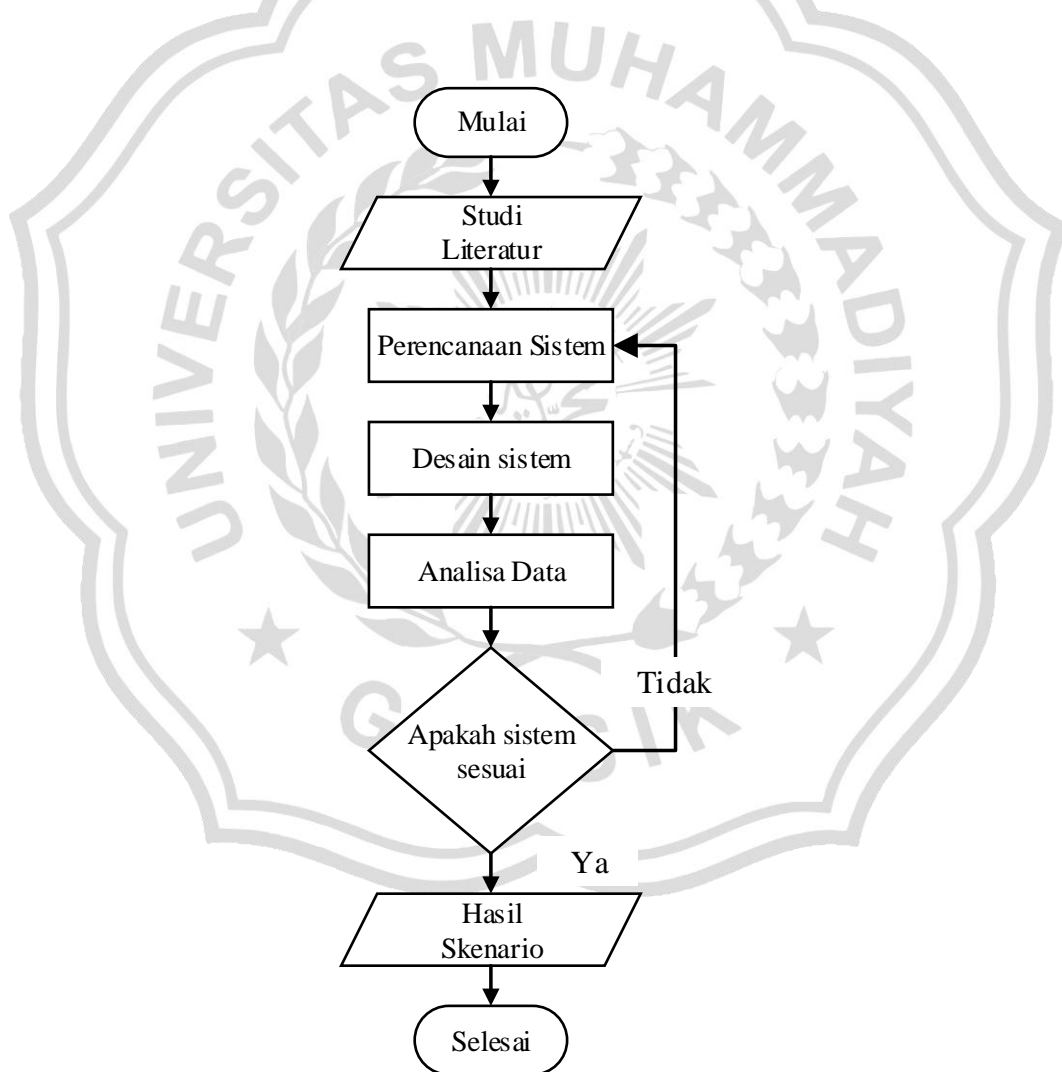


## BAB III

### METODE PENELITIAN

#### 3.1. Secara Umum

Pada penelitian ini metode yang dipakai adalah seperti yang di gambarkan pada *flowchart*, dengan melakukan metode seperti pada **Gambar 3.1** diharapkan penelitian ini dapat memenuhi hasil yang di inginkan.



**Gambar 3.1** Flowchart Penelitian

### 3.2.Studi Literatur

Tahapan ini yang perlu dilakukan adalah mengumpulkan referensi dan dasar teori yang diambil dari jurnal, internet, dan buku-buku yang berhubungan dengan elemen-elemen yang dipakai dalam penelitian ini.

Berikut literatur-literatur yang dipelajari antara lain sebagai berikut:

- a. Node MCU ESP8266
- b. ESP32-CAM
- c. Sensor magnet MC-38
- d. Sensor PIR
- e. Telegram Bot

### 3.3.Perancangan Sistem

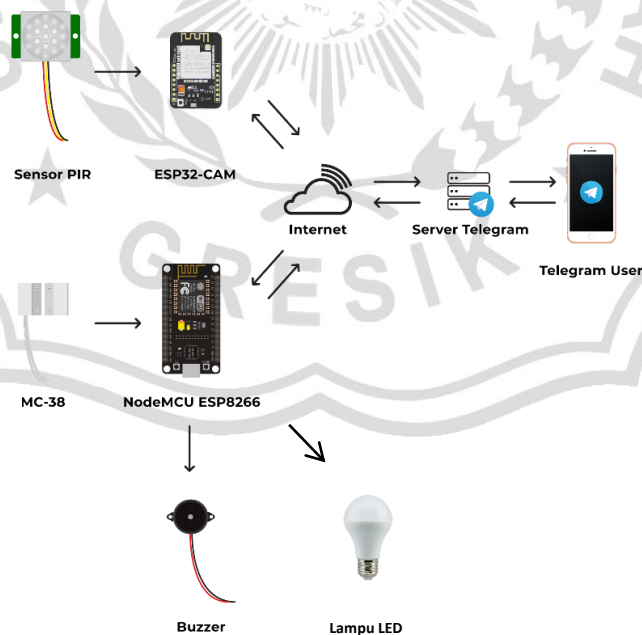
Tahap ini yang dilakukan adalah berupa pembuatan sistem monitoring keamanan kandang kambing dengan menggunakan NodeMCU ESP8266 dan Telegram *messenger* melalui bot yang telah dibuat sebelumnya. Alat ini dirancang dengan dua sistem, sistem kendali manual dan otomatis .

Pada sistem otomatis ini bisa memberitahukan kepada pemilik kandang melalui notifikasi melalui aplikasi telegram ketika sensor PIR (Passive Infrared Receiver) mendeteksi adanya aktifitas mencurigakan di sekitar kandang maka ESP32-CAM akan mengambil gambar dan mengirim hasil gambar ke Telegram pemilik kandang, sebagai pengamanan tambahan pada bagian pintu terpasang sensor magnet MC-38, jika pintu dibuka maka akan menyalakan buzzer yang fungsinya sinyal peringatan dan mengirim notifikasi peringatan pemilik kandang melalui

aplikasi Telegram. Sedangkan pada sistem manual pengguna atau pemilik kandang kambing dapat mengirimkan perintah melalui kode yang diterapkan pada sistem, misalkan perintah memotret, dan perintah pendukung lainnya.

### 3.3.1. Perancangan Hardware

Perancangan hardware ini meliputi pembuatan mekanik dan komponen, sedangkan untuk perangkat kerasnya terdiri dari NodeMCU ESP8266 sebagai mikrokontroler yang berfungsi mengontrol jalannya program, sensor PIR (Passive Infrared Receiver) sebagai pendeteksi gerakan manusia, sensor MC-38 sebagai saklar yang bekerja dengan menggunakan prinsip elektromagnetik, ESP32-CAM untuk menangkap gambar yang nantinya akan dikirim melalui aplikasi telegram penggunanya. Berikut alur kerja sistem seperti pada **Gambar 3. 2**.



**Gambar 3. 2** Alur Kerja Sistem

Prinsip kerja menurut alur kerja pada **Gambar 3. 2**

- a. NodeMCU ESP8266 berfungsi sebagai pengolah data yang mana diprogram dengan software Arduino IDE 1.8.10. dan dapat bekerja pada tegangan 5 V. Di sini NodeMCU sebagai pengambil data dari sensor magnet MC-38 serta buzzer dan mengirimkan data laporan ke *smartphone* melalui aplikasi Telegram menggunakan konsep IoT.
- b. Sensor PIR(Passive Infrared Receiver) berfungsi sebagai pendeteksi adanya benda atau tidak dengan sistem deteksi pancaran sinar infra-red dari objek tersebut.
- c. Sensor MC-38 atau sensor magnet pada alat ini digunakan sebagai sensor tambahan yang dipasang di pintu kandang
- d. Buzer berfungsi sebagai sirine penanda bahaya atau pemberi peringatan kepada pemilik kambing jika ada seseorang yang dicurigai masuk ke kandang, maka dengan adanya suara yang keras pemilik kambing akan terjaga atau bangun dari tidurnya jika malam hari ataupun siang hari
- e. ESP32-CAM sama seperti nodemcu dimana modul ini diprogram di software arduino , bekerja pada tegangan 3-5v, modul ini menggunakan input sensor PIR jika bekerja dalam sistem otomatis maka Sensor PIR akan mentrigger ESP32-CAM untuk mengambil gambar lalu mengirimkan data laporan ke telegram pengguna jika di perintah kondisi manual misalkan diperintah untuk mengambil gambar

maka kamera akan mengambil gambar selanjutnya dikirim ke aplikasi telegram pada smartphone pemilik kambing

- f. Lampu LED untuk penerangan bantuan pada ESP32-CAM pada saat malam hari.
- g. Telegram berfungsi untuk menerima data yang dibutuhkan dan mengirim perintah yang di kirim dari Telegram pengguna ke NodeMCU ESP8266, dan ESP32-CAM melalui Fitur Telegram Bot.
- h. Kabel *Jumper* Secukupnya untuk menghubungkan pin-pin pada semua komponen sensor dan komponen pendukung lainnya.

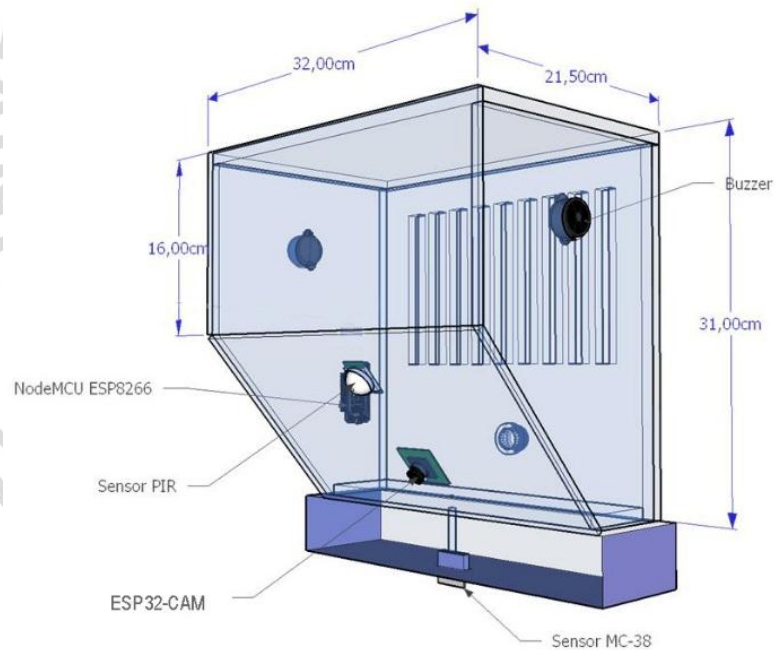
Alat dan bahan yang digunakan dalam perancangan *hardware* sistem Informasi keamanan kandang kambing menggunakan Telegram berbasis NodeMCU ESP32 adalah sebagai berikut :

1. Mikrokontroler NodeMCU ESP8266 sebanyak satu buah.
2. Modul Sensor PIR sebanyak satu buah.
3. Modul ESP32-CAM satu buah.
4. Buzzer sebanyak dua buah
5. Lampu LED satu buah
6. Komponen pendukung elektronika lainnya dan kabel.
7. Bahan bangunan seperti triplek dan kayu reng

Gambar 3. 3 merupakan skema *hardware* pada alat sistem informasi keamanan kandang kambing menggunakan Telegram berbasis NodeMCU ESP8266 yang berkonsep *internet of things*.

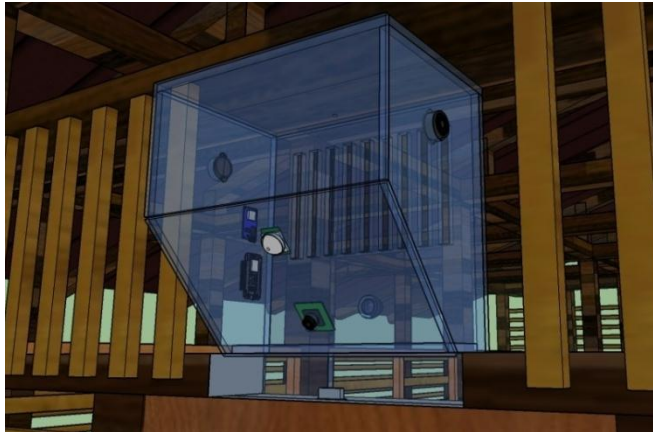


**Gambar 3. 3 Skema Hardware**

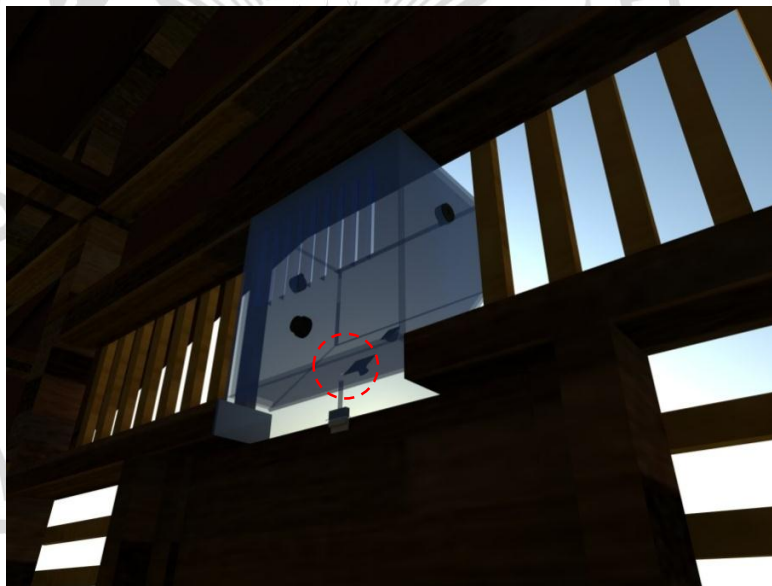


*Gambar 3. 4 Skema Hardware beserta ukuran dan letak komponen*

Dari skema hardware pada Gambar 3. 3 jika diaplikasikan di atas pintu kandang kambing seperti pada Gambar 3.5



**Gambar 3.5** Skema Hardware yang diaplikasikan dikandang  
Berikut adalah peletakkan sensor MC-38 yang diletakkan di daun pintu dan sisi lainnya diletakkan didalam kusen pintu seperti yang ditunjukkan pada **Gambar 3. 6**



**Gambar 3. 6** Letak sensor MC-38

### 3.3.2. Perancangan Software

Perancangan software disini adalah bagaimana alur sistem dan cara kerja alat yang digambarkan dalam flowchart pada **Gambar 3.9**.

Sedangkan perangkat lunak yang digunakan adalah aplikasi Arduino IDE 1.8.10, yang telah diinstal board NodeMCU ESP8266, dan board ESP32-CAM berfungsi sebagai aplikasi programming atau koding dari Modul tersebut sebagai pengontrol program. selain board untuk menghubungkan ke telegram yaitu menggunakan library UniversalTelegramBot. Pada sistem informasi keamanan kandang kambing ini nantinya akan bekerja dengan dua sistem, sistem auto dan sistem kendali manual, pada saat sistem auto, semua input dan output akan bekerja secara otomatis seperti saat sensor PIR (*Passive Infrared Receiver*) dalam keadaan *standby*, pada saat sensor PIR mendeteksi adanya gerakan manusia maka ESP32-CAM secara otomatis akan mengambil gambar lalu akan dikirim ke server telegram selanjutnya peternak kambing tersebut menerima pelaporan data berupa teks beserta foto yang diambil melalui aplikasi telegramnya, selain sensor PIR sensor magnet MC-38 yang terdapat di pintu dalam kondisi *standby* dan jika objek yang dicurigai tersebut membuka pintu maka sistem akan menganggap bahwa itu tindakan ilegal selanjutnya sensor MC-38 akan menunjukkan *logic* “1” maka buzzer alarm akan mengeluarkan bunyi sirine dan mengirim teks peringatan Bahwa pintu dalam kondisi terbuka ke pemilik kambing. Sedangkan pada sistem kendali manual, pengguna atau pemilik kandang bisa mengirimkan kode perintah dengan beberapa kode yang telah di terapkan pada kodingan sebelumnya, berikut macam-macam kode yang diterapkan beserta penjelasannya mulai dari awal:

1. Kode “/Start” maka akan muncul tampilan awal



2. Kode “/Masuk” maka akan menampilkan menu perintah awal

Pada menu perintah awal terdapat kode perintah diantaranya :

1. Kode “/AutoON” untuk mengaktifkan sistem otomatis
2. Kode “/AutoOFF” untuk menonaktifkan sistem otomatis
3. Kode “/Manual” untuk menampilkan tampilan menu pada perintah manual

4. Kode “/About” untuk menampilkan penjelasan mengenai Alat

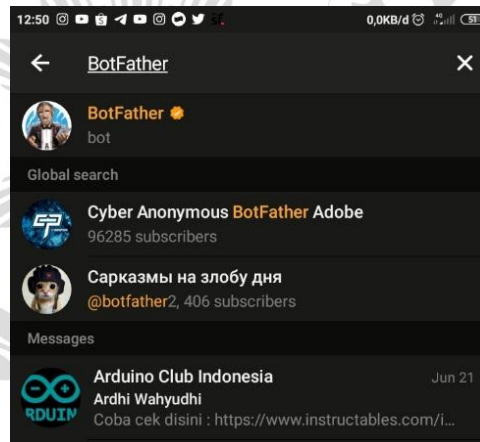
Pada tampilan menu pada perintah manual terdapat kode diantaranya :

1. Kode “/Memotret” ESP32-CAM untuk mengambil gambar ,
2. Kode “/NyalakanLEDbantuan” untuk mengaktifkan lampu kandang.
3. Kode “/MatikanLEDbantuan” untuk menonaktifkan lampu kandang.
4. Kode “/MatikanBuzzer” untuk menonaktifkan buzzer.
5. Kode “/StatusMC38” untuk mengecek keadaan sensor aktif atau tidak.
6. Kode “/StatusLED” untuk mengecek keadaan lampu kandang

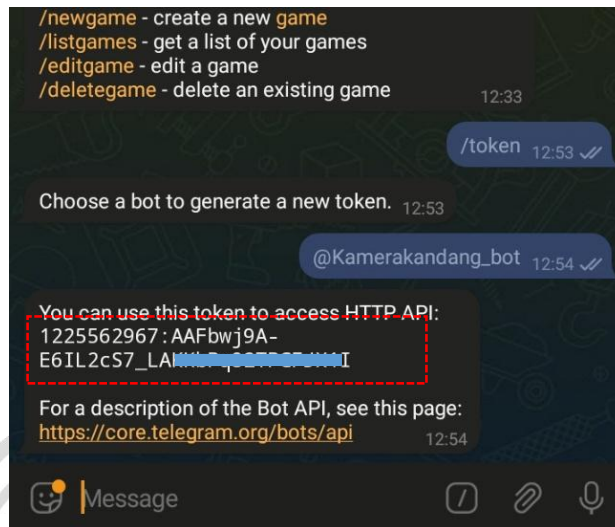
#### **3.3.2.1. Bot Telegram**

Pada dasarnya telegram bot tidak memerlukan nomor telepon tambahan untuk diatur. Pengguna bisa berkomunikasi dengan bot dengan kirim pesan dan perintah ke bot dengan membuka obrolan dengannya atau dengan menambahkannya ke grup. Jadi semua

pesan, perintah dan permintaan yang dikirim oleh pengguna diteruskan ke *software* yang berjalan di server melalui *Interface* HTTPS versi API Telegram yang disebut *Interface* API Bot. Untuk membuat bot tersebut yang pertama cari username @BotFather di kolom pencarian seperti pada Gambar 3.7 setelah itu buat bot telegram, agar mengetahui token bot pilih perintah `"/token"` pilih bot kalian selanjutnya akan mendapatkan token `"1225562967:AAFbwj9A-E6IL2c57_LAKKbPQ82*****"` seperti pada Gambar 3. 8 yang nantinya token tersebut untuk mengakses HTTP API dan dimasukkan ke kodingan pada arduino



**Gambar 3.7** pencarian BotFaher



**Gambar 3. 8** cek token bot telegram

### 3.3.2.2. Program kode Arduino

Berdasarkan perencanaan sistem selanjutnya dilakukan sebuah pengodingan pada board NodeMCU ESP8266 dan ESP32-CAM, sebelumnya terlebih dahulu menginstall board ESP8266 v.5.2.0, ESP 32 v.1.0.4 dan untuk librarynya menggunakan UniversalTelegramBot v.1.1.0, ArduinoJson v.5.13.5 yang akan dipakai dimana program tersebut terdiri dari :

#### a. Menghubungkan Wifi

Untuk menghubungkan ke WiFi Router hal pertama yang dibutuhkan adalah *library* ESP8266WiFi setelah itu untuk mengatur koneksi masukkan SSID dan password WiFi didalam koding arduino seperti dibawah ini :

```
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
```

```

// Atur SSID Dan Password Wifi Router / Wifi Hotspot / Wifi Tethring
Dari HP
char ssid[] = "bismillah"; // Masukan Nama SSID (Besar Kecil
Pengaruh)
char password[] = "arirehan123"; // Masukan Password Yang Di
Pakai Router
// sedangkan pada void terdapat koding seperti berikut
void setup() {
Serial.begin(115200);
// Set WiFi to station mode and disconnect from an AP if it was
Previously
// terhubung
WiFi.mode(WIFI_STA);
WiFi.disconnect();
delay(100);
// mencoba untuk terhubung ke wifi:
Serial.print("Connecting Wifi: ");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
Serial.print(".");
delay(500);
}
client.setInsecure();
Serial.println("");
Serial.println("WiFi connected");
Serial.print("IP address: ");
Serial.println(WiFi.localIP());

```

#### **b. Setting Perintah Telegram**

Pada pengaturan perintah telegram library yang digunakan adalah UniversalTelegramBot v.1.1.0 terdapat beberapa kode pilihan yang dimasukkan didalam program seperti pada kodingan dibawah ini:

```

#include <UniversalTelegramBot.h>
#define BOTtoken
"1225562967:AAfBwj9E6IL2cS7_LAKKbPq82TPCFJX1I" // Masukan
Token Kalian (Silakan Copy Dari Botfather)
WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);
int Bot_mtbs = 1000; //Untuk lamanya Scan pesan baru
long Bot_lasttime; // Pesan Terakhir
bool Start = false;
// untuk setting perintah pada telegram ganti nama dalam tanda kutip
sesuai apa yang direncanakan
const char* Menu_1 = "/Start";
const char* Menu_2 = "/Masuk";
const char* Menu_3 = "/about";
const char* Menu_4 = "/Manual";
const char* Menu_5 = "/AutoON";
const char* Menu_6 = "/AutoOFF";
//perintah manual & ON
const char* Perintah_1 = "/Memotret";
const char* Perintah_2 = "/MatikanBuzzer";
const char* Perintah_ON_2 = "/NyalakanLEDbantuan";
//perintah OFF
const char* Perintah_OFF_2 = "/MatikanLEDbantuan"; //sudah set
berdasarkan kodingan di atas maka selanjutnya akan diatur
fungsinya pada setiap perintah seperti kodingan di bawah ini:

void handleNewMessages(int numNewMessages) {
Serial.println("handleNewMessages");
Serial.println(String(numNewMessages));
for (int i=0; i<numNewMessages; i++) {
String chat_id = String(bot.messages[i].chat_id);
String Pesan = bot.messages[i].text;

```

```

String from_name = bot.messages[i].from_name;

if (from_name == "") from_name = "Guest";

if (Pesan == (Menu_1))

{

String start1 = "Selamat datang di Cam-bing " + from_name + ".\n";

start1 += "Sistem Informasi Keamanan Kandang Kambing Berbasis
IOT\n";

start1 += "Cam-bing by Electro Ndeso\n\n";

start1 += "Sistem keamanan ini dilengkapi\n";

start1 += " -Sensor PIR\n";

start1 += " -Sensor MC-38\n";

start1 += " -ESP32-CAM\n";

start1 += " -NODEMCU ESP8266\n";

start1 += " -Buzzer\n\n";

start1 += "Menggunakan media aplikasi Telegram Messenger\n";

start1 += "\n";

start1 += "Manfaat bagi para peternak kambing\n";

start1 += "1. Untuk membantu para peternak kambing menjaga hewan
ternaknya dari";

start1 += " orang yang berniat jahat\n";

start1 += "2. Agar mengetahui siapa yang berada disekitar area
kandang\n";

start1 += "3. Melalui alarm yang menandakan sinyal peringatan bahaya,
pemilik";

start1 += " kandang akan terjaga dari tidurnya.\n\n";

start1 += "Untuk melanjutkan ke Pengendalian alat anda dapat mengklik
(/Masuk)\n";

bot.sendMessage(chat_id, start1, "Markdown");

```

```

}

else if (Pesan == (Menu_2))

{

String Masuk = "Selamat Datang di sistem keamanan cam-bing " +
from_name + ".\n";

Masuk += "Gunakan Kode Berikut Untuk Mengontrol\n\n";

Masuk += "- /AutoON : mengaktifkan Mode pintar (disini proses
pengambilan";

Masuk += "gambar dan pengiriman data akan berlangsung otomatis \n";
Masuk += "- /AutoOFF : menonaktifkan Mode pintar \n";
Masuk += "- /Manual : menuju ke pengendalian manual \n" ;
Masuk += "- /about : Penjelasan mengenai alat keamanan Cam-bing
ini \n\n";

bot.sendMessage(chat_id, Masuk, "Markdown");
}

// kodingan untuk menampilkan text menu awal di telegram
if (Pesan == (Menu_4))
{

String welcome = "Selamat Datang di sistem keamanan cam-bing, " +
from_name + ".\n";

welcome += "Gunakan Kode Berikut Untuk Mengontrol Cam-bing
:\n\n";

welcome += (Perintah_1);

welcome += " <- Tekan untuk memotret\n\n";

welcome += "Kontrol buzzer & LED :\n\n";

welcome += (Perintah_ON_2);

welcome += " <- Tekan ON\n";

welcome += (Perintah_OFF_2);

```

```

welcome += " <- Tekan OFF \n\n";

welcome += (Perintah_2);

welcome += " <- Tekan OFF \n\n";

welcome += "Cek Status Sensor & LED :\n\n";

welcome += (Status_MC38);

welcome += " <- Cek Status\n";

welcome += (Status_LED);

welcome += " <- Cek Status \n\n";

bot.sendMessage(chat_id, welcome, "Markdown");
}

//-----settingan pada perintah autoon-----

else if (Pesan == (Menu_5)) //autoon
{
statusmc38 = 0;
digitalWrite (autoPin, HIGH);
bot.sendMessage(chat_id, "proses mengaktifkan auto...");
kondisimc = 1;
kondisiauto1 = 1;
delay(100);
}

//-----settingan untuk autooff-----

else if (Pesan == (Menu_6)) //autooff
{
statusmc38 = 1;
digitalWrite (autoPin, LOW);

```



```

digitalWrite (buzPin, LOW);

bot.sendMessage(chat_id, "proses menonaktifkan auto...."); // bot
mengirim balasan

kondisimc = 0;

kondisiauto1 = 0;

delay(100);

}

//-----settingan mematikan buzzer-----
else if (Pesan == (Perintah_2)) //perintah mematikan buzzer
{
digitalWrite (buzPin, LOW);
statusbuz = 0;
kondisimc = 0;
bot.sendMessage(chat_id, "Buzzer dimatikan");
}

//-----koding untuk kendali lampu LED-----
else if (Pesan == (Perintah_OFF_2)) //menonaktifkan LED
{
statusled = 1;

digitalWrite (ledPin, LOW);
bot.sendMessage(chat_id, "LED bantuan dimatikan");
}

else if (Pesan == (Perintah_ON_2)) //mengaktifkan LED
{

statusled = 0;

digitalWrite (ledPin, HIGH);

bot.sendMessage(chat_id, "LED bantuan dinyalakan");
}

```

```

}

//-----kodingan untuk mengecek status-----

else if (Pesan == (Status_MC38))

{

if(statusmc38){

bot.sendMessage(chat_id, "Lapor Mas Ari, MC38 dalam keadaan
nonaktif", "");

}

else

{

bot.sendMessage(chat_id, "Lapor Mas Ari, MC38 dalam keadaan aktif",
"");

}

}

else if (Pesan == (Status_LED))

{

if(statusled){

bot.sendMessage(chat_id, "Lapor Mas Ari, LED Bantuan dalam keadaan
nonaktif", "");

}

else

{

bot.sendMessage(chat_id, "Lapor Mas Ari, LED Bantuan dalam keadaan
aktif", "");

}

}

}

```

### c. Settingan Sensor MC38 dan PIR

Pada settingan di sensor ini memiliki konsep yang sama yaitu seperti pada kasus *pushbutton* ditekan berikut adalah kodingan pada sensor MC-38 dan PIR:

```
//---koding pada Sensor PIR---  
int mcPin = D2; //Pin Mc-38  
pinMode(mcPin, INPUT_PULLUP);  
int v = digitalRead(mcPin);  
if( (v==1)&&(kondisimc == 1)) {  
digitalWrite (buzPin,HIGH);  
Serial.println ("Lawang mbukak!");  
bot.sendMessage(chat_id, "pintu dalam keadaan terbuka, buzzer menyala!!!!");  
delay(5000);  
}  
else if (kondisimc == 0) {  
v = 0;  
}  
//---koding pada Sensor MC-38---  
int gpioPIR = 15; //PIR Motion Sensor  
pinMode(gpioPIR, INPUT_PULLUP);  
int v = digitalRead(gpioPIR);  
if( (v==1)&&(kondisi == 1)) {  
sendCapturedImage2Telegram(token, chat_id);  
delay(5000);  
}  
else if (kondisi == 0) {  
v = 0;  
}  
delay(1000);  
}
```

Pada kodingan di atas jika kutub mc-38 saling melepas maka buzzer akan menyala, berikutnya bot akan mengirim notifikasi bahwa pintu dalam keadaan terbuka. Sedangkan pada sensor PIR mendeteksi gerakan ESP32-CAM akan mengambil dan mengirim gambar ke pengguna telegram.

#### **d. Mengambil dan mengirim gambar ke telegram pada ESP32-CAM**

Dalam proses pengambilan pada ESP32-CAM dimulai dari proses mengambil gambar selanjutnya menghubungkan ke domain “api.telegram.org” setelah itu dikirim ke aplikasi telegram yang berada di smartphone pemilik kandang melalui token telegram bot yang telah dibuat sebelumnya apabila pengguna saat itu offline maka data nantinya akan dikirim pada saat pengguna telegram online kembali. Berikut adalah kodingannya :

```
//---- koneksi ESP32-CAM ke wifi
void setup()
{
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
  Serial.begin(115200);
  delay(10);
  WiFi.mode(WIFI_STA);
  Serial.println("");
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  long int StartTime=millis();
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    if ((StartTime+10000) < millis()) break;
  }
}
```

```

Serial.println("");
Serial.println("STAIP address: ");
Serial.println(WiFi.localIP());
Serial.println("");
if (WiFi.status() != WL_CONNECTED) {
  Serial.println("Reset");
  ledcAttachPin(4, 3);
  ledcSetup(3, 5000, 8); // indikator koneksi wifi
  ledcWrite(3,10);
  delay(200);
  ledcWrite(3,0);
  delay(200);
  ledcDetachPin(3);
  delay(1000);
  ESP.restart();
}
else {
  ledcAttachPin(4, 3);
  ledcSetup(3, 5000, 8);
  for (int i=0;i<5;i++) {
    ledcWrite(3,10);
    delay(200);
    ledcWrite(3,0);
    delay(200);
  }
  ledcDetachPin(3);
}
camera_config_t config; // pengaturan kamera
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = Y2_GPIO_NUM;
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;

```

```

config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;
//init with high specs to pre-allocate larger buffers
if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10; //0-63 lower number means higher quality
    config.fb_count = 2;
} else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12; //0-63 lower number means higher quality
    config.fb_count = 1;
}

//-----setting kualitas kamera ESP32-CAM-----
//drop down frame size for higher initial frame rate
sensor_t * s = esp_camera_sensor_get();
s->set_framesize(s,FRAMESIZE_SVGA); //
UXGA|SXGA|XGA|SVGA|VGA|CIF|QVGA|HQVGA|QQVGA
s->set_vflip(s, 1); // vertikal flip
//-----proses pengiriman gambar ke telegam-----
String sendCapturedImage2Telegram(String token, String chat_id) {
const char* myDomain = "api.telegram.org";
String getAll="", getBody = "";

```

```

//----- pengambilan gambar-----
camera_fb_t * fb = NULL;
fb = esp_camera_fb_get();
if(!fb) {
Serial.println("Camera capture failed");
delay(1000);
ESP.restart();
return "Camera capture failed";
}
//-----koneksi ke telegram-----
Serial.println("Connect to " + String(myDomain));
WiFiClientSecure client_tcp;
if (client_tcp.connect(myDomain, 443)) {
Serial.println("Connection successful");
String head = "--Taiwan\r\nContent-Disposition: form-data; name=\"chat_id\";
\r\n\r\n" + chat_id + "\r\n--Taiwan\r\nContent-Disposition: form-data;
name=\"photo\"; filename=\"esp32-cam.jpg\"\r\nContent-Type:
image/jpeg\r\n\r\n";
String tail = "\r\n--Taiwan--\r\n";
uint16_t imageLen = fb->len;
uint16_t extraLen = head.length() + tail.length();
uint16_t totalLen = imageLen + extraLen;
client_tcp.println("POST /bot"+token+"/sendPhoto HTTP/1.1"); //mengirim foto
ke telegram melalui bot.
client_tcp.println("Host: " + String(myDomain));
client_tcp.println("Content-Length: " + String(totalLen));
client_tcp.println("Content-Type: multipart/form-data; boundary=Taiwan");
client_tcp.println();
client_tcp.print(head);
uint8_t *fbBuf = fb->buf;

size_t fbLen = fb->len;

for (size_t n=0;n<fbLen;n=n+1024) {

if (n+1024<fbLen) {

```

```

client_tcp.write(fbBuf, 1024);

fbBuf += 1024;

}

else if (fbLen%1024>0) {

size_t remainder = fbLen%1024;

client_tcp.write(fbBuf, remainder);

}

}

client_tcp.print(tail);
esp_camera_fb_return(fb);

int waitTime = 10000; // timeout 10 seconds

long startTime = millis();

boolean state = false;

while ((startTime + waitTime) > millis())

{

Serial.print(".");

delay(100);

while (client_tcp.available())

{

char c = client_tcp.read();

if (c == '\n')

{

if (getAll.length()==0) state=true;

getAll = "";

}

}

```



```

else if (c != '\r')

getAll += String(c);

if (state==true) getBody += String(c);

startTime = millis();

}

if (getBody.length()>0) break;

}

client_tcp.stop();

Serial.println(getBody);

}

else {

getBody="Connected to api.telegram.org failed.";

Serial.println("Connected to api.telegram.org failed.");

}

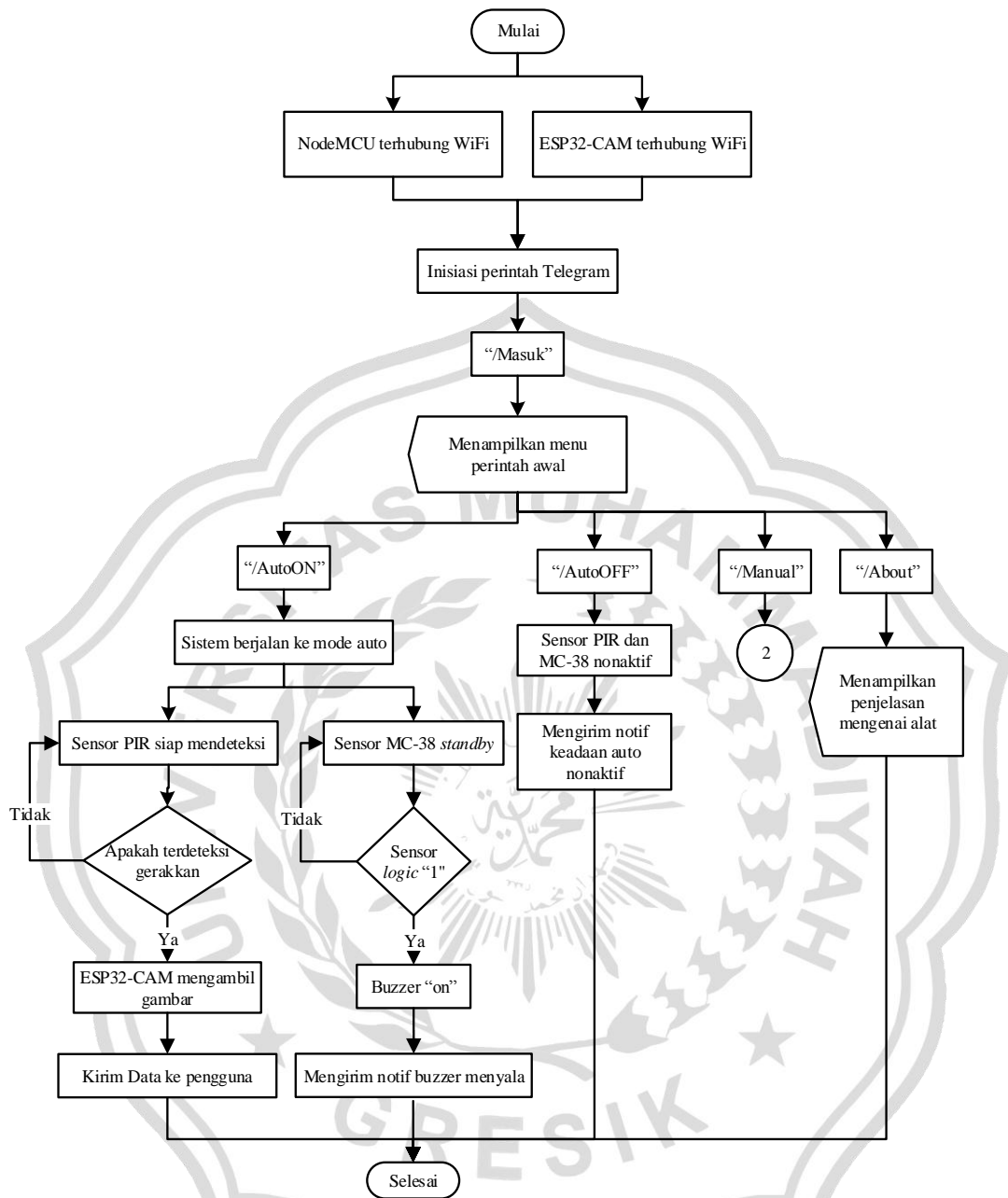
return getBody;

}

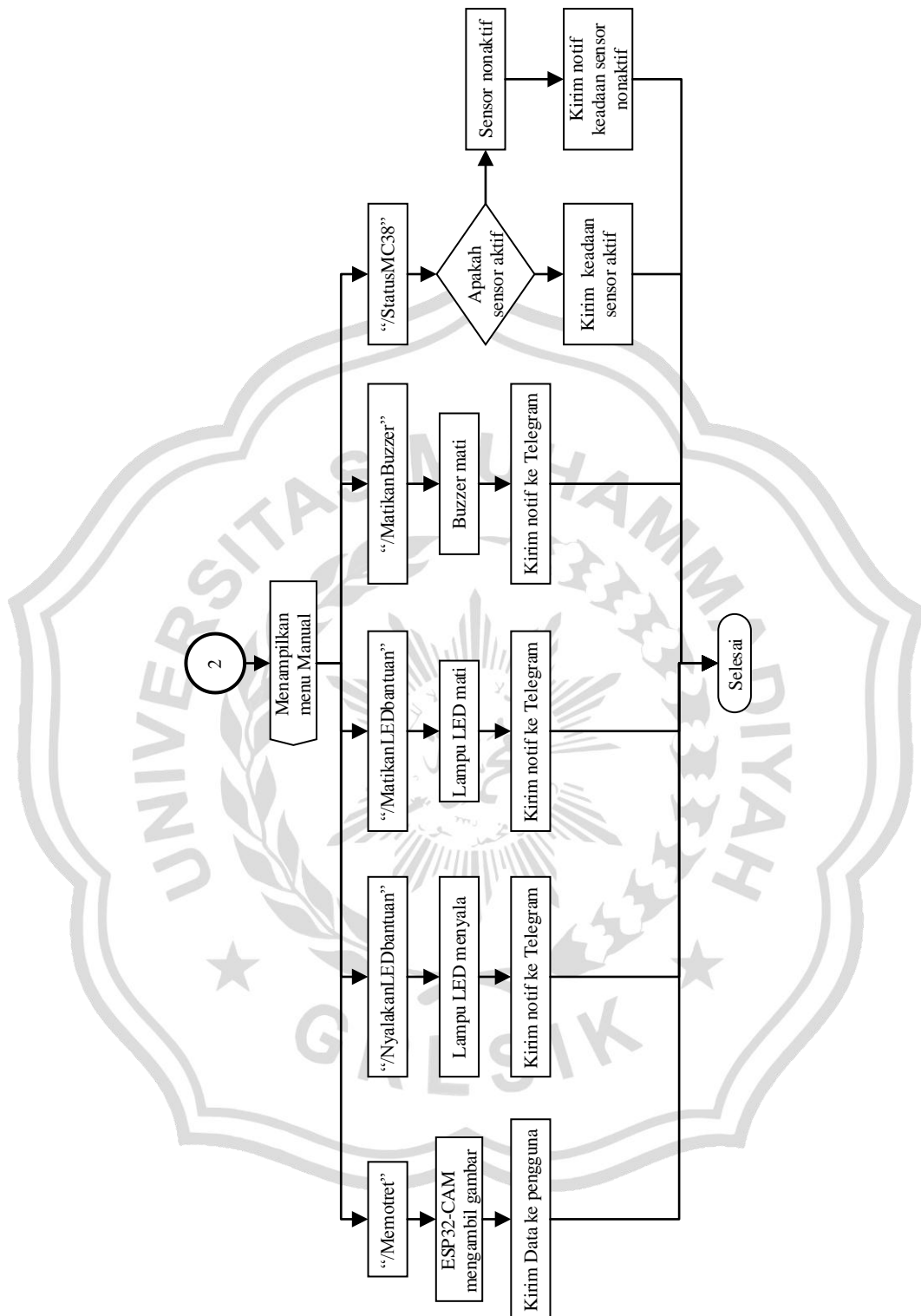
```

Berdasarkan dari kode-kode dasar tersebut diharapkan nantinya akan berjalan lancar sesuai rencana alat yang akan dibuat.

Berikut adalah flowchart dari perancangan software dari sistem informasi keamanan kandang kambing berbasis internet of things seperti yang ditunjukkan pada **Gambar 3.9** bagian1 dan **Gambar 3. 10** bagian 2



Gambar 3.9 Flowchart Perancangan Software bagian 1



Gambar 3. 10 Flowchart Perancangan Software bagian 2