LAMPIRAN

Lampiran 1 list Program

1. Program Pengujian Kamera VC0706 dengan Arduino Uno

```
// This is a basic snapshot sketch using the VC0706 library.
// On start, the Arduino will find the camera and SD Card and
// then snap a photo, saving it to the SD Card.
// Public domain.

// If using an Arduino Mega (1280, 2560 or ADK) in conjunction
// with an SD Card shield designed for conventional Arduinos
// (Uno, etc.), it's necessary to edit the library file:
//   libraries/SD/utility/Sd2Card.h
// Look for this line:
//   #define MEGA_SOFT_SPI 0
// change to:
//   #define MEGA_SOFT_SPI 1
// This is NOT required if using an SD Card breakout interfaced
// directly to the SPI bus of the Mega (pins 50-53), or if using
// a non-Mega, Uno-style board.

#include <Adafruit_VC0706.h>
#include <SPI.h>
#include <SD.h>

// comment out this line if using Arduino V23 or earlier
#include <SoftwareSerial.h>

// uncomment this line if using Arduino V23 or earlier
// #include <NewSoftSerial.h>

// SD Card chip select line varies among boards/shields:
// Adafruit SD shields and modules: pin 10
// Arduino Ethernet shield: pin 4
// Sparkfun SD shield: pin 8
// Arduino Mega w/hardware SPI: pin 53
// Teensy 2.0: pin 0
// Teensy++ 2.0: pin 20
#define chipSelect 10

// Pins for camera connection are configurable.
// With the Arduino Uno, etc., most pins can be used, except for
// those already in use for the SD Card (10 through 13 plus
// chipSelect, if other than pin 10).
// With the Arduino Mega, the choices are a bit more involved:
```

```
// 1) You can still use SoftwareSerial and connect the camera to
//    a variety of pins...BUT the selection is limited.  The TX
//    pin from the camera (RX on the Arduino, and the first
//    argument to SoftwareSerial()) MUST be one of: 62, 63, 64,
//    65, 66, 67, 68, or 69.  If MEGA_SOFT_SPI is set (and using
//    a conventional Arduino SD shield), pins 50, 51, 52 and 53
//    are also available.  The RX pin from the camera (TX on
//    Arduino, second argument to SoftwareSerial()) can be any
//    pin, again excepting those used by the SD Card.
// 2) You can use any of the additional three hardware UARTs on
//    the Mega board (labeled as RX1/TX1, RX2/TX2, RX3,TX3),
//    but must specifically use the two pins defined by that
//    UART; they are not configurable.  In this case, pass the
//    desired Serial object (rather than a SoftwareSerial
//    object) to the VC0706 constructor.

// Using SoftwareSerial (Arduino 1.0+) or NewSoftSerial (Arduino 0023 &
prior):
#if ARDUINO >= 100
// On Uno: camera TX connected to pin 2, camera RX to pin 3:
SoftwareSerial cameraconnection = SoftwareSerial(2, 3);
// On Mega: camera TX connected to pin 69 (A15), camera RX to pin 3:
//SoftwareSerial cameraconnection = SoftwareSerial(69, 3);
#else
NewSoftSerial cameraconnection = NewSoftSerial(2, 3);
#endif

Adafruit_VC0706 cam = Adafruit_VC0706(&cameraconnection);

// Using hardware serial on Mega: camera TX conn. to RX1,
// camera RX to TX1, no SoftwareSerial object is required:
//Adafruit_VC0706 cam = Adafruit_VC0706(&Serial1);

void setup() {

  // When using hardware SPI, the SS pin MUST be set to an
  // output (even if not connected or used).  If left as a
  // floating input w/SPI on, this can cause lockuppage.
#if !defined(SOFTWARE_SPI)
#if defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__)
  if(chipSelect != 53) pinMode(53, OUTPUT); // SS on Mega
#else
  if(chipSelect != 10) pinMode(10, OUTPUT); // SS on Uno, etc.
#endif
#endif

  Serial.begin(9600);
  Serial.println("VC0706 Camera snapshot test");
```

```
// see if the card is present and can be initialized:
if (!SD.begin(chipSelect)) {
  Serial.println("Card failed, or not present");
  // don't do anything more:
  return;
}

// Try to locate the camera
if (cam.begin()) {
  Serial.println("Camera Found:");
} else {
  Serial.println("No camera found?");
  return;
}
// Print out the camera version information (optional)
char *reply = cam.getVersion();
if (reply == 0) {
  Serial.print("Failed to get version");
} else {
  Serial.println("-----------------");
  Serial.print(reply);
  Serial.println("-----------------");
}

// Set the picture size - you can choose one of 640x480, 320x240 or 160x120
// Remember that bigger pictures take longer to transmit!

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

// You can read the size back from the camera (optional, but maybe useful?)
uint8_t imgsize = cam.getImageSize();
Serial.print("Image size: ");
if (imgsize == VC0706_640x480) Serial.println("640x480");
if (imgsize == VC0706_320x240) Serial.println("320x240");
if (imgsize == VC0706_160x120) Serial.println("160x120");

Serial.println("Snap in 3 secs...");
delay(3000);

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

// Create an image with the name IMAGExx.JPG
```

```
  char filename[13];
  strcpy(filename, "IMAGE00.JPG");
  for (int i = 0; i < 100; i++) {
    filename[5] = '0' + i/10;
    filename[6] = '0' + i%10;
    // create if does not exist, do not open existing, write, sync after write
    if (! SD.exists(filename)) {
      break;
    }
  }

  // Open the file for writing
  File imgFile = SD.open(filename, FILE_WRITE);

  // Get the size of the image (frame) taken
  uint16_t jpglen = cam.frameLength();
  Serial.print("Storing ");
  Serial.print(jpglen, DEC);
  Serial.print(" byte image.");

  int32_t time = millis();
  pinMode(8, OUTPUT);
  // Read all the data up to # bytes!
  byte wCount = 0; // For counting # of writes
  while (jpglen > 0) {
    // read 32 bytes at a time;
    uint8_t *buffer;
    uint8_t bytesToRead = min(32, jpglen); // change 32 to 64 for a speedup but
may not work with all setups!
    buffer = cam.readPicture(bytesToRead);
    imgFile.write(buffer, bytesToRead);
    if(++wCount >= 64) { // Every 2K, give a little feedback so it doesn't appear
locked up
      Serial.print('.');
      wCount = 0;
    }
    //Serial.print("Read  ");     Serial.print(bytesToRead,  DEC);  Serial.println("
bytes");
    jpglen -= bytesToRead;
  }
  imgFile.close();

  time = millis() - time;
  Serial.println("done!");
  Serial.print(time); Serial.println(" ms elapsed");
}

void loop() {
```

```
}
```

2. **Program Sistem Keamanan Ruangan**

```
#include<Wire.h>
#include<LCD.h>
#include<LiquidCrystal_I2C.h>
#include <Adafruit_VC0706.h>
#include <SPI.h>
#include <SD.h>
#include <SoftwareSerial.h>
LiquidCrystal_I2C lcd(0x27,2,1,0,4,5,6,7);

#define pinEcho 3
#define pinTrig 2
#define chipSelect 53
int dataIn = 5;
int buttonState = 0;
int PIR;
long duration, distance;
int ledAlarm  = 6;
int pinBuzzer = 7 ;
int calibTime = 10;
char fileName[13];
boolean keluar = false;

Adafruit_VC0706 cam = Adafruit_VC0706(&Serial3);

void setup() {
  Serial.begin(9600);
  //digitalWrite(ledAlarm,HIGH);
  lcd.begin(16,2);
  lcd.setBacklightPin(3,POSITIVE);
  lcd.setBacklight(HIGH);
  pinMode(dataIn, INPUT);

  pinMode(pinEcho, INPUT);
  pinMode(pinTrig, OUTPUT);
  pinMode(ledAlarm, OUTPUT);
  pinMode(pinBuzzer, OUTPUT);
  pinMode(53, OUTPUT);


  lcd.setCursor(0,0);
  lcd.print("Suhendro A U");
  lcd.setCursor(0,1);
  lcd.print("5215111740");
```

```
delay(5000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Kalibrasi Sensor");
delay(1000);


//Serial.println("Please wait, now calibrating the sensor....");
  for(int i = 0; i <= calibTime; i++){
    //Serial.print(((i*100)/calibTime));
    //Serial.print("% ");
    lcd.setCursor(12,1);
    lcd.print(((i*100)/calibTime));
    lcd.print("%");
   // Serial.println("DONE.....");
    delay(1000);
    }
 // Serial.println("Calibration Succesfully Done.");
 // Serial.println("** SENSOR ACTIVE **");
 // delay(50);
lcd.clear();
lcd.setCursor(2,0);
lcd.print(" SELESAI ");
delay(1000);
lcd.clear();
lcd.setCursor(1,0);
lcd.print("SENSOR ACTIVE!");
delay(2000);


// see if the card is present and can be initialized:
if (!SD.begin(chipSelect)) {
 Serial.println("Card failed, or not present");
 lcd.setCursor (2,0);
 lcd.print("Card Failed");
 delay (2000);
  // don't do anything more:
  return;
}

// Try to locate the camera
if (cam.begin()) {
 Serial.println("Camera Found:");

  lcd.clear();
  lcd.setCursor(2,0);
  lcd.print("Camera Found");
  delay(2000);
```

```
} else {
 Serial.println("No camera found?");
  lcd.clear();
  lcd.setCursor(2,1);
  lcd.print("NO CAMERA");

  delay(2000);
  return;
}
// Print out the camera version information (optional)
char *reply = cam.getVersion();
if (reply == 0) {
  Serial.print("Failed to get version");
  lcd.clear();
  lcd.setCursor(1,0);
  lcd.print("Fail get versi");
  delay(2000);
  return;
} else {
  Serial.println("-----------------");
  Serial.print(reply);
  lcd.clear();
  lcd.setCursor(1,0);
  lcd.print("versi didapat");
  delay(2000);
  Serial.println("-----------------");
}



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

uint8_t imgsize = cam.getImageSize();
//Serial.print("Image size: ");
//if (imgsize == VC0706_640x480) Serial.println("640x480");
//if (imgsize == VC0706_320x240) Serial.println("320x240");
//if (imgsize == VC0706_160x120) Serial.println("160x120");

lcd.clear();
lcd.setCursor(0,0);
lcd.print("Image size : ");
lcd.setCursor(0,1);
if (imgsize == VC0706_640x480) lcd.print("640x480");
if (imgsize == VC0706_320x240) lcd.print("320x240");
if (imgsize == VC0706_160x120) lcd.print("160x120");
```

```
  delay(3000);



  cam.setMotionDetect(true);          // turn it on


  //Serial.print("Motion detection is ");
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Deteksi gerakan");

  if (cam.getMotionDetect()) {
    //Serial.println("ON");
    lcd.setCursor(0,1);
    lcd.print("ON ");
    delay(2000);
}
  else {
   // Serial.println("OFF");
    lcd.setCursor(0,1);
    lcd.print("OFF");
    delay(2000);
    return;
  }
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("SISTEM KEAMANAN");
  lcd.setCursor(4,1);
  lcd.print("RUANGAN");
  delay(3000);

}

void sensor_ultrasonik(){
  digitalWrite (pinTrig, LOW);
  delayMicroseconds(2);
  digitalWrite (pinTrig, HIGH);
  delayMicroseconds(10);
  digitalWrite(pinTrig,LOW);
  duration = pulseIn(pinEcho, HIGH);
  distance = duration/58.2;
  Serial.println(distance);
  delay(500);
}
void pir(){
  PIR = digitalRead(4);
  if (PIR == HIGH){
```

```
    Serial.println("1");
    lcd.setCursor(6,1);
    lcd.print("1");
    delay(100);
  }
  else
  {
    lcd.setCursor(6,1);
    Serial.println("0");
    lcd.print("0");
    delay(100);
  }
}

void capture(){

   cam.setMotionDetect(false);

  if (! cam.takePicture()){
   // Serial.println("Failed to snap!");
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("FAILID TO SNAP");
    return;

  }
  else {
    //Serial.println("Picture taken!");
    lcd.clear();
    lcd.setCursor(1,0);
    lcd.print("PICTURE TAKEN");

    delay(1000);
  }
  char filename[13];
  strcpy(filename, "IMAGE00.JPG");
  for (int i = 0; i < 100; i++) {
    filename[5] = '0' + i/10;
    filename[6] = '0' + i%10;

    // create if does not exist, do not open existing, write, sync after write
    if (! SD.exists(filename)) {
      break;
    }
  }

  File imgFile = SD.open(filename, FILE_WRITE);
```

```
  uint16_t jpglen = cam.frameLength();
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(filename);

  while (jpglen > 0) {
    // read 32 bytes at a time;
    uint8_t *buffer;
    uint8_t bytesToRead = min(32, jpglen);
    buffer = cam.readPicture(bytesToRead);
    imgFile.write(buffer, bytesToRead);

    jpglen -= bytesToRead;
  }
  imgFile.close();
 // Serial.println("...Done!");
  lcd.setCursor(0,0);
  lcd.print("Done.......!");
  delay(1000);
  cam.resumeVideo();
  cam.setMotionDetect(true);
}


void alarm(){
   digitalWrite(ledAlarm, HIGH);
   digitalWrite(pinBuzzer,HIGH);
   lcd.clear();
   lcd.setCursor(0,0);
   lcd.print(" ALARM! ALARM! ");
   lcd.setCursor(0,1);
   lcd.print("ADA PENYUSUP !!!");
   delay(700);


   cam.setMotionDetect(false);

  if (! cam.takePicture()){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("FAILID TO SNAP");
  }
  else {
    lcd.clear();
    lcd.setCursor(1,0);
    lcd.print("PICTURE TAKEN");

    delay(1000);
```

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

    // create if does not exist, do not open existing, write, sync after write
    if (! SD.exists(filename)) {
      break;
    }
  }

  File imgFile = SD.open(filename, FILE_WRITE);
  uint16_t jpglen = cam.frameLength();
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(filename);




  while (jpglen > 0) {
    // read 32 bytes at a time;
    uint8_t *buffer;
    uint8_t bytesToRead = min(32, jpglen);
    buffer = cam.readPicture(bytesToRead);
    imgFile.write(buffer, bytesToRead);

    //Serial.print("Read   ");     Serial.print(bytesToRead,   DEC);   Serial.println("
bytes");

    jpglen -= bytesToRead;
  }
  imgFile.close();
 // Serial.println("...Done!");
  lcd.setCursor(0,0);
  lcd.print("Done.......!");
  delay(1000);
  cam.resumeVideo();
  cam.setMotionDetect(true);
}



void loop() {

  buttonState = digitalRead(dataIn);
```

```
  if (buttonState == HIGH){
   lcd.clear();
   lcd.setCursor(0,0);
   lcd.print("Sistem Keamanan");
   lcd.setCursor(4,1);
   lcd.print(" OFF ");
   delay(2000);
   digitalWrite(pinBuzzer,LOW);
   capture();
   delay(1000);
    keluar = true;
  }
  else
 {
   if (buttonState == LOW){
     keluar = false;
     digitalWrite(pinBuzzer,LOW);
     pir();
     sensor_ultrasonik();
     Serial.print(distance);
     lcd.clear();
     lcd.setCursor(2,0);
     lcd.print("Sistem Aktif");
     lcd.setCursor(8,1);
     lcd.print(distance);
     digitalWrite(ledAlarm, LOW);

  if (cam.motionDetected()|| digitalRead(4)== HIGH||distance < 10) {
    alarm();
     }
   }
 }

  while (keluar){
   lcd.clear();
   lcd.setCursor(0,0);
   lcd.print("Sistem Keamanan");
   lcd.setCursor(4,1);
   lcd.print(" OFF ");
   delay(500);
   digitalWrite(pinBuzzer,LOW);
   buttonState = digitalRead(dataIn);
   if (buttonState == LOW){
     keluar = false;

  }
  }
}
```

3. **Program RFID *reader***

```
#include <SPI.h>
#include <RFID.h>

#define SS_PIN 10
#define RST_PIN 9
#define ledIDsalah 7
#define ledIDbenar 8
int kirimData = 6;
boolean keluar = false;



long int data;

RFID rfid(SS_PIN,RST_PIN);

void setup()
{
 Serial.begin(9600);
 SPI.begin();
 rfid.init();

 pinMode (ledIDbenar, OUTPUT);
 pinMode (ledIDsalah,OUTPUT);
 pinMode (kirimData, OUTPUT);
 digitalWrite(kirimData,LOW);

}

void loop()
{

 if(rfid.isCard())
 {
  if(rfid.readCardSerial())
  {
   String serial;
   for (int i=0; i<= 4; i++)
   {
    serial += String(rfid.serNum[i],DEC);
   }
   Serial.println("Card ID: " + serial);
```

```
  //digitalWrite(led1, HIGH);
  //Card ID: 136411988163 = KTP
  //Card ID: 217416521347 = A1
  //Card ID: 2451333844122 = A2
  //Card ID: 229160454468 = X
  //Card ID: 872510115 = G.KUNCI

   if ( serial == "136411988163" || serial =="217416521347" || serial ==
"2451333844122"){
    // Serial.println(DataAnalog1);
     digitalWrite(kirimData,HIGH);
     digitalWrite(ledIDbenar,HIGH);
     delay(3000);
     serial = "";
     keluar = true;
   }
   else
   {
      digitalWrite(kirimData,LOW);
     for(int i = 0; i <= 3; i++){
       digitalWrite (ledIDsalah, HIGH);
       delay(200);
       digitalWrite (ledIDsalah, LOW);
       delay(200);
     }
   }
    rfid.halt();
 delay(1000);
  }
 }

   while (keluar){
   if(rfid.isCard()){
   if(rfid.readCardSerial())
   {
   String serial;
   for (int i=0; i<= 4; i++)
   {
    serial += String(rfid.serNum[i],DEC);
   }
   Serial.println("Card ID: " + serial);

    if ( serial == "136411988163" || serial =="217416521347" || serial ==
"2451333844122"){
    // Serial.println(DataAnalog1);
     digitalWrite(kirimData, LOW);
     digitalWrite(ledIDbenar,LOW);
```

```
    keluar = false;
  }
  else
  {
    for(int i = 0; i <= 3; i++){
      digitalWrite (ledIDsalah, HIGH);
      delay(200);
      digitalWrite (ledIDsalah, LOW);
      delay(200);
    }
  }
}
 rfid.halt();
delay(1000);
}
}
}
```

Lampiran 2 Spesifikasi Alat

Alat sistem keamanan ruangan memiliki spesifikasi sebagai berikut :

1. Memiliki kamera VC0706 yang dapat memfoto dan mendeteksi gerakan
2. Mampu mendokumentasikan gambar
3. Terdapat sensor PIR yang dapat mendeteksi manusia dengan menerima pancaran infra merah dari tubuh manusia.

4. Terdapat sensor ultrasonik untuk mendeteksi seseorang yang memasui ruangan melali pintu
5. Supplay 9 Volt DC
6. Menggunakan RFID reader untuk membaca UID RFID tag sehingga alat dapat membedakan penyusup atau bukan.
7. Mampu menyalakan *Buzzer* sebagai bentuk alarm.

Lampiran 3 Dokumentasi

Lampiran 4 *Datasheet*

1. **Sensor Passive Infra Red (PIR) HC – SR501**
2. **Sensor Ultrasonik HC – SR04**
3. **Kamera VC0706**