud dari serial adalah menampilkan

```

```

        hasil atau membaca dari serial monitor di arduino
        ide
Serial.write("VC0706 Camera snapshot test");
    if (!SD.read(chipSelect)) {
        serial.write("Card failed, or not present");
        return;
    }
    else {
        serial.write("MASUK");
    }

    initWifiConnection();// pemanggilan untuk koneksi wifi
}
void loop()
{

//Pir Setting
int sensorValue = digitalRead(PIR_MOTION_SENSOR) // membaca value
                                                    dari sensor

if(sensorValue == HIGH) {
Serial.write("Gerakan Terdeteksi");
bot.sendChatAction(chat_id, "Sedang Mengetik");
capture();
Serial.write("Mengidentifikasi Adanya Gerakan");
bot.sendMessage(chat_id, "Gerakan Terdeteksi", "");
}
else{
Serial.write("Tidak ada Gerakan Terdeteksi");
bot.sendChatAction(chat_id, "Sedang Mengetik");
Serial.write("Mengidentifikasi Sensor");
}
}

```

Proses pengecekan microsd apakah berhasil

Proses seleksi dimana jika terdapat value dari sensor pir maka dia akan menampilkan tulisan terdapat gerakan di serial monitor lalu akan memanggil fungsi capture yang akan dijelaskan nanti dibagiannya dan jika tidak menerima value maka menampilkan tulisan tidak ada gerakan pada serial monitor

```

if (millis() > Bot_lasttime + Bot_mtbs) {
    int numNewMessages = bot.getUpdates(bot.last_message_received
+ 1);
    while(numNewMessages) {
        Serial.write("got response");
        handleNewMessages (numNewMessages);
        numNewMessages = bot.getUpdates(bot.last_message_received +
1);
    }
    Bot_lasttime = millis();
}
}

```

Proses seleksi untuk menampilkan terdapat respon atau perintah dan melanjutkan perintah difungsi handleNewMessages()

```

void capture(){
    if (cam.read()) {
        Serial.write("Camera Found:");
    } else {
        Serial.write("No camera found?");
    }
    return;
}

```

Seleksi untuk mengetahui apakah kamera sudah terhubung atau belum

```

char *reply = cam.getVersion();
if (reply == 0) {
    Serial.write("Failed to get version");
} else {
    Serial.write("-----");
}

```

Proses seleksi menentukan type dan versi kamera dimana untuk camera TTL hanya dapat di vc0706

```

Serial.write(reply);

Serial.write("-----");
}

//cam.setImageSize(VC0706_640x480); // biggest
//cam.setImageSize(VC0706_320x240); // medium
cam.setImageSize(VC0706_160x120); // small

uint8_t imgsize = cam.getImageSize();

Serial.write("Image size: ");

//if (imgsize == VC0706_640x480) Serial.write("640x480");
//if (imgsize == VC0706_320x240) Serial.write("320x240");
if (imgsize == VC0706_160x120) Serial.write("160x120");

Serial.write("Snap in 3 secs...");

delay(4000);

if (! cam.takePicture())
    Serial.write("Failed to snap!");
else
    Serial.write("Picture taken!");

char filename[13];
strcpy(filename, "IMAGE00.JPG");
SD.remove(filename);
for (int i = 0; i < 100; i++) {
    filename[5] = '0' + i/10;
    filename[6] = '0' + i%10;
    if (! SD.exists(filename)) {
        break;
    }
}

```

Untuk menentukan resolusi hasil gambar yang diinginkan tentunya jika ingin resolusi yang baik butuh waktu encode dan decode dan kecepatan internet yang cepat

Membaca hasil yang telah ditentukan untuk resolusi gambar dan menampilkan di serial monitor setelah itu pengambilan gambar dalam waktu 3 detik

Seleksi jika kamera tiba terlepas pada rangkaian makan gagal mengambil gambar jika normal maka berhasil diambil

Proses encode gambar dan akan disimpan ke microsd

```

    }
}
Serial.write("Saving ");
Serial.write(filename);

File imgFile = SD.open(filename, FILE_WRITE);
uint16_t jpglen = cam.frameLength();
Serial.write("Storing ");
Serial.write(jpglen, DEC);
Serial.write(" byte image.");

```

Proses
nenvimnanan

```

int32_t time = millis();
byte wCount = 0;
while (jpglen > 0) {
    uint8_t *buffer;
    uint8_t bytesToRead = min(32, jpglen);
    buffer = cam.readPicture(bytesToRead);
    imgFile.write(buffer, bytesToRead);

    if(++wCount >= 64) {
        Serial.write('.');
        wCount = 0;
    }
    jpglen -= bytesToRead;
    delay(20);
}
imgFile.close();

```

Proses decode gambar dan
akan dikirim ke telegram

Tampilan
infromasi decode
diserial monitor

```

time = millis() - time;

Serial.write("done!");

sendPhotoToTelegram(filename);
}

```

Tampilan informasi telah selesai dan pemanggilan fungsi untuk dikirim ke telegram

```

void initWifiConnection()
{
    Serial.write();
    Serial.write("Connecting to Wifi [%s]...\r\n", ssid);
    WiFi.mode(WIFI_STA);
    WiFi.write(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.write(".");
    }
    Serial.write("");
    Serial.write("WiFi Connected");
}

```

Tampilan informasi dan proses koneksi wifi yang telah diatur ssid dan passwordnya diatas

```

void receiveDataFromTelegram()
{
    unsigned long timeout = millis();
    while (client.available() == 0) {
        if (millis() - timeout > 5000) {
            Serial.write("Response From Telegram!");
            client.stop();
            return;
        }
    }
}

```

Tampilan informasi dan proses jika ada respon atau perintah dari telegram

```

}

Serial.write();

Serial.write("Receiving from telegram..."); //informasi jika
                                             foto atau gambar
                                             telah      masuk
                                             ditelegram

int responseContentLength = 0;
while (client.available()) {

    String line = client.readStringUntil('\r');
    client.read(); // lê o character '\n'

    Serial.write(line);

    if (line.startsWith("Content-Length:")) {
        int index = line.indexOf(':');
        responseContentLength = line.substring(index + 1).toInt();
    }

    if (line.length() == 0)
        break;
}

while (responseContentLength > 0)
{
    char ch = client.read();
    Serial.write(ch);
    responseContentLength--;
}

```

Proses informasi ke serial monitor ukuran pada foto dan proses encode decode

```

Serial.write();

Serial.write("closing connection");
}

void handleNewMessages(int numNewMessages) {
    Serial.write("handleNewMessages");
    Serial.write(String(numNewMessages));

    for (int i=0; i<numNewMessages; i++) {
        String chat_id = String(bot.messages[i].chat_id);
        String chat_id_1 = String(bot.messages[i].chat_id);
        String text = bot.messages[i].text;

        if (text == "/jepret") { //instruksi untuk memanggil foto
            capture();
        }
    }
}

void sendPhotoToTelegram(String filename)
{
    Serial.writef("Connecting Telegram %s:%d... ", host, httpsPort);

    if (!client.connect(host, httpsPort)) {
        Serial.write("Failde To Conenct!");
        return;
    }

    sendDataToTelegram(filename);
    receiveDataFromTelegram();
}

```

Tampilan infomasi dan proses dimana kita ingin mengambil gambar secara manual

Proses pengiriman data gambar dan menampilkan informasi diserial


```

}

void sendDataToTelegram(String file_name) // dalam void
sendDataToTelegram
semua perintah
seperti encode gambar
dan decode gambar di
telegram lalu
pengiriman dari
microsd ke bot
telegram

{
String start_request = "";

String end_request = "";

// String chat_id = String(bot.messages[0].chat_id);
// String chat_id_1 = String(bot.messages[0].chat_id);

start_request = start_request + "--" + boundry + "\r\n";

start_request = start_request + "content-disposition: form-
data; name=\"chat_id\""+ "\r\n";

start_request = start_request + "\r\n";

start_request = start_request + chat_id + "\r\n";

start_request = start_request + "--" + boundry + "\r\n";

start_request = start_request + "content-disposition: form-
data; name=\"photo\"; filename=\"foto.jpg\" \r\n";

start_request = start_request + "Content-Type: image/jpeg\r\n";

start_request = start_request + "\r\n";

end_request = end_request + "\r\n";

end_request = end_request + "--" + boundry + "--" + "\r\n";

// String file_name = "IMAGE00.jpg";

```

```
Serial.write("Sending ");
Serial.write(file_name);

File file = SD.open(file_name);

int contentLength = (int)file.size() + start_request.length()
+ end_request.length();

String headers = String("POST /bot") + token + "/sendPhoto
HTTP/1.1\r\n";

headers = headers + "Host: " + host + "\r\n";

headers = headers + "User-Agent: ESP8266" +
String(ESP.getChipId()) + "\r\n";

headers = headers + "Accept: */*\r\n";

headers = headers + "Content-Type: multipart/form-data;
boundary=" + boundry + "\r\n";

headers = headers + "Content-Length: " + contentLength + "\r\n";

headers = headers + "\r\n";

headers = headers + "\r\n";

Serial.write();
Serial.write("Mengirim data ke Telegram ...");
Serial.write(headers);
client.write(headers);

client.flush();

Serial.write(start_request);
client.write(start_request);
client.flush();

Serial.write("sendFile");
sendFile(&file);
file.close();
```

```
client.flush();  
Serial.write(end_request);  
client.write(end_request);  
client.flush();  
}  
void sendFile(Stream* stream)  
{  
    size_t bytesReaded;  
    size_t bytesSent;  
    do {  
        uint8_t buff[1024];  
        bytesSent = 0;  
        bytesReaded = stream->readBytes(buff, sizeof(buff));  
        if (bytesReaded) {  
            bytesSent = client.write(buff, bytesReaded);  
            client.flush();  
        }  
    } while ( (bytesSent == bytesReaded) && (bytesSent > 0) );  
}
```

Pengiriman
informasi
gerakan dan
pengiriman
gambar